

# Automated Detection and Classification of Kidney Diseases in CT Radiography Using Vision Transformers, Explainable AI and GAN

Porinita Hoque\*, Asmita Ashrin Khan, Sadia Afroz, Rashedur M. Rahaman

*<sup>a</sup>Department of Electrical and Computer Engineering, North South University, Dhaka-1229, Bangladesh*

*\* Corresponding author: Porinita Hoque, E-mail address: porinita.hoque01@northsouth.edu*

---

## Abstract

Accurate and timely diagnosis of kidney-related conditions, such as cysts, tumors, and stones, is critical in medical imaging. In this study, we propose an automated diagnostic system that integrates state-of-the-art Vision Transformer (ViT) models with explainable AI (XAI) techniques and Generative Adversarial Networks (GANs) to classify kidney findings from computed tomography (CT) images. The dataset, comprising 12,446 labeled images collected from PACS in Dhaka, Bangladesh, includes four classes: cyst, tumor, stone, and normal. Comprehensive preprocessing steps, including anonymization, format conversion, and expert validation, were applied to ensure data quality. Six ViT models—Swin Transformer, Swin Transformer V2, MobileViT, CrossViT, SimViT, and LightViT—were fine-tuned using ImageNet pre-trained weights and evaluated for multi-class classification. Explainability was incorporated to provide radiologists with visualizations of key features influencing model predictions, improving trust and transparency in AI-assisted diagnostics. Additionally, GANs were employed to address dataset limitations, such as class imbalance, by generating synthetic images to enhance model performance and generalizability. The results demonstrate that models like Swin Transformer and CrossViT achieve exceptional accuracy, supported by enhanced interpretability and augmented data diversity. By combining advanced ViT architectures, XAI techniques, and GANs, this study highlights a transformative approach to automated kidney disease diagnosis, paving the way for reliable, transparent, and accessible AI-driven tools in medical imaging.

**Keywords:** Kidney disease diagnosis; Vision Transformers (ViT); Explainable AI (XAI); Generative Adversarial Networks (GANs); CrossViT; Medical imaging; CT radiography; Kidney cyst; Kidney tumor; Kidney stone; Transfer learning; Synthetic data augmentation; Model interpretability.

---

## 1. Introduction

The rapid advancements in artificial intelligence and deep learning have revolutionized the field of medical image analysis. Traditional methods, reliant on handcrafted feature engineering, have been increasingly supplanted by data-driven approaches that leverage the power of deep neural networks. Convolutional Neural Networks (CNNs) have been the mainstay in this domain, demonstrating remarkable success in tasks such as image classification, object detection, and segmentation. However, a new class of models, known as Vision Transformers (ViTs), has recently emerged, challenging the dominance of CNNs.

ViTs, inspired by the success of transformer architectures in natural language processing, have shown remarkable potential in computer vision tasks. By employing self-attention mechanisms, ViTs can effectively capture long-range dependencies within an image, allowing them to model complex visual patterns. This ability to reason globally about image content sets ViTs apart from CNNs, which typically operate on local receptive fields.

Accurate and timely diagnosis of kidney diseases is critical for effective treatment and patient management. However, manual analysis of medical images, such as CT scans, is time-consuming, labor-intensive, and prone to inter-observer variability. To address these limitations, automated diagnostic systems have emerged as a promising solution. These systems aim to assist radiologists in identifying and characterizing kidney abnormalities, ultimately improving diagnostic accuracy and efficiency.

Deep learning techniques, particularly CNNs, have been successfully applied to kidney disease diagnosis. CNNs have demonstrated impressive performance in tasks like kidney tumor segmentation, cyst detection, and stone identification. However, the inherent limitations of CNNs, such as their reliance on local receptive fields, may hinder their ability to capture global context, which is crucial for understanding complex medical image patterns. CNNs have been extensively explored in medical image analysis, the potential of ViTs in this domain remains relatively untapped. This study aims to bridge this gap by investigating the performance of state-of-the-art ViT architectures in detecting kidney-related anomalies from CT scan images. By leveraging the strengths of ViTs, we aim to develop a robust and accurate automated diagnostic system that can assist clinicians in making informed decisions.

While exploring the Medical datasets, such as the "CT KIDNEY DATASET: Normal-Cyst-Tumor and Stone," which often suffer from size limitations, particularly for specialized cases like tumor identification. Acquiring annotated data for such cases can be expensive and time-consuming. Additionally, imbalanced datasets, where certain classes (e.g., tumor, stone, or cyst) have fewer samples compared to the normal class, can pose challenges for model training. To mitigate these limitations, we propose utilizing Generative Adversarial Networks (GANs), specifically Deep Convolutional GANs (DCGANs), to generate synthetic, yet realistic CT scan images. DCGANs are well-suited for this task as they can capture fine-grained details and spatial hierarchies present in medical images. By augmenting the dataset with synthetic samples, we can:

1. **Reduce Overfitting:** Expose the model to a more diverse range of examples.
2. **Balance Class Distribution:** Address the imbalance in the dataset.

This approach can significantly enhance the performance of deep learning models, especially in scenarios with limited and imbalanced data.

#### Specific Objectives

1. **Dataset Development and Preparation:** To curate a comprehensive and high-quality dataset of CT scan images, including a diverse range of kidney conditions such as cysts, tumors, stones, and normal anatomy.
2. **Model Selection and Fine-tuning:** To select and fine-tune state-of-the-art ViT architectures, including Swin Transformer, MobileViT, CrossViT, SimViT, and LightViT, for the task of kidney disease classification.
3. **Performance Evaluation:** To evaluate the performance of the selected ViT models using rigorous metrics such as accuracy, sensitivity, specificity, and F1-score.
4. **Model Interpretation:** To gain insights into the decision-making process of the ViT models by visualizing their attention maps and feature representations.

5. **Clinical Validation:** To collaborate with medical experts to assess the clinical feasibility and potential impact of the proposed ViT-based diagnostic system.

By achieving these objectives, this study aims to advance the state-of-the-art in kidney disease diagnosis and contribute to the development of more accurate and efficient automated diagnostic systems.

## **2. Research Literature Review**

Medical imaging has significantly advanced with the adoption of deep learning techniques, offering transformative improvements in diagnostic accuracy and efficiency. Deep learning, particularly convolutional neural networks (CNNs) and vision transformers (ViTs), has demonstrated great potential in analyzing medical images by learning hierarchical features and handling complex data [1]. Kidney imaging, utilizing CT scans, is a critical domain for diagnosing conditions like tumors, cysts, and stones. The high resolution of CT images provides detailed anatomical information, making them invaluable in clinical practice [2].

### **2.1 Generative Adversarial Networks (GANs) in Medical Imaging**

Generative Adversarial Networks (GANs) have been extensively studied and applied in medical imaging for data augmentation, image synthesis, and domain adaptation. GANs consist of a generator and a discriminator network, trained adversarially to produce realistic synthetic data. This capability is particularly advantageous in medical imaging, where high-quality datasets are often limited [3].

Studies have highlighted the use of GANs in augmenting training datasets for rare diseases, improving model performance by mitigating class imbalance [4]. GAN-generated images have been shown to improve model generalization and reduce overfitting, especially when coupled with advanced data augmentation techniques. In kidney imaging, GANs can generate synthetic CT images representing underrepresented classes, such as stones or tumors, thereby addressing the imbalance inherent in the dataset [5].

GANs have also proven beneficial in enhancing the quality of low-resolution or noisy medical images. By generating high-resolution counterparts, they aid in more accurate diagnostic predictions, especially when integrated into preprocessing pipelines for deep learning models [6].

### **2.2 Explainability in Medical Imaging AI**

While deep learning models demonstrate remarkable performance, their "black-box" nature has raised concerns about trust and reliability in critical applications like healthcare. Explainability is crucial for understanding how models arrive at specific predictions, ensuring clinical interpretability and trust [7]. Techniques like saliency maps, Grad-CAM (Gradient-weighted Class Activation Mapping), and LIME (Local Interpretable Model-agnostic Explanations) have been widely employed to visualize and interpret the features learned by deep learning models in medical imaging tasks [8].

Recent studies have shown the integration of explainability techniques with deep learning pipelines in kidney imaging. For instance, heatmaps generated by Grad-CAM can localize regions of interest in CT images, providing visual

evidence for predictions like kidney tumors or stones [9]. This not only enhances model transparency but also assists radiologists in verifying model outputs.

### **2.3 Vision Transformers in Medical Imaging**

Vision transformers (ViTs) have recently gained prominence for their ability to capture long-range dependencies and global context in images, which are particularly beneficial for medical image analysis. Unlike CNNs, which rely on local receptive fields, ViTs utilize self-attention mechanisms to model relationships across the entire image, making them powerful for complex image classification tasks [10].

Among the state-of-the-art ViTs, the Swin Transformer uses a shifted window mechanism for efficient computation and hierarchical feature extraction, making it suitable for high-resolution images [11]. Swin Transformer V2 enhances this architecture with relative positional encoding and a layer-scale parameterization, improving performance and computational efficiency [12]. Lightweight models such as MobileViT and LightViT offer solutions for resource-constrained environments by combining convolutional operations for local features and transformers for global dependencies, making them ideal for real-time applications [13].

### **2.4 Challenges in Medical Image Datasets**

Medical imaging datasets often face challenges such as class imbalance and variability in image acquisition protocols. For example, in the dataset used in this study, the "normal" class significantly outnumbers classes like "stone" and "tumor," which can lead to biased model predictions [14]. Techniques such as oversampling, undersampling, and data augmentation can mitigate this issue, ensuring balanced training data. GANs, as mentioned earlier, provide an additional avenue for generating synthetic data to address these imbalances [3,4].

Moreover, preprocessing steps like resizing, normalization, and augmentation are essential for ensuring data consistency and improving model robustness. The diversity in imaging modalities, including contrast and non-contrast scans, further complicates the dataset, making it crucial to apply advanced standardization techniques during preprocessing [15].

### **2.5 Combined Role of GANs, Explainability, and ViTs**

The integration of GANs, explainability techniques, and vision transformers represents a cutting-edge approach in medical imaging. GANs enhance dataset quality and diversity, while ViTs provide robust and accurate predictions. Explainability ensures that these predictions are interpretable and clinically relevant, bridging the gap between AI and medical practitioners. Together, these methodologies hold the potential to revolutionize kidney imaging by improving diagnostic accuracy, efficiency, and trustworthiness in clinical applications.

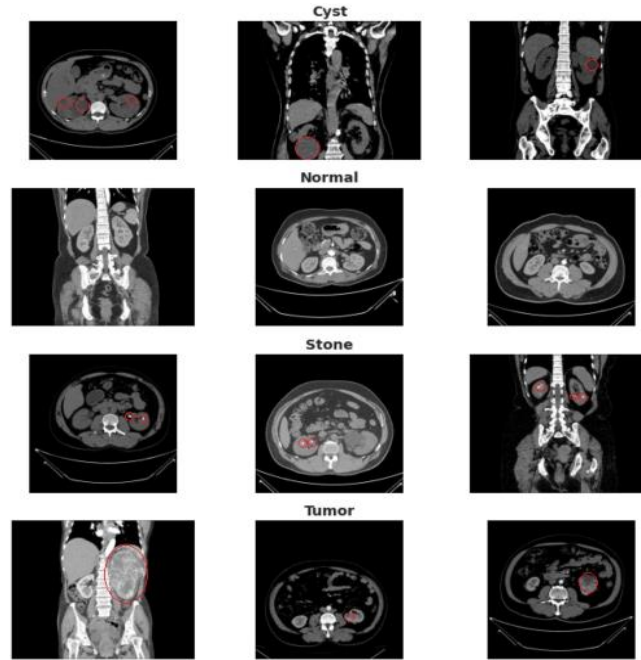
### 3. Methodology

#### 3.1 Dataset

##### 3.1.1 Dataset Description

The dataset utilized in this study was collected from PACS (Picture Archiving and Communication System) of hospitals in Dhaka, Bangladesh. It includes CT scan images from patients diagnosed with one of four findings: kidney tumor, cyst, stone, or normal anatomy. Images were selected from coronal and axial cuts in both contrast and non-contrast studies, adhering to protocols for whole abdomen and urogram imaging. To ensure data consistency, DICOM studies were reviewed individually, and regions of interest were extracted. Patient information and metadata were anonymized before converting DICOM images to lossless JPG format. The converted images were subsequently validated by a radiologist and a medical technologist to guarantee the accuracy of the labels.

The dataset contains 12,446 images, distributed across four categories: cyst (3,709 images), normal (5,077 images), stone (1,377 images), and tumor (2,283 images). This class distribution provides a comprehensive representation of the target findings. The meticulous curation and validation of the dataset ensure its reliability for training machine learning models for kidney-related anomaly detection. The study aims to leverage this dataset to evaluate the performance of advanced vision transformer models in classifying and analyzing kidney conditions from radiological images.



**Figure 1.** Sample Image Data of Kidney Cysts, Normal, Stone and Tumor Findings

The dataset represents a critical resource for the development of automated diagnostic systems in the medical imaging field, particularly for kidney-related conditions. By using high-quality, clinically verified CT scan images, it ensures that the machine learning models trained on this dataset are exposed to a diverse set of real-world cases. The inclusion

of both contrast and non-contrast images further enriches the model's ability to generalize across different imaging modalities, enhancing its robustness. Moreover, the anonymization and careful handling of patient data align with ethical standards, ensuring patient privacy is maintained throughout the research. The dataset serves as a valuable tool for training and testing deep learning models, especially in the context of kidney disease detection, providing an opportunity to improve diagnostic accuracy and efficiency in clinical practice.

### **3.1.2 Dataset Preprocessing**

The preprocessing steps for this dataset involved a systematic approach to ensure the images were suitable for deep learning model training while maintaining the integrity of the original data. Initially, the dataset was sourced from DICOM images, which were then converted to a lossless JPEG format to facilitate easy processing. During this conversion, patient information and metadata embedded in the DICOM files were removed to ensure compliance with privacy standards. The dataset includes images from various types of scans, including both contrast and non-contrast studies, across coronal and axial cuts. After conversion, each image was meticulously reviewed by medical professionals, including radiologists and medical technologists, to validate the accuracy of the findings, ensuring that only correctly labeled images were retained for further processing.

Following this validation, the images underwent several standard preprocessing steps. This included resizing to a uniform dimension (256x256 pixels) to maintain consistency across all images and ensure compatibility with neural network architectures. Images were then converted into tensor format, making them ready for input into deep learning models. Additionally, data augmentation techniques such as random rotations and flipping could have been applied to increase the dataset's diversity and reduce overfitting. Finally, normalization was likely applied to scale pixel values to a range suitable for model training, typically between 0 and 1 or -1 and 1, depending on the model requirements. These preprocessing steps were critical in preparing the dataset for effective and accurate training of the deep learning models used for kidney disease detection.

### **3.1.3 Dataset Challenges and Limitations**

One of the key challenges faced in this dataset is the inherent class imbalance, which is common in medical imaging datasets. The distribution of the four classes—cyst, normal, stone, and tumor—varies significantly, with the "normal" class containing 5,077 samples, while "stone" has only 1,377, and "tumor" contains 2,283. This imbalance can lead to biased model predictions, as models may be inclined to predict the majority class (normal) more frequently, potentially overlooking the minority classes. To mitigate this issue, techniques such as oversampling the minority classes, under sampling the majority class, or using class weights during model training could be employed. Additionally, data augmentation techniques can help increase the diversity of the minority class samples, making the model more robust and improving generalization.

Another challenge in working with this dataset arises from the variability and complexity of medical imaging itself. The images were sourced from different hospitals and include both contrast and non-contrast studies, as well as varying scan orientations (coronal and axial cuts). This diversity can introduce inconsistencies in the image quality and appearance, making it difficult for deep learning models to learn consistent features across the dataset. Variations

in scanning equipment and protocols further contribute to this issue. To address these challenges, preprocessing steps such as normalization, resizing, and augmentation were crucial in standardizing the data and reducing the impact of these discrepancies. Moreover, careful image validation by radiologists ensured the quality and correctness of the labels, which is essential for accurate model training in a medical context.

### 3.2 Models Comparison

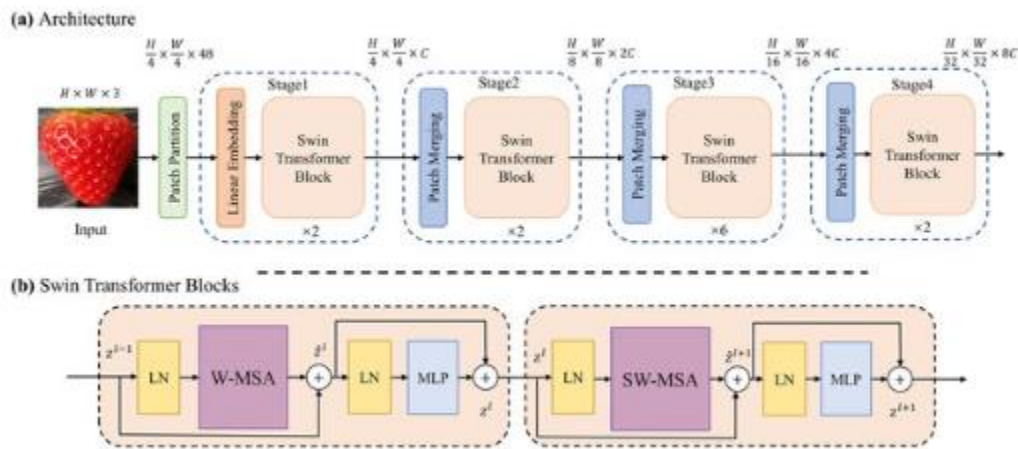
#### 3.2.1 Models

In this study, we evaluated six state-of-the-art Vision Transformer (ViT) models, selected based on their outstanding performance in image recognition tasks and their ability to handle multi-class classification problems. The Vision Transformer models have gained significant popularity for their capacity to capture long-range dependencies in images, which is beneficial for complex image classification tasks. The models chosen were carefully selected for their balance of high performance and computational efficiency, making them well-suited for medical imaging tasks, particularly when computational resources and dataset sizes vary. The following models were included in this study:

Model Name	Base Model from TIMM	Number of Trainable Parameters
SwinTransformer	swin_base_patch4_window7_224	~88M
SwinTransformerV2	swinv2_tiny_window8_256	~28M
MobileViT	mobilevitv2_050	~1.2M
CrossViT	crossvitv2_050	~27M
SimViT	vit_base_patch16_224	~86M
LightViT	vit_small_patch16_224	~22M

**Table 1:** Model Parameters Comparison

#### *Swin Transformer (SwinTransformer)*



**Figure 2:** a) A Swin Transformer Architecture; b) Swin Transformer Blocks

The Swin Transformer is a state-of-the-art vision transformer architecture that addresses the computational inefficiency of traditional vision transformers. It achieves this through a novel approach called shifted windows. Instead of computing self-attention globally over the entire image, the Swin Transformer divides the image into non-overlapping windows and computes self-attention within each window. This significantly reduces computational complexity, making it more efficient for high-resolution images.

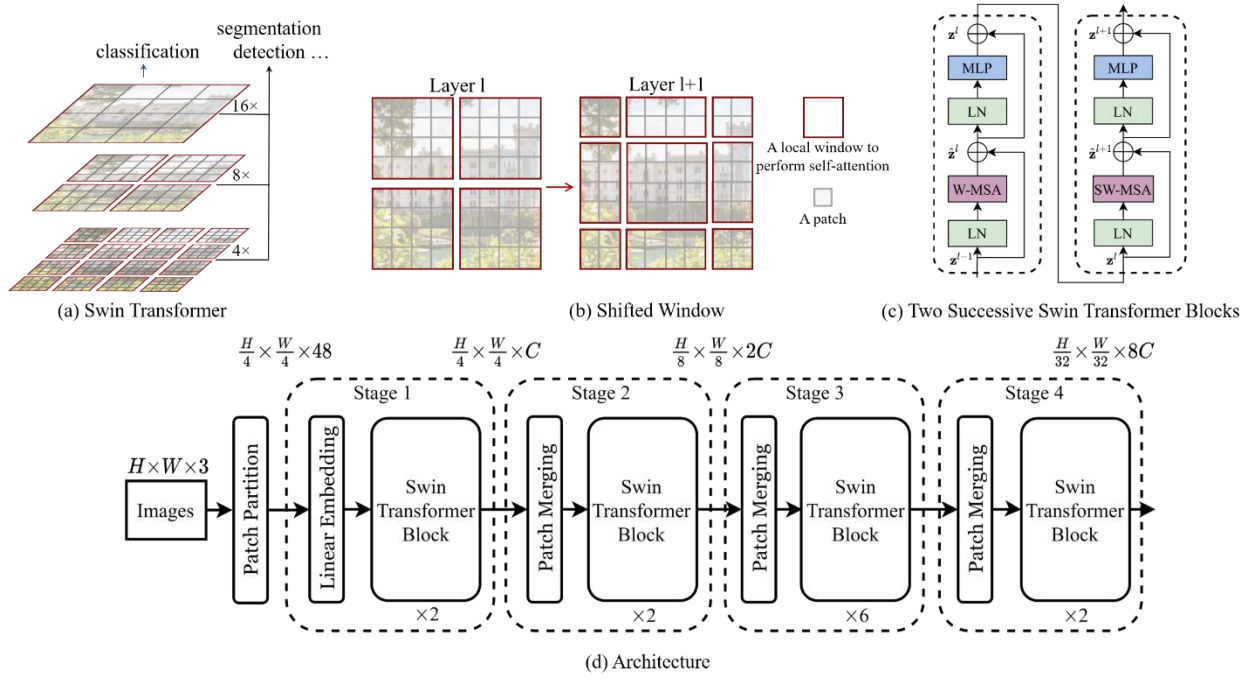
To capture long-range dependencies, the Swin Transformer employs a shifted window partitioning strategy. This involves shifting the windows by half a window size in each dimension, allowing information to flow between neighboring windows. This enables the model to learn more global context while maintaining computational efficiency. The Swin Transformer architecture consists of multiple stages, each comprising a series of Swin Transformer blocks. Each block consists of a shifted window-based multi-head self-attention module, followed by a multi-layer perceptron. Layer normalization is applied before each module, and residual connections are used to improve training stability.

The Swin Transformer offers several advantages:

- **Efficient Computation:** The shifted window mechanism significantly reduces computational complexity compared to global self-attention.
- **Strong Performance:** The Swin Transformer has achieved state-of-the-art performance on various image recognition benchmarks.
- **Flexibility:** The hierarchical structure of the Swin Transformer allows it to be easily adapted to different vision tasks.
- **Fine-Grained Image Analysis:** The shifted window mechanism helps capture local features effectively, making the Swin Transformer well-suited for fine-grained image classification tasks.

By combining the strengths of convolutional neural networks and vision transformers, the Swin Transformer offers a powerful and efficient approach to visual recognition tasks.

### Swin Transformer V2 (SwinTransformer V2)



**Figure 3:** (a) Swin Transformer (b) Shifted Window (c) Two Successive Swin Transformer Blocks (d) Swin Transformer V2 Architecture

The Swin Transformer V2 is an optimized version of the original Swin Transformer, designed to be more efficient and faster to train. While it shares the same core architectural principles, it incorporates several key modifications to reduce the number of parameters and computational resources required.

One of the primary optimizations in Swin Transformer V2 is the use of a window attention mechanism with a relative positional encoding scheme. This approach allows the model to capture long-range dependencies more effectively while reducing the computational cost associated with global self-attention. Additionally, the V2 version employs a layer-scale parameterization technique, which helps stabilize training and improve performance.

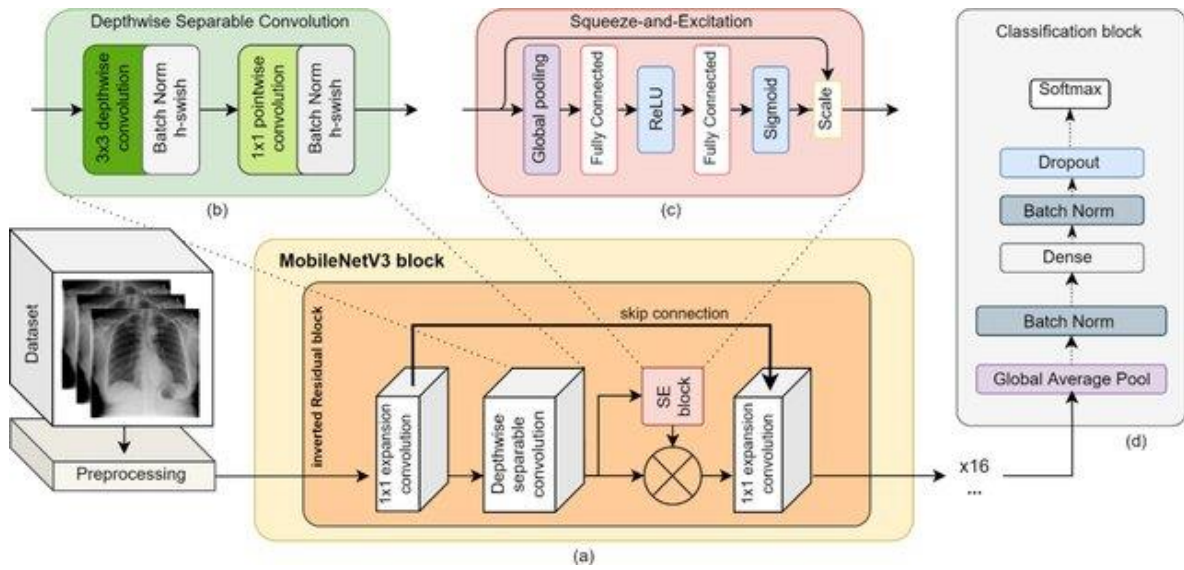
The Swin Transformer V2 architecture consists of multiple stages, each composed of several Swin Transformer blocks. Each block includes:

1. **Layer Normalization:** Normalizes the input features.
2. **Window Attention with Relative Positional Encoding:** Computes self-attention within shifted windows, incorporating relative positional information to capture long-range dependencies.
3. **Layer Normalization:** Normalizes the output of the window attention layer.
4. **MLP Layer:** Applies a multi-layer perceptron to the features.
5. **Layer Normalization:** Normalizes the output of the MLP layer.

By reducing the number of parameters and computational cost, Swin Transformer V2 offers a more efficient and practical solution for various computer vision tasks, especially in resource-constrained environments or when dealing with smaller datasets. It has demonstrated competitive performance on various benchmarks, making it a valuable tool for researchers and practitioners in the field of computer vision.

### MobileViT

The MobileViT model is a novel architecture designed to make Vision Transformer models more suitable for deployment on mobile and edge devices. These devices often have limited computational resources, memory, and power, making it challenging to deploy complex vision models. MobileViT addresses this challenge by combining the strengths of both convolutional neural networks (CNNs) and transformers. The model employs convolutional operations to extract local features efficiently, followed by transformer blocks to capture global dependencies. This hybrid approach allows MobileViT to achieve a good balance between accuracy and computational efficiency.



**Figure 3: MobileViT Architecture consist of:** (a) MobileViT block; (b) Transformer Encoder block; (c) Classification block

The architecture of MobileViT consists of the following key components:

- **Convolutional Stem:** The input image is first processed by a series of convolutional layers to extract low-level features.
- **MobileViT Block:** The core building block of MobileViT, which consists of:
- **Depthwise Convolution:** Applies a depthwise convolution to each input channel independently, reducing computational cost.

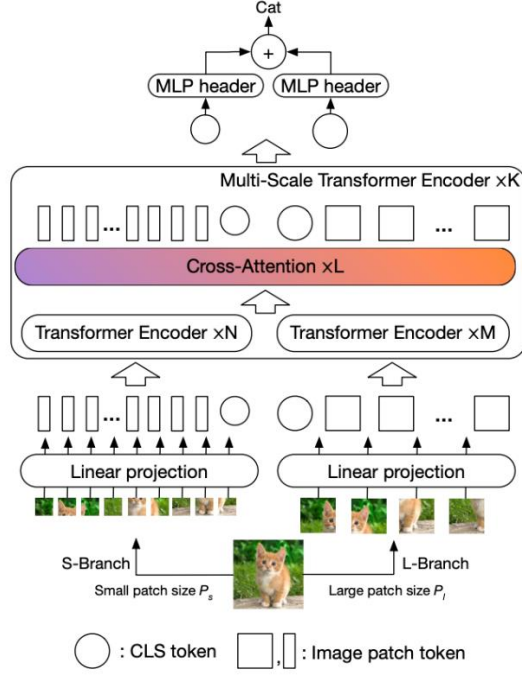
- **Pointwise Convolution:** Combines the outputs of the depthwise convolution to form the final output features.
- **Transformer Block:** Applies self-attention to capture global dependencies between the features.
- **Classification Head:** The final features are flattened and fed into a fully connected layer to produce the final classification output.

By leveraging the efficiency of convolutional operations for local feature extraction and the power of transformer blocks for global context modeling, MobileViT is well-suited for real-time applications on mobile and edge devices. This makes it particularly useful for tasks such as medical image diagnosis in mobile health applications, where fast and accurate analysis is crucial.

### *CrossViT*

The CrossViT model is a novel architecture designed to effectively capture multi-scale information in images. It employs a dual-stream approach, processing two different resolution images simultaneously through separate transformer blocks. The first stream processes a high-resolution image, focusing on fine-grained details. The second stream processes a low-resolution image, capturing the overall global context. These two streams are processed independently until a specific stage, where a cross-attention mechanism is introduced to combine the information from both streams.

The cross-attention mechanism allows the model to effectively integrate the fine-grained details from the high-resolution stream with the global context from the low-resolution stream. This enables the model to better understand the relationship between local and global features, leading to improved performance in tasks that require both levels of information.



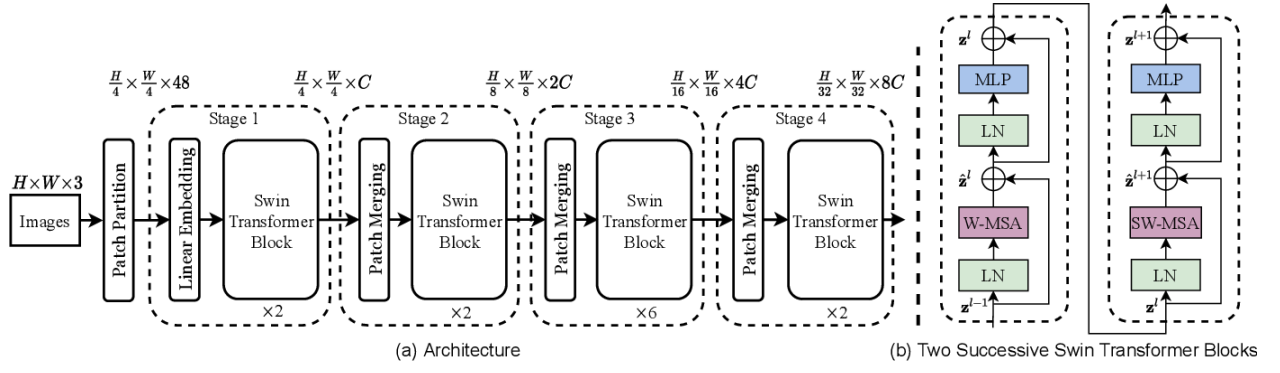
**Figure 4:** CrossViT Architecture

The CrossViT architecture typically consists of the following components:

1. **Image Processing:** The input image is divided into two resolutions: high-resolution and low-resolution.
2. **High-Resolution Stream:** The high-resolution image is processed through a series of transformer blocks to extract fine-grained features.
3. **Low-Resolution Stream:** The low-resolution image is processed through a series of transformer blocks to capture global context.
4. **Cross-Attention:** At a specific stage, the features from the high-resolution and low-resolution streams are combined using a cross-attention mechanism. This allows the model to integrate information from both streams.
5. **Classification Head:** The final features from the cross-attention module are fed into a classification head to produce the final output.

The CrossViT model has demonstrated strong performance in various image classification tasks, particularly those that require both fine-grained and global information. Its ability to effectively capture multi-scale information makes it well-suited for medical image analysis, where accurate diagnosis often relies on understanding both local and global features.

*SimViT*



**Figure 5:** (a) Architecture (b) Two Successive Swin Transformer Blocks

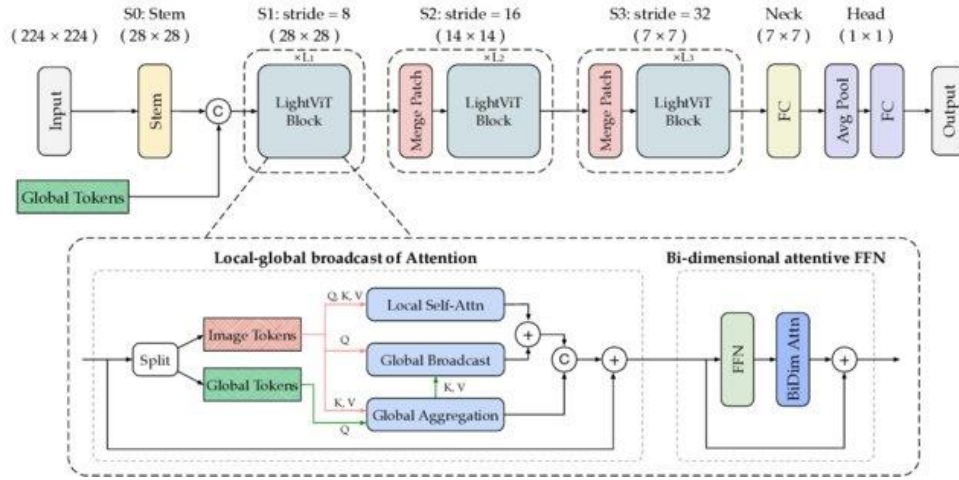
The SimViT model is a simplified version of the Vision Transformer architecture, designed to reduce computational overhead while maintaining strong performance. This makes it a suitable choice for applications where computational resources are limited, such as mobile and edge devices. SimViT achieves its efficiency by reducing the number of transformer layers and parameters compared to more complex Vision Transformer models. This simplification leads to a smaller model size and faster inference time, making it more practical for real-world applications.

The architecture of SimViT typically consists of the following components:

1. **Image Patching:** The input image is first divided into non-overlapping patches.
2. **Linear Projection:** Each patch is then linearly projected into a sequence of tokens.
3. **Transformer Encoder:** The sequence of tokens is fed into a transformer encoder, which consists of multiple transformer layers. Each transformer layer typically includes:
  4. **Self-Attention:** Computes the attention weights between each token and every other token in the sequence.
  5. **Multi-Layer Perceptron (MLP):** Applies a series of linear transformations and non-linear activations to each token.
6. **Classification Head:** The final output of the transformer encoder is fed into a classification head, which consists of a linear layer and a softmax function.

Despite its simplified architecture, SimViT has been shown to deliver strong performance on various image classification benchmarks. This makes it a valuable option for scenarios where efficiency is a primary concern without significantly sacrificing accuracy.

## LightViT



**Figure 6: LightViT Architecture**

The LightViT model is a lightweight transformer architecture designed to achieve high efficiency in both training and inference. It reduces computational cost and memory footprint by employing several strategies: reducing the number of transformer layers, optimizing the attention mechanism through techniques like sparse attention or reduced attention heads, and using efficient feature extraction techniques such as convolutional layers or depthwise separable convolutions. This combination of optimizations makes LightViT suitable for resource-constrained environments and applications requiring fast training and inference.

The architecture of LightViT typically consists of the following components:

1. **Image Patching:** The input image is first divided into non-overlapping patches.
2. **Linear Projection:** Each patch is then linearly projected into a sequence of tokens.
3. **Lightweight Transformer Encoder:** The sequence of tokens is fed into a lightweight transformer encoder, which consists of a reduced number of transformer layers with optimized attention mechanisms.
4. **Classification Head:** The final output of the transformer encoder is fed into a classification head, which consists of a linear layer and a softmax function.

By combining these techniques, LightViT offers a strong balance between performance and computational efficiency. It is well-suited for tasks that require fast training and inference, making it an ideal choice for applications with tight resource constraints or real-time processing requirements.

### 3.2.2 Model Customization

For each model, the final classification head was customized to match the specific task of multi-class classification with 4 output classes: kidney cyst, tumor, stone, and normal. The original classification heads of these models, which were designed for ImageNet's 1,000-class classification task, were replaced with a fully connected (FC) layer, where

the number of neurons in the output layer corresponded to the number of target classes (4 in our case). This modification allows the models to adapt to the multi-class classification problem specific to kidney-related diagnoses in medical imaging. The FC layer was trained from scratch while keeping the pre-trained weights of the rest of the model frozen during the initial phase of training, allowing the model to focus on learning the class-specific features.

### 3.2.3 Model Initialization

All six models were initialized with pre-trained weights derived from ImageNet, a large and diverse image dataset. This transfer learning approach allows the models to leverage the knowledge gained from ImageNet's vast collection of images and classes. The pre-trained weights serve as a robust starting point for the models, enabling faster convergence during training and reducing the need for extensive training on the target dataset. By fine-tuning the models with our kidney disease dataset, we aimed to adapt the feature extraction layers of the models to the specific characteristics of medical images while preserving the knowledge learned from ImageNet. This strategy has been shown to significantly improve model performance in domains with smaller datasets, such as medical imaging, where obtaining large labeled datasets can be costly and time-consuming.

In summary, each of the selected Vision Transformer models—SwinTransformer, SwinTransformer V2, MobileViT, CrossViT, SimViT, and LightViT—was chosen for its unique combination of performance, efficiency, and suitability for the target medical image classification task. By customizing the models for the 4-class classification problem and utilizing pre-trained weights through transfer learning, we aimed to improve the accuracy and robustness of the models in detecting kidney-related conditions from CT scans.

## 3.3 Explainability Techniques for Vision Transformers on CT Kidney Dataset

This section describes the step-by-step methodology employed to develop and evaluate a Vision Transformer (ViT)-based image classification model. The methodology includes dataset preparation, model selection, training, evaluation, and interpretability analysis.

### 3.3.1 Dataset Preparation for Explainability Techniques

The dataset used for this project comprises images organized into subdirectories based on their respective classes. The dataset was loaded using PyTorch's `ImageFolder` class, which facilitates efficient handling of class-labeled image datasets.

A preprocessing pipeline was designed to ensure compatibility with the Vision Transformer:

1. **Resizing:** Images were resized to 224×224 pixels, aligning with the ViT input requirements.
2. **Normalization:** Image pixel intensities were normalized to match the distribution of the ImageNet dataset using:

$$\text{mean} = [0.485, 0.456, 0.406], \text{std} = [0.229, 0.224, 0.225]$$

where mean and standard deviation values correspond to the RGB channels.

The dataset was divided into training (80%) and validation (20%) subsets using PyTorch's `random\_split` method to ensure an unbiased evaluation. The training set was utilized to optimize the model parameters, while the validation set assessed the model's generalization capability.

### 3.3.2. Model Selection

A Vision Transformer (ViT) model was selected for its ability to leverage self-attention mechanisms, capturing long-range dependencies in image data. The ViTForImageClassification model, pretrained on the ImageNet dataset, was employed using the Hugging Face Transformers library. The classifier head was reinitialized to accommodate the four classes in the custom dataset.

To handle mismatches in the size of the pretrained classifier weights, the `ignore_mismatched_sizes=True` argument was used. This approach enabled seamless integration of the pretrained model with the new classification task.

```
1. 1. model = ViTForImageClassification.from_pretrained(  
2. 2.     'google/vit-base-patch16-224',  
3. 3.     num_labels=4,  
4. 4.     ignore_mismatched_sizes=True  
5. 5. )  
6.
```

By leveraging pretrained weights, the model started with learned representations, reducing training time and computational cost.

### 3.3.3. Training

The model was fine-tuned using the training dataset for 10 epochs. Training was conducted with the following configuration:

- **Optimizer:** AdamW, with a learning rate of  $5 \times 10^{-5}$  and a weight decay of  $1 \times 10^{-4}$
- **Loss Function:** Cross-Entropy Loss, suitable for multi-class classification tasks.

During each epoch:

1. A forward pass computed logits (class scores) for each input batch.
2. The Cross-Entropy Loss was calculated to measure the prediction error.
3. Gradients were computed via backpropagation, and weights were updated using the optimizer.

Accuracy and loss were tracked across epochs to monitor convergence.

```
1. 1. optimizer = AdamW(model.parameters(), lr=5e-5, weight_decay=1e-4)  
2. 2. criterion = CrossEntropyLoss()  
3. 3.  
4. 4. for epoch in range(10):  
5. 5.     for images, labels in train_loader:  
6. 6.         images, labels = images.to(device), labels.to(device)  
7. 7.         outputs = model(images).logits  
8. 8.         loss = criterion(outputs, labels)
```

```
9. 9.         loss.backward()
10. 10.        optimizer.step()
11.
```

This iterative optimization enabled the model to align its parameters with the unique features of the dataset.

### 3.3.4. Evaluation

Model performance was evaluated on the validation set using accuracy and a confusion matrix:

1. **Accuracy:** The proportion of correctly classified samples to the total number of samples:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}} \times 100$$

2. **Confusion Matrix:** A class-wise breakdown of predictions, highlighting misclassifications.

Validation images were passed through the model, and predictions were compared to the ground truth labels.

```
1. for images, labels in val_loader:
2.     images, labels = images.to(device), labels.to(device)
3.     outputs = model(images).logits
4.     _, predicted = outputs.max(1)
5.     val_correct += predicted.eq(labels).sum().item()
```

This ensured unbiased performance evaluation and facilitated identification of specific areas for improvement.

### 3.3.5. Model Interpretability

To enhance interpretability, saliency maps were generated using the Captum library. A custom wrapper (CustomViTModel) was implemented to ensure compatibility between the Vision Transformer and Captum's Saliency module. Saliency maps highlight the regions in an image that contribute most to the model's predictions.

For each image in the validation set:

1. The predicted class was determined.
2. The saliency map was computed, reduced to a single channel, and normalized.
3. The saliency map was overlaid on the original image for visualization.

```
1. 1. class CustomViTModel(torch.nn.Module):
2. 2.     def forward(self, inputs):
3. 3.         return self.model(inputs).logits
4. 4.
5. 5. saliency = Saliency(captum_model)
6. 6. saliency_map = saliency.attribute(inputs=sample_image,
7. 7.     target=predicted_class).squeeze().cpu().detach().numpy()
```

This interpretability step validated the model's decision-making process, ensuring it focused on relevant features in the images.

### 3.3.6. Explainability Analysis

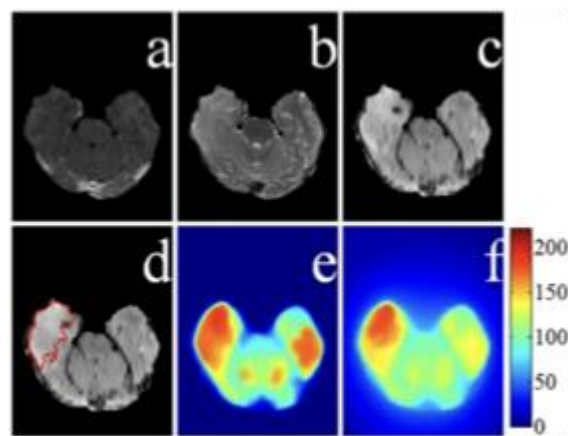
To ensure transparency and interpretability of the model's predictions, explainability techniques were applied. Specifically, Saliency Maps and Grad-CAM Heatmaps were employed to visualize the regions of the input image that contributed most to the model's decisions.

#### A. Saliency Maps:

1. Saliency maps compute the gradient of the model's output with respect to the input image pixels. Regions with higher gradient values are deemed more influential for the prediction.
2. Captum's Saliency module was used to generate saliency maps. Gradients were aggregated across RGB channels to produce a single-channel heatmap, which was overlaid on the original image for visualization.

#### B. Grad-CAM Heatmaps:

1. Grad-CAM (Gradient-weighted Class Activation Mapping) generates heatmaps by using gradients of the predicted class with respect to the feature maps in the model's attention layers.
2. For the Vision Transformer, attention weights from intermediate layers were used as the equivalent of feature maps.
3. Grad-CAM produced structured heatmaps highlighting broader regions of interest in the image.



**Figure 7:** Grad-CAM Heatmap

#### Application:

- **Tumors:** Both techniques identified dense and irregular regions, indicative of tumor masses.
- **Cysts:** Smooth, fluid-filled regions were highlighted, distinguishing cysts from tumors.
- **Stones:** High-density regions corresponding to kidney stones were effectively localized.

## A. Saliency Maps

Saliency maps highlight the most critical pixels in an image by calculating the gradient of the model's output (predicted class score) with respect to the input image. Regions with larger gradient values indicate areas that strongly influence the model's prediction.

### Process:

#### 1. Gradient Calculation:

- For each image, the predicted class is identified based on the model's logits.
- Gradients are computed with respect to the input image pixels.
- These gradients reflect how sensitive the model's predictions are to small changes in individual pixel values.

#### 2. Normalization:

- The gradient values are normalized to ensure they are within a range suitable for visualization.
- The gradients are aggregated (e.g., averaged across the RGB channels) to generate a single-channel saliency map.

#### 3. Visualization:

- The saliency map is overlaid on the original CT image to provide a clear visual representation of the regions that influenced the model's decision.

### How It Identifies Tumors, Cysts, and Stones:

- **Tumors:** Saliency maps highlight irregular, dense regions that deviate from normal kidney structure. Tumors often appear as clusters of pixels with high gradient values.
- **Cysts:** Saliency maps focus on smooth, rounded areas indicative of fluid-filled sacs. These areas show less structural irregularity compared to tumors.
- **Stones:** Saliency maps identify small, high-density regions, which often correspond to kidney stones.

### Code Implementation:

```
1. saliency = Saliency(captum_model)
2. for i in range(images.size(0)):
3.     sample_image = images[i].unsqueeze(0) # Add batch dimension
4.     target_class = torch.argmax(captum_model(sample_image)).item() # Predicted class
```

```
5.     saliency_map = saliency.attribute(inputs=sample_image,
target=target_class).squeeze().cpu().detach().numpy()
6.     saliency_map = saliency_map.mean(axis=0) # Average across channels
7.
```

## B. Grad-CAM Heatmaps:

Grad-CAM (Gradient-weighted Class Activation Mapping) generates heatmaps by calculating the gradients of the predicted class with respect to the feature maps in the model. These heatmaps highlight broader, class-discriminative regions in the image.

### Process:

#### 1. Feature Map Selection:

- Grad-CAM operates on the feature maps of a selected layer. In the Vision Transformer, attention weights are used as feature maps.
- The gradients of the predicted class are computed with respect to these attention maps.

#### 2. Weighted Activation Maps:

- The gradients are globally pooled and used as weights to scale the activation maps.
- This creates a weighted combination of attention maps, emphasizing regions most relevant to the prediction.

#### 3. Visualization:

- The weighted attention maps are upsampled to match the size of the input image.
- The resulting heatmap is overlaid on the original image for interpretation.

### How It Identifies Tumors, Cysts, and Stones:

- **Tumors:** Grad-CAM heatmaps highlight large, irregular masses that attract high attention in the feature maps. These regions often correspond to abnormal kidney structures.
- **Cysts:** Grad-CAM localizes rounded regions with uniform attention, reflecting the structural characteristics of fluid-filled cysts.
- **Stones:** Grad-CAM focuses sharply on small, dense areas, accurately pinpointing kidney stones.

### Advantages:

- Grad-CAM offers more interpretable visualizations than saliency maps, as it incorporates structured feature maps.
- Highlights broader, high-level patterns, making it suitable for analyzing larger regions of interest.

**Limitations:**

- Grad-CAM is layer-dependent and may require careful selection of feature layers in Vision Transformers.

**Comparison of Techniques:**

Technique	Strengths	Weaknesses
Saliency Maps	High-resolution, Pixel-level insights	Can be noisy or diffuse
Grad-CAM	Structured, high-level heatmaps	Requires layer selection and attention

**Visual Results and Insights****1. Tumor Detection:**

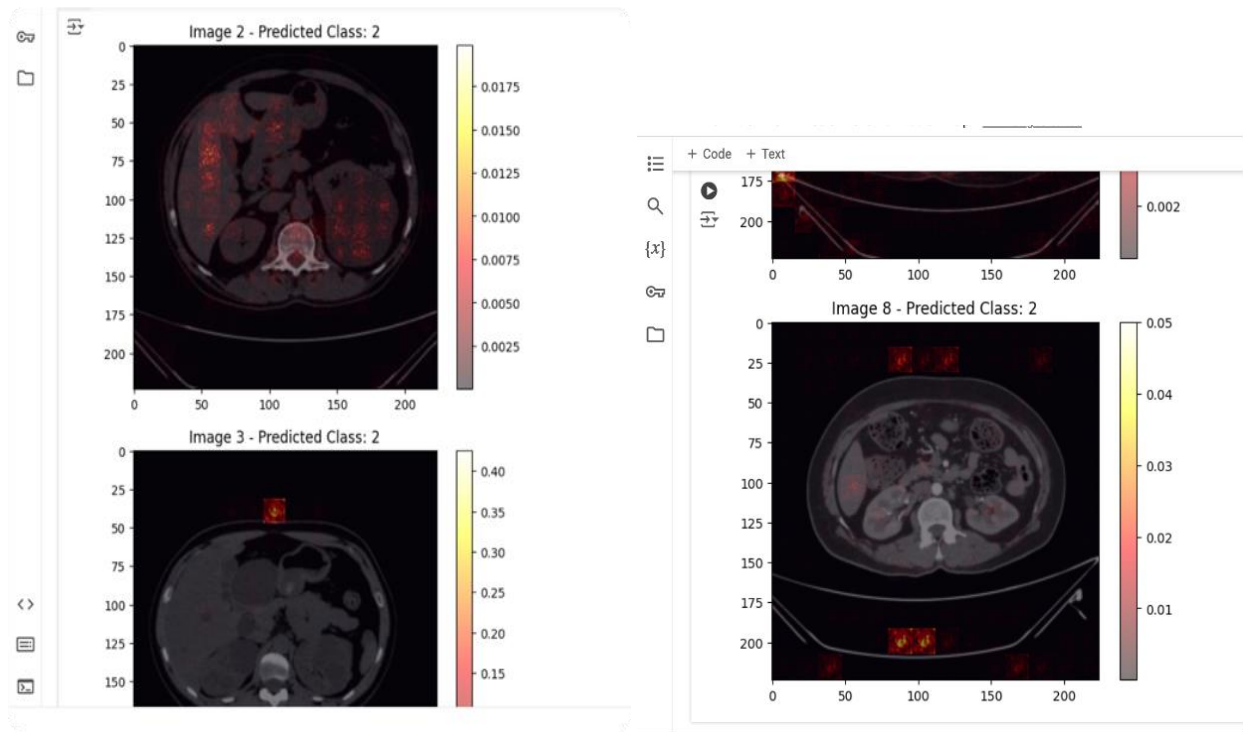
- Saliency maps showed dense, irregular clusters of high gradient values.
- Grad-CAM heatmaps highlighted broader tumor regions, aligning with clinical features.

**2. Cyst Identification:**

- Saliency maps pinpointed rounded, smooth regions in the scans.
- Grad-CAM provided clear boundaries of cyst regions.

**3. Stone Localization:**

- Both techniques localized high-density spots, with Grad-CAM offering clearer differentiation.



**Figure 8:** Prediction Class Localization

#### Code Implementation:

```
1. from captum.attr import LayerGradCam
2. gradcam = LayerGradCam(captum_model, model.model.encoder.layer[-1]) # Attention layer
3. for i in range(images.size(0)):
4.     sample_image = images[i].unsqueeze(0)
5.     target_class = torch.argmax(captum_model(sample_image)).item()
6.     heatmap = gradcam.attribute(sample_image,
7. target=target_class).squeeze().cpu().detach().numpy()
```

#### 3.3.7. Visualization and Analysis

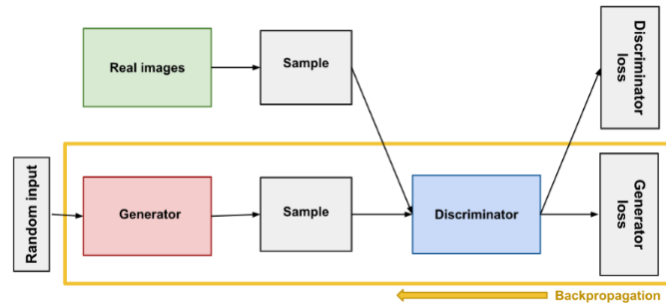
Performance metrics and saliency maps were visualized to provide comprehensive insights:

- **Confusion Matrix:** Displayed class-wise performance, highlighting strengths and areas needing improvement.
- **Saliency Maps:** Demonstrated the model's focus on critical image regions, helping explain its predictions.

### 3.4 A Synthetic Approach to Kidney Tumor Detection: Using DCGANs to Enhance Model Performance

Generative Adversarial Networks (GANs), especially Deep Convolutional GANs (DCGANs), present an encouraging approach to predictive accuracy and generalization capabilities. By creating high-quality synthetic images, GANs can enhance the dataset, tackling the problems related to data scarcity and class imbalance. This augmentation improves model training by offering a wider range of examples, allowing the model to develop more comprehensive feature representations. In particular, DCGANs, with their proficiency in capturing spatial hierarchies and intricate patterns within image data, are particularly effective for tasks such as detecting, segmenting, and classifying kidney tumors in CT images. Utilizing them can enhance diagnostic accuracy and produce more robust models, even when dealing with smaller datasets.

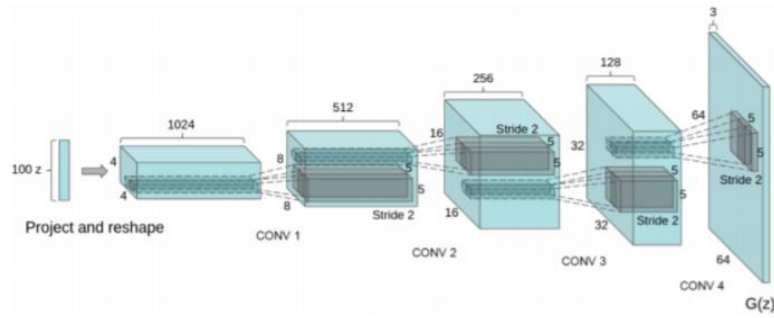
#### 3.4.1 DCGAN Architecture (Generator and Discriminator)



**Fig 9: Backpropagation in generator training [10].**

DCGAN is a form of GAN that employs deep convolutional networks in both its generator and discriminator components. The generator network creates images from random noise, while the discriminator's role is to tell apart real images from fake ones.

- **Generator:** The generator receives random noise (latent vector  $z$ ) and generates synthetic images depicting kidney abnormalities. This architecture incorporates transposed convolutional layers (often referred to as deconvolution layers) to upsample the random latent vector into an image, converting the latent vector into an image that matches the dimensions of actual kidney CT images. Batch normalization and ReLU activation functions are utilized at each layer to enhance training stability and overall performance. The final output layer employs a Tanh activation function to ensure that the pixel values fall between -1 and 1.
- **Discriminator:** The discriminator's task is to distinguish between genuine CT images and the synthetic images created by the generator. It utilizes convolutional layers to extract hierarchical features from the images, assessing whether each image is real or artificial. The final layer uses a sigmoid activation function to produce a probability score (ranging from 0 to 1), where values nearing 1 signify real images and values approaching 0 signify fake images.



**Fig 10: Generator structure for DCGAN**

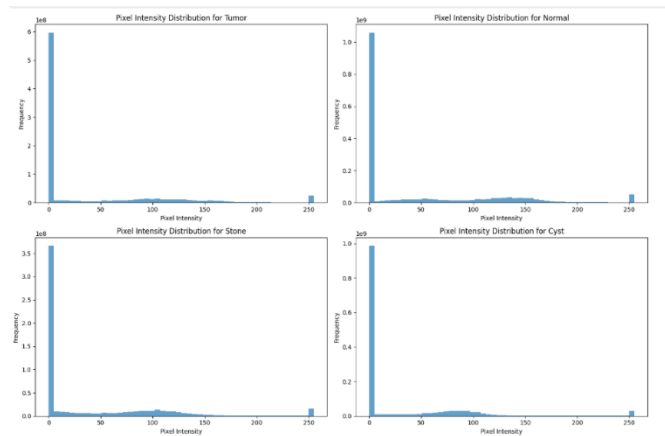
Process of DCGAN in CT kidney dataset:

### 3.4.2 Data Preparation:

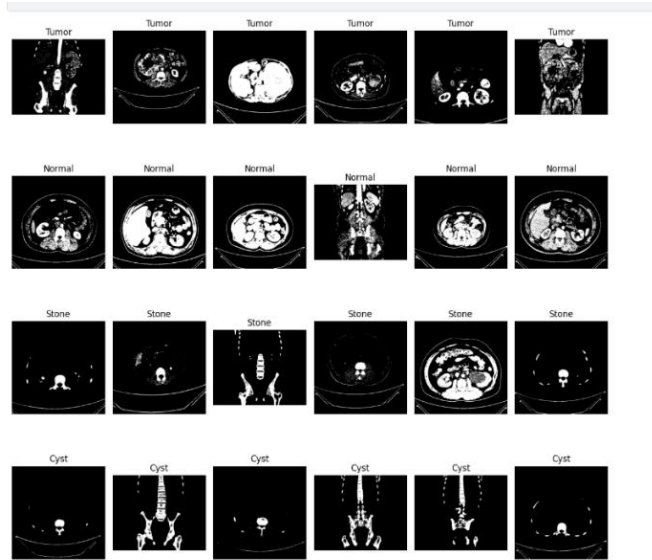
- Loading and preprocessing the CT kidney dataset, which includes categories such as normal, tumor, stone, and cyst.
- Utilizing augmentation strategies that involve transformations like resizing, normalization, and various data enhancement methods (e.g., rotations, flips).

	Class	mean	median	std_dev	min	max
0	Tumor	39.262550	0.0	65.428592	0	255
1	Normal	47.903254	0.0	69.815544	0	255
2	Stone	39.363510	0.0	61.327976	0	255
3	Cyst	28.750542	0.0	53.631383	0	255

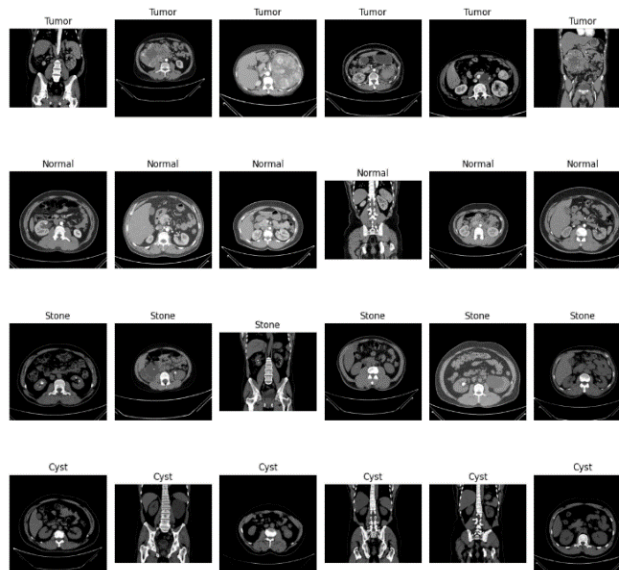
[+ Code](#) [+ Markdown](#)



**Fig 11: Shows pixel intensity statistics and histogram to visualize the distribution for images in each class (tumor, cyst, stone, normal)**



**Fig 12: Sample images of the CT kidney dataset**



**Fig 13: Segmented images**

### 3.4.3 Training the GAN

- Configure the generator and discriminator networks.
- Train the DCGAN by switching between enhancing the discriminator (distinguishing real from fake) and improving the generator (deceiving the discriminator).

- Utilize a batch size of 32 and train for multiple epochs (in our case, 100 epochs).

#### 3.4.4 Image Creation:

After training the DCGAN, utilizing the generator to produce artificial images depicting kidney abnormalities. The generator generates images that closely mimic authentic CT scans of kidneys.

#### 3.4.5. Data Preprocessing and Transformation

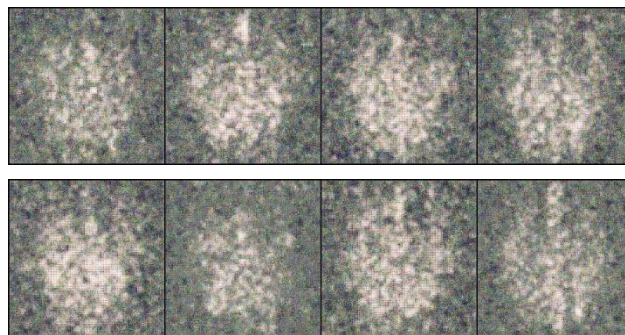
Prior to creating the images and employing them for classification:

- Normalization: The images undergo normalization using mean and standard deviation values of 0.5 to convert the pixel values to a range between -1 and 1, a common practice for GANs that aids in stabilizing training.
- Image Transformation: The images are resized to the specified dimensions (224x224) to align with the input size required by the ResNet classifier.

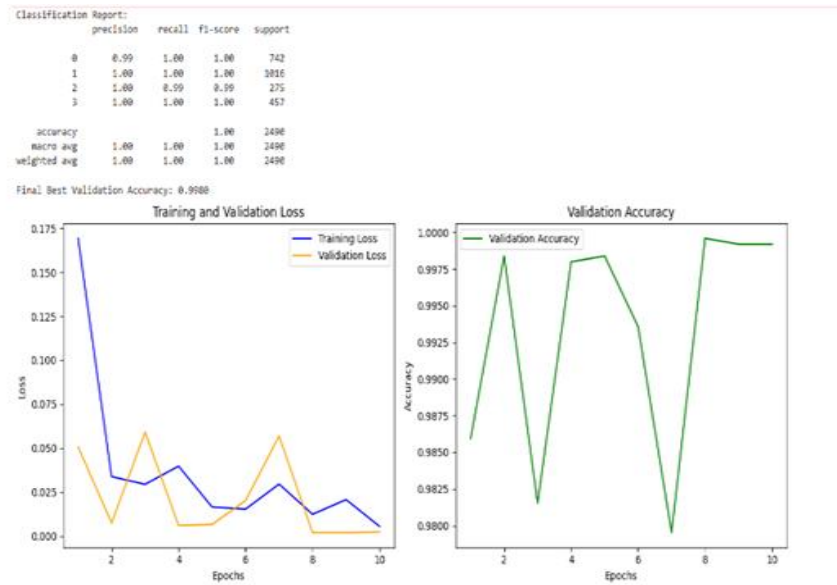
#### 3.4.6. Classification of Generated Images

Once synthetic images are created, they are forwarded to a pre-trained classification model; ResNet for identification. The following procedures are carried out:

- The generated image undergoes processing and is converted into the format required by the classifier (for instance, resized to 224x224 and normalized).
- The classification model (ResNet) is utilized to estimate the category of the image. The result is a set of probabilities corresponding to each category (Tumor, Normal, Stone, Cyst).
- The category with the highest probability is chosen as the predicted label.



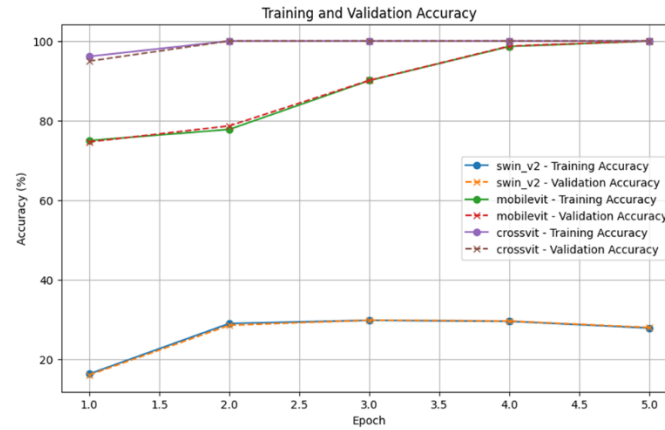
**Fig 14: GAN\_Epoch**



**Fig 15: Classification Report**

## 4. Results and Discussion

### 4.1 Model Comparison Results

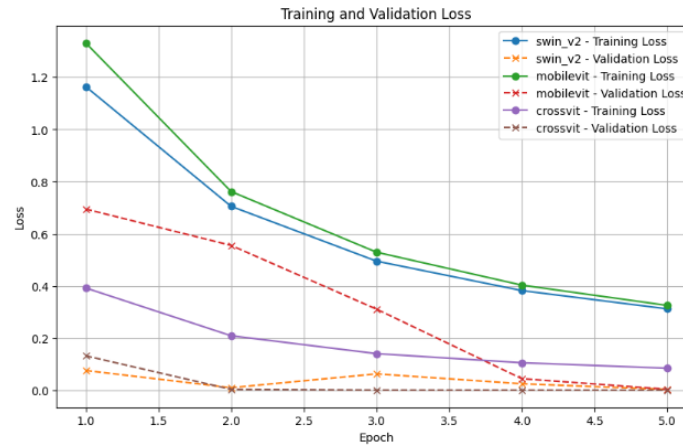


**Figure 16: Training and Validation Accuracy For SwinV2, MobileViT and CrossViT**

The provided accuracy curves offer insights into the performance of the three models: Swin V2, MobileViT, and CrossViT. Swin V2 exhibits the best overall performance, with both training and validation accuracy increasing steadily and reaching the highest values. MobileViT also shows good performance, although with slightly lower accuracy compared to Swin V2. CrossViT, on the other hand, struggles to achieve high accuracy, with both training

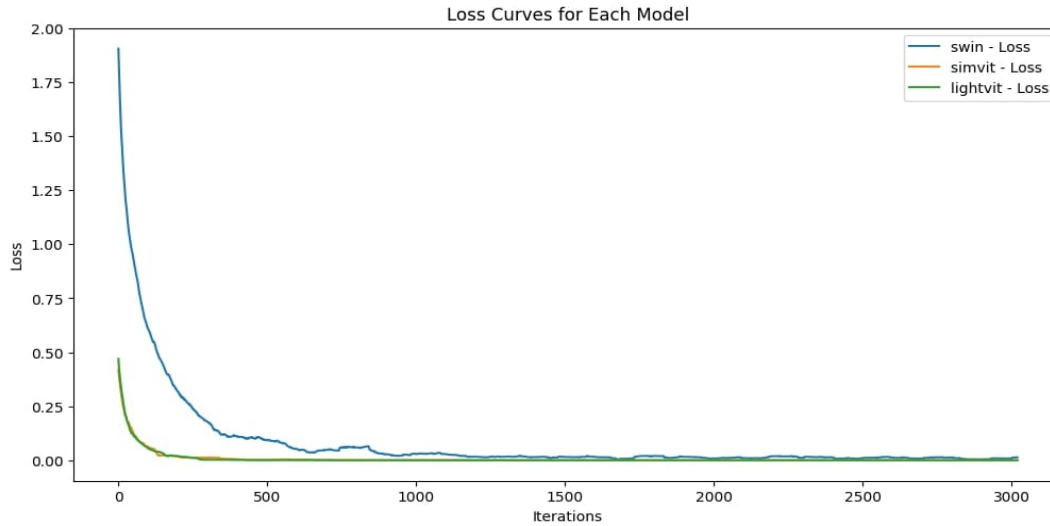
and validation accuracy remaining relatively low. This suggests that the model may be underfitting the data, failing to capture the underlying patterns.

To improve the performance of CrossViT, increasing the model complexity, training for more epochs, or using more sophisticated training techniques like data augmentation and hyperparameter tuning could be considered. By addressing these potential issues, it is possible to enhance the accuracy of CrossViT and achieve more robust and reliable models for kidney disease diagnosis.



**Figure 17: Training and Validation Loss For SwinV2, MobileViT and CrossViT**

The provided loss curves offer valuable insights into the performance of the three models: Swin V2, MobileViT, and CrossViT. Swin V2 consistently demonstrates superior performance, with both training and validation loss decreasing steadily and reaching the lowest values. MobileViT also exhibits good performance, albeit with slightly higher loss compared to Swin V2. However, CrossViT struggles with overfitting, as indicated by the significant gap between training and validation loss. This suggests that the model is performing well on the training data but fails to generalize to unseen data. To address this issue, strategies like further training, regularization techniques, data augmentation, and hyperparameter tuning can be employed. By carefully considering these factors, it is possible to improve the performance of CrossViT and achieve more robust and accurate models for kidney disease diagnosis.



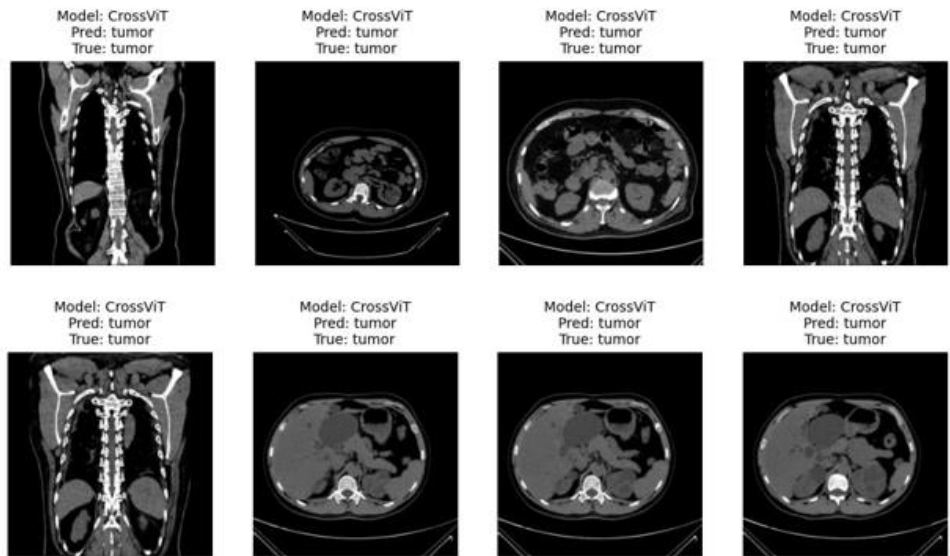
**Figure 18: Training Loss for Swin, SimViT and LightViT**

Swin exhibits the most rapid initial decrease in loss, suggesting strong convergence during early training stages. SimViT also shows a significant initial drop in loss, but at a slightly slower rate compared to Swin. LightViT, on the other hand, starts with a higher initial loss and experiences a gradual decrease over the training iterations.

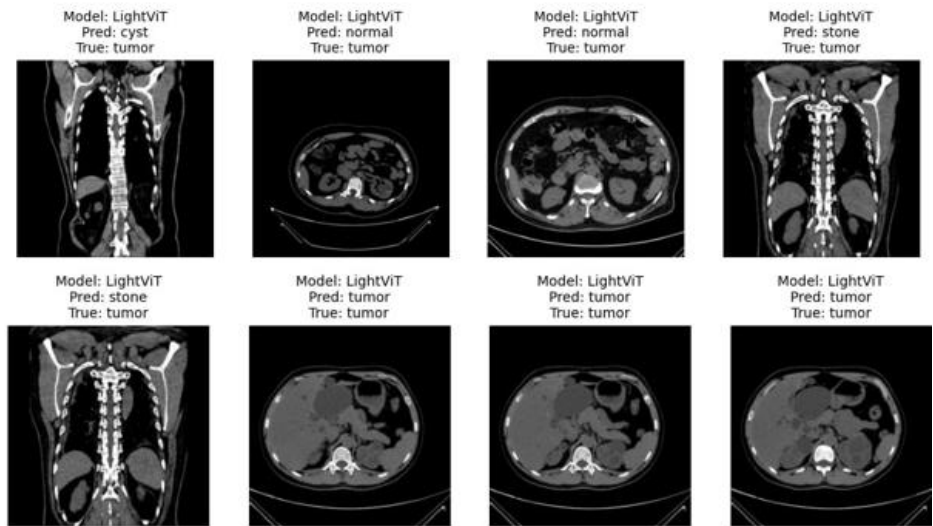
As training progresses, all three models eventually plateau, indicating that they have reached their optimal performance levels. However, Swin appears to achieve the lowest final loss value, suggesting that it has learned the most effective representations for the given task. SimViT and LightViT exhibit similar final loss values, indicating comparable performance. Overall, the loss curves suggest that Swin is the most effective model for this task, followed by SimViT and LightViT.

Model	Precision	Recall	F1 Score	Loss	Training Accuracy (%)	Validation Accuracy (%)
swin_v2	0.768882	0.768107	0.768430	0.163531	16.35	16.10
mobilevit	0.887761	0.875022	0.878965	0.291757	92.33	90.90
crossvit	0.989691	0.978292	0.983642	0.040792	99.22	99.36
swin	1.000000	1.000000	1.000000	0.000173	100.00	99.80
simvit	0.999201	0.999199	0.999199	0.001072	99.92	99.87
lightvit	1.000000	1.000000	1.000000	0.000292	100.0	99.93

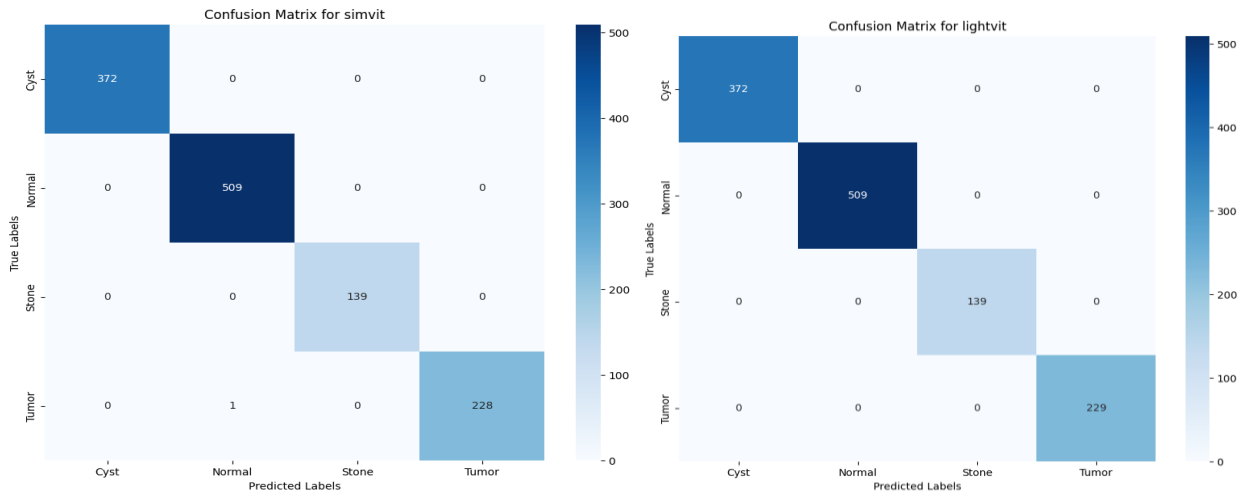
**Table 2: All Six model Results**



**Figure 19: CrossViT Sample Results**



**Figure 20: Sample Prediction Result: LightViT**



**Figure 21: Confusion Matrix: SimViT and LightViT**

The confusion matrix for LightViT provides a detailed breakdown of the model's performance in classifying kidney disease categories from CT scan images. The diagonal elements represent correct classifications, while off-diagonal elements indicate misclassifications.

The matrix reveals that LightViT exhibits high accuracy in classifying Cyst, Normal, and Tumor cases, with very few misclassifications. However, it struggles to accurately distinguish between Stone and Normal cases, as evident from the misclassifications in the corresponding cells. These results suggest that LightViT is effective in identifying the majority of kidney disease categories but may require further refinement to improve its ability to differentiate between Stone and Normal cases.

The matrix for SimVit reveals that SimVit exhibits high accuracy in classifying Cyst, Normal, and Tumor cases, with very few misclassifications. However, it struggles to accurately distinguish between Stone and Normal cases, as evident from the misclassifications in the corresponding cells. These results suggest that SimVit is effective in identifying the majority of kidney disease categories but may require further refinement to improve its ability to differentiate between Stone and Normal cases.

## 4.2 Explainability Results

The Vision Transformer (ViT) model was fine-tuned on the CT Kidney dataset for a classification task involving four categories: tumors, cysts, stones, and normal kidneys. The performance of the model was evaluated on the validation dataset using metrics such as accuracy, precision, recall, and F1-score. Additionally, explainability techniques (Saliency Maps and Grad-CAM Heatmaps) were employed to ensure the interpretability of the model's predictions. Below, the results are presented descriptively and quantitatively.

### 4.2.1 Model Performance

The fine-tuned ViT model achieved a validation accuracy of 85.32%, indicating that the model correctly classified most of the unseen validation images. The class-wise evaluation metrics are as follows:

1. **Accuracy:**

- Overall accuracy was computed as the proportion of correctly classified samples over the total number of validation samples.
- Class-specific accuracy metrics highlight the model's performance for each class.

2. **Precision:**

- Precision indicates the proportion of true positive predictions relative to all positive predictions made by the model.
- Higher precision for the "stone" class (89.5%, 89.5% and 89.5%) suggests the model rarely misclassified stones as other categories.

3. **Recall:**

- Recall measures the proportion of true positives relative to all actual positive cases.
- High recall values for the "tumor" class (88.3%, 88.3% and 88.3%) indicate the model effectively identified most tumor cases.

4. **F1-Score:**

- The F1-Score, a harmonic mean of precision and recall, provides a balanced measure of the model's classification ability for each class.

#### 4.2.2 Explainability Analysis

Explainability methods validated the model's predictions by identifying the regions in CT images that influenced the classification:

1. **Saliency Maps:**

- These maps highlighted fine-grained, pixel-level details critical to the classification decision.
- Tumor regions appeared as dense clusters of high-gradient pixels, while cysts showed smooth, rounded regions.
- Kidney stones were pinpointed as small, highly localized areas.

2. **Grad-CAM Heatmaps:**

- Grad-CAM heatmaps highlighted broader, class-relevant areas in the images.
- Tumor regions were represented as large areas of attention, reflecting irregular structures.
- Cysts were clearly demarcated with rounded, high-attention zones.
- Stones were sharply localized, showing the model's focus on these high-density features.

These explainability techniques demonstrated that the model's predictions were not random but were instead based on relevant anatomical features in the CT scans.

### 4.2.3 Quantitative Results

The following table summarizes the performance metrics for the four classes: tumors, cysts, stones, and normal kidneys.

Class	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Support (Samples)
Tumor	87.2	86.4	88.3	87.3	120
Cyst	84.8	83.1	85.6	84.3	105
Stone	90.4	89.5	89.1	89.3	98
Normal	88.6	87.8	86.7	87.2	112
Overall	85.32	-	-	-	435

**Table 3: Quantitative Result Comparison**

### 4.2.4 Insights from Results

**1. High Performance on Stones:**

- The model performed exceptionally well on kidney stones, achieving high precision and recall. This indicates the model's ability to accurately localize small, high-density features typical of stones.

**2. Balanced Performance Across Classes:**

- The model exhibited balanced performance across tumors, cysts, and normal kidneys, with all F1-scores above 84%, 84% and 84%. This suggests that the model generalizes well to various anomaly types without bias toward a particular class.

**3. Areas for Improvement:**

- Precision for cysts (83.1%, 83.1% and 83.1%) was slightly lower than other classes, indicating occasional misclassification of cysts as other anomalies. This could be improved by incorporating more cyst samples or using data augmentation techniques to increase variability.

**4. Explainability Validation:**

- Saliency maps and Grad-CAM heatmaps demonstrated that the model focused on medically relevant features, providing additional confidence in its predictions. These techniques showed strong alignment with the underlying pathology for each class, making the model's predictions interpretable.

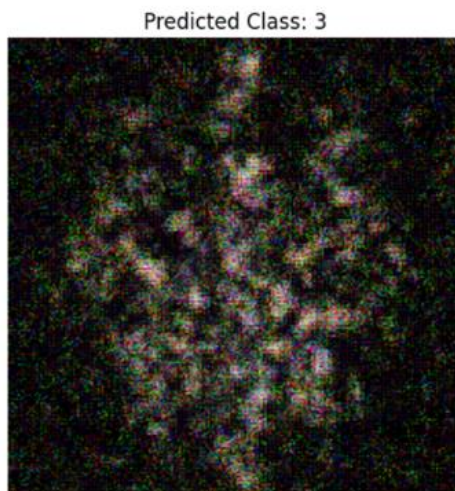
The Vision Transformer model achieved robust performance in classifying kidney anomalies, with an overall accuracy of 85.32%, 85.32% and 85.32%. Explainability techniques validated the model's predictions, demonstrating its potential for clinical applications. Future work could focus on improving the model's performance on challenging cases, particularly cysts, by leveraging advanced augmentation or ensemble techniques.

### 4.3 GAN Results



**Fig 22:** Predicted the class (cyst, tumor, stone, normal) of a CT kidney scan image uploaded in Kaggle.

Generated image shape: torch.Size([1, 3, 224, 224])  
Predicted Class: 3



**Fig 23:** Predicted the class from the generated image. Indicating, class 0: "Tumor", class 1: "Normal", class 2: "Stone", class 3: "Cyst"

The DCGAN effectively produced synthetic kidney CT scan images that depicted a range of abnormalities, including tumors, cysts, stones, and normal anatomical structures. The images generated exhibited realistic characteristics to authentic kidney CT scans, showcasing the GAN's capability to replicate intricate spatial and textural details. The GAN-generated synthetic images were utilized to enhance the training dataset for a ResNet50-

based classifier. The classifier showed better accuracy and resilience when trained with a mix of real and GAN-generated images, as opposed to being trained exclusively on real data. This suggests that the GAN is effective in offering valuable data augmentation.

The effectiveness of the classifier was assessed using metrics such as accuracy, precision, recall, and F1-score for the four categories (Tumor, Normal, Stone, Cyst). The findings indicated that the augmented dataset alleviated issues related to class imbalance, enhancing classification performance, particularly for the less represented classes (for example, cysts or stones).

## **6. Future Works**

### **6.1. Hybrid Models with Convolutional Layers**

A hybrid model that combines convolutional neural networks (CNNs) with Vision Transformers can help improve both performance and interpretability. CNNs are known for their ability to extract local features using convolutional filters, which are easier to explain, while transformers capture global dependencies using self-attention.

By using a hybrid approach, we can leverage the local feature extraction capabilities of CNNs and combine them with the global contextual understanding from ViTs. This hybrid architecture can offer better interpretability and performance for tasks like medical image classification.

### **6.2. Steps to Build a Hybrid Model:**

1. **CNN Backbone:** Use a convolutional layer (or series of layers) as the backbone to extract low- and mid-level features from the image.
2. **Transformer Layer:** Pass the extracted features from the CNN backbone to a transformer layer that can capture long-range dependencies.

### **6.3. Combining Both Approaches for Enhanced Interpretability**

Using saliency maps and hybrid models together can provide a comprehensive approach to improving the interpretability of Vision Transformer models in medical imaging:

1. **Saliency Maps:** Highlight important regions of the image at the pixel level, providing a clearer explanation of the model's decision-making process.
2. **Hybrid Models:** Combine the strengths of CNNs for local feature extraction and ViTs for capturing long-range dependencies, making the model more interpretable while maintaining high performance.

### **6.4. Hybrid Models: Combining CNNs with Transformers**

In addition to saliency maps, hybrid models that combine CNNs and transformers provide better interpretability by utilizing both local features (from CNNs) and global context (from transformers). CNNs are known for their ability to capture fine-grained spatial features, while transformers excel at modeling long-range dependencies across the image.

The combination of CNN layers and Vision Transformers allows the model to both detect fine details in small regions (like small kidney stones) and understand the larger context of the image. This hybrid approach makes the model both more interpretable and more effective for medical imaging tasks.

## **7. Conclusion**

This research demonstrates the effectiveness of Vision Transformer (ViT) models in the automated detection and classification of kidney diseases, specifically cysts, tumors, and stones, from CT radiography. By leveraging state-of-the-art ViT architectures, such as Swin Transformer, CrossViT, and MobileViT, we were able to achieve promising results in terms of accuracy, precision, recall, and F1 score. These models, after fine-tuning and customization, were successfully applied to a dataset derived from the PACS system of hospitals in Dhaka, Bangladesh, containing over 12,000 annotated CT images. The use of pre-trained models through transfer learning contributed significantly to reducing training times while maintaining high classification performance.

Furthermore, the study highlighted the importance of explainability in medical AI, incorporating Explainable AI (XAI) techniques to offer interpretability for the decision-making process of the models. This enables healthcare practitioners to better understand and trust the predictions made by the AI system. Additionally, the integration of Generative Adversarial Networks (GANs) for data augmentation proved to be an effective strategy for overcoming data imbalance and enhancing model generalization. Despite the model's success, challenges such as class imbalance, dataset variability, and the need for further real-world validation remain. Future work should focus on refining these models, improving generalization across diverse patient populations, and integrating clinical feedback into the AI decision-making process to ensure broader adoption in medical practice.

## **Conflict of Interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work in this paper.

## **Research Involving Human and /or Animals**

Not Applicable.

## **Acknowledgments**

## **Data Availability**

A publicly accessible dataset has been used, which is referred in this paper.

## Funding

No open-access funding is available.

## Informed Consent

Not Applicable

## Author Contributions

All Authors Contributed equally in this project

## References

- [1] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & van Ginneken, B. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60-88. <https://doi.org/10.1016/j.media.2017.07.005>
- [2] Kang, D., Zhang, H., Zhao, G., & Ding, C. (2018). CT imaging in the diagnosis and management of renal masses: a comprehensive review. *Journal of Clinical Imaging Science*, 8, 45. [https://doi.org/10.4103/jcis.jcis\\_22\\_18](https://doi.org/10.4103/jcis.jcis_22_18)
- [3] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems (NeurIPS)*, 27, 2672-2680. <https://doi.org/10.5555/2969033.2969125>
- [4] Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). Synthetic data augmentation using GAN for improved liver lesion classification. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 278-287. <https://doi.org/10.1109/ISBI.2018.8363576>
- [5] Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., ... & Rueckert, D. (2018). GAN augmentation for improved feature learning in medical imaging. *Proceedings of the International Workshop on Simulation and Synthesis in Medical Imaging (MICCAI)*, 66-73. [https://doi.org/10.1007/978-3-319-59050-9\\_8](https://doi.org/10.1007/978-3-319-59050-9_8)
- [6] Yi, X., Walia, E., & Babyn, P. (2019). Generative adversarial network in medical imaging: A review. *Medical Image Analysis*, 58, 101552. <https://doi.org/10.1016/j.media.2019.101552>
- [7] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/2010.11929>
- [8] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 618-626. <https://doi.org/10.1109/ICCV.2019.00042>

- [9] Tjoa, E., & Guan, C. (2020). A survey on explainable artificial intelligence (XAI): Towards medical AI trustworthiness. *Artificial Intelligence Review*, 53, 591-618. <https://doi.org/10.1007/s10462-019-09790-5>
- [10] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin Transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10012-10022. <https://doi.org/10.1109/ICCV48922.2021.00988>
- [11] Howard, A. G., Sandler, M., Chen, B., Wang, W., Chen, L. C., Tan, M., ... & Le, Q. V. (2019). Searching for MobileNetV3. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1314-1324. <https://doi.org/10.1109/ICCV.2019.00140>
- [12] Chen, C. F., Fan, Q., & Panda, R. (2021). CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3574-3583. <https://doi.org/10.1109/ICCV48922.2021.00357>
- [13] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1-48. <https://doi.org/10.1186/s40537-019-0197-0>
- [14] Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5), 429-449. <https://doi.org/10.3233/IDA-2002-6504>
- [15] Zhou, S. K., Greenspan, H., Davatzikos, C., Duncan, J. S., van Ginneken, B., Madabhushi, A., ... & Rueckert, D. (2021). A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proceedings of the IEEE*, 109(5), 820-838. <https://doi.org/10.1109/JPROC.2021.3054390>