**Data Preparation/Feature Engineering**

## 1. Overview

In this project, comprehensive data preparation and feature engineering were pivotal in optimizing the CT kidney image dataset for machine learning tasks. Techniques such as resizing ensured uniformity in image dimensions, facilitating consistent model inputs. Balancing addressed class imbalance, enhancing the model's ability to generalize across diverse classes. Data augmentation enriched the dataset by diversifying image variations, bolstering the model's robustness to unseen data. Normalization standardized pixel intensity values across images, ensuring consistent scale and aiding in convergence during model training. The application of binarization masks effectively highlighted regions of interest like stones, enabling focused analysis and precise feature extraction. Collectively, these preprocessing steps not only ensured high data quality but also empowered the machine learning model to achieve superior performance and reliability in classification and segmentation tasks.

## 2. Data Collection

The dataset utilized in this study, the "CT KIDNEY DATASET: Normal-Cyst-Tumor and Stone," comprises 12,446 annotated CT images, including normal kidneys, cysts, stones, and tumors. This dataset, publicly available on Kaggle, provides a comprehensive resource for training and evaluating the proposed model. It worth mentioning that no preprocessing techniques were applied at this stage by us.

## 3. Data Cleaning

In this step duplicated images were removed as will be explained in more details in Exploratory Data Analysis section.

## 4. Exploratory Data Analysis (EDA)

In this study, a comprehensive Exploratory Data Analysis (EDA) was conducted on a dataset of CT kidney images. Key EDA processes applied include duplication checking, class distribution analysis, mask application, image statistics calculation, multiple image display, and pixel value distribution investigation. These processes help understand the dataset's characteristics, ensure data quality, and identify areas for further analysis.

### Duplication Checking

Duplication checking involves identifying and removing duplicate images to ensure dataset integrity. Duplicates can bias results if not addressed. By comparing pixel values, duplicates were identified and removed, maintaining dataset uniqueness and accuracy.

A computer screen shot of a program code

Description automatically generated

A screenshot of a computer

Description automatically generated

As a result, 525 duplicated images were found and removed.

### Class Distribution Analysis

Class distribution analysis provides insights into the balance or imbalance of the dataset, crucial for classification tasks. The dataset was categorized into four classes: Normal, Cyst, Tumor, and Stone. Frequencies of each class were calculated and visualized using a bar chart, revealing imbalances that need to be addressed for robust model development.

A screen shot of a computer

Description automatically generated

A graph of different colored bars

Description automatically generated

* 1. **Mask Application**

Segmentation masks highlight regions of interest (ROIs), such as pathological areas. Binary masks were applied to delineate kidney regions, which were then overlaid on the original images for verification. This process is vital for tasks requiring precise localization of anatomical structures or abnormalities. For instance, Binarization of the mask in the stone condition example below effectively isolates these regions, enhancing visibility and enabling precise localization within the images. This approach is crucial for tasks requiring detailed examination or segmentation of specific structures, ensuring accurate and targeted analysis of medical imaging data.

A screen shot of a computer program

Description automatically generated

A close-up of an x-ray

Description automatically generated

* 1. **Image Statistics Calculation**

Calculating image statistics provides a quantitative summary of the dataset's characteristics. Key statistics such as mean, standard deviation, minimum, and maximum pixel values were computed. These metrics offer insights into the intensity distribution and contrast of the images, aiding in understanding the dynamic range and variability.

A screen shot of a computer program

Description automatically generated

* 1. **Display of Multiple Images**

Visualizing multiple images allows for qualitative assessment of the dataset. A grid of sample images was displayed to provide an overview of the dataset's diversity, helping detect inconsistencies or artifacts that may affect analysis.

A group of images of the body

Description automatically generated

* 1. **Pixel Value Distribution Investigation**

Investigating pixel value distribution is essential for understanding image quality and contrast. A histogram of pixel intensities was plotted, revealing the range and frequency of pixel values. This analysis aids in determining appropriate preprocessing techniques, such as normalization or contrast enhancement.

**A black screen with text

Description automatically generated**

**A graph of a number of data

Description automatically generated**

**5. Data Transformation**

The code balances the dataset to ensure that each class has an equal number of samples, preventing model bias towards more frequent classes. It then normalizes the data by scaling it to a range between 0 and 1, which aids in model convergence during training. The balanced and normalized data is converted into TensorFlow datasets, shuffled, and batched, making it ready for efficient training and evaluation within the TensorFlow framework

A screen shot of a computer

Description automatically generated

**Model Exploration**

**1. Model Selection**

The proposed combination of CNN and Transformer architectures aims to leverage the strengths of both approaches: the CNN's ability to extract various features and the Transformer's capability in sequence modeling. This combination is intended to improve the detection of kidney abnormalities in CT scans. The use of Transformer attention maps highlights the regions of the image that the Transformer focuses on, thereby enhancing the interpretability of the detection results. Explainable AI (XAI) is then employed to further improve the model's interpretability by providing justifications for the classifications made and highlighting the dependencies of its decisions using Gradient-weighted Class Activation Mapping (Grad-CAM). Integrating Transformers and XAI in the TransConvNet project is crucial because Transformers address the challenge of capturing complex patterns in CT images of kidneys, which is essential for accurately detecting abnormalities. XAI ensures that the model's predictions are interpretable, fostering trust among clinicians and aiding in decision-making processes. This hybrid approach enhances diagnostic accuracy and provides detailed explanations for each prediction, thereby improving the system's overall reliability.

**2. Model Architecture**

The hybrid model integrates a CNN followed by Transformer layers and CNN layer, with XAI utilized to gain a more comprehensive understanding of the trained model.

* **CNN Layers**: The CNN model consists of three convolutional layers, each followed by max-pooling. The convolutional layers have 32, 64, and 128 filters, respectively, with a filter size of 3x3. These layers are designed to progressively extract complex features from the image while reducing the spatial dimensions using max-pooling. The Rectified Linear Unit (ReLU) activation function is used to introduce non-linearity into the model.
* **Transformer Layers**: The features extracted by the CNN are reshaped and fed into a multi-head attention layer with four heads to capture complex relationships and dependencies within the CT images. This is followed by a dropout layer (0.1) to mitigate overfitting, a CNN layer to provide interability by integrating it with the Grad-CAM model.Finally a normalization layer to stabilize and speed up the training process. Subsequently, there are two dense layers connected to another dropout layer (0.1). The first dense layer has 128 neurons and employs the ReLU activation function, followed by a second dense layer that projects back to the original input dimension. Another normalization layer follows the dropout layer.
* **Global Pooling and Output**: The transformed features pass through a global pooling layer to generate a single feature map for each corresponding category. Finally, the output is fed into a SoftMax layer with four neurons to produce the final classification.

It is worth mentioning that the last CNN layer of the model contains an optimal combination of detailed spatial information and higher-level semantics. Grad-CAM calculates the gradient of this information to produce a coarse localization map highlighting the crucial regions in the CT image, assisting in predicting the target class.

**2. Model Training**

The model was trained using the architecture detailed on table 1. It worth mentioning that the data has been split into three datasets: training, validation, and testing datasets.

Table 1: The model’s architecture.

|  |  |  |  |
| --- | --- | --- | --- |
| Layer | Filter | Kernel Size | Activation Function |
| Conv-1 | 32 | 3x3 | ReLU |
| Max-pooling-1 | - | 2x2 | - |
| Conv-2 | 64 | 3x3 | ReLU |
| Max-pooling-2 | - | 2x2 | - |
| Conv-3 | 128 | 3x3 | ReLU |
| Max-pooling-3 | - | 2x2 | - |
| Multi-Head Attention (4) | - | - | - |
| Dropout (0.1) | - | - | - |
| Normalization | - | - | - |
| Dense-1 | 128 | - | ReLU |
| Dense-2 | 128 | - | ReLU |
| Dropout (0.1) | - | - | - |
| Normalization | - | - | - |
| Conv-4 | 256 | 3x3 | ReLU |
| Global Pooling | - | - | - |
| Dense | 4 | - | Softmax |

**3. Model Evaluation**

Present the evaluation metrics used to assess the model's performance. Include confusion matrices, ROC curves, or any other relevant visualizations.

The confusion matrix in table 2 shows how well the model performs for each class. Here’s the breakdown:

* **Cyst:** Out of 746 instances of Cyst, 732 were correctly classified. There were 14 misclassifications (3 as Normal, 6 as Stone, and 5 as Tumor).
* **Normal:** Out of 742 instances of Normal, 734 were correctly classified. There were 8 misclassifications (3 as Cyst, 4 as Stone, and 1 as Tumor).
* **Stone:** Out of 783 instances of Stone, 729 were correctly classified. There were 54 misclassifications (12 as Cyst, 18 as Normal, and 24 as Tumor).
* **Tumor:** Out of 762 instances of Tumor, 733 were correctly classified. There were 29 misclassifications (4 as Cyst, 9 as Normal, and 16 as Stone).

This matrix reveals that the model has high accuracy for all classes but shows slightly more confusion between Stone and other classes. The largest number of misclassifications occurs in the Stone class, indicating potential areas for model improvement.

Table 2: The confusion matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cyst | Normal | Stone | Tumor |
| Cyst | 732 | 3 | 6 | 5 |
| Normal | 3 | 734 | 4 | 1 |
| Stone | 12 | 18 | 729 | 24 |
| Tumor | 4 | 9 | 16 | 733 |

Table 3 shows the performance metrics for all classified classes.

* **Precision:** High for all classes, indicating most predictions for each class are correct.
* **Recall (Sensitivity):** Also high, particularly for Normal and Tumor, indicating the model correctly identifies most true instances.
* **Specificity:** Very high for all classes, showing the model accurately identifies non-instances.
* **F1 Score:** Reflects a balanced performance across Precision and Recall, with slightly lower values for the Stone class due to higher misclassifications.

Table 3: The performance metrics.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | Specificity | F1 Score |
| Cyst | 0.9813 | 0.9747 | 0.9939 | 0.9780 |
| Normal | 0.9607 | 0.9892 | 0.9870 | 0.9747 |
| Stone | 0.9655 | 0.9310 | 0.9885 | 0.9479 |
| Tumor | 0.9607 | 0.9709 | 0.9870 | 0.9658 |

**4. Code Implementation**

A computer screen shot of text

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen with text and images

Description automatically generated

A screenshot of a computer program

Description automatically generated