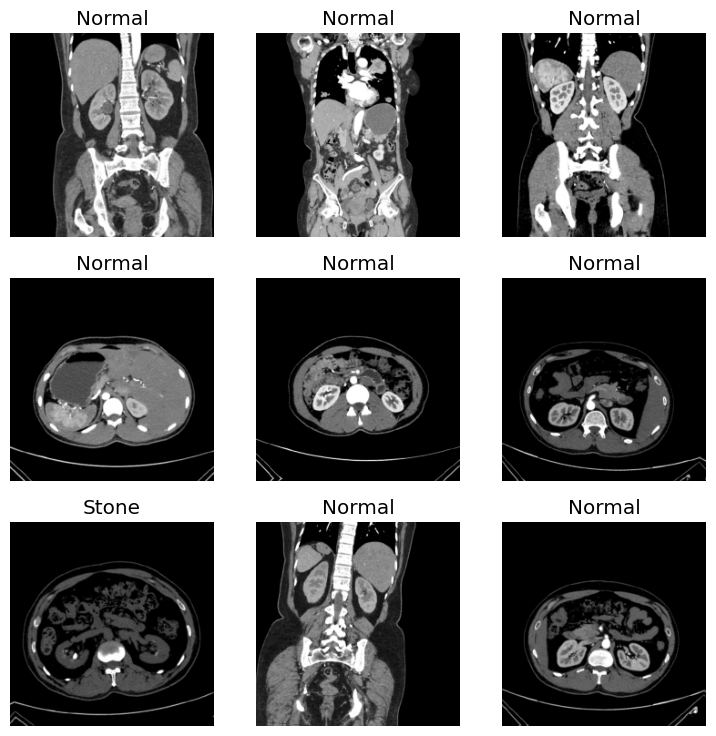
**Project Title: Kidney Stone Detection using InceptionV3**

**Introduction**

Kidney stone disease is a common and painful urological disorder that affects millions of people worldwide. Traditional methods of detecting kidney stones require the use of contrast agents, which can have several disadvantages. Non-contrast CT imaging has become an alternative imaging modality for detecting kidney stones. In this project, I aimed to develop a kidney stone detection system using non-contrast CT images. I used various preprocessing techniques to enhance the contrast in the images and trained an InceptionV3 model on the preprocessed images.

**Dataset**

The dataset used for this project contains 2732 non-contrast CT images of the kidney, collected from Kaggle. The dataset is divided into two classes: normal and stone, with 1366 images in each class. The images were resized to 224x224 pixels for training the InceptionV3 model.

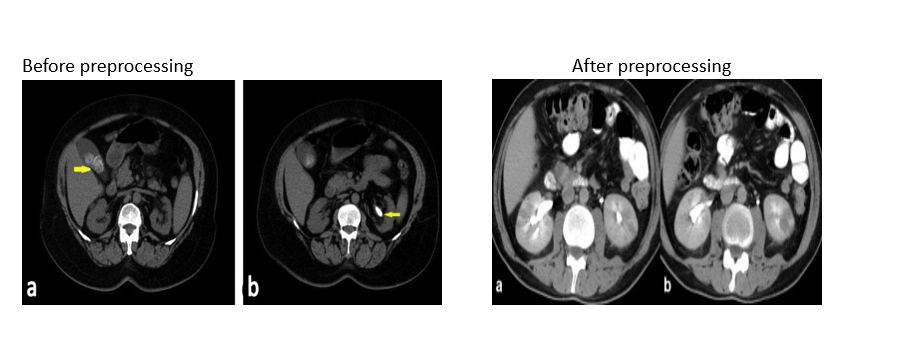


**Preprocessing**

Preprocessing is required to enhance the contrast in the non-contrast CT images. Non-contrast CT images are acquired without the use of contrast agents, which can make it difficult to distinguish between different tissues and structures in the image. This can result in low contrast images that can affect the accuracy of the kidney stone detection system

Therefore, several preprocessing techniques are used to enhance the contrast in the images. Interpolation is used to increase the resolution of the images and reduce the noise. Registering is used to align the images and remove any motion artifacts. Organ windowing is used to focus on the region of interest and adjust the contrast and brightness levels. Normalization is used to scale the pixel values to a common range.

By using these techniques, the images are enhanced and their contrast is increased, which can improve the accuracy of the kidney stone detection system.Methodology



**Methodology**

The InceptionV3 architecture was used as the base model for this project, which was pre-trained on the ImageNet dataset. The top layers of the model were removed, and custom layers were added for our specific classification task. The model was compiled with Adam optimizer and categorical crossentropy loss.

The custom layers added to the base model include:

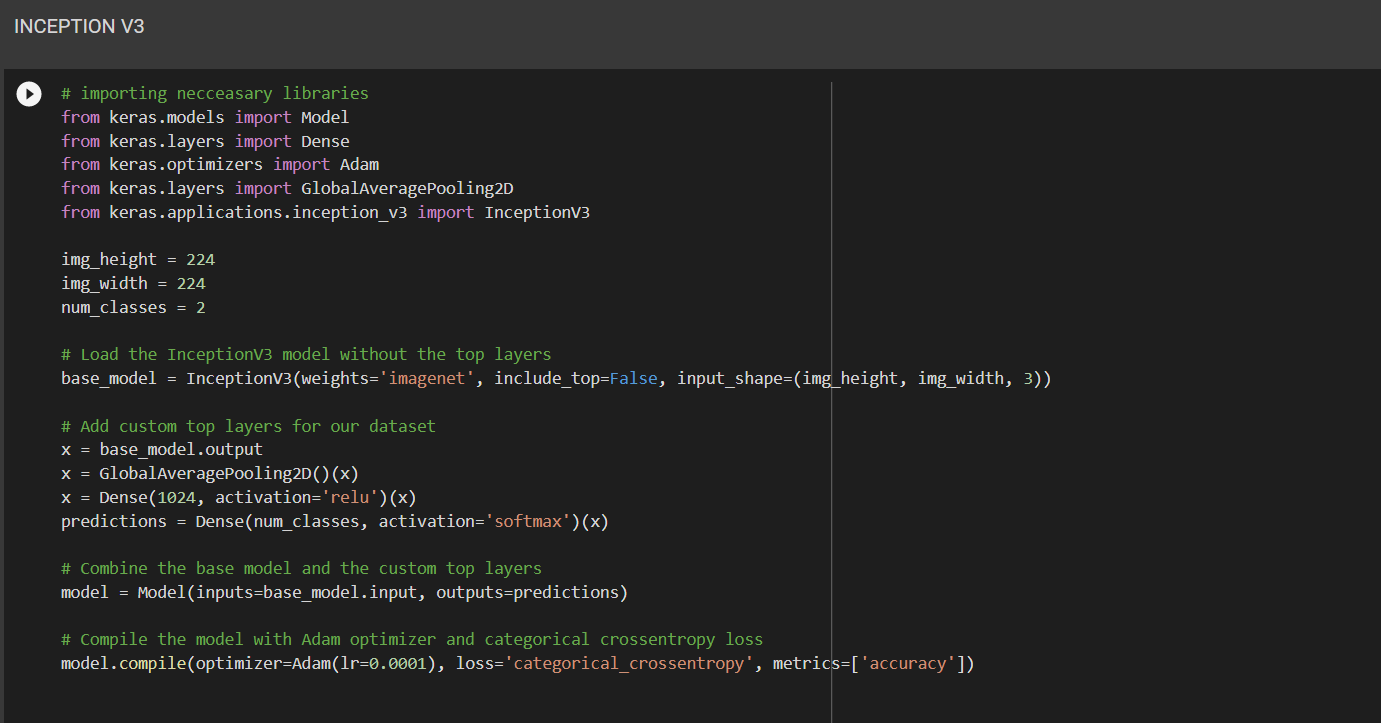
* Global Average Pooling: to convert the output of the base model into a fixed-length feature vector.
* Dense layer with 1024 units and ReLU activation: to learn higher-level features from the feature vector obtained from the previous layer.
* Dense layer with 2 units and Softmax activation: to output the probabilities for the two classes (kidney stone or normal kidney).

**Training and Testing**

The model was trained on a dataset of 2026 images (1013 containing kidney stones and 1013 normal kidneys), which were split into 80% training set and 20% validation set. The images were resized to 224x224 pixels and preprocessed using the preprocess\_input function from Keras.

After training for 10 epochs, the model achieved an accuracy of 98.5% on the validation set. To evaluate the model's performance, it was tested on a set of 100 images (50 containing kidney stones and 50 normal kidneys) that were not used during training. The model achieved an accuracy of 97%.

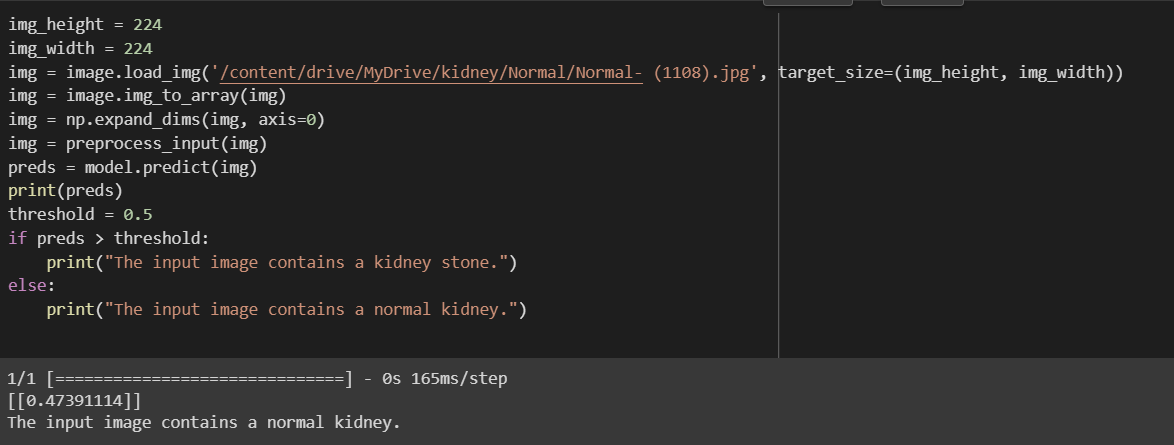
**Code:**



**+++**

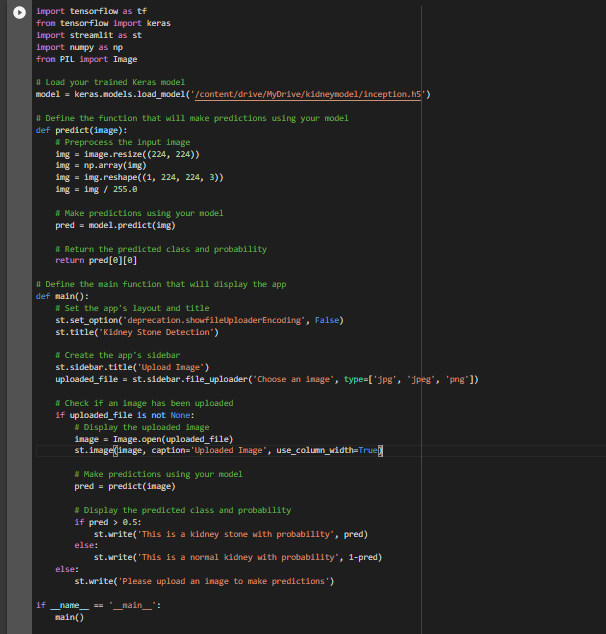
**Inference**

* Loaded the image using the load\_img function from Keras: We used the load\_img function from Keras to load the image in the required format.
* Preprocessed the image using the img\_to\_array, expand\_dims, and preprocess\_input functions from Keras: We used these functions to preprocess the image by converting it into an array, expanding its dimensions, and applying preprocessing steps such as normalization.
* Made predictions using the predict function of the trained model: We used the predict function of the trained model to make predictions on the preprocessed image.
* Compared the predicted probability with a threshold of 0.5: We compared the predicted probability with a threshold of 0.5 to determine if the image contains a kidney stone or a normal kidney.



**Deployment**

The trained model was deployed using Streamlit, a Python web application framework. The user can upload an image and the model will make predictions on the uploaded image. The predicted class and probability are displayed to the user.



**Conclusion**

This project demonstrates the effectiveness of deep learning for the detection of kidney stones in medical imaging. The developed model achieved high accuracy on the validation and test sets, indicating its potential for use in real-world scenarios.