Kidney Disease Classification

Team Members: Shriya Gandotra, Swayam Mehta, Smit Patel, Luis Funes

INTRODUCTION

Proposal Changes:

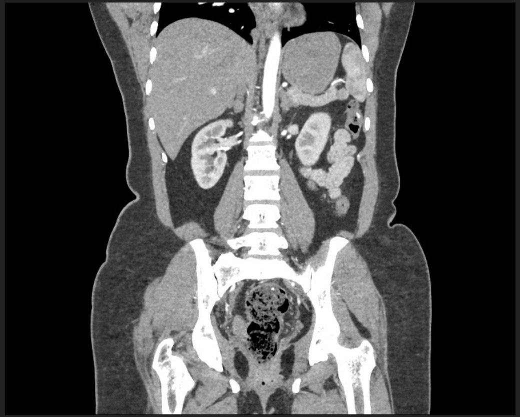
Revised original dataset from 1 class - kidney stone classification to multi-class dataset.

This project employs classical and modern learning techniques to classify kidney diseases from CT images, differentiating normal, cyst, tumor, and stone conditions. By leveraging convolutional neural networks (CNN) and support vector machine (SVM), we aim to enhance diagnostic accuracy and support timely medical intervention, advancing medical imaging technology in healthcare.

Impact:

Enhanced Diagnostic Accuracy, Automated Image Analysis, Predictive Analytics, Cost Reduction.

NORMAL

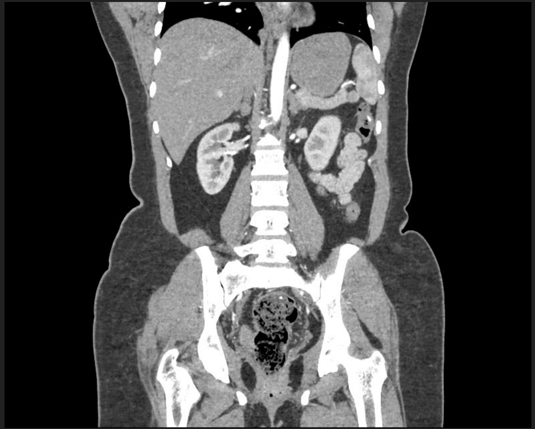


STONES

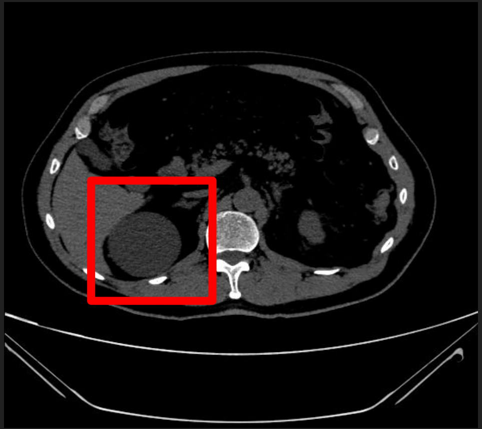
DATASET - 4 Classes

The dataset was collected from PACS (Picture archiving and communication system) from different hospitals in Dhaka, Bangladesh where patients were already diagnosed with having a kidney tumor, cyst, normal or stone findings. Both the Coronal and Axial cuts were selected from both contrast and non-contrast studies with protocol for the whole abdomen and urogram.

Normal



Cyst



Stone

Tumor

PRE-PROCESSING

No need to Grayscale: CT images were already B/W

Reduced image size from 705x569 to 96x96 for processing efficiency.

<u>SVM:</u>

Randomly flipped images and rotated images to introduce noise and variable complexity to the learning model.

<u>CNN</u>:

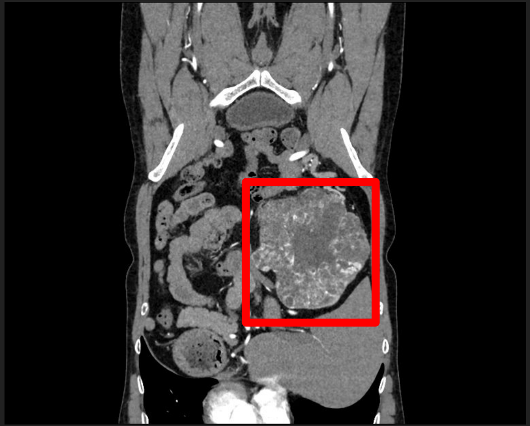
Random flip/rotation, tilt 10°, change image brightness, contrast, and saturation.

<u>Normal</u>

<u>Cyst</u>

<u>Horizontal</u>

<u>Vertical</u>



SVM (Support Vector Machine)

```
print("Training the SVM model...")
start_time = time.time()
svm_model = svm.SVC(kernel='poly')
svm_model.fit(X_train, y_train)
print("SVM model trained.")
```

Classification Report:

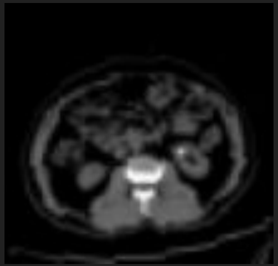
	precision	recall	f1-score	support
Cyst	0.72	0.97	0.82	737
Normal	0.87	0.90	0.89	1001
Stone	0.76	0.32	0.45	280
Tumor	0.83	0.59	0.69	471
accuracy			0.80	2489
macro avg	0.79	0.70	0.71	2489
weighted avg	0.80	0.80	0.78	2489

```
Model evaluation completed.
Accuracy: 0.798
Precision: 0.804
Recall: 0.798
F1 Score: 0.782
```

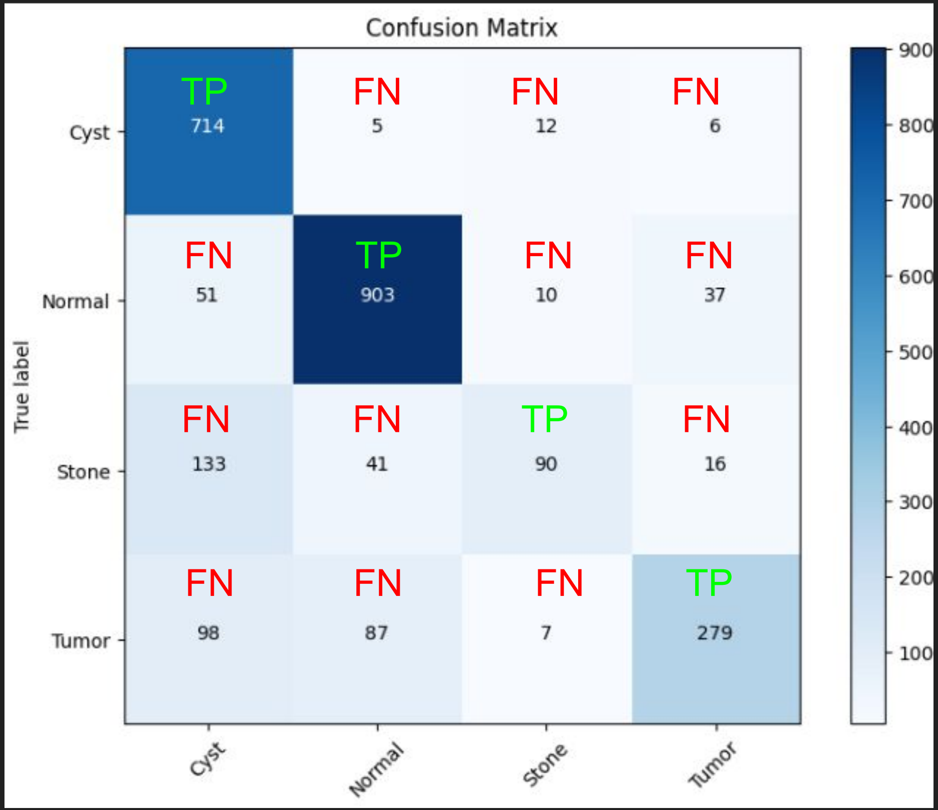
- Between rbf, linear, poly and sigmoid, 'poly' gave the best metrics across the board
- Poly can handle higher dimensional spaces without computing new feature space
- Poly also works well with regularization over simpler models like linear

CONFUSION MATRIX

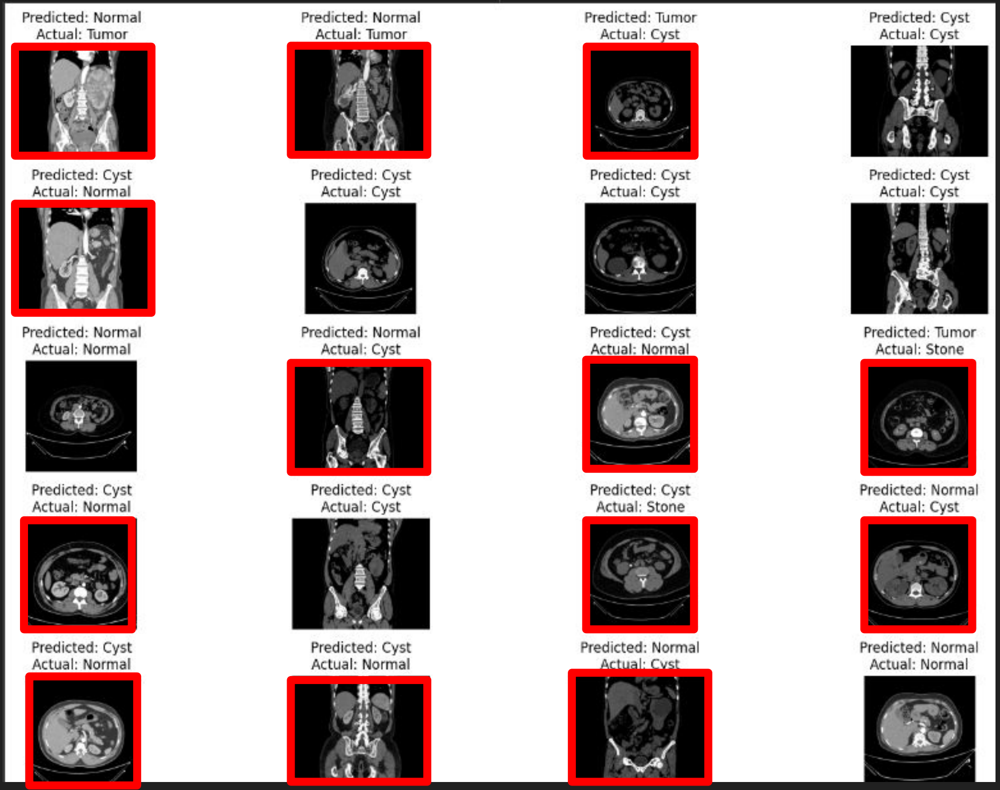
Stone class was the most difficult to classify by our model. This is due to the fact that each stone variety in size, position and overall resolution.

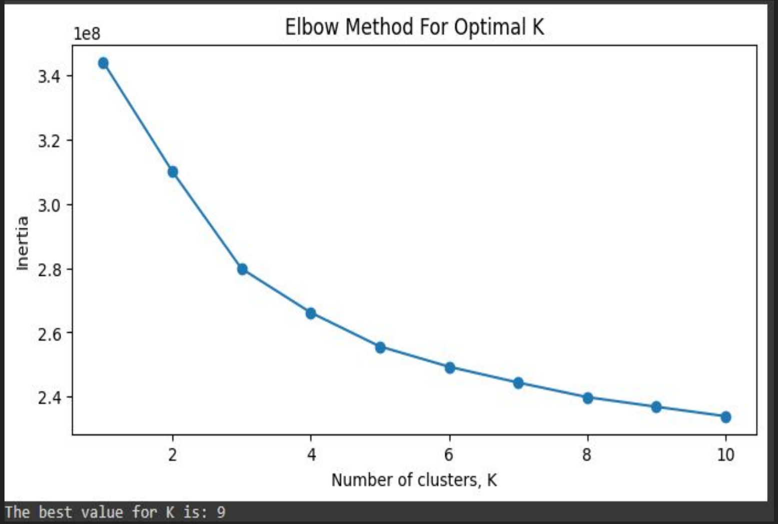


96x96 Stone Image



SVM VALIDATION





CNN (Convolutional Neural Network)

```
num_classes = 4

def cnn_model(num_classes):
    model = nn.Sequential(
        nn.Conv2d(3, 64, kernel_size=3, padding=1),
        nn.Tanh(),
        nn.Conv2d(64, 128, kernel_size=3, padding=1),
        nn.Tanh(),
        nn.MaxPool2d(2, 2),
        nn.Flatten(),
        nn.Linear(128 * 16 * 16, 256),
        nn.Tanh(),
        nn.Dropout(0.5),
        nn.Linear(256, num_classes)
    )
    return model
```

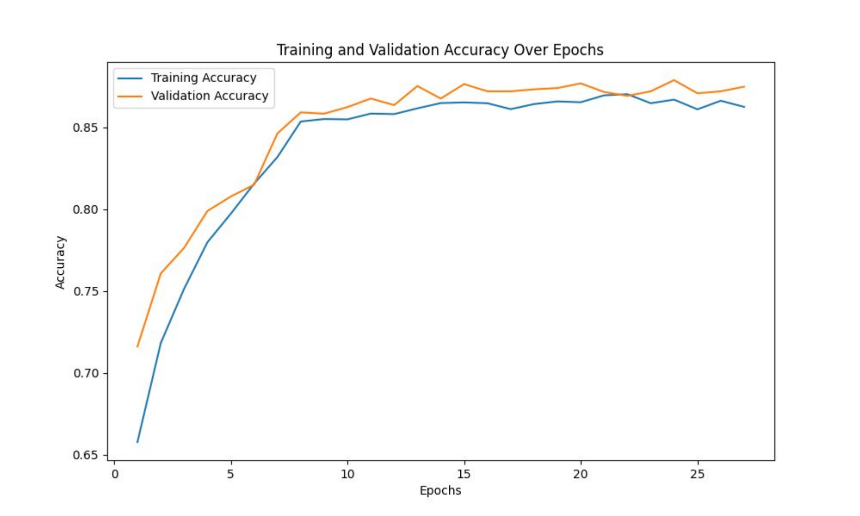
Kernel Size:
Fine-grained details and are computationally more efficient.

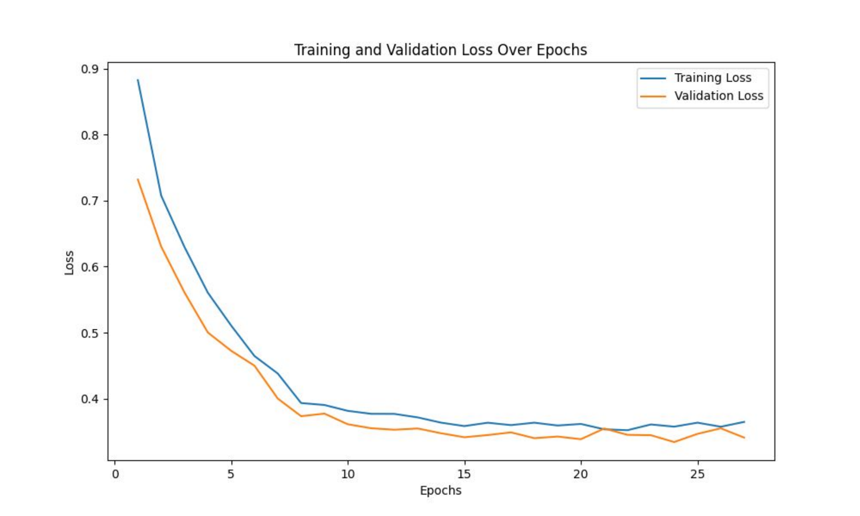
ReLU:
Introduces non-linearity into the model.

Tanh:
Distinguishes features, provided better benchmark results as a whole (**Selected Activation**).

Dropout:
Prevents Overfitting.

TRAINING





Training and Validation Accuracy Over Epochs

Training and Validation Loss Over Epochs

CNN VALIDATION

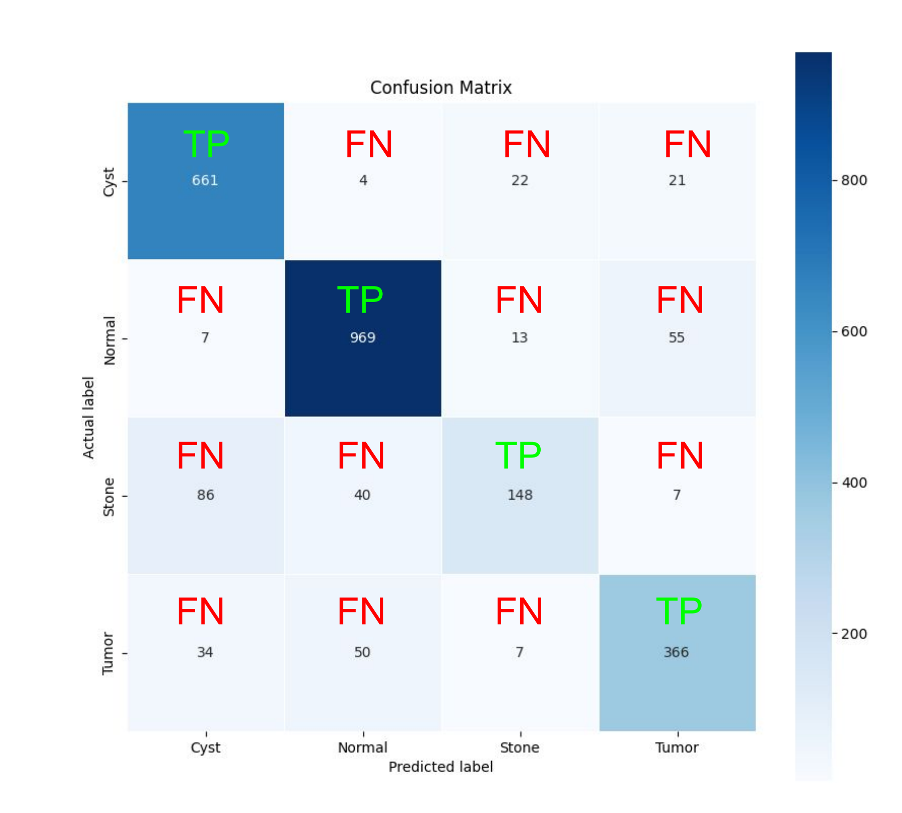
Our model was overall successful in classify the correct class. The main issue came across when the class feature was overall weak and even hard to detect by eye. This is due to the resolution of 96x96 that reduces the overall clarity of the images causing the model to miss some of the key class elements.

```
Finished Training in 1722.88 seconds
Evaluating model...
Accuracy: 0.876
Precision: 0.875
Recall: 0.876
F1 Score: 0.872
```

Classification Report:

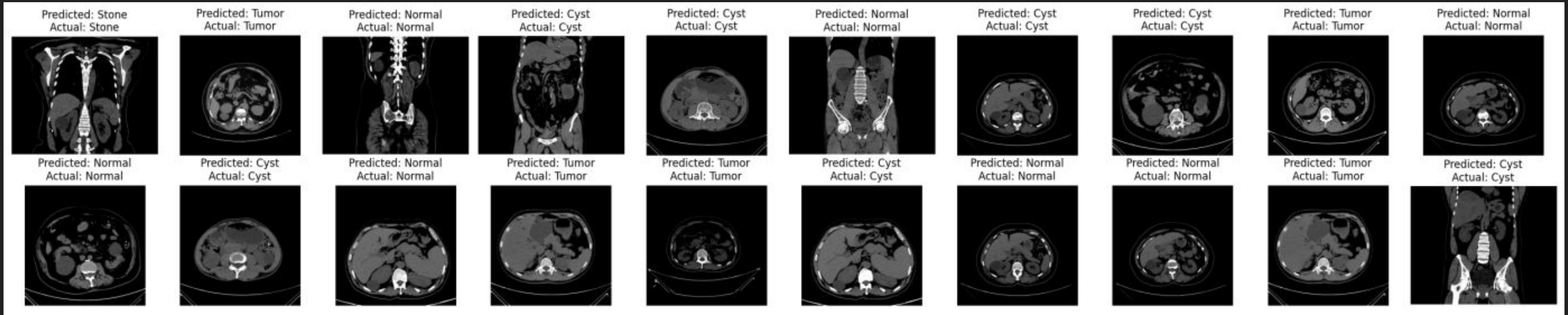
	precision	recall	f1-score	support
Cyst	0.86	0.95	0.90	726
Normal	0.91	0.94	0.93	1041
Stone	0.85	0.59	0.70	253
Tumor	0.83	0.77	0.80	470
accuracy			0.88	2490
macro avg	0.86	0.81	0.83	2490
weighted avg	0.87	0.88	0.87	2490

CONFUSION MATRIX



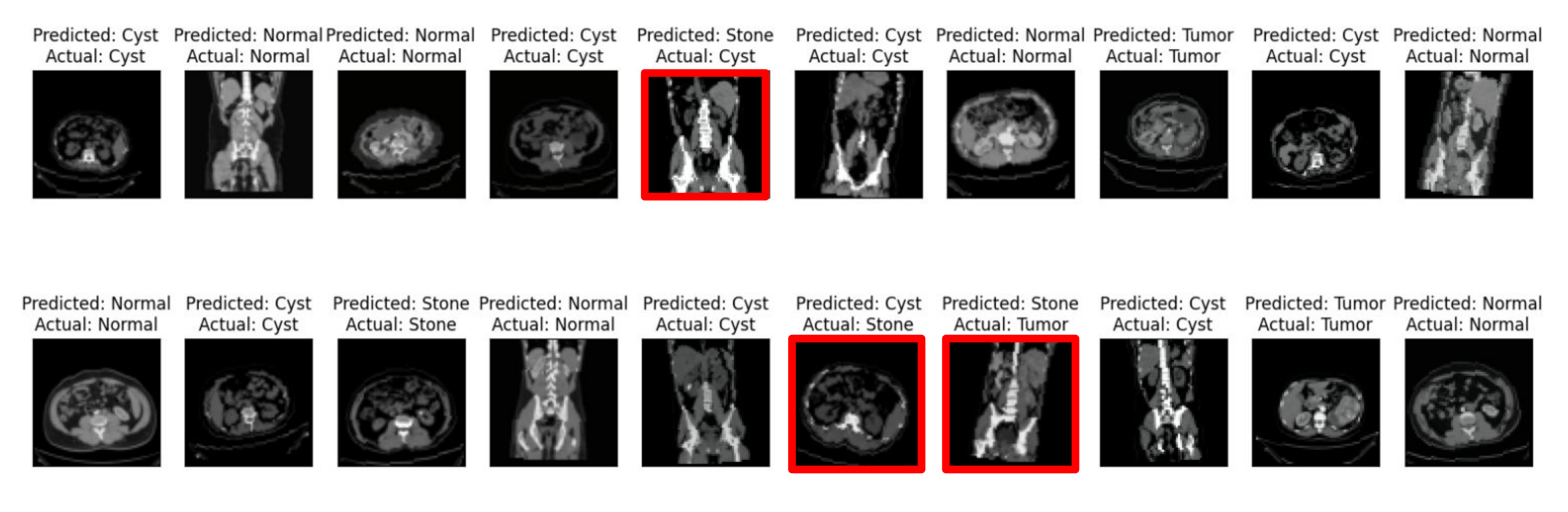
CNN VALIDATION

First Iteration



CNN VALIDATION

Second Iteration



COMPARISON

SVM: Training Time - 11.5 mins	CNN: Training Time - 28 mins
Recall: 0.798	Recall: 0.876
Precision: 0.804	Precision: 0.875
Accuracy: 0.798	Accuracy: 0.876
F1 Score: 0.782	F1 Score: 0.872

CNN model outperformed SVMs in image classification due to their use of the Tanh activation function and their structured architecture. Tanh helps CNNs model complex data patterns, and the layered design effectively extracts features from raw images, a capability SVMs with polynomial kernels lack

APPLICATION

This model can be implemented into any hospital or lab where CT image analysis is required. CT analysis can expedited using machine learning to process and detect kidney diseases given ample data. This could cut overall medical costs and improve disease outlook.

AREAS OF IMPROVEMENT

- Increase resolution of images during pre-processing. Given more capable hardware
- Implement more modern ML techniques(Real-time, unsupervised learning)
- Include a variety of CT images from different sources.
- Kidney stone showed to have scored lowest across precision, recall, f1 score. This could be improved if more images were implemented into the data set and resolution used during preprocessing is increased.

GITHUB

https://github.com/Swayyum/Intro-to-ML--4105/tree/main/Final%20Project

Thank You !

Any questions ?