National College of Ireland

Configuration Manual

MSc Research Project
Artificial Intelligence

Haameem Shimar Yusuff Nazzer
Student ID: 23330155

School of Computing
National College of Ireland

Supervisor: Anderson Simiscuka

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Haameem Shimar Yusuff Nazzer

Student ID: x23330155

Programme: MSc in Artificial Intelligence **Year:** 2025

Module: MSc (Research) Practicum/Internship Part 2

Lecturer: Anderson Simiscuka

Submission Due Date: 11-August-2025

Project Title: Predicting Chronic Kidney Disease (CKD) Using Histopathological Images and Clinical Data

Word Count: 700 **Page Count:** 11

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Haameem Shimar Yusuff Nazzer

Date: 11-August-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual: Kidney Disease Prediction System

Haameem Shimar Yusuff Nazzer
Student ID: x23330155

1 Introduction

This project is designed to develop and deploy a **Kidney Image Classifier & Chronic Kidney Disease (CKD) Prediction System**.It integrates **deep learning models** for image classification (chronic, normal, stone, tumor) and a **multi-output machine learning model** for predicting CKD classification, Glomerular Filtration Rate (GFR), and CKD stage based on clinical and textual data.The system is built using **Python** and **TensorFlow**, with a **Streamlit-based web interface** for user interaction.
It also uses **SQLite** for user authentication and supports image uploads alongside structured and text data inputs for predictions.

This configuration manual provides a **step-by-step guide** to setting up the environment, installing required software, preparing datasets, running training scripts, and deploying the web application.

2 Specifications

This section mentions the hardware and software specifications of the system that was used for implementing the research.

2.1 Hardware Specification

The research is implemented on the system with the following hardware configuration as shown in the Table 1

Specification	Details
Machine	Victus by HP Gaming Laptop 15-fb0xxx
Operating System	Windows 11 (22H2)
Processor	AMD Ryzen 5 5600H with Radeon Graphics
Installed RAM	16.0 GB DDR4 (15.3 GB usable)
System Type	64-bit operating system, x64-based processor
GPU	Multiple GPUs installed (Dedicated GPU with 4 GB VRAM; exact model can be NVIDIA or AMD, check in Device Manager)
Storage	477 GB total (322 GB used, 155 GB free)

Table 1 : Hardware Specifications

2.2 Software Requirements

Component	Requirement
Operating System	Windows 11 (22H2) – Compatible with Windows 10, macOS 12+, Ubuntu 20.04+
Python Version	Python 3.8 – 3.10
IDE / Tools	Thonny IDE (recommended for running scripts) Command Prompt / Terminal Optional: VS Code / PyCharm
Python Libraries	**numpy, pandas, matplotlib, seaborn, tensorflow, scikit-learn, imblearn, joblib, opencv-python, scikit-image, scipy, streamlit**
Database	SQLite3 (pre-installed with Python)
Browser	Google Chrome / Microsoft Edge / Firefox

Table 2 : Software Requirements

3 Dataset Description

1. CT Kidney Dataset – Normal, Cyst, Tumor, and Stone

- A labeled collection of CT scan images segmented into four categories: **Normal**, **Cyst**, **Tumor**, and **Stone**, sourced from hospital archives in Dhaka, Bangladesh
- Intended for training image classification models to detect renal anomalies using visual data.

Sample:

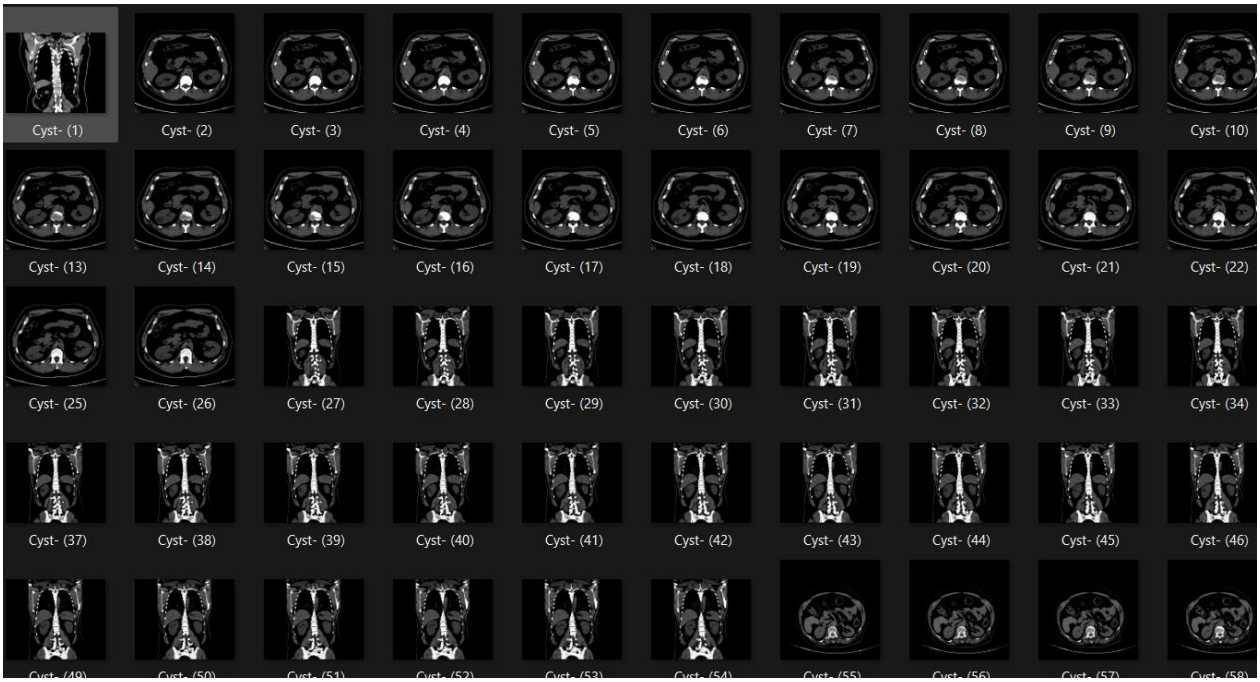


Figure 1 : Sample dataset kidney images

2. Clinical Kidney Disease Dataset (Kaggle by Amanik000)

- A comprehensive clinical dataset featuring structured and laboratory data collected from patients, designed to support analysis and prediction of **Chronic Kidney Disease (CKD)**
- Includes key clinical features—such as RBC count, blood pressure, albumin, etc.—used in conjunction with textual notes or numeric data for predictive modeling.

Sample:

id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet
0	48	80	1.02	1	0		normal	notpresent	notpresent	121	36	1.2			15.4	44	7800	5.2	yes	yes	no	good
1	7	50	1.02	4	0		normal	notpresent	notpresent		18	0.8			11.3	38	6000		no	no	no	good
2	62	80	1.01	2	3	normal	normal	notpresent	notpresent	423	53	1.8			9.6	31	7500		no	yes	no	poor
3	48	70	1.005	4	0	normal	abnormal	present	notpresent	117	56	3.8	111	2.5	11.2	32	6700	3.9	yes	no	no	poor
4	51	80	1.01	2	0	normal	normal	notpresent	notpresent	106	26	1.4			11.6	35	7300	4.6	no	no	no	good
5	60	90	1.015	3	0			notpresent	notpresent	74	25	1.1	142	3.2	12.2	39	7800	4.4	yes	yes	no	good
6	68	70	1.01	0	0		normal	notpresent	notpresent	100	54	24	104	4	12.4	36			no	no	no	good
7	24		1.015	2	4	normal	abnormal	notpresent	notpresent	410	31	1.1			12.4	44	6900	5	no	yes	no	good
8	52	100	1.015	3	0	normal	abnormal	present	notpresent	138	60	1.9			10.8	33	9600	4	yes	yes	no	good
9	53	90	1.02	2	0	abnormal	abnormal	present	notpresent	70	107	7.2	114	3.7	9.5	29	12100	3.7	yes	yes	no	poor
10	50	60	1.01	2	4		abnormal	present	notpresent	490	55	4			9.4	28			yes	yes	no	good
11	63	70	1.01	3	0	abnormal	abnormal	present	notpresent	380	60	2.7	131	4.2	10.8	32	4500	3.8	yes	yes	no	poor
12	68	70	1.015	3	1		normal	present	notpresent	208	72	2.1	138	5.8	9.7	28	12200	3.4	yes	yes	yes	poor
13	68	70						notpresent	notpresent	98	86	4.6	135	3.4	9.8				yes	yes	yes	poor
14	68	80	1.01	3	2	normal	abnormal	present	present	157	90	4.1	130	6.4	5.6	16	11000	2.6	yes	yes	yes	poor
15	40	80	1.015	3	0		normal	notpresent	notpresent	76	162	9.6	141	4.9	7.6	24	3800	2.8	yes	no	no	good
16	47	70	1.015	2	0		normal	notpresent	notpresent	99	46	2.2	138	4.1	12.6				no	no	no	good
17	47	80						notpresent	notpresent	114	87	5.2	139	3.7	12.1				yes	no	no	poor
18	60	100	1.025	0	3		normal	notpresent	notpresent	263	27	1.3	135	4.3	12.7	37	11400	4.3	yes	yes	yes	good
19	62	60	1.015	1	0		abnormal	present	notpresent	100	31	1.6			10.3	30	5300	3.7	yes	no	yes	good
20	61	80	1.015	2	0	abnormal	abnormal	notpresent	notpresent	173	148	3.9	135	5.2	7.7	24	9200	3.2	yes	yes	yes	poor
21	60	90						notpresent	notpresent		180	76	4.5		10.9	32	6200	3.6	yes	yes	yes	good

Figure 2 : dataset csv

4 Project Structure

Organize your files and folders as follows, ensuring **all Python scripts** and **datasets** are placed in the correct locations:

Figure 3 : Folder Structure

```
project_root/
│
├── app.py                      # Streamlit frontend
├── CNN.model/                  # CNN model directory
├── ResNet50_model.h5           # Trained ResNet50 model
├── VGG16_model.h5              # Trained VGG16 model
├── ckd_model.h5                # Multi-output prediction model
├── scaler.pkl                  # Trained StandardScaler
├── label_encoders.pkl          # Label encoders (joblib)
├── tfidf_vectorizer.pkl        # Trained TF-IDF vectorizer
├── users.db                    # SQLite database (auto-created)
│
├── image_dataset/              # 📁 Folder with class-named subfolders of images
│   ├── normal/
│   ├── infected/
│   └── ...
│
├── csv_dataset/                # 📁 Folder containing the clinical dataset
│   └── dataset.csv             # CSV file with features, notes, labels
```

5 Preaparing the dataset

Clinical Dataset

- The clinical dataset is already included in the project folder (root_directory/) since it is small in size.
- No additional download is required for this file.

Image Dataset

- Download the image dataset from the link provided in the ethics form.
- Once downloaded, extract the files.
- Organize the images into the image_dataset/ folder, creating **four subfolders**:
- Place the images in their corresponding subfolder based on their category.

```
image_dataset/
├── chronic/
├── normal/
├── stone/
└── tumor/
```

6 Running the Web Application

You will use **Command Prompt (cmd)** to launch the app.

1. Open **Command Prompt**
2. Navigate to the project directory:

   cd path\to\your\project_root
3. Launch the app by passing the below:
   streamlit run app.py
4. The app will open in your browser (usually at http://localhost:8501)

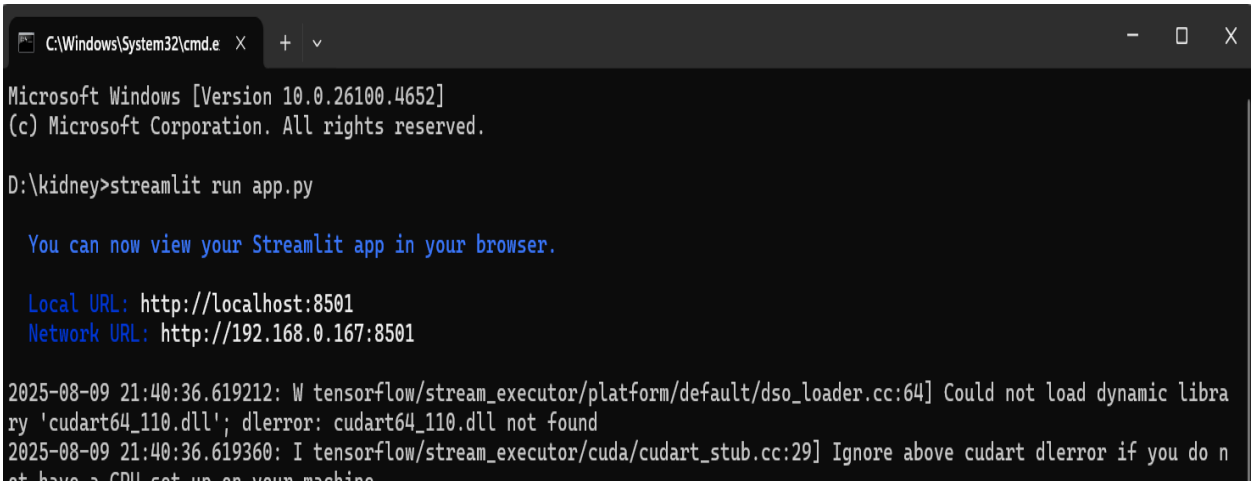
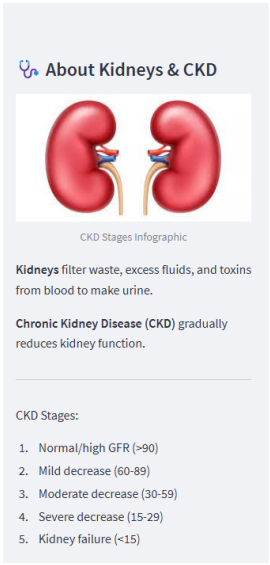
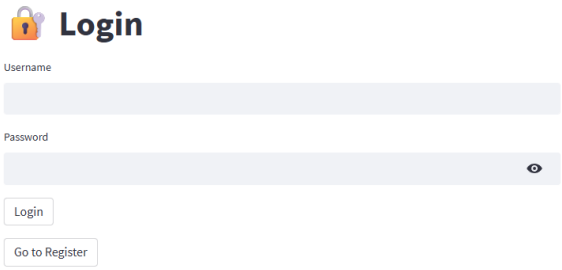


Figure 4 : launching of the Application

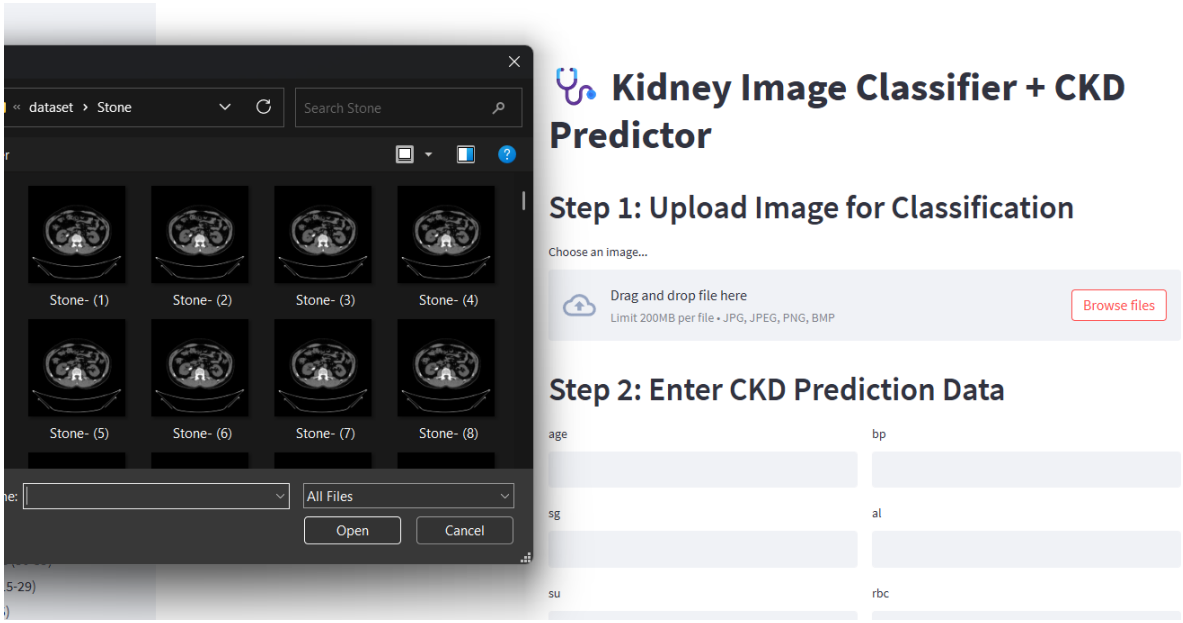
7 After launching the app

1. Log in or register (credentials stored in users.db)

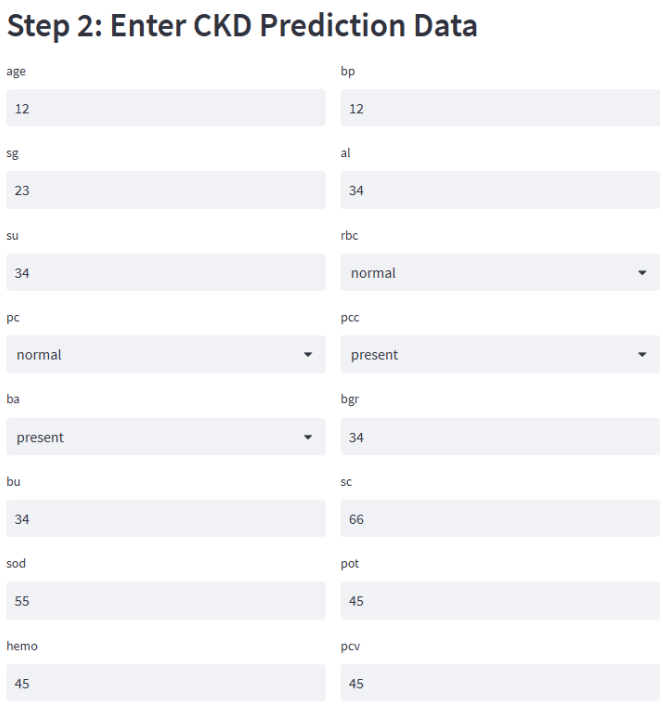




2. Upload a kidney image



3. Enter clinical details + notes:

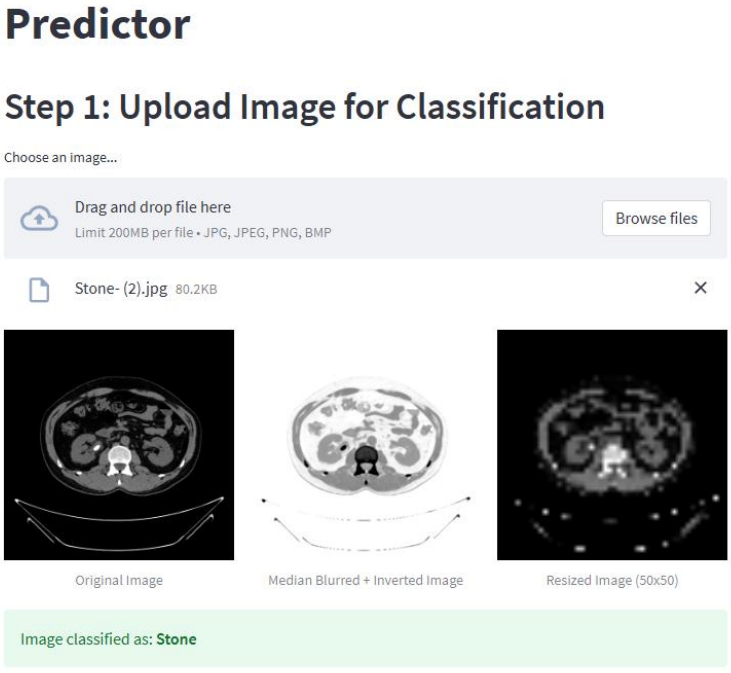


4. Click **"Predict CKD"** to see:
    - CKD classification (Yes/No)
    - CKD Stage
    - GFR value

8 Results :

8.1 Result 1:

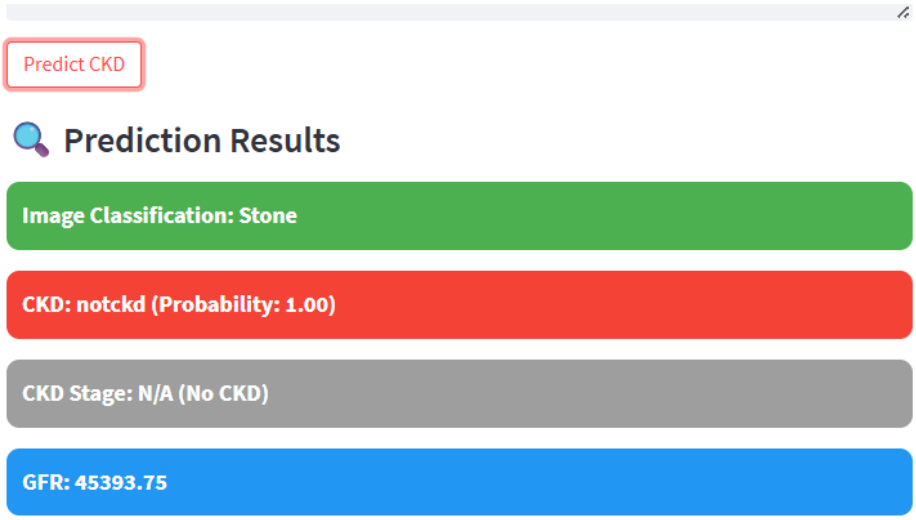
The trained CNN model will classify it into one of these four categories: **Normal**, **Tumor**, **Stone**, or **Chronic**.



8.2 Result 2 :

The system will generate the following information:

- **Image Classification Result** (Normal, Tumor, Stone, or Chronic)
- **CKD Prediction** with probability score
- **CKD Stage** (if applicable)
- **Estimated GFR (Glomerular Filtration Rate)** value



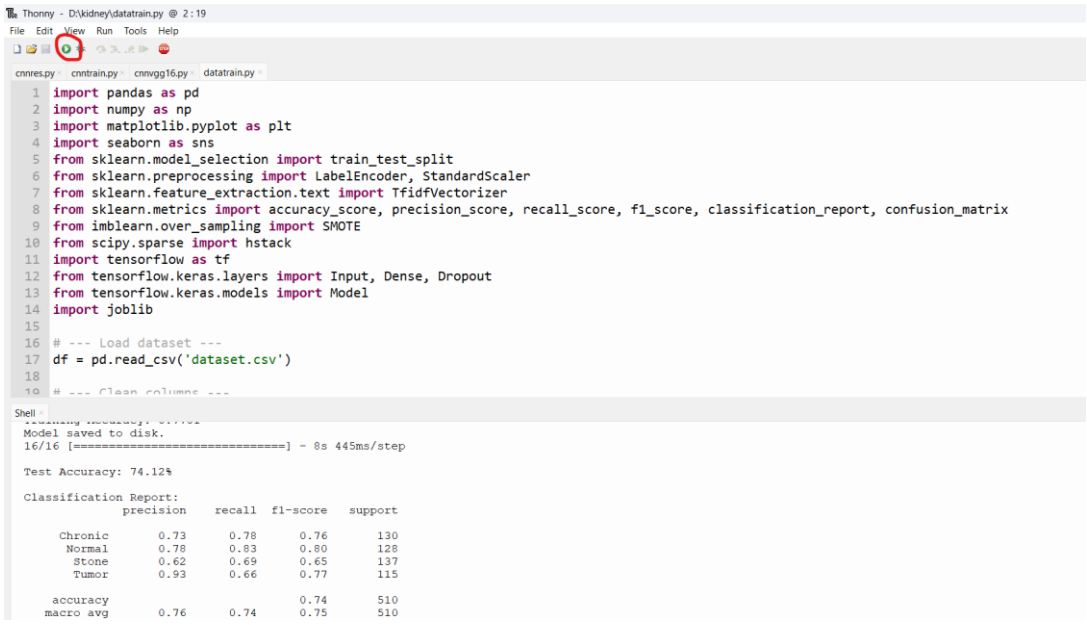
9 Model Training Scripts (via Thonny IDE)

All training should be done by opening the scripts in **Thonny**, not from terminal.

Script	Script Name	Framework
ResNet50 transfer learning	cnnres.py	TensorFlow/Keras
Custom CNN (3×Conv)	cnntrain.py	TensorFlow/Keras
VGG16 transfer learning	cnnvgg16.py	TensorFlow/Keras
CKD multimodal (tabular + text)	datatrain.py	TensorFlow/Keras, scikit-learn, imblearn

How to Run:

- Open Thonny
- Load each script via File → Open
- Click **Run** or press F5
- Wait for training to complete and view output



Note: All visuals, including accuracy plots, loss curves, confusion matrices, heatmaps, and other performance metrics, will be automatically generated and displayed within the **Thonny IDE** during script execution.

10 Conclusion

This configuration manual provides all the necessary steps, software requirements, hardware specifications, dataset organization guidelines, and execution instructions to set up and run the kidney disease prediction system. By following this manual, users can seamlessly install dependencies, configure the environment, organize datasets, and execute the project in **Thonny IDE** or via the **web application interface**. Proper adherence to the outlined setup ensures smooth operation, accurate predictions, and reproducible results.
Ask ChatGPT