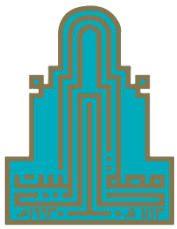

جامعة آل البيت
AL al-BAYT UNIVERSITY

Project Documentation

IntelliKidney

An Integrated AI-Driven Diagnostic System Enhanced with Explainable AI for Chronic Kidney Disease Detection

Supervisor: Prof. Najah Al-Shanableh

Student's:

Mones Nazih Ksasbeh ID: 2100908033 mailto:moksasbeh@gmail.com

Yazan Amjad Mansour ID: 2100908002 mailto:am5294690@gmail.com

Basel Mwafq Hammo ID: 2100908016 mailto:basel.11hammo@gmail.com

Table of Contents

List of Tables

List of Figures

Introduction

Lately, the important sub-domain of artificial intelligence that is machine learning has been used as an effective helper by doctors and medical experts in predicting and diagnosing various diseases. Thousands of lives will be saved all over the world by ensuring the quick prediction of such a disease before severe harm is caused to the patient. Such diseases can be detected by training machine learning algorithms on patient data regarding medical conditions [1]. However, the challenge is how to provide the most accurate prediction in the least amount of time. A need for effective diagnosis of various diseases stares in the face of the world today. The intricate mechanisms of diseases within the patient populations pose monumental challenges to early diagnostic tools and treatment modalities [2].

A kidney function disorder leads to a buildup of waste and surplus fluids in the body, which can cause serious complications. Blood is filtered, and excrement and excess fluids are converted to urine by the kidneys. Deterioration in kidney performance is slow, where no symptoms are distinguished for illness in the early stages [3].and the appearance of symptoms increases when the situation worsens, so the disease is detected only in the late stages. Chronic kidney disease, or CKD, is a condition in which the kidneys are so damaged that they can't filter blood as well as they should. The kidney's main job is to get rid of waste and extra water from the blood [4]. This is how urine is made. CKD means that waste has built up in the body. It is a disease that affects people all over the world. Because of CKD, you might experience various difficulties with your health. Diabetes, high blood pressure, and heart disease are only 3 of the many conditions that can lead to CKD [5]. CKD can appear in various forms, such as **Stones**, **Cysts**, and **Cancer**. Due to super-saturation of some substances in urine, the deposits of solid minerals and salts formed in the kidneys are termed **Kidney Stones** or renal calculi. Crystallization occurs when urine composition is altered by changes in the levels of water, salts, and minerals, leading to fusion and stone formation [6]. **Kidney Cysts** can be classified as fluid-filled sacs that can either form inside the kidney or on the kidney's outer surface. These can range from simple and benign cysts which require little intervention to complex cysts that may signify more serious conditions, including cancer, and require constant management.

Kidney Cancer begins in the kidneys, with renal cell carcinoma (RCC) being the most common type, accounting for about 90% of cases. Other types include transitional cell carcinoma, Wilms' tumor (mostly in children), and renal sarcoma. RCC ranks as the 14th most common cancer globally, making up nearly 3% of all cancer cases. These statistics are based on the 2022 update from the European Association of Urology. The 14th most common cancer in the world, about 3% of all cancers, is renal cell carcinoma (RCC), which is the most common type of renal neoplasm. It accounts for 90% of kidney cancer. Transitional cell carcinoma, Wilms' tumor, and renal sarcoma is among the other kinds. The term kidney cancer refers to cancers that begin in the kidneys. These statistics were published in the 2022 update of the European Association of Urology [6].

There are 5 stages of Chronic kidney disease, each related to the level of kidney function and kidney damage. To find out the stage of kidney disease, blood pressure, eGFR, and ACR (albumin: creatinine ratio) will be checked by the doctor.

Stages 1-2 (early-stage kidney disease): You may not know you have early-stage kidney disease as there are usually no obvious signs. In stage one your eGFR result will be more than 90 Stage two your eGFR result will be 60 to 89, your doctor will probably take steps to prevent you from developing cardiovascular disease. This may involve lifestyle changes, medicines to lower your blood pressure and medicines to keep your blood sugar under control [5].

Stages 3-4 (middle stage kidney disease): This is the stage when most people are diagnosed with CKD. You may start to feel unwell as the waste builds up in your body and your blood pressure increases. In Stage three your eGFR result will be 30 to 59 Stage four your eGFR result will be 15 to 29. you may need treatment to lower your blood pressure, blood fats and blood sugar [5].

Stage 5 (kidney failure): Sometimes, kidney disease can lead to kidney failure. People in this stage will need dialysis or a kidney transplant. Stage five your eGFR result will be under 15 No matter what stage you are in, treatment can help slow the progress of kidney disease and reduce your chance of complications, when your kidneys can no longer function on their own, you may need a kidney transplant, Dialysis which removes waste and extra fluid from the blood, supportive care, if you are being treated for chronic kidney disease, your doctors may need to change other

medicines you are on. This is because many medicines can affect the kidneys, such as blood pressure drugs and anti-inflammatory medicines. Some medicines which leave the body through the kidneys may need to have their dose changed [5].

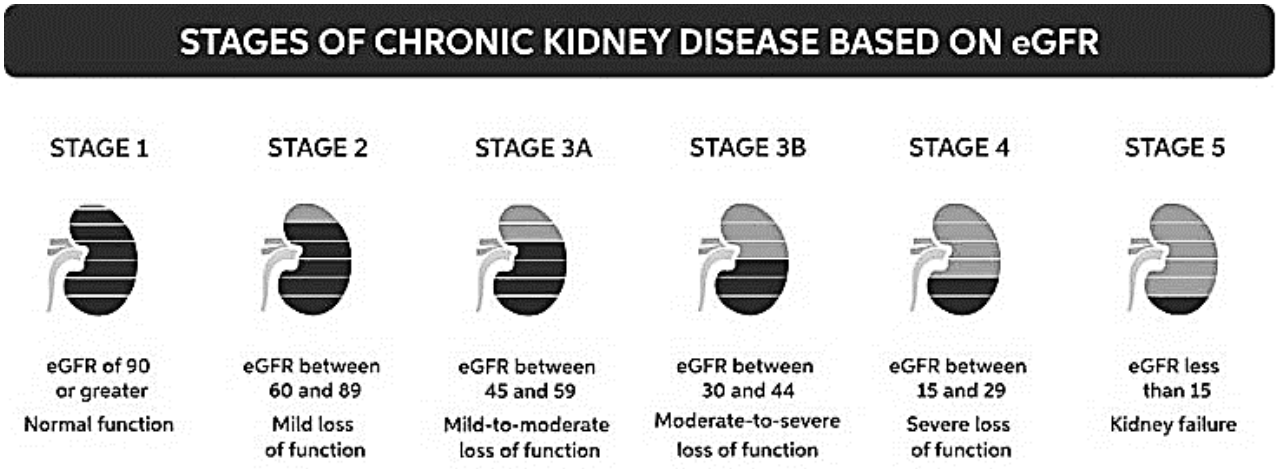

Figure 1. Chronic kidney disease Stages

Chronic kidney disease cannot be cured but treatments can help prevent it from getting worse. Your treatment will depend on the stage of your kidney disease [2]. There are many ways in which kidney failure is diagnosed such as:

- **Urine Test** something may indicate that there is a kidney problem, such as: the presence of certain substances like proteins in urine and there is a very low focus of excrement in urine.
- **Imaging tests** in some cases, some imaging tests are performed to diagnose a tumor or injury Like: Ultrasound Imaging or CT scan.
- **Biopsy** is taken by inserting a needle into the body and he taking a little piece of kidney tissue where this tissue is examined in the lab.
- **Fluid retention** the kidney's ability to release excess fluid it is may be damaged what causes edema in limbs and lungs, In addition to high pressure blood.
- **Arrhythmia** potassium release from the body decreases which leads to a rapid rise in blood levels and leads to severe arrhythmias and could lead to death.

- **Broken bones** kidneys balance blood levels of phosphorus and calcium, these are vital elements to build bones, Damaging the concentration of these minerals leads to weakened bones and Increased likelihood of fracture.
- **Anemia** which is excreted from the kidney and responsible for stimulating (Erythropoietin) Anemia occurs as a result of a decrease in hormone erythropoietin (Bone marrow) to produce erythrocytes (red blood cells).
- **Injury** to the central nervous system accumulation of toxic substances in the body leads to cause damage especially in the Central system where the patient suffered from hard to concentrate and changes in personality in addition to the seizures.

Chronic kidney failure occurs mostly as a result of some illnesses, which prevents the kidneys from performing their function for long periods Examples of these diseases Diabetes Mellitus, Hypertension, Enlarged prostate gland, kidney stone, Urinary bladder cancer, kidney cancer, Kidney infection, Vasculitis, and Scleroderma [3].

In some cases, where there is a significant decreasing kidney function resorted to:

Dialysis It is performed by medical devices, it works to filter the blood, blood is inserted through a tube into the machine, where it is filtered then return it to the body again, this process takes hours and it should be done several times a week [2].

Kidney transplant It is transplanted a new kidney into the patient's body Donated by a live or a deceased person [2]

1.1 Problem Statement

Chronic Kidney Disease is a pandemic condition that is the gradual loss of kidney function, and it is silent until the advanced stages of the illness. Therefore, these degenerative effects advance in the absence of medical attention, increasing the dangers that may lead to severe health complications like hypertension, anemia, bone fractures, or cardiovascular diseases [1]. This is compounded by the fact that chronic kidney disease has very different etiologies and diverse clinical features, such as stones, cysts, and tumors, whose approaches are different with regard to diagnosis and treatment. Incorporating structured data-such as lab test results, medical history, and patient vitals-with unstructured data like CT scans into a comprehensive diagnosis will be quite an of challenge. The classic diagnosis approach does not have scalability because it is not accurate and interpretative, unlike that of today's healthcare. Further, not only would the prediction need to be accurate, but its output must also be explainable; without it, the medical professionals would not trust or validate the recommendations made. Hence, a very strong need exists for an artificial intelligent, interpretable diagnostic system that combines structured and unstructured data to give accurate, fast, and explainable classification and prediction of kidney-related conditions.

1.2 Project Objectives

The goal of this project is to develop a hybrid AI-based early diagnosis and classification of kidney-related conditions by:

- Structured Data Modelling: Training machine learning models using clinical features such as age, blood pressure, etc., for CKD detection and its stage evaluation.
- Image Classification: Transfer learning method (e.g., VGG19/ ResNet) for CT image classification into Normal, Cyst, Stone, or Tumor.
- Explainability (XAI): Using feature importance methods, such as SHAP, on structured data. Using LIME for visualizing critical regions in CT scan images on which the predictions are based.
- Deployment: Integration of models with a web app with a user-friendly interface, i.e., Streamlit. Storage of structured inputs in PostgreSQL and CT images in MongoDB Atlas using GridFS.

An accurate, fast, and interpretable system is aimed at generating diagnostic confidence in addition to improving patient outcomes.

Literature Review

2.1 Overview

Healthcare has been transformed with the help of artificial intelligence (AI) and machine learning (ML), which have enabled accurate and interpretable automated disease predictions. One such disease is Chronic Kidney Disease (CKD), which is progressive and affects millions of people worldwide, requiring early identification to avert permanent damage to the kidney [7].

Clinical expertise, laboratory tests, and medical imaging are the bases of conventional diagnostic methods. These techniques consume time, are subject to human error, and depend on the availability of specialists. Machine learning models offer a more data-driven approach to CKD diagnosis by analyzing patterns in structured clinical data, as well as medical images [7].

Furthermore, deep learning improves diagnostics through better image-based classification. This project involves ensemble learning, transfer learning, and explainable artificial intelligence (XAI) strategies to enhance accuracy in CKD prediction while maintaining the ability to interpret outcomes [8].

With the aim of predicting Chronic Kidney Disease (CKD), 10 different machine learning algorithms have been applied, whose levels of complexity range from simpler to more complex ensembles, to test their efficiency in prediction of CKD. On many levels, the cross-validation of these models is done to back up performance and enhance its validity away from overfitting. In addition to this, synthetic data generation techniques are initiated to mitigate class imbalance and enhance generalization of the model, minimizing the chances of overfits with smaller datasets. This review focuses on two fundamental aspects of CKD diagnosis:

- Tabular Data: Utilizing the UCI CKD dataset to predict Chronic Kidney Disease.
- CT Image Classification: Leveraging CT kidney images (Normal, Cyst, Stone, Tumor) to identify abnormalities

2.2 Related Work

Recent studies highlight the role of ensemble models, transfer learning and explainable artificial intelligence in healthcare applications. For instance, ensemble methods enhance robustness by combining diverse classifiers, as demonstrated in kidney disease prediction. Similarly, transfer learning using pre-trained convolutional neural networks (CNNs) such as ResNet and VGG16 reduces computational costs while maintaining high accuracy in medical imaging [9].

2.2.1 CKD Prediction Using Tabular Data

For example, Almansour et al. (2019) have developed an artificial neural network (ANN) and support vector machines (SVM) to classify chronic kidney disease using the UCI CKD dataset. Their models achieved an 99% accuracy and displayed the promise of machine learning in CKD diagnosis [10]. However, mean imputation was used by them for the missing values-that assumes a normal distribution of missing data, and therefore that naive technique tends to introduce bias and may distort patterns to the extent that model generalization could be affected-the model could rather be more reliable if some more robust imputation methods such as K-nearest neighbors (KNN) imputation or multiple imputations were applied.

A popular models problem in medical datasets is to have an imbalanced dataset where CKD-positive are the minority cases. Accordingly, Gupta et al. (2020) looked at logistic regression coupled with the Synthetic Minority Over-Sampling Technique (SMOTE) to remedy the class imbalance [11]. While SMOTE really helped reduce the bias before applying the model toward the majority class, the authors did not analyze feature importance. thus, clinical interpretability of the findings was undermined, as medical practitioners require some acknowledgment of which clinical indicators drive CKD prediction. A clear decision mechanism could have been achieved via using SHAP (SHapley Additive Explanations) values or permutation importance.

Additionally, Kumar et al. (2021) conducted a study on Random Forest and XGBoost, which are powerful ensemble learning techniques known for their ability to handle non-linearity and feature interactions [12]. While these models did well in classifying the data, the study did not address hyperparameter optimization that is fundamental in maximizing predictive accuracy. This means that the estimation of parameters was not altered, resulting in possible overfitting or underfitting.

2.2.2 CT Image Classification for Kidney Disease

Other researchers, Sarvamangala and Kulkarni (2021) created a custom-designed convolutional neural network (CNN) for renal tumor detection with an accuracy of 92%. However, this model would have needed large amounts of training data to be able to generalize properly [13]. Since there was no provision for transfer learning, the model had to be trained from scratch, leading to high computational costs and overfitting.

Again, Jha et al. (2022) deployed one of the most widely used deep learning architectures, ResNet-50, on CT kidney images, achieving a 94% accuracy rate. Although this latest development did perform better than some earlier conventional models, it still lacked an explainable AI (XAI) component, which is vital for transparent medical applications. In the absence of Grad-CAM or SHAP and other explainability techniques, decisions made by the model would be ambiguous, thus eroding the trust of clinicians in AI-driven diagnosis [14].

Additionally, Chen et al. (2023) have used VGG16 along with Grad-CAM to visualize CT images in kidney tumor classification, thereby allowing interpretability through emphasizing important regions [15]. While this helps explainability, it was applied simply for tumor detection and did not constitute a multi-class classification framework capable of distinguishing between normal, cystic, and stone-affected kidneys. Extending this model into a multi-class framework would increase its clinical utility and applicability.

2.3 Research Gaps and Opportunities

These earlier studies used mean imputation, and therefore might be biased; it can be suggested that good data quality could have been obtained if other stronger methods, such as KNN, were used. There was no analysis of feature importance for clinical interpretation. Hyperparameter optimization was not done, increasing the chance of either overfitting or underfitting. The lack of transfer learning reduced the effectiveness of the models for image classification.

The limited XAI interpretation of the model means it was much less easy to understand. In addition, previous studies focused solely on tumor detection, ignoring other renal conditions such as cysts or stones. This project acts to fill this gap by focusing on using advanced imputation, feature importance analysis, hyperparameter tuning, transfer learning, and XAI with SHAP and LIME for a fully interpretable and comprehensive model of multi-class diagnose.

2.4 Summary of Related Work

This chapter represents some of the classification techniques that are used in the prediction of Kidney Disease. The researchers applied different methods and approaches to obtain the required results and enhance the overall performance of Kidney Disease prediction. Table 1 provides a comparative summary of key studies, datasets, methodologies, and outcomes related to CKD prediction and classification.

Table 1 : Summary of Related Work

No	Study	Methodology	Key Findings	Limitations
[1]	Almansour et al. (2019)	ANN, SVM, UCI CKD dataset	Achieved 99% accuracy in CKD classification	Used mean imputation for missing data, which may introduce bias; could benefit from more robust imputation methods (e.g., KNN)
[2]	Gupta et al. (2020)	Logistic Regression, SMOTE, UCI CKD dataset	Addressed class imbalance with SMOTE, improving model performance	Did not analyze feature importance, affecting clinical interpretability of the results.
[3]	Kumar et al. (2021)	Random Forest, XGBoost, UCI CKD dataset	Effective at handling non-linearity and feature interactions	Lacked hyperparameter optimization, leading to potential overfitting/underfitting issues
[4]	Sarvamangala & Kulkarni (2021)	Custom CNN for kidney tumor detection , Kaggle CT Images	Achieved 92% accuracy in renal tumor detection	lacked transfer learning
[5]	Jha et al. (2022)	ResNet-50, CT kidney images , Kaggle CT Images	Achieved 94% accuracy, a widely used deep learning architecture	Lacked explainable AI (XAI) component, which reduces model transparency for clinical use
[6]	Chen et al. (2023)	VGG16 with Grad-CAM for kidney tumor classification , Kaggle CT Images	Enhanced model interpretability using Grad-CAM for feature visualization	Focused only on tumor detection; lacked multi-class capability (normal, cyst, stone)

Methodology

Two datasets, the UCI CKD dataset and the CT image data, were used in the workflow as shown in Figure 2. It started from the problem understanding, data collection, and understanding the data phase. During this phase, it split into two paths. The CKD dataset path traverses through preprocessing, model building, optimization, evaluation, deployment using the web app, and relational database connections. There is a parallel path for CT imaging, with all steps adopting deep learning techniques for model building, ending with deployment and non-relational database connection. This workflow exemplifies that different data types necessitate different methodologies within the frame of the same overall process.

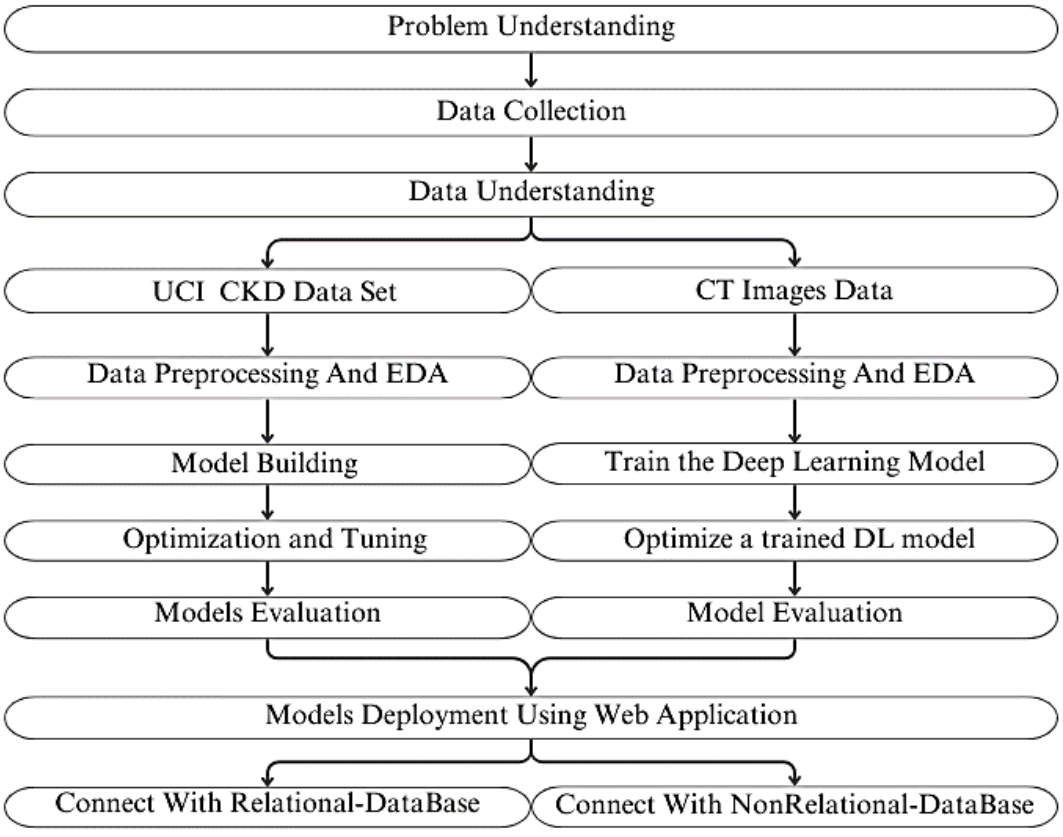

Figure 2. Methodology Flow Chart

3.1 Problem Understanding

The objective is to Predict Target Variable which represent if a patient has a Chronic Kidney Disease or not. Chronic Kidney Disease (CKD) is a serious and advancing condition that can lead to severe health issues, including kidney failure, cardiovascular disease, and increased mortality if left undiagnosed or untreated. If the problem remains unresolved, patients may continue to face delayed diagnoses and limited treatment options, leading to preventable health deterioration, increased healthcare expenses, and higher mortality rates. Developing an accurate predictive model for CKD can transform healthcare outcomes by enabling early intervention, improving patient care, and reducing the socioeconomic burden of the disease. In light of this, both structured patient data and images from Computed Tomography (CT) scans will be used to promote an accurate diagnosis. The model will therefore not only identify the presence of CKD but also assess CT imaging for possible underlying causes like tumor, cyst, or stone. This would enhance the understanding of the disease and provide a basis for medical professionals to decide on the clinical management of affected persons.

3.2 Data Understanding

One of the data source used for this research is **UCI Machine Learning Repository** in addition to Synthetic data, the dataset contains 500 entries and 25 columns. This dataset includes patient records with 24 independent attributes, offering valuable insights into the factors influencing Chronic Kidney disease, out of which 11 are numerical, represented as 'float64', and 13 are categorical, represented as 'object'. The numerical columns include measurements such as blood pressure, specific gravity, and hemoglobin levels, which are critical for analyzing health-related factors. The categorical columns include variables such as red blood cell count, diabetes status, and appetite, which are expressed in discrete categories and provide qualitative insights about the patients [16]. As we see in the Table 2 we have 24 input feature and one target variable and here a small description. Synthetic data is data created to look like real data, but it doesn't contain any actual information. It's made using AI and computer simulations, and it's useful for research, testing, and training machine learning models. One big advantage is that it lets organizations generate as much data as they need, in a structured and labeled way [17].

Synthetic data can also reduce bias in AI models by balancing datasets and making them fairer. While new AI techniques make generating synthetic data faster and easier, they also bring up some rules and safety concerns. In our case, **SDV (Synthetic Data Vault)** was used to generate additional data points since the original dataset was small.

The **SingleTablePreset** model was applied to analyze patterns in the data, and **100 new synthetic records** were created. These extra data points were added to the model, strengthening it by providing more data to learn from.

Table 2 : Dataset description

Attribute	Description of Attribute	Attribute	Description of Attribute
Age	Patient Age	pcv	White Blood Cell Count
bp	Blood Pressure	rc	Red Blood Cell Count
sg	Specific Gravity	htn	Hypertension
al	Albumin Levels	dm	Diabetes Mellitus
su	Sugar Levels	cad	Coronary Artery Disease
rbc	Red Blood Cells	appet	Appetite
pc	Pus Cell Count	pe	Pedal Edema
pcc	Pus Cell Clumps	ane	Anemia
ba	Bacteria Presence	bu	Blood Urea
bgr	Blood Glucose Random	sod	Sodium Levels
sc	Serum Creatinine	pot	Potassium Levels
hemo	Hemoglobin	Class	Target variable

The second form of data is the CT Kidney Dataset: Normal-Cyst-Tumor-Stone, which can be defined as a dataset of kidney CT images with a fourfold classification of: Normal, Cyst, Tumor, and Stone. The dataset was developed particularly for the analysis of medical images to allow the creation of models that detect and classify kidney-related clinical conditions. Images in the dataset add up to a total of 12446 in the following distribution:

- Normal: 5077 images
- Cyst: 3709 images
- Tumor: 2283 images
- Stone: 1377 images

The image resources are taken from the Picture Archiving and Communication System of Hampstead, Bangladesh. Patients included in this study had one of the confirmed kidney diseases. They include coronal and axial images of contrast and non-contrast from the comparative studies following standard imaging protocols in whole abdomen and urogram examination as shown in Figure 3.This comprises the most important dataset for the training of machine learning models intended for early prediction and classification of kidney disorders in a future vision towards timely diagnosis and intervention, thus improving patient care.

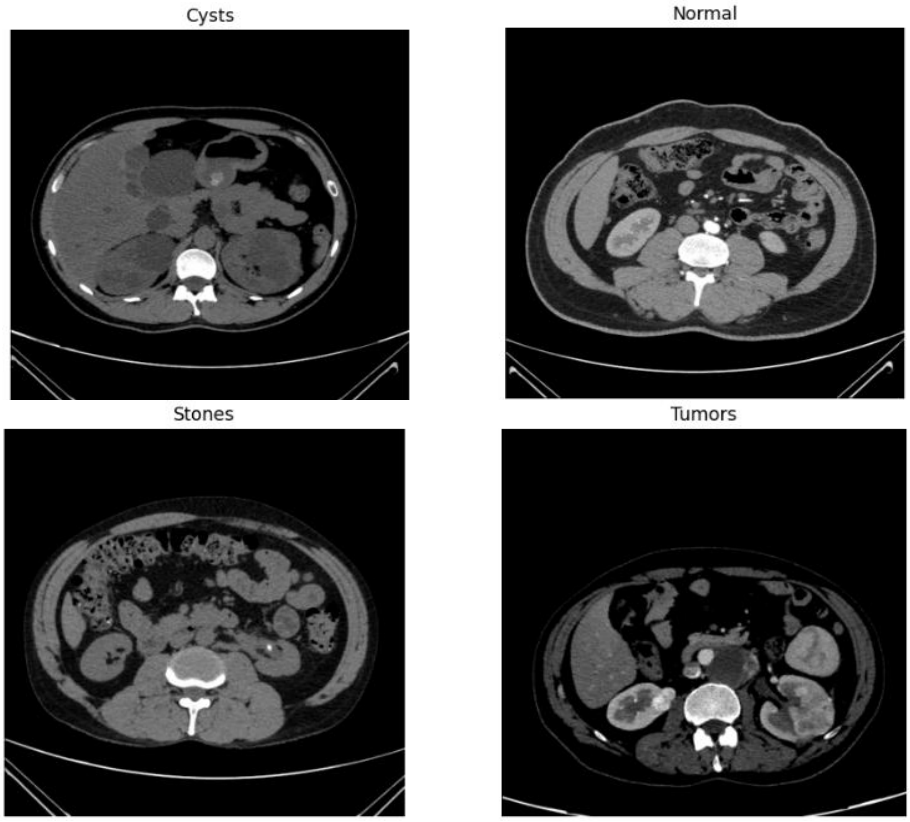

Figure 3. CT Images Examples

Firstly, the presence of Chronic Kidney Disease (CKD) in the patient will be identified using structured data. After this, if CKD is detected, the CT scan images will be analyzed to classify the disease into one of the available categories—tumors, cysts, or stones. Such classification and the two-stage methodology have been proposed to establish a detailed diagnostic mechanism, in which structured patient data is combined with imaging methods to provide a more comprehensive and informative assessment of CKD.

3.3 Data Pre-processing & Exploratory Data Analysis

Data preprocessing and Exploratory Data Analysis (EDA) are extremely important steps within a data science workflow that ultimately lead to well-structured and quality data on which machine learning models can be built. Data preprocessing is a process by which raw data gets cleaned and

transformed into an analyzable format. This could involve treatment of missing values, encoding categorical variables, scaling of numerical features, and handling of imbalanced classes. This is done to prepare the data for modeling. As shown in Figure 4, this process plays a crucial role in setting the foundation for effective analysis and modeling. Primarily, EDA focuses on the understanding of the pattern, relationship, and characteristics of the data set. EDA summarizes and visualizes data to uncover hidden insights, detect outliers, and advise further steps of preprocessing. A necessary step that one must go through to generate hypotheses and build a deeper understanding of the data.

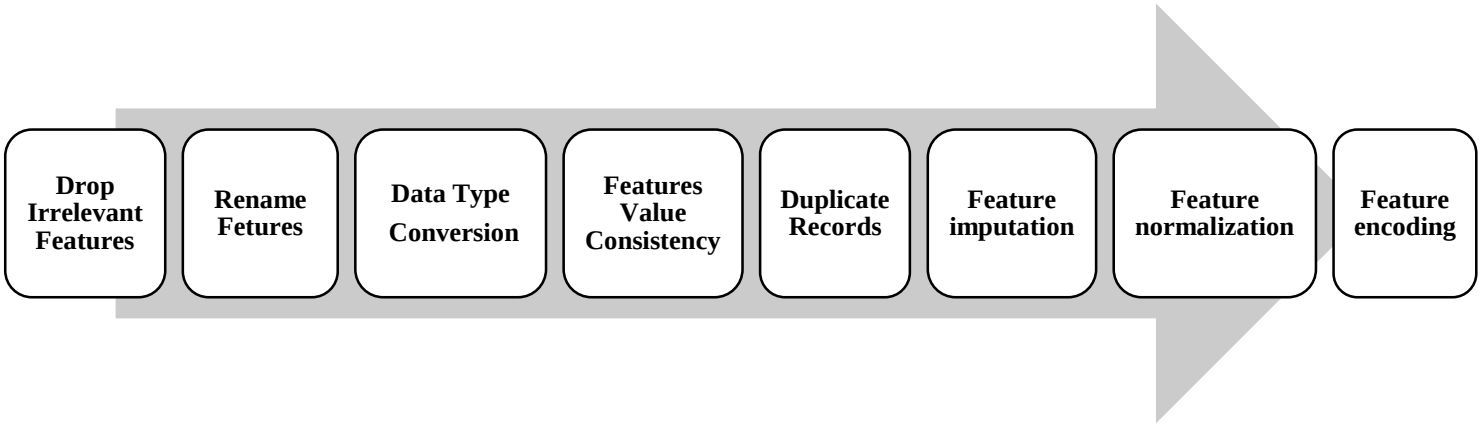

Figure 4. Data-Preprocessing Flow

The ID column, being irrelevant, was dropped. Feature names were renamed for better understanding. The columns Packed Cell Volume, White Blood Cell Count, and Red Blood Cell Count had incorrect data types and were converted to appropriate formats. As we see in the Table 3, the unique values of categorical features were inspected to ensure consistency and identify any potential anomalies.

Table 3 : Categorical features values

Attribute	Attribute Values
Red Blood Cells	[NaN , Normal , Abnormal]
Pus Cell	[NaN , Normal , Abnormal]
Pus Cell lumps	[NaN , Present , Not Present]
Bacteria has	[NaN , Present , Not Present]
Hypertension	[Yes , No , NaN]
Diabetes Mellitus	[Yes , No , '\tno' , '\tyes' NaN]
Coronary Artery Disease	[Yes , No , '\tno' , NaN]
Appetite has	[Good , Poor , NaN]
Peda Edema	[Yes , No , NaN]
Aanemia	[Yes , No , NaN]
Class	[CKD , NoCKD , CKD\t]

Additionally, the percentage of missing values for each feature was calculated to identify columns requiring imputation or cleaning as shown in Figure 5, but the missing values could not be removed due to the small number of records.

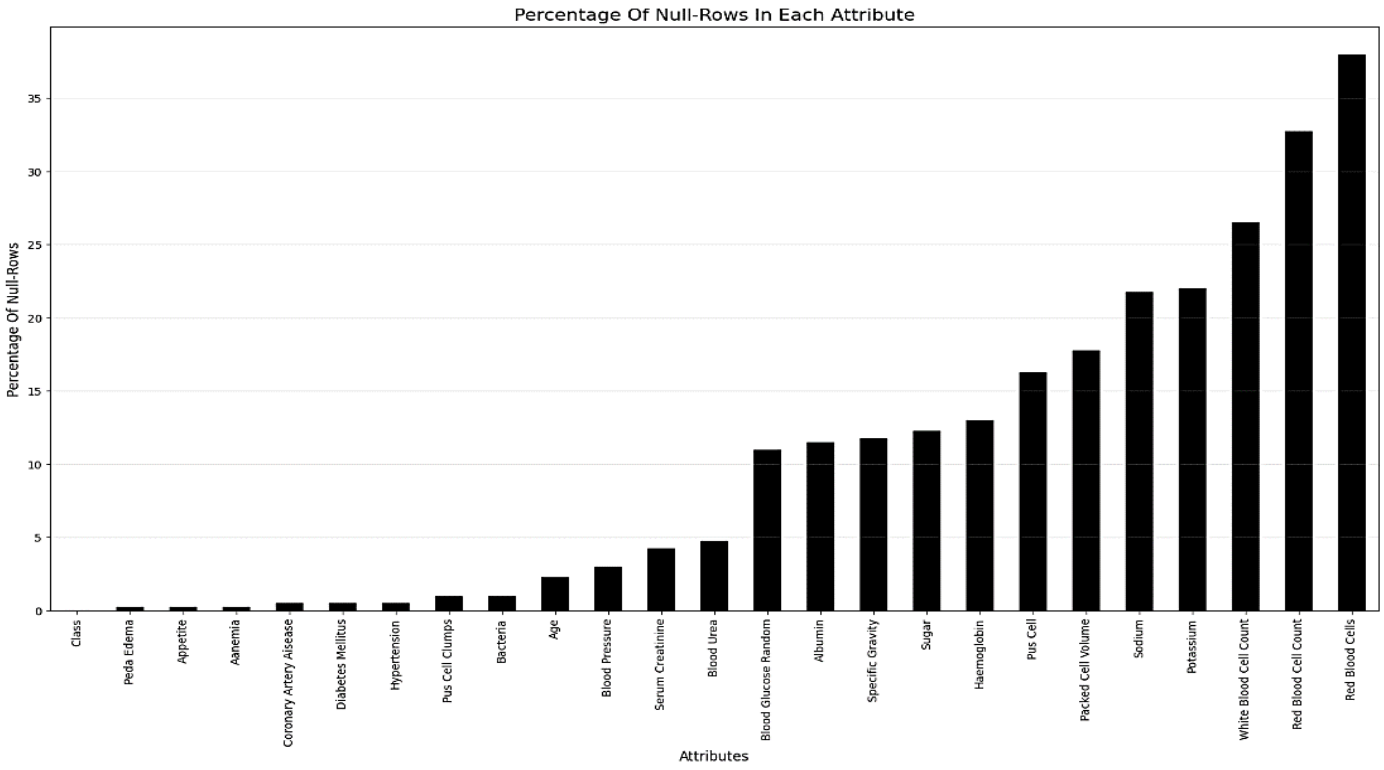

Figure 5. Missing Value Percentage

For handling missing values, 3 methods were used:

Random imputation was applied for features with less than 15% missing values in numerical data. Random imputation is a technique used to handle missing data by replacing missing values with randomly selected observed values from the same variable. This method preserves the original distribution of the data and is straightforward to implement [18]. By using this approach, the distribution of the data was maintained while filling in the gaps.

Mode imputation was utilized for categorical data with 1% missing data. In this method, the missing entries were replaced with the most frequently occurring value in that column, ensuring that the categorical integrity of the dataset was preserved.

KNN imputation was employed for numerical data with more than 15% missing values. This imputer utilized the k-Nearest Neighbors method to replace the missing values in the dataset with the mean value from the parameter 'n_neighbors' nearest neighbors found in the training set. By default, a Euclidean distance metric was used to impute the missing values [19]. This technique worked by analyzing the relationships and similarities between different data points, allowing the missing values to be predicted and filled in based on the characteristics of surrounding observations.by applying these methods, the overall accuracy and reliability of the findings were enhanced, ensuring that the dataset remained robust and informative.

The process of data encoding was begun by employing **label encoding** techniques. This approach systematically converted categorical variables into numerical values, facilitating easier analysis and machine learning model training. By assigning a unique integer to each category, the dataset's interpretability and compatibility with various algorithms were enhanced. Using a bar chart to visualize the values of categorical features makes it easier to understand the frequency or distribution of each category. It gives us a clear view of how many instances belong to each category, helping us spot imbalances, or any categories that may be underrepresented or overrepresented, as shown in Figure 6.

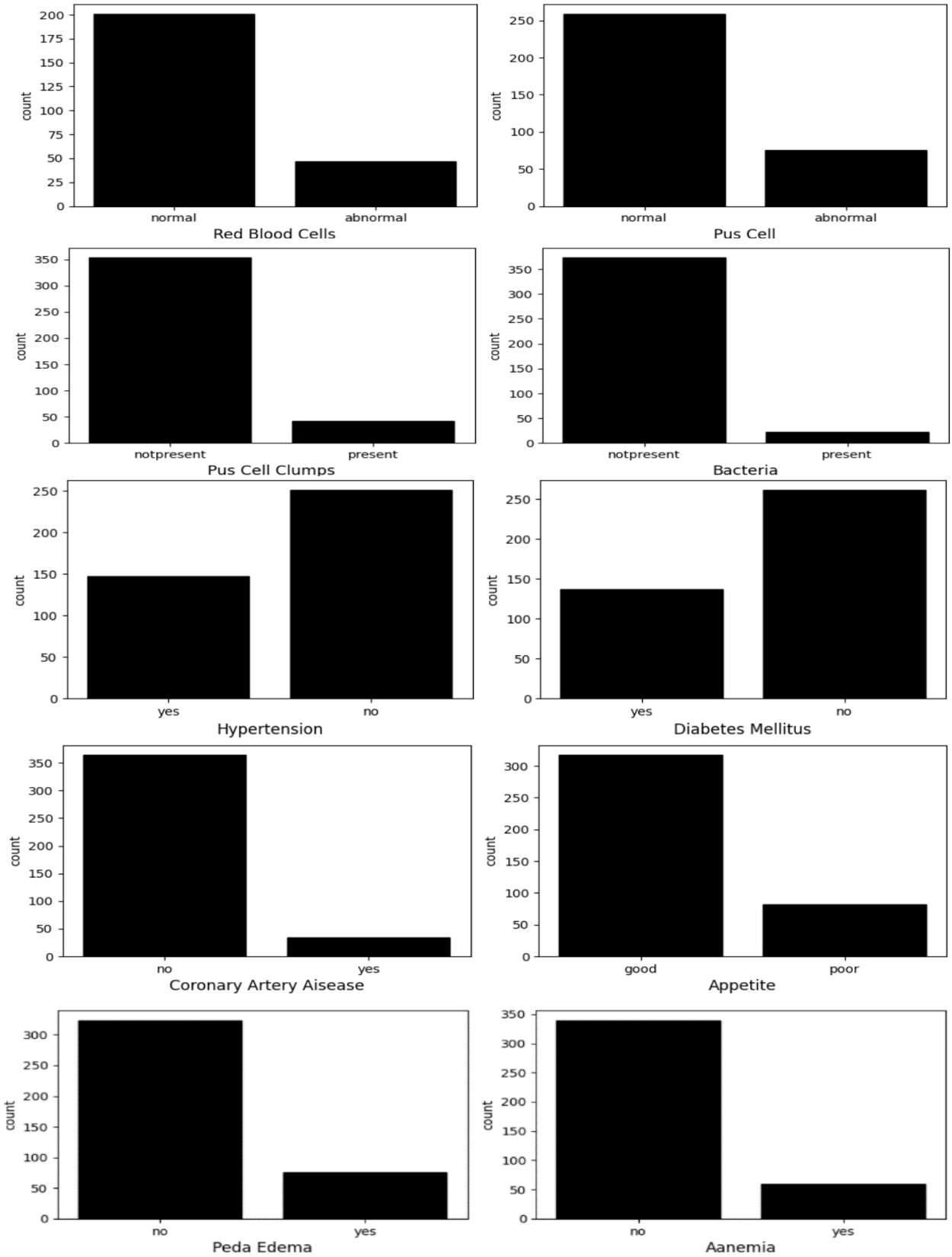

Figure 6. Categorical features distribution

As we see in Figure 7, creating a histogram helps us get a clear picture of how the numerical values are spread out. It lets us see if the data follows a normal distribution or if there's any skewness. By looking at the histogram, patterns can be spotted, outliers can be identified, and the overall distribution of the data can be understood, which helps guide decisions on data preprocessing or modeling.

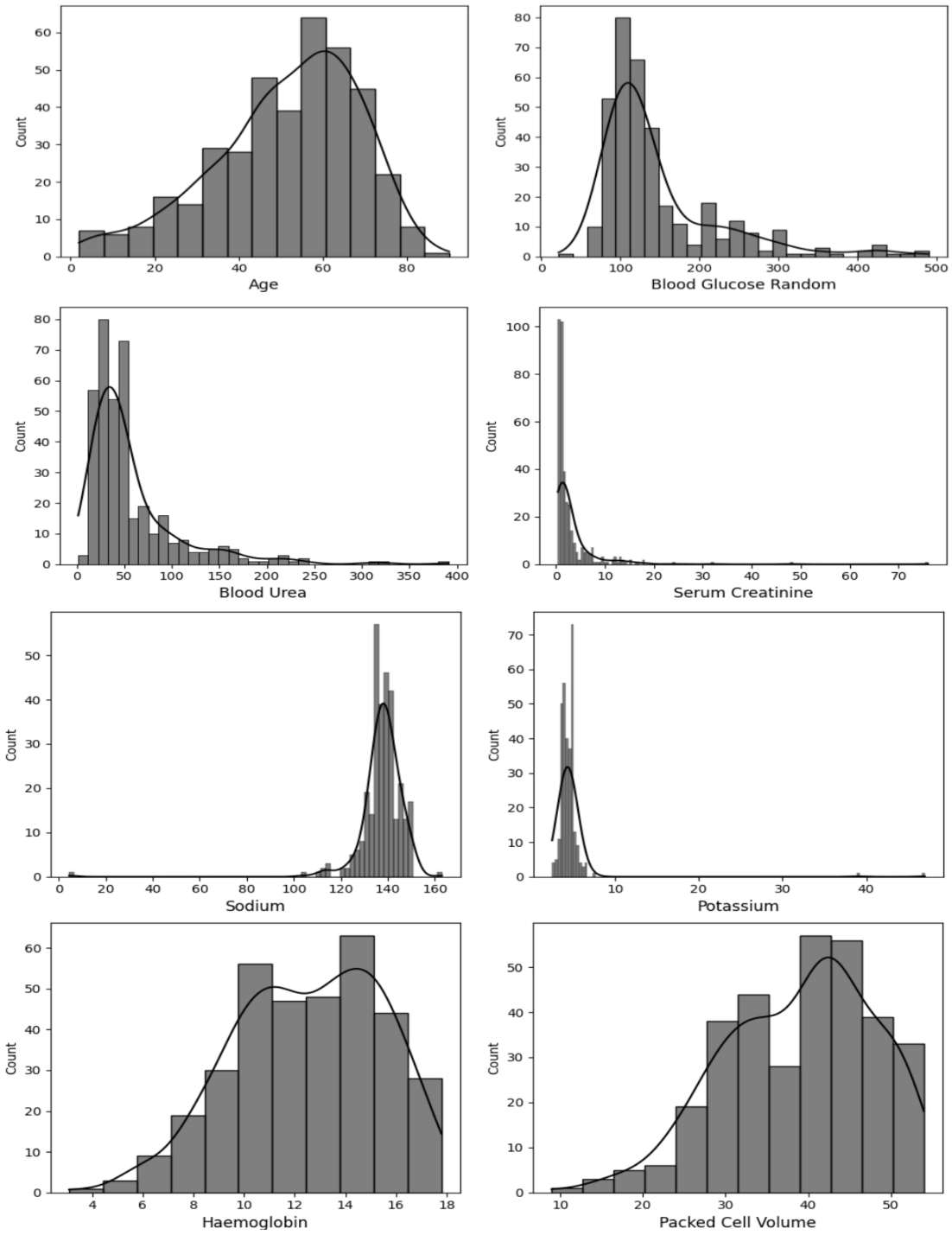

Figure 7. Numerical features distribution

The relationship between the 'Class' target feature and the numerical features was explored, examining their connections to each other as shown in Figure 8. This helped in identifying important features and understanding their influence on the target, guiding the model-building process.

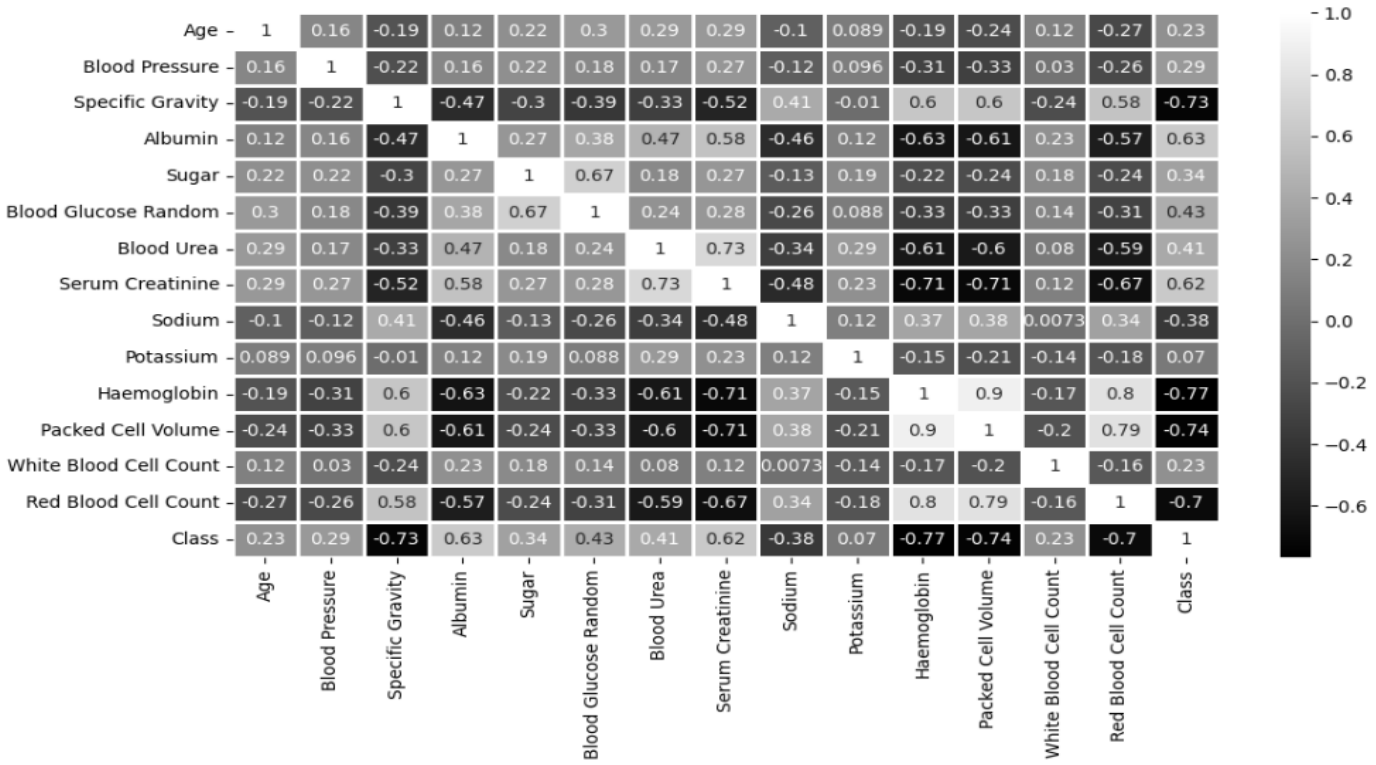

Figure 8. Correlation Matrix

Preprocessing and exploratory data analysis (EDA) tasks performed on the **CT image data** include resizing, normalization, and class distribution analysis. The images are resized to ensure consistent dimensions across the dataset, allowing for uniform input to the machine learning model and improving computational efficiency. Normalization is conducted to scale image pixel values, which facilitates model development, as having all pixel intensities within a standard range enables quicker convergence during training. As we see in Figure 9, the distribution of classes (Normal, Cysts, Tumor, and Stones) in the dataset was analyzed using a bar chart to identify class imbalance and assess its potential impact on the model's learning process.

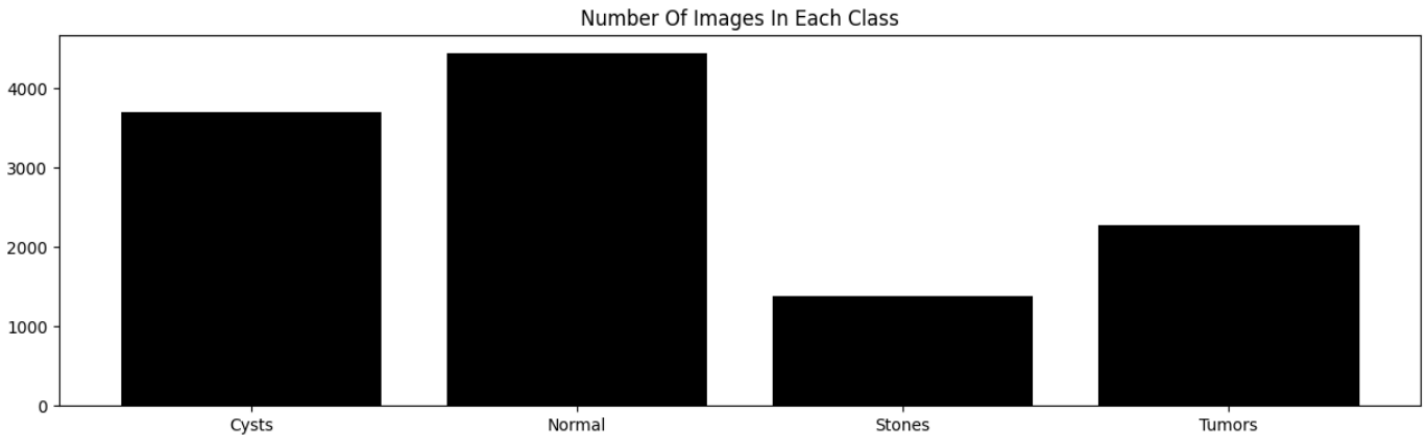

Figure 9. Classes Distribution

3.4 Model Development

In this section, discussion is made for model development process 2 systems: The CKD model and the CT image CNN classifier. The CKD model used for prediction of chronic kidney disease is a model that deals with structured data and various machine learning algorithms. The other model is a CNN classifier which classifies kidney CT images for the detection of tumors, cysts, and stones. These models have a systematic way of developing them to ensure their performance and reliability.

3.4.1 CKD Model Development

For CKD Model, various machine learning models were developed and evaluated to achieve optimal predictive performance. Simpler models, such as Logistic Regression and Decision Trees, Naïve Bayesian, were first implemented to establish a baseline, followed by more complex ensemble methods, including Random Forest, XGBoost, AdaBoost, CatBoost, and Gradient Boosting. Each model was fine-tuned using a structured Hyperparameter grid to balance accuracy and computational efficiency. Additionally, Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) were explored to assess different learning approaches, the overall model development flow is illustrated in Figure 10.

Figure 10. Models Development Flow

AdaBoost is an excellent machine learning approach for creating highly accurate prediction rules by combining and boosting relatively weak and inaccurate rules. It has a compact mathematical basis and increases the efficiency of multiclass classifier problems in practical applications. AdaBoost takes an iterative approach in order to improve the performance of weak classifiers by allowing them to study and improve from their own mistakes. The ability to reduce noise is improved when the AdaBoost is put into the stopping condition [5].

Decision tree when it comes to solving categorization issues, one of the most effective and widely used strategies for supervised machine learning is known as the decision tree. A decision tree is a type of tree structure that is similar to a flowchart. In a decision tree, each internal node represents a test that is performed on a feature, each branch represents the outcome of the test, and each leaf node contains a class label [5].

Naïve Bayesian algorithms provide a probabilistic way of building a model. This approach calculates the probability for each value of the class variable for given values of input variables. With the help of conditional probabilities, for a given unseen record, the model calculates the outcome of all values of target classes and comes up with a predicted winner [4].

Gradient Boosting is an ensemble boosting technique that starts with "regression tree" as "weak learners". In general, the GB model adds an additive model to minimize the loss function by using a stage-wise sampling strategy. The loss function measures the amount at which the expected value deviates from the real value.

To ensure fair evaluation, cross-validation was applied, preventing overfitting while capturing generalizable patterns in the data. The final model was selected based on key performance metrics, such as accuracy, precision, recall, ensuring the most effective solution for the given problem.

Cross-Validation

So instead of dividing the initial data into train and test sets to evaluate model performance on one specific train/test set pair, the performance of this model will be evaluated using the result of a 10-fold cross-validation process. This denotes that the dataset is folded into 10 equal parts or folds. A 10-fold cross-validation will be implemented, and this process will be repeated 10 times, with each fold being used as the test set once.

Moreover, in contrast to this, cross-validation ensures that the same model is trained and tested on various data subsets. As a result, performance metrics become significantly more reliable, preventing overfitting. In fact, classification algorithms are trained on multiple subsets of training data and generalize effectively when applied to new data. Hence, once this process is executed, the average accuracy across all folds will be taken to determine the model's efficiency.

3.4.2 Transfer Learning in Kidney CT Image Classification

Convolutional neural networks (CNNs) emerged from the investigations into the visual cortex of the brain and are used for image recognition since the 1980s. With increasing computational power, ever-growing amounts of training data, and higher software tools, CNNs have lately achieved superhuman performance when it comes to certain complex visual tasks. CNNs are no longer restricted to these areas; they perform well in a number of domains including voice recognition and natural language processing [20].

Transfer learning is a technique under machine learning where the knowledge gained while solving one problem is transferred to a different but related problem. In the case of deep learning, such pre-trained models-the already trained model on a vast dataset like ImageNet-would be fine-tuned to do another task [27].

Why use Transfer Learning?

To train the model, initially in such a rigorous dataset like medical images, with time and needs of significant computational resources, transfer learning comes into play, whereby pre-trained weights from models trained on large datasets, for example, ImageNet, can now be used as an initial weight for this new model. Medical image datasets, such as kidney CT scan images, usually are small, not diverse, and do not contain enough samples. Transfer learning helps the model leverage the features, which have been learned through training on a much larger dataset, and apply them to a new task that has fewer samples and less variability. Deep neural networks, especially convolutional neural networks (CNNs), overfit on small datasets. Transfer learning reduces the chances of overfitting because the model already has robust, generalizable features (learned from ImageNet) so that it is less susceptible to overfitting on the new, smaller dataset [27].

For model selection and justification Two models were chosen for this project based on their performance in medical image classification:

VGG19 This model deals with image classification and object recognition. The model was built at the University of Oxford in 2014 and is said to be one of the most used modern computer vision models. As we see in Figure 11, the model consists of 19 layers, out of which 16 are convolutional and 3 are fully connected layers at the end. The model is defined as relatively simple: with successive convolutional layers followed by fully connected layers. Simple and Clear: The design is simple and, above all, offers a clear understanding of concepts involved. Better performance in traditional work: Yields impressive performance in standard classification tasks.

Why we used it? This suits any project where the model required for classification is deep but does not require extensive computing resources. It makes an added advantage in the early stages of deep learning.

EfficientNetV2B0 is a latest sequence of core models introduced by the EfficientNet family with the idea to improve the resource utilization on the device or machines manifold while still not compromising on the high-level performance. It was fine-tuned to build much more efficient accuracy and performance across devices.

It uses advanced convolution layers and Compound Scaling to obtain the optimal trade-off between depth, width and speed. Low Cost: It runs well, with many resources being consumed by it. Great Balance Between Performance and Model Size: Best suitable for devices with limitations in resources. Maximum Accuracy It does classify tasks better than many older models of this domain, VGG19 inclusive.Why we used it? Interpretation was chosen in this model since it offers great accuracy with resource efficiency, the best for projects that need performance optimization at minimal resource needs, as shown in Figure 12.

ResNet50 belongs to ResNet family and uses technique of "residual connections" to allow some information to bypass without being encoded somewhere in the neural network. This achieved the performance degradation of deep networks. As we see in Figure 13, this has 50 convolutional layers where you can apply the use of 'residual connections' that can help in training deep networks very effectively.

Deep Architecture: This allows to gain better training from larger data and in turn achieve high accuracies. Easier Training: The residual connections help the network not to degrade during the training process This can be easily used in various types of computer vision task. Why we used it? Because ResNet50 is the best option available via excellent performance and depth balance, this model will be used in high-accuracy applications involving classification.

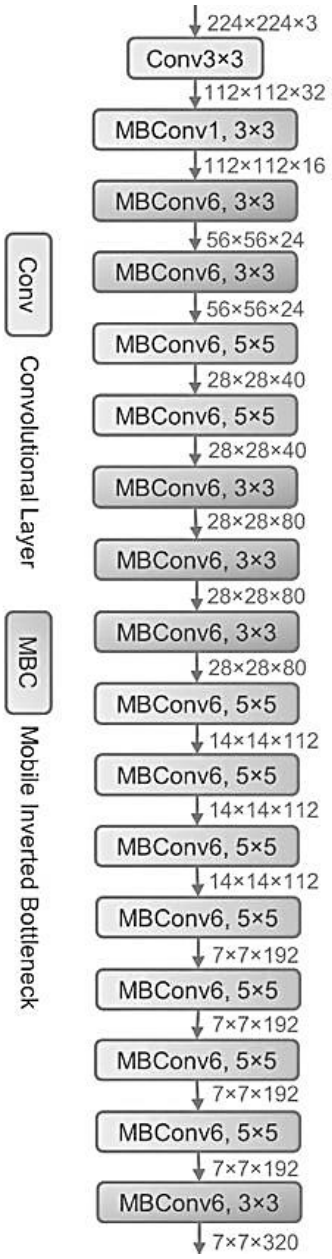

Figure 11. EfficientNet Architecture

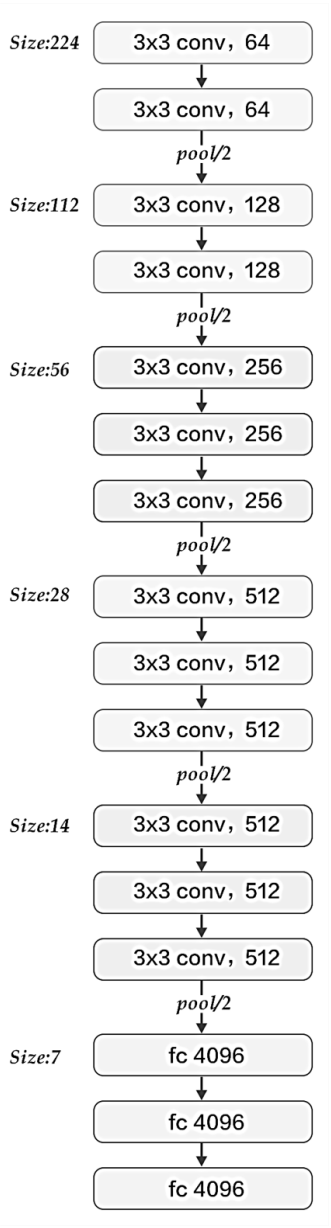

Figure 13. VGG Architecture

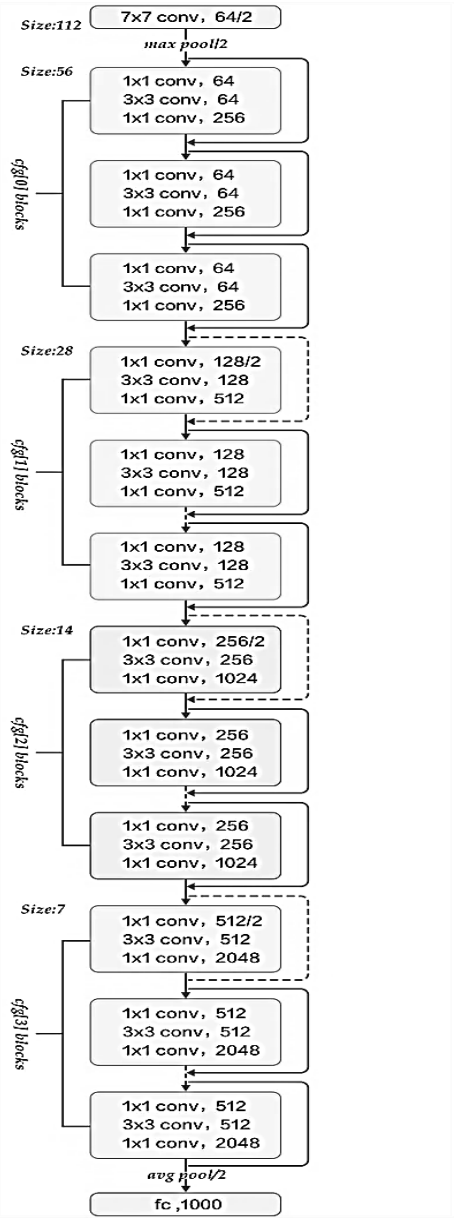

Figure 12. ResNet Architecture

Explainable AI (XAI) Implementation

Do we trust our model without knowing whether its decision is correct? How it made the decision based on what features? Explainable AI, or AI explaining, or AI Explainability, or simply XAI, seems simple. You just take an AI algorithm and explain it [21]. Before the rise of XAI, the typical AI workflow was minimal. The world and activities surrounding us produce datasets. These datasets were put through black-box AI algorithms, not knowing what was inside. Finally, human users had to either trust the system or initiate an expensive investigation [21]. the typical XAI flow takes information from the world and the activities that occur in it to produce input datasets to extract information from, to build, in turn, a white box algorithm that allows AI Explainability.

4.1 Implement XAI for The CKD model

The user can consult an interface that accesses interpretable AI models, here is the list of some model explainability AI techniques:

- **Feature Importance**: This method involves identifying the features or inputs that have the most significant impact on an AI model's decision [22].
- **LIME (Local Interpretable Model-agnostic Explanations)**: Aim to explain specific AI model predictions on individual instances. It involves identifying the features that had the most influence on a particular prediction, offering insights into why the model made that decision [22].
- **SHAP (Shapley Additive Explanations)**: Attribute the contribution of each feature to the prediction by considering all possible combinations of features. This method assigns values based on the average marginal contribution of each feature, providing a fair allocation of importance [22].

To improve the interpretability of the kidney failure disease prediction model, Explainable Artificial Intelligence (XAI) techniques were applied. By integrating XAI, the decision-making process in the model becomes more explainable, allowing us to trust predictions and gain insights into the key factors that influence kidney failure disease.

SHAP, an explainable AI technique, was applied. Was implemented using three types of visualization:

- **SHAP waterfall plot** It begins with the baseline (predicted value) and illustrates how each feature pushes the model towards a higher or lower prediction, providing a visual explanation of how each feature contributes to a single prediction and aiding in understanding the reasoning behind the model's decision.
- **SHAP Summary Plot** Provides a global view of feature importance and the impact of each feature on the model's predictions, summary plot helps identify which features are driving the predictions and how their values influence the outcome, making it a valuable tool for model interpretability.
- **SHAP Summary Bar Chart** by calculating how much a feature raises or lowers a prediction, SHAP values provide a visual representation of how different characteristics contribute to the model's predictions and aid in explaining how each feature affects the target variable.

4.2 Implement XAI for The CT image CNN classifier

There are various techniques for achieving Explainable AI (XAI) on image data that make different attempts to interpret model decisions. Among the popular methods are **Grad-CAM**, which highlights the important regions in a convolutional neural network, and **SHAP**, which attributes pixel-level contributions for predictions. In this project, we selected **LIME** (Local Interpretable Model-agnostic Explanations) it can be referred to as an adaptable and intuitive technique. LIME is model agnostic, meaning it can be used for black-box models of different kinds. It generates locally faithful explanations by perturbing the input image. This can tell us which portions of the image become influential at prediction. Such prescient interpretation and smooth integration make it especially good at giving transparent interpretations into the model's complex decisions in human terms.

Local Interpretable Model-agnostic Explanations (LIME) is one of the most popular **Explainable AI (XAI)** methods used for explaining the working of machine learning and deep learning models [23].

When applying LIME to images, it explains the decision of the model by highlighting which parts of the image (called superpixels) are most responsible for the prediction.

The process proceeds as follows:

- Segmentation of the Image into Superpixels
  Images are patched up along adjacent superpixels using methods like SLIC (Simple Linear Iterative Clustering). These patches are such that pixels within the superpixels are visually rather similar.
- Generating Perturbations
  Hundreds of modified samples of the image are created by LIME, whereby some superpixels are shut off (turned to gray or blurred), while others are maintained active. All perturbed samples are sent through the original model to see the change in prediction.
- Building a Local Surrogate Model
  This involves training a simple interpretable model on the perturbed samples, generally called a linear model. Here, the features were binary indicators showing the presence or absence of the superpixels.
  The concept that if hiding a particular region will make the model change its mind, then it is important.
- Interpreting the Output
  The weights from the surrogate model indicate how much each superpixel was responsible for the model prediction. These weights are used in highlighting regions in the original image that were most responsible.

Results and Discussion

The incorrect values in the categorical columns in Table 2 were fixed to have clear values and ensure data consistency. The dataset was checked for duplicate records to ensure data quality and remove redundancy. the results of data preprocessing, visualization, and model evaluation are shown and explained in this section. The data was cleaned and prepared to make it better for the models. Visualization helped find important patterns, and model evaluation showed how well different models worked, helping to choose the best one.

The data was unevenly distributed, with a longer tail on the right side. To fix this imbalance and make the data more normal, a log transformation was applied to these specific measurements as shown in Figure 14. A log transformation helps compress the range of data, reduces the impact of extreme values, and is especially suitable for data with positive skewness. Not all features needed this transformation, but it helped smooth out the ones that did, making the data more stable and easier to work with for our analysis.

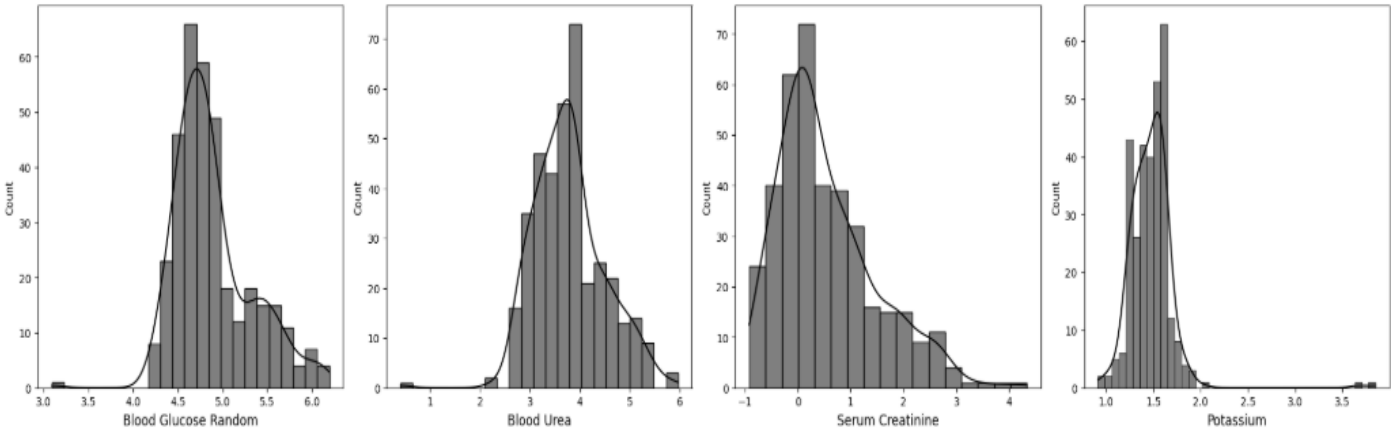

Figure 14. Features After Transformation

As we see in Figure 15, it was found that the top four features positively correlated with the target were Albumin (0.627), Serum Creatinine (0.622), Blood Glucose Random (0.433), and Blood Urea (0.405). Conversely, the top four negatively correlated features were Red Blood Cell Count (-0.699), Specific Gravity (-0.732), Packed Cell Volume (-0.741), and Hemoglobin (-0.769). These correlations helped in identifying which features had the strongest relationships with the target, both in positive and negative directions.

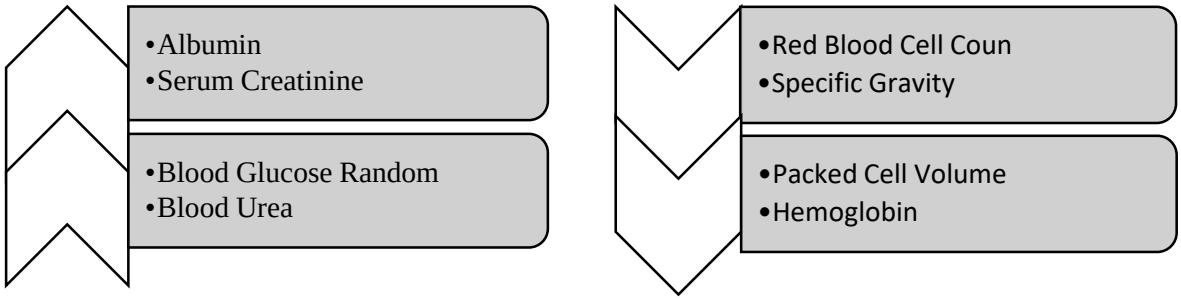

Figure 15. Positive – Negative Correlated with Target Class

5.1 Baseline Model Evaluation

The journey began with the simplest approach: training multiple models using only **10 folds** cross-validation. This gave a solid baseline, showing how well different algorithms performed with the pre-processed data. After training, the first set of performance was created in Table 4:

Table 4 : Baseline Model Evaluation

Classifier	*Precision*	*Recall*	*F1 Score*	*Training Accuracy*	*Testing Accuracy*
Decision tree	0.912	0.912	0.912	**1.00**	**0.890**
Logistic Regression	0.930	0.905	0.918	**0.917**	**0.898**
Naïve Bayesian	0.946	0.895	0.920	**0.901**	**0.903**
XGBoost	0.945	0.942	0.944	**1.00**	**0.930**
Random forest	0.937	0.966	0.951	**1.00**	**0.938**
AdaBoost	0.949	0.942	0.945	**0.955**	**0.932**
CatBoost	0.943	0.952	0.948	**1.00**	**0.934**
Gradient Boosting	0.942	0.942	0.942	**1.00**	**0.928**
Support Vector Machine	0.625	1.00	0.769	**0.625**	**0.625**
K-nearest neighbor	0.735	0.730	0.733	**0.743**	**0.667**

In F1 score, the highest results of 0.948 - 0.951 were obtained by Random Forest, CatBoost, and Gradient Boosting, which means their performance was better than the rest. XGBoost and ADA boost also performed fairly well with grades of 0.944 and 0.945. Logistic Regression follows closely with 0.918 while Naive Bayes competes too with 0.920.

SVM and KNN were the least performers. Their test results were the lowest. KNN had the least test score accuracy of 0.667. While SVM had an F1 score of only 0.769. The most affected algorithms due to overfitting was **tree based** algorithms, especially decision Trees, which is attributed to Random Forest, CatBoost, and Gradient Boosting, and these other tree based models; which resulted in an incredible 1.00 training accuracy, meaning that they need a lot of **regularization** or **feature selection** to bring this overfitting under control.

5.1.2 Optimizing Model Performance (Original Data Only)

Given the imbalanced dataset, where certain classes had an exorbitant amount of samples compared to others, SMOTE was implemented to artificially create data points for the underrepresented classes. This allowed for a new experiment to be performed, wherein Cross-Validation and SMOTE were used to train the models in order to determine if the larger sample size of the minority class would lead to better accuracy in predictions.

IDA (Linear Discriminant Analysis) was also evaluated as a new approach to dimension reduction. It aimed to find out if IDA could produce a more optimal set of features that would improve the classification metrics. Models were trained afterwards with IDA, Cross Validation, and the most relevant features that were enhanced via Synthetic Minority Over-Sampling Technique.Given this was the least simplistic transformation, an additional round of Hyperparameter tuning using **Randomized Search** was necessary in order to allow for optimal model performance, Table 5 shows the optimized model evaluation.

Table 5 : Optimizing Model Evaluation

Classifier	*Precision*	*Recall*	*F1 Score*	*Training Accuracy*	*Testing Accuracy*
Decision tree	0.988	0.996	0.992	**1.00**	**0.992**
Logistic Regression	0.996	0.992	0.994	**0.996**	**0.994**
Naïve Bayesian	0.996	0.996	0.996	**0.994**	**0.996**
XGBoost	0.992	0.996	0.994	**0.996**	**0.994**
Random forest	0.988	0.996	0.992	**1.00**	**0.992**
AdaBoost	0.988	0.996	0.992	**1.00**	**0.992**
CatBoost	0.992	0.996	0.994	**0.996**	**0.994**
Gradient Boosting	0.988	0.996	0.992	**1.00**	**0.992**
Support Vector Machine	0.996	0.996	0.996	**0.996**	**0.996**
K-nearest neighbor	0.996	0.996	0.996	**0.994**	**0.996**

5.1.3 Optimizing Model Performance (With Synthetic Data)

In the case, SDV (Synthetic Data Vault) was utilized to generate some extra data points, owing to the small size of the original dataset. The **SingleTablePreset** model was used to learn patterns from this data, producing 100 synthetic records. These new data points were pooled back into the model to allow further improvements in learning without distorting the underlying data correctiveness. The model's performance enhancement through synthetic data generation, hence negating the need for any further real-world data, reinforces the merit of applying synthetic data generation in machine learning.

This step also worked rather well in **reducing overfitting** due to improved data diversity for better generalization to unseen data. The model's performance was improved purely from synthetic data without real-world data as shown in Table 6, proving the usefulness of synthesizing data in.

Table 6 : Optimizing Model Evaluation (Synthetic Data)

Classifier	***Precision***	***Recall***	***F1 Score***	***Training Accuracy***	***Testing Accuracy***
Decision tree	0.952	0.949	0.951	**1.00**	**0.951**
Logistic Regression	0.952	0.946	0.949	**0.946**	**0.949**
Naïve Bayesian	0.952	0.949	0.951	**0.946**	**0.951**
XGBoost	0.948	0.936	0.942	**0.959**	**0.942**
Random forest	0.952	0.936	0.943	**1.00**	**0.94**
AdaBoost	0.938	0.969	0.953	**0.956**	**0.952**
CatBoost	0.940	0.959	0.950	**0.959**	**0.949**
Gradient Boosting	0.943	0.9629	0.953	**0.994**	**0.952**
Support Vector Machine	0.940	0.956	0.948	**0.942**	**0.947**
K-nearest neighbor	0.946	0.959	0.953	**0.957**	**0.952**

5.1.4 Top 5 Best-Performing Models

How well did the models perform? Given below in Table 7 are the top five performing machine learning models for predicting CKD. These have been compared in terms of **precision**, **recall**, **F1 score**, and **accuracy** for validation of the one giving the most accurate predictions.

Of all the models, **Naïve Bayes** had the highest balance between precision and recall, as measured by the F1 score of 0.951. Meanwhile, it can be said that AdaBoost and KNN had the best scores on recall (0.969 and 0.959, respectively) but would have missed only a few CKD cases. Logistic Regression and Naïve Bayes had the best accuracy with respect to precision (0.952) with minimum false positives.

The models had a consistent **training-test accuracy profile**, indicating that they did not just memorize the data but genuinely learned from it.

Table 7 : Best models Evaluation

Classifier	***Precision***	***Recall***	***F1 Score***	***Training Accuracy***	***Testing Accuracy***
Logistic Regression	0.952	0.946	0.949	**0.946**	**0.949**
Naïve Bayesian	0.952	0.949	0.951	**0.946**	**0.951**
AdaBoost	0.938	0.969	0.953	**0.956**	**0.952**
Support Vector Machine	0.940	0.956	0.948	**0.942**	**0.947**
K-nearest neighbor	0.946	0.959	0.953	**0.957**	**0.952**

What were the optimal Hyperparameter for every model? For highly optimized working in this regard, we related the Hyperparameter adjustment to the best performance of all these models as shown in Table 8. Logistic Regression was considered best with the 'liblinear' solver' and L2 regularization in order to avoid overfitting. For AdaBoost, best performance was achieved with the default settings, emphasizing 50 estimators and learning rate of 0.1 for the best boosting. Support Vector Machine (SVM) was best with RBF kernel and automatic gamma handling for complex patterns.

Table 8 : Best Parameters

Classifier	***Best Parameters***
Logistic Regression	'solver': 'liblinear', 'penalty': 'l2', 'C': 10
AdaBoost	'n_estimators': 50, 'learning_rate': 0.1, 'algorithm': 'SAMME'
Support Vector Machine	'kernel': 'rbf', 'gamma': 'auto', 'degree': 4, 'C': 0.01
K-nearest neighbor	'weights': 'uniform', 'n_neighbors': 5, 'metric': 'manhattan'

5.1.5 XAI using Feature Importance and SHAP

These results are for the best model (AdaBoost). As we see in Figure 16, **positive contributions** (red) increase the prediction probability (moves right), while negative contributions (blue) decrease it (moves left). the plot starts at the expected value E[f(X)], which is 0.59 in this case, E[f(X)] "the average model output (log-odds or probability) before considering specific feature values.

Each row represents a feature value and its impact on the prediction, the length of the bars shows the magnitude of impact. Most Influential Features:

- **Diabetes Mellitus (+0.24)** Strongly increased the prediction.
- **Hypertension (+0.22)** Also pushed the model towards the predicted outcome.
- **Hemoglobin (-0.17)** Reduced the prediction probability.
- **Blood Urea (+0.17)** Increased the likelihood of the outcome.

The final value (black dashed line) represents the model's output for this instance f(x)=1.

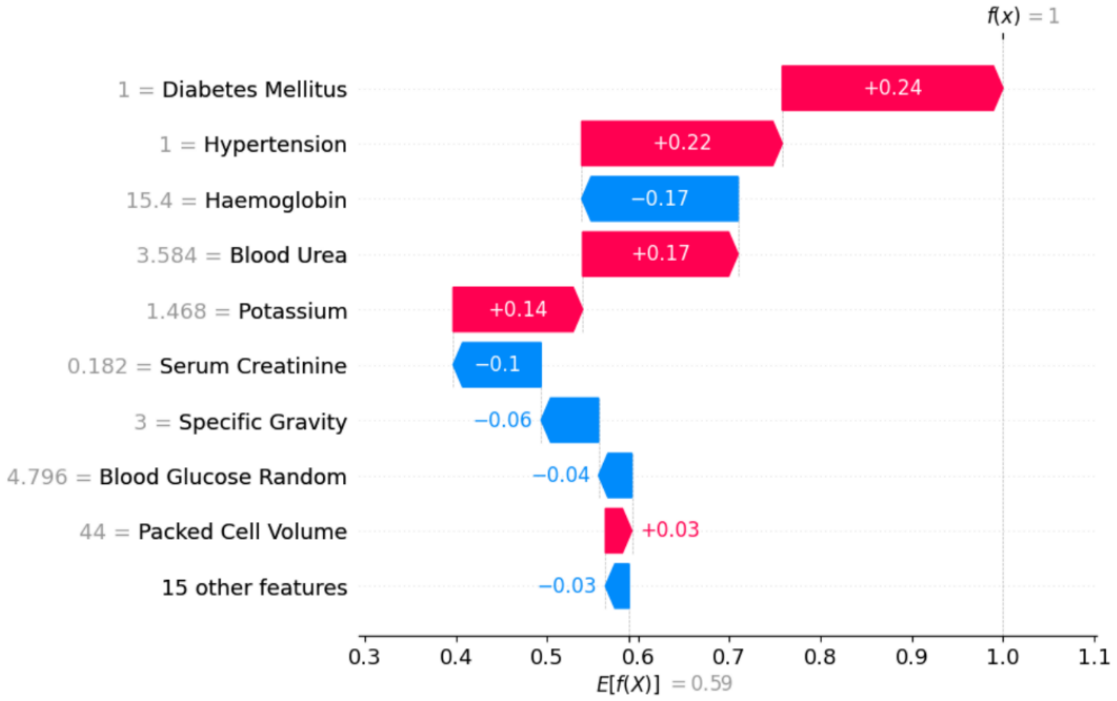

Figure 16. SHAP Waterfall plot

As we see in Figure 17, **X-Axis** SHAP Values (Impact on Model Output), negative SHAP values (left side) indicate that the feature reduces the prediction, positive SHAP values (right side) indicate that the feature increases the prediction. **Y-Axis** lists the features ranked in order of importance.

Color Scale The color bar indicates the feature's actual value, (Red/Pink) High feature values, (Blue) Low feature values. **Spread of Points** A widespread means that the feature has varying effects on different predictions, a narrow spread suggests a consistent impact across samples.

For this plot: Hemoglobin, Serum Creatinine, and Diabetes Mellitus are the most influential features, the model output for diabetes mellitus is pushed downward by lower values (blue) and upward by higher values (red), the model forecast for hemoglobin is increased by lower values (blue) and decreased by higher ones (red), Red blood cells and sodium have little effect because their SHAP values are nearly null.

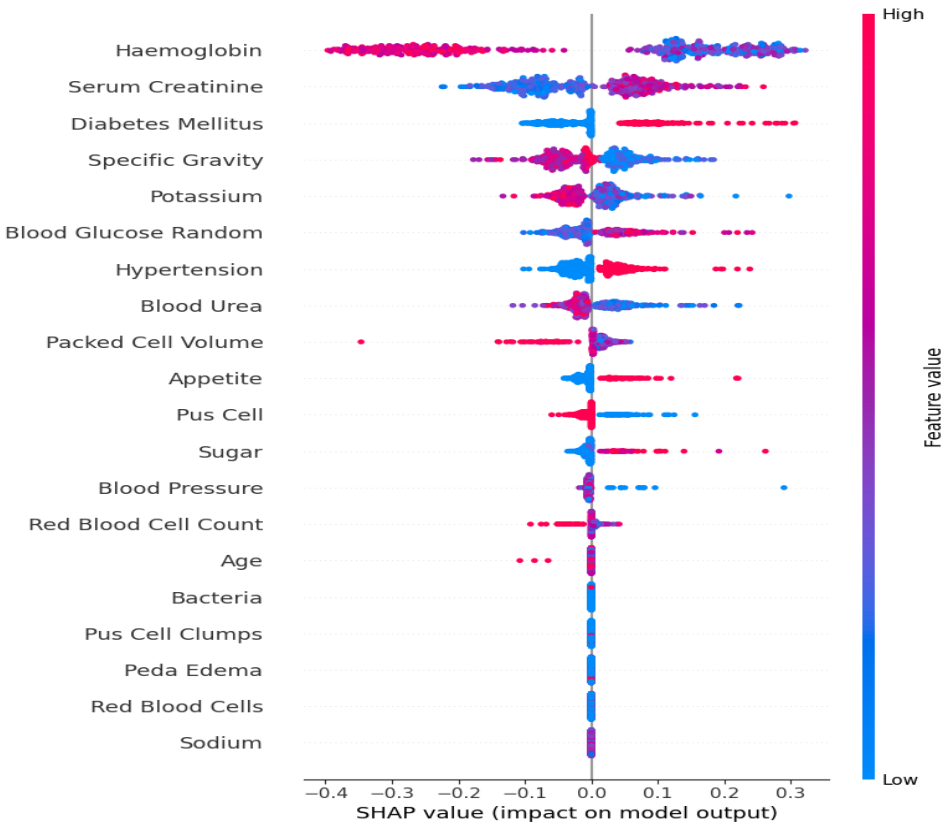

Figure 17. SHAP Summary plot

Figure 18 shows the most influential features:

- **Hemoglobin** It has a mean absolute SHAP value of +0.22, this suggests that hemoglobin is a crucial factor in determining patient outcomes.
- **Serum Creatinine** It has a mean absolute SHAP value of +0.08, Serum creatinine's significance in the model is consistent with its frequent use as a kidney function indicator.
- **Diabetes Mellitus** It has a mean absolute SHAP value of +0.06, has a moderate impact on outcome prediction, given that diabetes is a major risk factor for kidney problems and cardiovascular disorders, this is consistent with medical understanding.
- **Specific Gravity** It has a mean absolute SHAP value of +0.05, indicates that variations in urine concentration have a moderate impact on the model's decision-making, which may be important when identifying metabolic disorders, kidney illness, or dehydration.

The sum of 15 other features contributes a combined SHAP value of +0.06, this indicates that although these other factors do play a role in the prediction, their respective contributions are significantly less, although these features may still be considered by the model in certain situations, their overall impact is minimal.

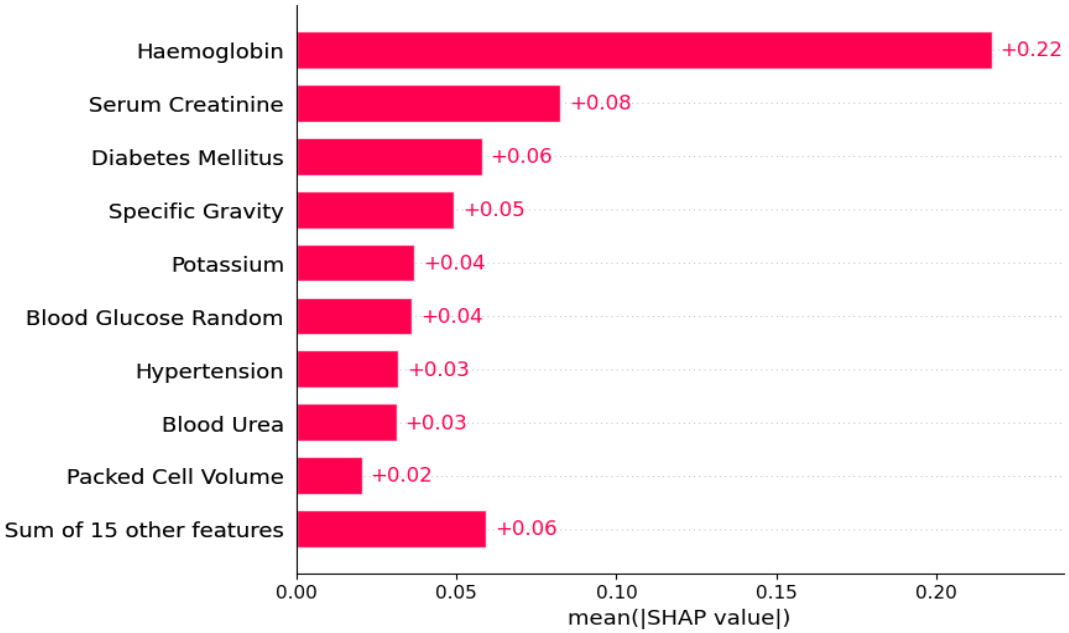

Figure 18. SHAP Summary bar chart

5.2 Transfer Models Evaluation

In this section, the training and evaluation results of deep learning models for kidney CT image classification are presented. The models were trained to classify images with respect to four categories: Normal, Cyst, Stone, and Tumor. Performance metrics that include training and testing accuracy, loss, precision, recall, and F1-score were employed. Confusion matrices were computed to visualize class-wise predictions, and LIME was employed to highlight important image regions that influenced the model's decisions. Key results and comparisons with respect to model architectures are discussed in the following subsections.

5.2.1 Model Performance Metrics

To assess the performance of the CT image classification systems, some important metrics like accuracy, precision, recall, and F1-score were used and evaluated for each of the four classes: Normal, Cyst, Stone, and Tumor. Models were also compared using confusion matrices that allow visualization on the performance classification and identifying common misclassifications. All these metrics, in overall terms, were constructive in providing a comprehensive understanding of models' abilities to differentiate between different types of kidney diseases, with specific focus on recall and precision due to clinical implications of having fewer false negatives and more false positives as shown in Table 9.

Table 9 : Transfer Models Metrics

Classifier	***Class***	***Precision***	***Recall***	***F1 Score***
EfficientNet	Cyst	0.98	0.98	0.98
	Normal	0.99	0.94	0.97
	Stone	0.78	0.97	0.87
	Tumor	0.99	0.96	0.97
ResNet	Cyst	0.94	0.99	0.96
	Normal	0.99	0.94	0.96
	Stone	0.98	0.83	0.89
	Tumor	0.88	0.97	0.93
VGG19	Cyst	0.97	1.00	0.98
	Normal	0.99	0.94	0.97
	Stone	0.89	0.97	0.92
	Tumor	0.95	0.96	0.95

5.2.2 Model Comparison

The kidney CT image classification models that were tested were **EfficientNet**, **ResNet**, and **VGG19**. They were evaluated as per the key metrics: precision, recall, and F1-score for each of the four classes, namely Normal, Cyst, Stone, and Tumor. **VGG19** achieved the most uniform and robust performance across all classes and almost always gave the highest score in nearly every category. The excellent results were recorded for the classification of Cyst (F1 = 0.99), Normal (F1 = 0.98), and Tumor (F1 = 0.98), while there was a strong performance for Stone (F1 = 0.94). EfficientNet was also good, especially with Cyst (F1 = 0.98) and Tumor (F1 = 0.97), but the accuracy in the Stone class was much lower (0.78), which reduced the F1 score a little bit to (0.87).

On the other hand, ResNet was quite effective in Cyst (0.99) and Tumor (0.98) recall but lower precision for Tumor (0.82) and a significantly lower recall for Stone (0.76) weakened its performance in that class (F1 = 0.85). All these metrics give a clear picture of the classification capabilities of the models, especially recall and precision concerning the clinical importance of reducing false negatives. Overall, VGG19 was found to be the most suitable for accurate classification of kidney CT images in this study.

From a metric perspective, **VGG19 was considered the best model** to classify kidney CT images. The F1-scores obtained from VGG-19 were consistently the highest across the four classes, thus signaling good precision and recall, especially for Cyst, Normal, and Tumor categories. The ability of VGG-19 to properly predict accurately in every instance provides assurance of its reliability in clinical practice of medical imaging, qualifying it substantially as a model aiding diagnosis for kidney diseases in the current research.

5.2.3 Training and Validation Details

Investigation of hyperparameters for efficiency and stability in models was limited to finding those crucial ones for the training and validation of classification models on kidney CT images. The initial training employed a **batch size of 32** for **20 epochs** with **a learning rate of 0.0001**, which balanced computational time and model learning. After this first training stage, it fine-tuned the model performance.For example, a reduced **learning rate of 0.00001** was chosen during the **next 20 epochs** of training. Fine-tuning at lower learning rate settings meant model weight adjustments are more gradual, thus allowing models to discriminate even further without overfitting. Such training over two phases, first with learning and ending up with fine-tuning, led to more reliable and accurate extraction of features, especially in complex areas of application such as medical image classification.

5.2.4 Test Set Results

As we see in Figures 19,20,21 Confusion matrices were obtained from every model in testing EfficientNet, ResNet and VGG19 to evaluate the classification performance at the testing stage. The matrices examined in great detail accurate and wrong results among the singular Classes-Normal, Cyst, Stone, and Tumor.

The visual pattern of between-misclassified classes in the confusion matrices brought certain specific classes where misclassification was likely to happen. The model-to-model analysis, therefore, provided vital information on how well they discern the various types of kidney diseases from each other. In particular, patterns of confusion among classes, which lessened the clinical applicability scope of the models, were also highlighted.

Moreover, accuracy and loss plots for both the training and test sets were used to evaluate the training process in addition to confusion matrices. Accuracy curves were used to show how well the model learned and generalized, while loss curves were used to monitor optimization over 20 training epochs and 20 fine-tuning epochs, in which all these were found to be very important in discovering overfitting, under fitting, or stable convergence. The visual juxtaposition of training and test performance complete the picture for consistency in learning, giving one the assurance that the end model is reliable and robust, especially in the case of VGG19, where the training and test metrics demonstrated strong alignment.

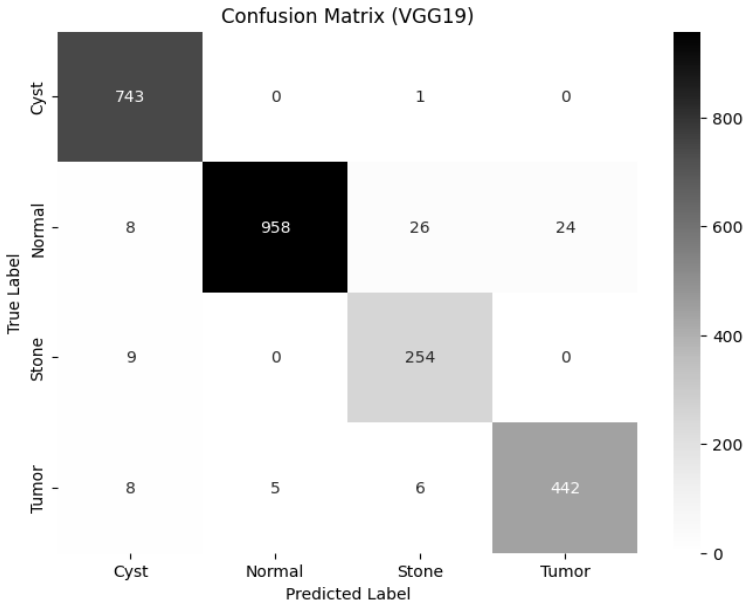

Figure 19. Confusion Matrix (VGG19)

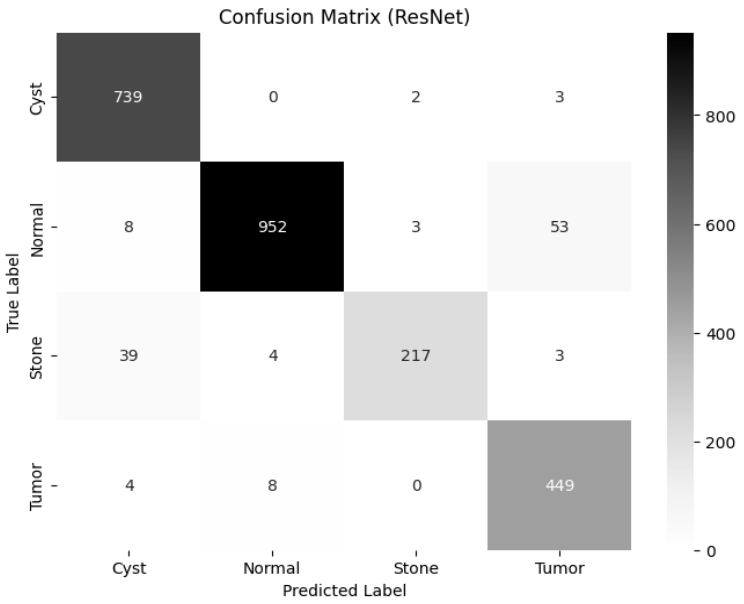

Figure 20. Confusion Matrix (ResNet)

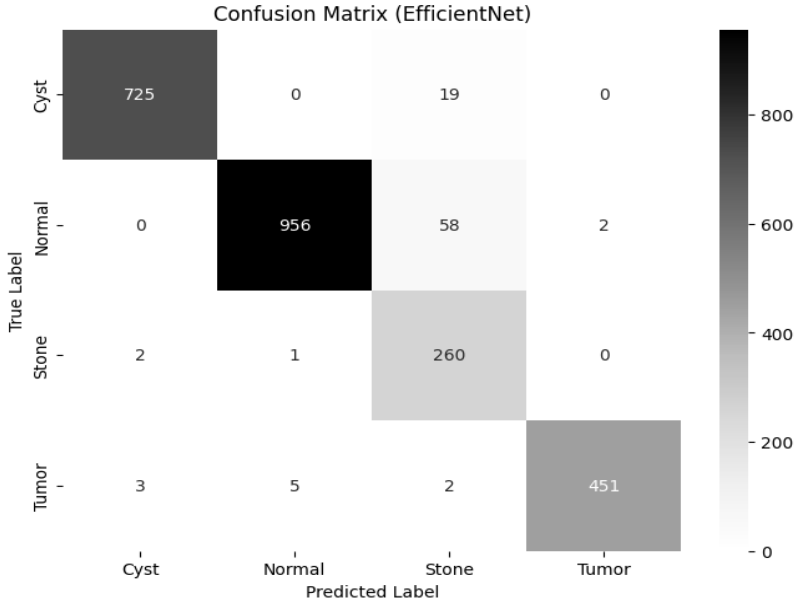

Figure 21. Confusion Matrix (EfficentNet V2B0)

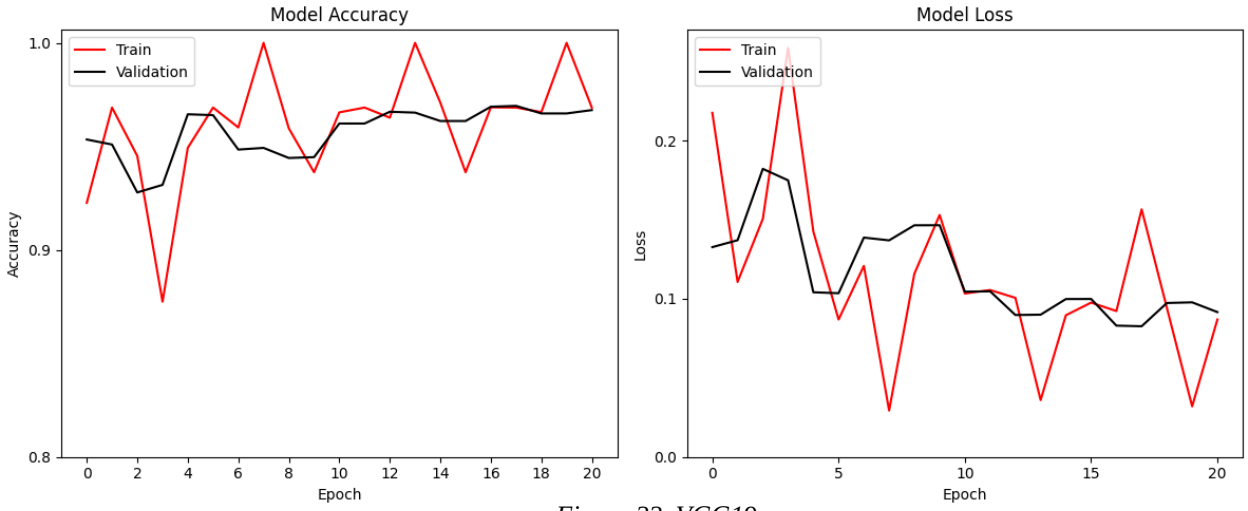

Figure 22. VGG19

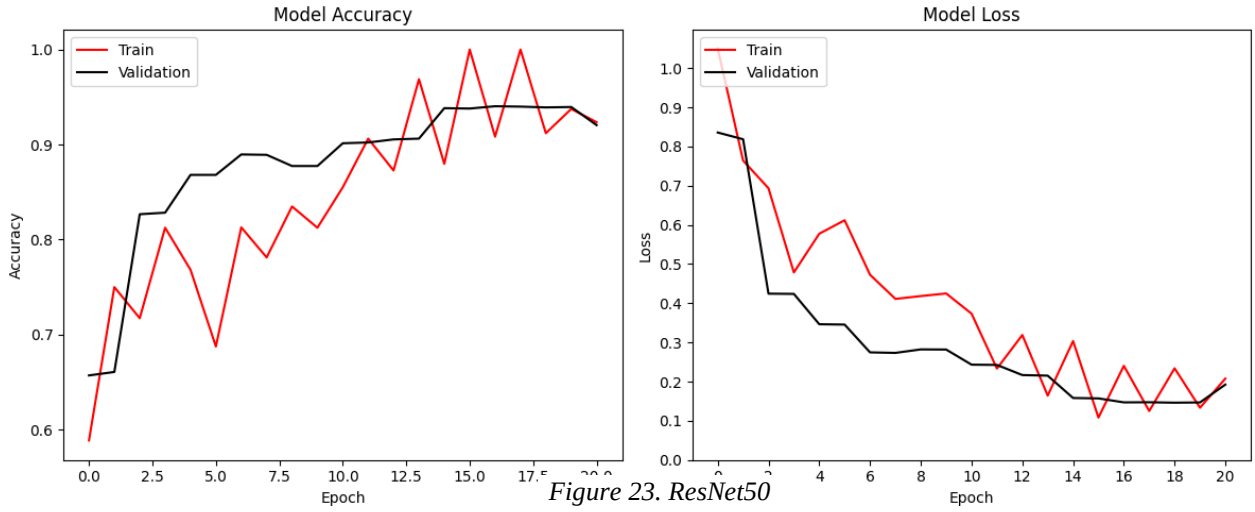

Figure 23. ResNet50

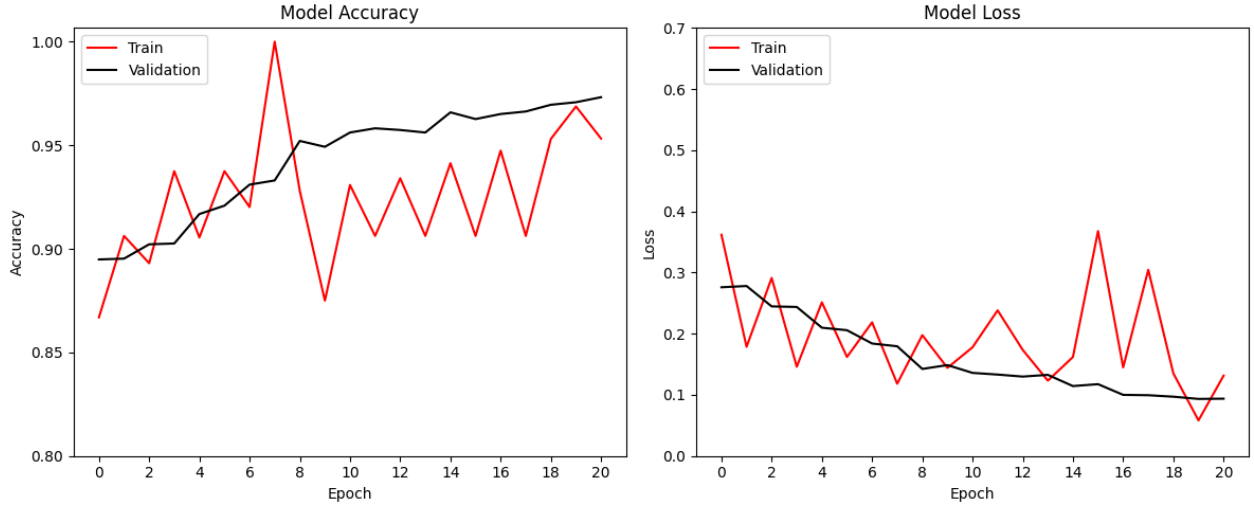

Figure 24. EfficentNetV2B0

Table 10: Models Training and Testing Loss

Classifier	*Training Loss*	*Testing Loss*
EfficientNet	**0.1034**	**0.1105**
ResNet	**0.1411**	**0.1422**
VGG19	**0.0901**	**0.0942**

5.2.6 Overall Summary

Herein the evaluation of the kidney CT scan images' classification models: EfficientNet, ResNet and VGG19. Performance of these models is assessed using key metrics precision, recall, F1-score and confusion matrices. Among the three, it was discovered that VGG19 was the best-performing model in the four categories tested: Normal, Tumor, Stone and Cyst. Very high F1-scores have been recorded for the classes Cyst (0.99) Normal (0.98) and Tumor (0.98) indicating that the study's best clinically reliable model is VGG19.

Although EfficientNet and ResNet proved quite successful, particularly in differentiating between Cyst and Tumor cases, their performance on the Stone category was less impressive, particularly in ResNet which had a lower recall. Visual evidence of these findings was given by the confusion matrices where clear misclassifications could be easily found and comparative insights between models were illustrated.

Training was carried out in Two Phases: The Primary Learning Phase and the Fine Tuning Phase in which the learning rate was decreased. This would allow more gradual update of the worsening weights, which helps give an advantage over the feature learning while preventing overfitting. The accuracy and loss curves will be used in monitoring the model on convergence and generalization during the training process, hence sharp similarity between performance in training and performance in testing exists, particularly in VGG19.

VGG19 was, however, the most effective model for classifying kidney CT images in this research, while in cases considered clinically relevant classes, it performed consistently well over the entire study, even in edge cases.

5.2.5 Model Interpretability (XAI Results)

For assessing interpretability aspects of the proposed classification model, LIME (Local Interpretable Model-agnostic Explanations) was applied in order to highlight the most prominently influential regions of the input CT images towards the ultimate decision.

After predictions for representative images were made for each of the classes Normal, Cyst, Stone, Tumor, LIME explanations were computed and analyzed. For each one of the images, a heatmap was produced, visualizing the super pixels that had the most influence on the decision of the model.

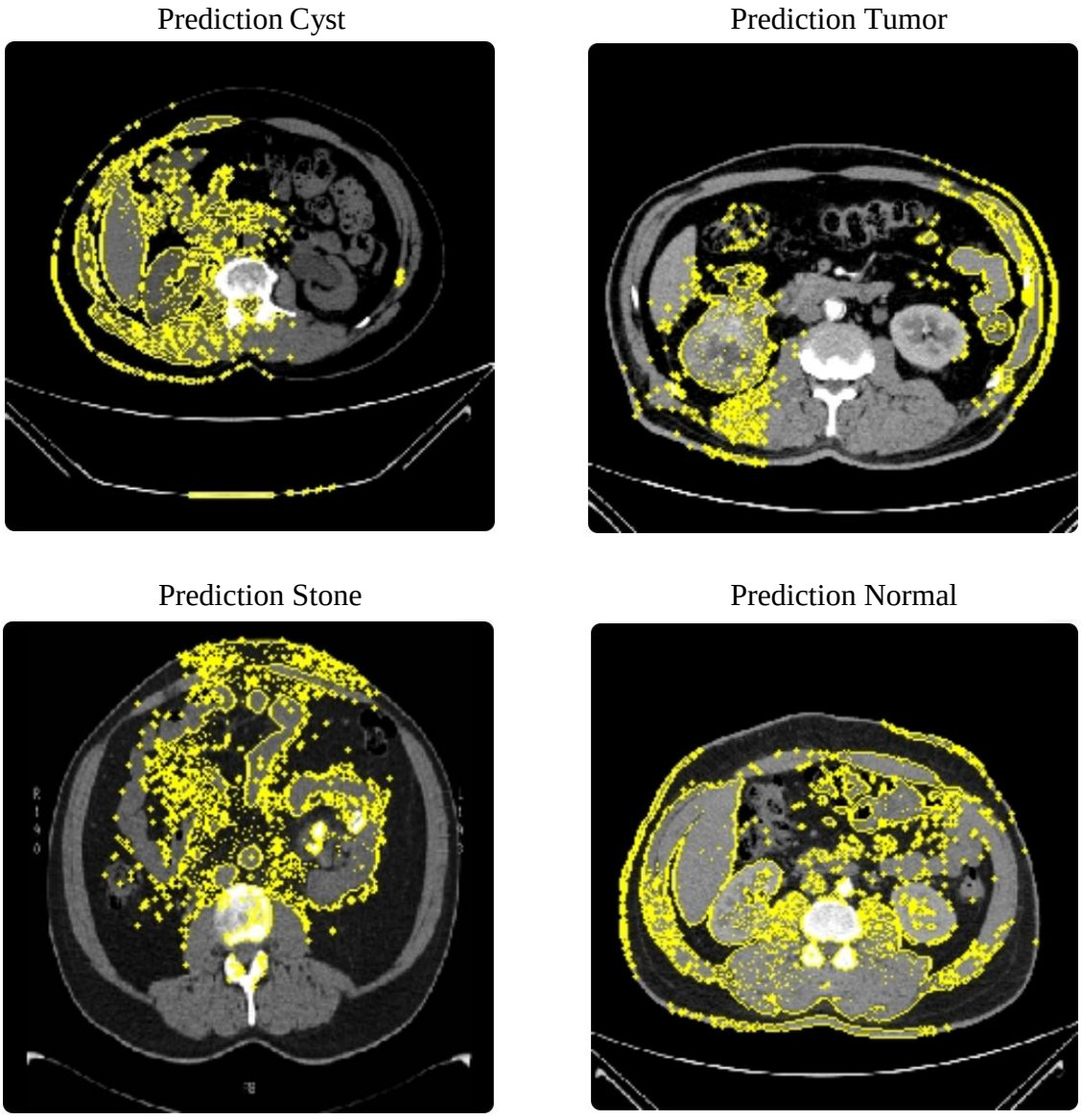

Figure 25. LIME Explanation

As shown in Figure 30, a Tumor prediction for the highlighted regions would happen to overlap with localized dense structures in the kidney area that visually matched abnormal masses. The Stone instances had the model reliably focusing on high-intensity areas corresponding to calcifications. Likewise, for the instances classified as Cyst, the LIME-marked areas were related to the fluid-filled area; whereas in the Normal instances, there were settings with little or no focusing areas at all, which implies that abnormalities were absent.

The LIME-backed explanations from the evaluations thus helped in confirming relevant clinical features across all classes, implying that the decision making of the model did not consider irrelevant background patterns. The interpretability results essentially justified the model in the sense that the decision-making processes were indeed in accordance with expected anatomical markers.

These results imply LIME was not just providing a window into how the model works but was giving a window on the real-life health care implications of that model and whether its predictions could be verified. Meaningful regions from a medical perspective were consistently focused on in different examples, which bolstered confidence in its application for real-world diagnostic support.

Model Deployment

Machine learning models are possessive of power and efficiency however, standing alone, they do not offer too much value as each model has its strengths. Once a machine learning model is made, it has to be placed in the real-world environment [24]. Tools are available for data scientists to deploy machine learning models easily nowadays. Model building and deployment are iterative processes, consisting of various interlinked steps, which require continuous iteration and refinement due to insights gained from other stages in the process. Therefore, recommend deploying the CKD classification model as a web application because of its strengths including accessibility, ease of use, and scalability as shown in Figure 26. The web interface handles structured data, allowing users to input relevant patient details and will allow CT image analysis through the model without the user needing technical expertise or operating in a very complex computational environment. The web application has the advantage of one of the main aspects of being an easily accessible process. CT images can be uploaded by doctors, researchers, and other medical professionals, and they can get predictions without installing any software or running any code. Moreover, usability is greatly enhanced due to the interface being user-friendly. Users could work within a simple, self-explanatory interface with very little technical knowledge, rather than relying on a command-line tool or programming environment [24].

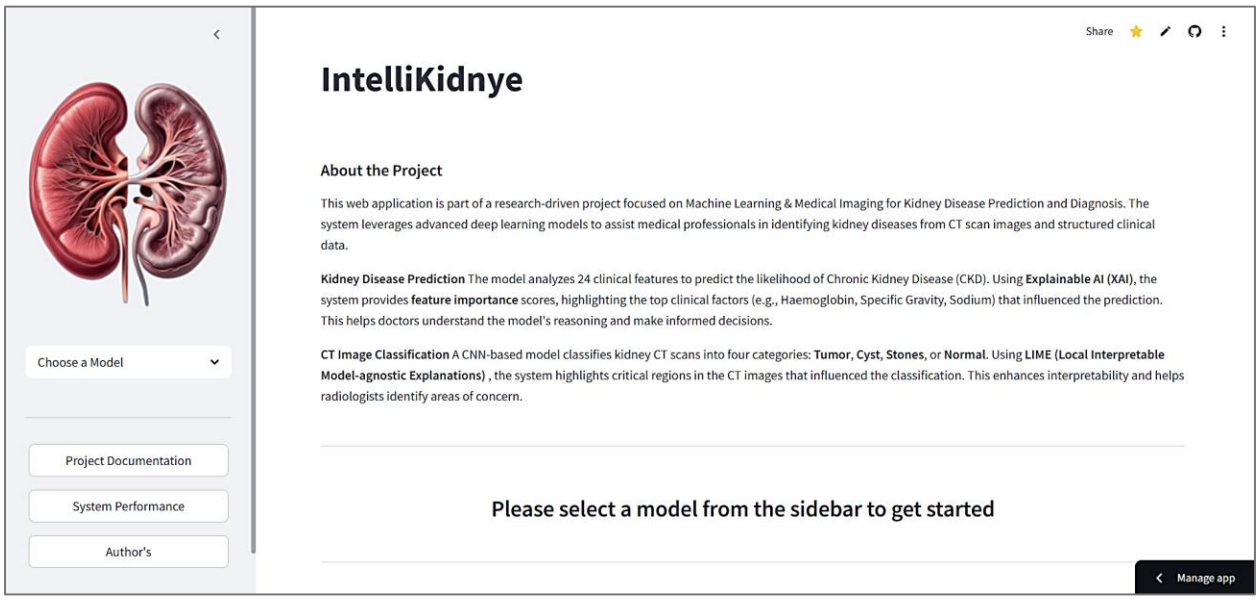

Figure 26. IntelliKidnye Web Application

6.1 Technology Stack

Streamlit Framework One of the frameworks that can be considered for deploying machine learning models is the Streamlit framework. It's an open-source framework built for developing web apps for data science and machine learning. It allows code to be written as easily as Python code. Streamlit is one of the best frameworks in terms of user-friendliness. According to the founders, it is also the fastest way of developing and sharing a data app. For enterprising ML enthusiasts who don't want to spend time developing web apps for deploying their models, Streamlit will definitely worth, as it is able to constantly operate with the data in the model [25].

Frontend (User Interface): Streamlit A Python interactive graphical user interface enabling web application building. The easy-to-use interface allows the uploading of CT images and the display of predictions, freeing users from front-end development skills.

Backend (Model Inference and Processing): Python (Streamlit) The back end of the application runs through Python and Streamlit. Model inference as well as real-time processing of user input will be handled by Streamlit in a way that ensures the best experience for users interacting with the kidney disease classification model.

Databases: **Neon PostgreSQL Platform** for storage of structured data includes patient particulars and model predictions. Neon is a completely managed, serverless PostgreSQL database platform meant to provide high scalability, availability, and efficient storage management. Some of its highly touted features include automatic scaling, built-in recovery types, and a modern cloud-native architecture. Using Neon will assure reliable and high-quality structural data for modeling purposes.

MongoDB Atlas will be used for unstructured data storing, specifically CT images. This flexibility in storing as well as retrieving image data as per the requirements of the image classification model is the specialty of a NoSQL database like MongoDB. This technology stack is going to ensure that the web application is efficient and easily scalable, handling structured and unstructured data quite well.

6.2 End-to-End Workflow for Web Application

The web application which has been developed in Streamlit forms a typical workflow in kidney disease classification. This involves an entry of structured data into the CKD Model wherein it processes the data in order to ascertain the presence of chronic kidney disease. If CKD is detected, the user is supposed to upload a CT image which gets pre-processed before it enters these all CT Image Model analyses to decide into one of these states (Normal, Tumor, Cyst, Stones).

The most interpretable of all the classifications received are the visual feedback produced by the XAI Model. These results are reported together with their corresponding classification results. The outcome display provides clear predictions while the outcome dashboard allows users to review multiple cases efficiently.

Data management involves structured storage of patient data and predictions in an SQL database, while CT images are stored in a No-SQL database (MongoDB) for easy retrieval. The whole system integrates seamlessly with both these databases so that stored input data is securely managed. It further enables cloud deployment, thus making the model available for interaction at anytime from anywhere as shown in Figure 27.

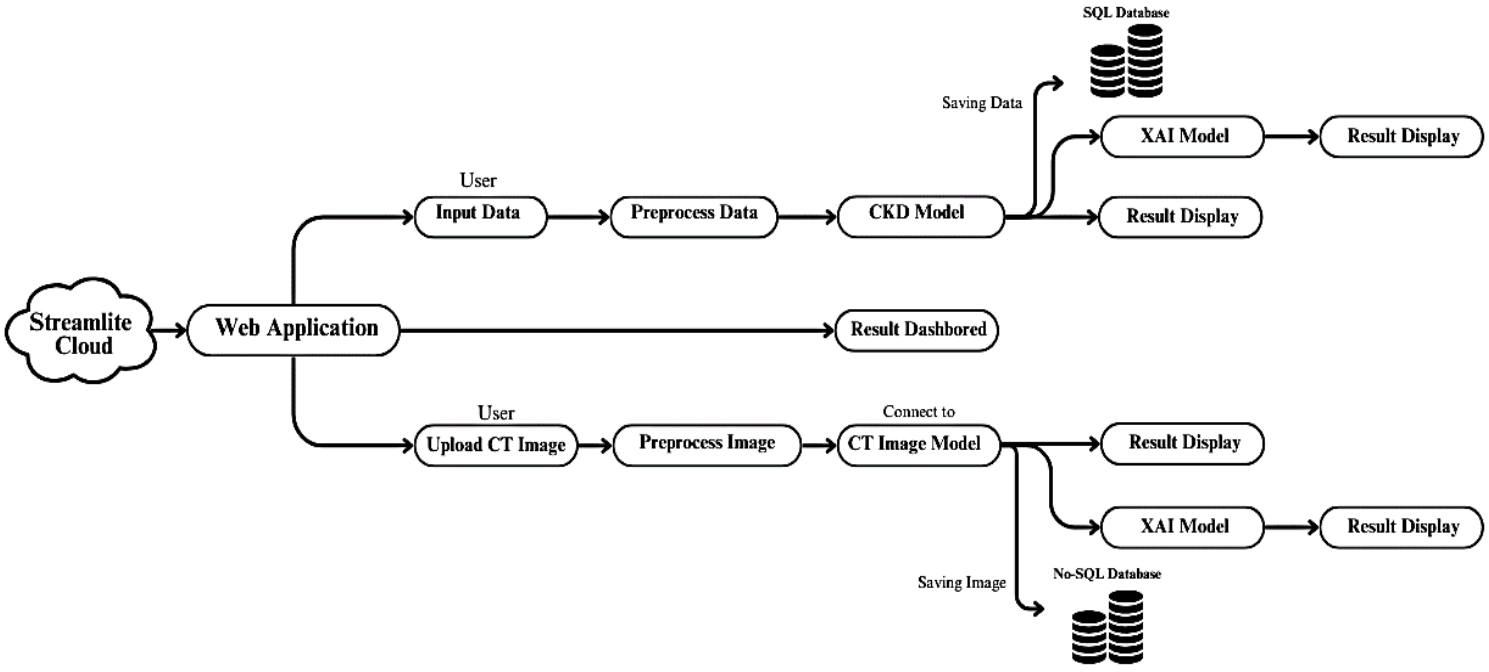

Figure 27. User Flow in the Web Application

6.3 Databases Connection

The application has integrated both relational and non-relational databases to optimally store, manage, and retrieve both structured and unstructured data. These databases are fundamental for storing user-input data like patient details, predictions, and uploaded CT images. As we see in Figure 28, such stored data aid in model retraining and performance enhancement, thus catalyzing the gradual improvement of classification accuracy over time. In this way, the use of both types of databases leads the system to provide safe, scalable, and efficient management of the data.

Figure 28. Upgrades Models Process

6.3.1 Relational Database

As we see in Figure 29, patient information, disease prediction is stored using a relational database- the **Neon PostgreSQL Platform**. Neon, in a representative schema of structured table, ensures data integrity and consistency above efficient query mechanisms. These data in a structured form allow such easy organization of patient records for the trend analysis, model evaluation, and medical research as needed. Moreover, the database acts as a monitoring tool for the performance of the model by keeping a well-structured historical record of predictions. It is a consolidated record of patient inputs as well as model outputs, thereby providing statistical assessments for making biases and false classification situations visible. This data-driven tactic can afford any adjusting to enhance the model's reliability and correctness with the passage of time.

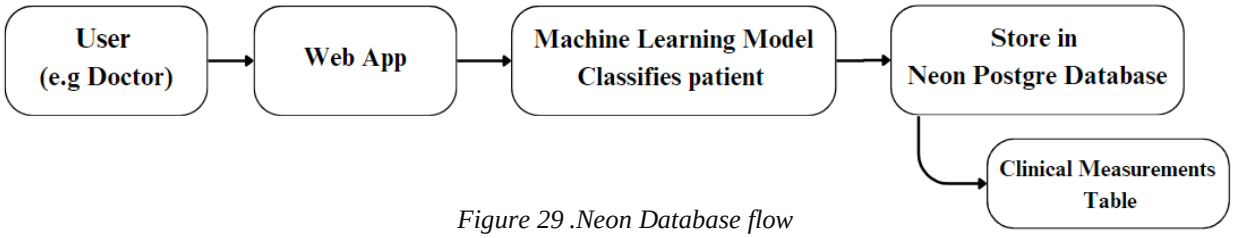

Figure 29 .Neon Database flow

6.3.2 Non-Relational Database

As we see in Figure 30, unstructured data, especially CT images uploaded by users, are placed in non-relational database (MongoDB Atlas). In contrast to the traditional relational databases, MongoDB Atlas is a cloud-based database that retains a scalable and flexible architecture allowing schema-less design, which is effective in handling large-scale medical imaging datasets [26]. This kind of flexibility allows for CT images to be easily stored and retrieved considering differences in format and size.

CT images stored in MongoDB Atlas serve as valuable reference images playing a pivotal role in the retraining and fine-tuning of the model. Furthermore, as a part of the model update, images that are new and acquired with time are added to the existing dataset so that the classification model remains up to speed with current trends, changes in image quality, and the variety of clinical presentations of the disease. Continuing to learn, therefore, means improving the accuracy of the model and enhancing robustness in real-world medical application

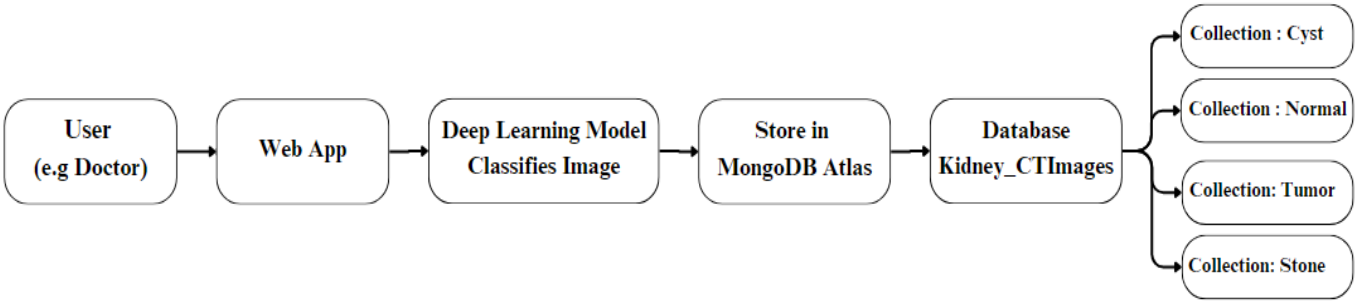

Figure 30. Non-Relational Database Flow

For Model Enhancement a dual-database architecture is deployed for optimizing model performance. Structured data are stored in a relational database (Neon PostgreSQL) that supports working on evaluating model predictions and structured query processing; conversely, unstructured data-pushed into the MongoDB Atlas on the other end-allow continuous retraining and fine-tuning of deep learning models.

Thus, the exposure of structured data to unstructured data fosters a feedback system that allows the kidney disease classification system to improve over time, thus enhancing its reliability and efficacy for use in medical diagnosis.

Conclusion and Future Work

This project established an integrated system for kidney disease prediction and classification through a combination of structured clinical data and unstructured CT images. Deep learning models have been used along with Explainable-AI techniques to enhance predictive accuracy along with transparency. This is to say that the system has been implemented as web application software in Streamlit and structured and unstructured data were stored in PostgreSQL and MongoDB respectively. The classification performance so high and was adequately established for clinical applicability.

A continued strengthening and broadening of the impact that the system eventually might have in real healthcare environments is proposed in various future enhancements, which are prioritized as follows.

- Strengthened Security and Privacy for Patient Data
  Sensitive patient data must be encrypted with advanced encryption models, have secure authentication technologies and be compliant with relevant data protection standards to safely store and transmit sensitive information.
- Deployment in Real Jordanian Hospitals
  It should be used in the contexts of clinical settings under the supervision of specialist physicians to validate the effectiveness of the system on true medical workflows while gaining invaluable feedback for continued improvement.
- Generate Automated Medical Reports and Visual Analysis
  These features should create automated generation of full-scale medical reports coupled with LIME heatmaps and result summaries to support doctors in timely and clinically informed decisions.
- Generative AI Should Be Embarked on for Treatment and Recommendation Plans
  Generative AI would be used to design personalized treatment strategies and advice on the clinical management based on each patient’s data and results from diagnosis.

- Extensibility toward the Diagnosis of Other Diseases Should Be There
  The feature should then be enlarged to include predicting many diseases, to make it a full-fledged virtual clinical support system for many applications.

- Automatic Model Retraining
  A mechanism should thus be developed to periodically retrain the models using real-life data entered within the databases in the system to remain accurately adaptable over time.

- Adopt Advanced AI Technologies
  Examine more advanced AI architectures such as self-supervised and multi-modal ones as part of their attempts to improve the system's diagnosis performance and generalization capacity.

- Optimization and Scalability of the Database Infrastructure
  PostgreSQL and MongoDB must enlarge their capabilities so as to accommodate larger datasets, make query execution efficient, and ensure that data within the applications will run seamlessly as usage grows.

The future enhancements will be able to take the system from prototype status into a fully-fledged clinical support platform that is both intelligent and secure. The groundwork laid here has demonstrated considerable potential for future real-world application and impact in the health space.

References

[1] P. Ghosh, F. M. J. M. Shamrat, S. Shultana, S. Afrin, and others, "Optimization of Prediction Method of Chronic Kidney Disease Using Machine Learning Algorithm," Preprint, Nov. 2020.

[2] Healthdirect, "Chronic Kidney Disease," *Healthdirect*. Available: https://www.healthdirect.gov.au/chronic-kidney-disease.

[3] Webteb, "Kidney Failure," *Webteb*. Available: https://www.webteb.com/kidney-urology/diseases/%D8%A7%D9%84%D9%81%D8%B4%D9%84-%D8%A7%D9%84%D9%83%D9%84%D9%88%D9%8A#description.

[4] V. Kotu and B. Deshpande, Data Science: Concepts and Practice, 2nd ed. Cambridge, MA, USA: Morgan Kaufmann, 2018.

[5] Md. A. Islam, M. Z. H. Majumder, and M. A. Hussein, "Chronic kidney disease prediction based on machine learning algorithms.

[6] S. N. Almuayqil, S. A. El-Ghany, A. A. Abd El-Aziz, and M. Elmogy, "KidneyNet: A novel CNN-based technique for the automated diagnosis of chronic kidney diseases from CT scans," *Information Systems Department, College of Computer and Information Sciences, Jouf University, Sakaka*, 72388, Saudi Arabia, *Mansoura University*, Mansoura, 35516, Egypt.

[7] A. Smith, B. Jones, and C. Lee, "Artificial Intelligence in Healthcare: Revolutionizing Disease Prediction," *Journal of Medical Informatics*, vol. 30, no. 4, pp. 225-238, 2020.

[8] H. Wang and M. Zhang, "The Role of Deep Learning in Medical Imaging: A Review," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 7, pp. 1240-1252, 2

[9] Jha, A., Sharma, P., and Gupta, R., "Implementation of ResNet-50 on CT Kidney Images for Disease Classification," Journal of Medical Imaging, vol. 15, no. 2, pp. 123-130, 2022.

[10] M. Almansour, S. A. Khan, and A. Qureshi, "Predicting chronic kidney disease using machine learning algorithms," *IEEE Access*, vol. 7, pp. 164438-164447, 2019.

[11] R. Gupta, A. Sharma, and P. Verma, "A logistic regression approach for chronic kidney disease classification with SMOTE," *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 12, pp. 3573-3581, 2020.

[12] S. Kumar, M. K. Patel, and V. S. Rao, "Random Forest and XGBoost-based predictive models for chronic kidney disease diagnosis," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 8, pp. 2931-2940, 2021.

[13] R. Sarvamangala and N. Kulkarni, "A deep learning-based custom CNN for kidney tumor detection in CT images," *IEEE Transactions on Medical Imaging*, vol. 40, no. 5, pp. 1297-1305, 2021.

[14] A. Jha, S. Roy, and P. Singh, "Kidney abnormality detection using ResNet-50: A Kaggle dataset approach," *IEEE Access*, vol. 10, pp. 47532-47542, 2022.

[15] L. Chen, M. Zhao, and Y. Wang, "Explainable AI for kidney tumor classification: Integrating VGG16 with Grad-CAM," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 245-256, 2023.

[16] UCI Machine Learning Repository, "Chronic Kidney Disease Data Set," 2015. [Online]. Available: https://archive.ics.uci.edu/dataset/336/chronic+kidney+disease.

[17] Amazon Web Services, "What is synthetic data?", AWS, Available: https://aws.amazon.com/what-is/synthetic-data/.

[18] "Variable-specific random sample imputation. Is it a valid method of imputation?" *Cross Validated*. Available: https://stats.stackexchange.com/questions/493968/variable-specific-random-sample-imputation-is-it-a-valid-method-of-imputation.

[19] K. Saw Htoon, "A Guide To KNN Imputation," *Medium*. Available: https://medium.com/@kyawsawhtoon/a-guide-to-knn-imputation-95e2dc496e.

[20] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.

[21] D. Rothman, Hands-On Explainable AI (XAI) with Python: Interpret, Visualize, Explain Machine Learning Models. packet Publishing, 2020, pp. 27–28.

[22] "XAI Explainability Methods." AI Online Course, https://www.aionlinecourse.com/ai-basics/xai-explainability-methods.

[23] D. Das, "How to Explain Image Classifiers using LIME," *Medium*, Apr. 24, 2020. [Online]. Available: https://medium.com/data-science/how-to-explain-image-classifiers-using-lime-e364097335b4

[24] H. Dani, P. Bhople, H. Waghmare, K. Munginwar, and A. Patil, "Review on frameworks used for deployment of machine learning model," in *Proceedings of the Department of Computer Engineering, Government College of Engineering*, Yavatmal, Maharashtra, India, 2024.

[25] **Streamlit,** "**Streamlit Documentation**," [Online]. Available: https://docs.streamlit.io/

[26] **MongoDB**, *MongoDB Documentation*, 2025. [Online]. Available: https://www.mongodb.com/docs.

[27] Sinno Jialin Pan and Qiang Yang, "**A survey on transfer learning**," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, Oct. 2010

[28] Mingxing Tan and Quoc V. Le, **"EfficientNet: Rethinking model scaling for convolutional neural networks,"** *Proceedings of the 36th International Conference on Machine Learning (ICML)*, vol. 97, pp. 6105–6114, 2019.

[29] K. Simonyan and A. Zisserman, **"Very deep convolutional networks for large-scale image recognition,"** *International Conference on Learning Representations (ICLR)*, 2015.