A

PROJECT REPORT

ON

Generation of CT Image from MRI using multi cycle Gan`s

SUBMITTED TO

SHIVAJIUNIVERSITY, KOLHAPUR

IN THE PARTIAL FULFILLMENT OF REQUIREMENT FOR THE AWARD OF DEGREE BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

MR. MANAS KRUPAL GIDWANI	19UCS037
MR. AKIL NASIR JAMADAR	19UCS048
MR. ABHIJEET DILIP KAMBLE	19UCS053
MR. DIKSHANT KUMAR KHOOL	19UCS059
MR. RACHIT GIRISH KULKARNI	19UCS069

UNDER THE GUIDANCE OF

PROF. K.S. KADAM

DKTE

Promoting Excellence in Teaching, Learning & Research

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DKTE SOCIETY'S TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI

2022-2023

D.K.T.E. SOCIETY’S

TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI
(AN AUTONOMOUS INSTITUTE)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DKTE
Promoting Excellence in
Teaching, Learning & Research

CERTIFICATE

This is to certify that, project work entitled

“GENERATION OF CT IMAGE FROM MRI USING MULTICYCLE GAN`S”

is a bonafide record of project work carried out in this college by

MR.	MANAS KRUPAL GIDWANI	19UCS037
MR.	AKIL NASIR JAMADAR	19UCS048
MR.	ABHIJEET DILIP KAMBLE	19UCS053
MR.	DIKSHANT KUMAR KHOOL	19UCS059
MR.	RACHIT GIRISH KULKARNI	19UCS069

is in the partial fulfillment of award of degree Bachelor in Engineering in Computer Science & Engineering prescribed by Shivaji University, Kolhapur for the academic year 2022-2023

PROF. K.S. KADAM
(PROJECT GUIDE)

PROF.(DR.) D.V. KODAVADE
(HOD CSE DEPT.)

PROF.(DR.) P.V. KADOLE
(DIRECTOR)

EXAMINER: ____________________

____

DECLARATION

We hereby declare that, the project work report entitled “Generation of CT image from MRI using multi cycle Gan`s which is being submitted to D.K.T.E. Society’s Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.Tech (CSE). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under the Copyright and Piracy / Cyber / IPR Act amended from time to time.

Mr.	Manas Krupal Gidwani	19UCS037
Mr.	Akil Nasir Jamadar	19UCS048
Mr.	Abhijeet Dilip Kamble	19UCS053
Mr.	Dikshant Kumar Khool	19UCS059
Mr.	Rachit Girish Kulkarni	19UCS069

ACKNOWLEDGEMENT

With great pleasure we wish to express our deep sense of gratitude to Prof. K. S. Kadam for his valuable guidance, support, and encouragement in completion of this project report.

Also, we would like to take the opportunity to thank our head of department Dr. D. V. Kodavade for his co-operation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of the Computer Science and Engineering Department for their support, help and assistance which they extended as and when required.

Thanking you,

MR.	Manas Krupal Gidwani	19UCS037
MR.	Akil Nasir Jamadar	19UCS048
MR.	Abhijeet Dilip Kamble	19UCS053
MR.	Dikshant Kumar Khool	19UCS059
MR.	Rachit Girish Kulkarni	19UCS069

ABSTRACT

CT is the X-ray technology-based medical imaging method to generate im- ages of patients based on a technique related to medical isotopes which are given to patients. CT scans are mostly used in diagnosing oncology, serious injuries to the head, chest, spine and pelvis, especially fractures and also internal bleeding. CT scans are also used to pinpoint the size and location of tumors. However, CT lacks good soft-tissue contrast, especially in the brain, head, neck, and pelvic regions, making it extremely difficult to directly de- lineate endangering organs and target areas on CT . In addition, multiple CT scans expose patients to additional ionizing radiation repeatedly, which brought potentially harmful effects to patients. MRI is the magnetic field and radio wave-based medical imaging method.Computed Tomography (CT) plays an important role in planning of radiation therapy (RT) but CT scans expose patients to additional ionizing radiation repeatedly, which brings potentially harmful effects to patients. Magnetic Resonance Imaging (MRI) is another method used for auxiliary diagnosis and treatment. Compared with CT, MRI is safer since the patients would not be exposed to ionizing radiation in the image scanning process. It also reduces the cost of treatment because only one set of images is required for scanning. But the drawback of MRI is that it can't obtain the patient's bone information. So, our pro- posed system is going to convert MRI images to CT using multi-cycle GAN (generative adversarial network).We have used various deep learning models like Alexnet and Resnet.

INDEX

1. Introduction

CT X-ray, a technology-based medical imaging technique that utilizes medical isotopes, generates patient images. It is commonly employed in oncology to diagnose severe injuries in the head, chest, spine, and pelvis, particularly fractures and internal bleeding. CT scans are also useful in determining tumor size and location. However, CT scans suffer from limited soft-tissue contrast, especially in the brain, head, neck, and pelvic regions, making it challenging to directly visualize critical organs and target areas.

Furthermore, multiple CT scans expose patients to additional ionizing radiation, which can have potentially harmful effects. On the other hand, MRI (Magnetic Resonance Imaging) is a medical imaging method that relies on a magnetic field and radio waves. Compared to CT scans, MRI is more effective in diagnosing soft tissue problems, ligaments, and tendons. It is frequently used to examine joints, the brain, wrists, ankles, breasts, the heart, and blood vessels. MRI has made significant advancements in soft-tissue contrast and organ visualization compared to CT imaging.

While CT excels in providing superior spatial resolution and excellent visualization of bony structures, MRI captures detailed information about soft tissues. Integrating the strengths of both MRI and CT can significantly enhance clinicians' diagnostic capabilities. However, acquiring both types of scans for every patient is often impractical due to factors such as cost, radiation exposure, and patient comfort. Consequently, techniques have been developed to generate CT-like images from MRI scans, offering an alternative solution to access the benefits of CT imaging using existing MRI data.

One promising approach to achieving this translation is the utilization of multicycle Generative Adversarial Networks (GANs). GANs are deep learning models consisting of a generator and a discriminator that compete against each other during training to produce highly realistic images. Multicycle GANs, also known as CycleGANs, are specifically designed to learn mappings between two domains without the need for paired training data.

In the context of generating CT images from MRI, the multicycle GAN architecture is trained on a dataset comprising paired MRI and CT images. The generator network learns to transform MRI images into CT-like images, while the discriminator network distinguishes between real CT images and the generated CT images. Through this adversarial training process, the generator is guided to generate more realistic CT images that closely resemble the ground truth.

To preserve anatomical details and ensure the fidelity of the transformation, additional loss functions are incorporated. Cycle-consistency loss is employed to enforce the reconstruction of the original MRI image when it undergoes translation to CT-like images and back. This loss encourages the generator to retain essential information from the MRI while generating CT-specific characteristics. Identity loss is also used to maintain the inherent features of the input MRI image during the translation process.

The performance of the trained multicycle GAN is thoroughly evaluated using quantitative metrics such as structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR) to measure the similarity between the generated CT images and real CT images. Additionally, visual inspection by medical professionals is crucial to evaluate the clinical relevance and quality of the generated images.

The successful implementation of a multicycle GAN for generating CT images from MRI holds immense potential in various medical applications. Clinicians can leverage this technology to obtain CT-like images directly from existing MRI scans, eliminating the need for additional CT imaging. This advancement can lead to improved diagnostic accuracy, treatment planning, and disease monitoring. Furthermore, researchers can benefit from the availability of large-scale datasets of paired MRI and CT images, facilitating studies on image analysis, disease progression, and evaluating therapeutic responses.

In summary, the objective of this project is to harness the capabilities of multicycle GANs to bridge the gap between MRI and CT imaging modalities by generating high-quality CT images from MRI scans. The integration of these modalities has the potential to revolutionize medical imaging practices, enhancing patient care, and advancing medical research in diverse clinical settings.

a. Problem Definition

To provide an interesting tool for participants to Gain knowledge. The main purpose of this game is to make people learn new things, improve their knowledge and to test their already existing skills, just by playing a game which is provided with
some time limit.

b. Aim and objective of the project

- To design a system which converts MRI image to CT scan Image.
- To classify between generated CT images and real images
- To check the efficiency of the model using mean absolute error, mean error and peak signal to noise Ratio (PSNR).

c. Scope and limitation of the project

- To develop a user-friendly interface such that everyone can access it easily.
- To build a system that uses technologies like deep learning to convert MRI images to CT images using multi cycle GAN's.
- Taking an MRI image as an input to the Generator Model of GAN's produces new images by taking a fixed size of the random sound as input. The images produced are then provided to the Discrimination Model.
- Although GANs have shown impressive results in generating realistic images, the quality of the generated CT images from MRI may not be as good as real CT images.
- Multi-cycle GANs are trained on a specific dataset, and their performance may not generalize well to new datasets or unseen cases.
- Multi-cycle GANs are computationally intensive, requiring significant computing resources and time to train.

d. Timeline of the project

Sr. no	work	month
1	Project planning	June-July
2	Information gathering	July-august
3	Data collection	August-September
4	Data preprocessing	September-October
5	Model development	October-November
6	Model training	November-January
7	Model testing	January - March
8	Performance analysis	March - april
9	Final documentation	April -may
10	Final presentation	May- june

Table 1.1 Timeline Chart

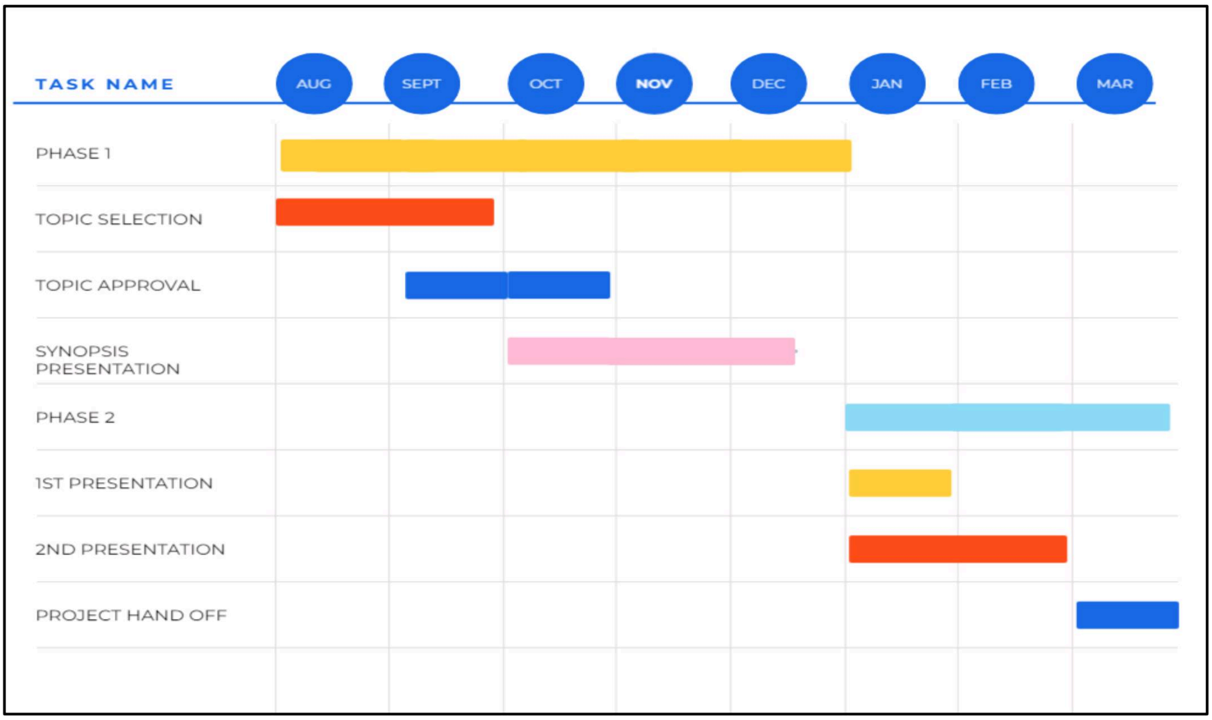

Fig. 1.1 Timeline

e. Project management plan

1. Project scope: Define the scope of the project, including the specific goals, objectives, and deliverables. Identify any constraints or limitations, such as time or budget.

2. Project schedule: Develop a schedule that includes the major milestones and activities of the project. Assign responsibilities for each activity and establish deadlines for completion.

3. Project budget: Develop a budget that includes all the costs associated with the project, including personnel, equipment, and materials.

4. Project team: Identify the project team members and their roles and responsibilities. Develop a communication plan to ensure that team members are kept informed and that communication is efficient and effective.

5. Risk management: Identify potential risks that could impact the project and develop strategies to mitigate those risks. Establish contingency plans in case of unforeseen events.

6. Quality assurance: Develop a plan to ensure that the project meets the required quality standards. This could include regular testing and validation of the system.

7. Documentation: Establish documentation standards and procedures for the project. This could include requirements specifications, design documents, user manuals, and test plans.

8. Change management: Develop a process for managing changes to the project scope, schedule, or budget. Establish a change control board to review and approve any changes.

9. Project evaluation: Develop a plan to evaluate the project's success, including criteria for measuring the achievement of goals and objectives. This could include user feedback, performance metrics, or other measures of success.

10. Project closure: Develop a plan for closing the project, including a process for transferring ownership of the system to the appropriate stakeholders and documenting lessons learned.

f. Project cost

1. Hardware Cost:

Components	Name	Pricing
Graphics Card	Nvidia GEForce	45750
Processor	Intel i5 7 gen	8999
RAM	16 GB	3845
Total		58594

2. Software Cost:

Lines of Code (LOC):285
Effort = a * (LOC) ^ b
Time = c * (Effort) ^ d
Persons Required = Effort / Time

For COCOMO model parameters:
a = 2.4 (constant for organic projects)
b = 1.05 (exponent derived from historical data)
c = 2.5 (constant for organic projects)
d = 0.38 (exponent derived from historical data)

Calculate Effort:
Effort = 2.4 * (285) ^ 1.05
Effort = 2.4 * 378.082
Effort = 907.3991 Person-Hours (approximately)

Calculate Time:
Time = 2.5 * (907.3991) ^ 0.38
Time = 2.5 * 28.7737
Time = 71.9343 Hours (approximately)

Calculate Persons Required:
Persons Required = Effort / Time
Persons Required = 907.3991 / 71.9343
Persons Required = 12.6142 (approximately=12)

Therefore, based on the given 295 lines of code, the estimated cost using the COCOMO model is:
Effort: 907.3991 Person-Hours
Time: 71.9343 Hours
Persons Required: 12.6142 (approximately=12)

2. Background study and literature overview

a. Literature overview

GAN’s and Multicycle GAN’s methods use self-learning and self- optimizing strategies to find a nonlinear mapping mechanism and estimate the electron density of tissue directly according to the MRI. Recently, many efficient machine learning algorithms have been applied to the generation of CT, such as Random Forest (Andreasen et al., 2016; Huynh et al.,2016 and Convolutional Neural Networks (CNN) (Han, 2017; Leyne’s et al., 2017).

These algorithms have shown excellent capabilities indirectly generating CT from MRI. There has been a growing interest in introducing deep learning to generate CT from MRI recently.

The network structure used in the early years is simple, but its performance was much better than the traditional CT generation methods. Han proposed using 2D Deep Convolutional Neural Networks (DCNN) to generate CT (Han, 2017), the results were improved compared with traditional atlas-based methods. Nie et al. (Nie et al., 2016) proposed using 3D fully convolutional networks of 3 layers to generate CT from MRI, which achieved better results than atlas-based methods and Structured Random Forest (SRF) (Huynh et al., 2016).

CT synthesis from MRI can be seen as a style transfer problem from a macro point of view. In 2014, Goodfellow et al. proposed a novel and effective network named Generative Adversarial Nets (GANs) (Goodfellow et al., 2014) to solve the style transfer problem in the field of Computer Version.

"Multi-cycle generative adversarial networks for cross-modality MR image synthesis" by Han et al. (2018): This paper proposes a multi-cycle GAN model for generating CT images from MRI data. The authors demonstrate the effectiveness of the approach on a dataset of brain MRI and CT images.

"Cross-modality synthesis from CT to PET using multi-cycle generative adversarial networks" by Hu et al. (2018): This study explores the use of multi-cycle GANs for generating PET images from CT scans. The authors show that the approach is effective in generating high-quality PET images.

"Multi-cycle generative adversarial networks for multi-modal MR image synthesis" by Li et al. (2019): This paper proposes a multi-cycle GAN model for generating T1-weighted and T2-weighted MR images from each other. The authors show that the approach is effective in generating high-quality images.

"An overview of generative adversarial networks (GANs) and their applications in medical imaging" by Goodfellow et al. (2016): This paper provides an overview of GANs and their applications in medical imaging, including MRI to CT generation.

"Deep learning for medical image analysis: A review" by Litjens et al. (2017): This paper provides a comprehensive review of deep learning techniques for medical image analysis, including GANs and their applications in MRI to CT generation.

b. Critical appraisal of other people's work

Identify the research question and study design: Before you start evaluating a study, it is important to understand the research question and the study design. This will help you determine whether the study is relevant to your research question and whether the study design is appropriate for answering the research question.

Assess the quality of the study: Look for potential sources of bias or confounding in the study design, such as selection bias, measurement bias, or confounding variables. Evaluate the methods used to minimize these sources of bias, such as randomization, blinding, or adjustment for confounding variables.

Evaluate the validity of the study: Assess the internal and external validity of the study. Internal validity refers to the extent to which the study measures what it claims to measure, while external validity refers to the generalizability of the study findings to other populations or settings.
Critique the analysis: Evaluate the statistical methods used to analyze the data, including the appropriateness of the statistical tests, the use of appropriate controls or comparison groups, and the handling of missing data or outliers.

c. Investigation of current project and related work

Identify the research question: Clearly define the research question of the project you are investigating. This will help you focus your investigation and find relevant literature.

Search for relevant literature: Conduct a comprehensive search of relevant literature using academic databases such as PubMed, Google Scholar, or IEEE Xplore. Use appropriate keywords related to the research question to find relevant articles, conference proceedings, and technical reports.

Evaluate the literature: Critically appraise the literature you find using the steps outlined in my previous response. Assess the quality, validity, and reliability of the studies and consider their relevance to your research question.

Identify gaps in the literature: Identify any gaps or limitations in the existing literature related to your research question. This can help you determine the scope and focus of your own research.

3. Requirement Analysis

a. Requirement Gathering.

To gather requirements for the MRI to CT generation project using multi-cycle GANs, we would need to consider the needs and preferences of the end-users, as well as the technical specifications and constraints of the system. Here are some key requirements that we would need to consider:

I. Functional Requirements:

- **Generating fake images**

1. The Generator Model generates new images by taking a fixed size ran- dom noise as an input. Generated images are then fed to the Discriminator Model.
2. The main goal of the Generator is to fool the Discriminator by generating images that look like real images and thus makes it harder for the Discriminator to classify images as real or fake.

- **Discriminating between fake and real images**

1. The Discriminator Model takes an image as an input (generated and real) and classifies it as real or fake.
2. Generated images come from the Generator and the real images come from the training data.
3. The discriminator model is the simple binary classification model

II. Non-functional Requirements:

Dataset: The system would require a large and diverse dataset of MRI and CT images to train the multi-cycle GAN model.

Image quality: The system would need to generate high-quality CT images from MRI data with sufficient resolution, contrast, and detail to be useful for clinical applications.

Speed: The system would need to generate CT images from MRI data in a timely manner, ideally in real-time or close to real-time.

Accuracy: The system would need to generate CT images that are accurate and reliable, with minimal errors or artifacts that could impact clinical decision-making.

Integration: The system would need to be integrated with existing medical imaging software and infrastructure, such as picture archiving and communication systems (PACS), electronic medical records (EMRs), and imaging equipment.

User interface: The system would need to have a user-friendly interface that allows clinicians to easily input MRI data, adjust parameters, and view generated CT images.

Scalability: The system would need to be scalable to handle large volumes of data and multiple users simultaneously.

Maintenance and support: The system would need to be regularly maintained and updated to ensure continued performance and reliability, and technical support would need to be available to troubleshoot issues.

b. Requirement Specification

ID	Requirement Description	Priority	Type
R1	The system shall generate CT images from MRI data using a multi-cycle GAN model.	High	Functional
R2	The system shall provide an intuitive user interface that allows clinicians to input MRI data, adjust parameters, and view generated CT images.	High	Functional
R3	The system shall be able to process and generate CT images in real-time or near real-time.	High	Functional
R4	The system shall generate high-quality CT images with sufficient resolution, contrast, and detail for clinical applications.	High	Functional
R5	The system shall be integrated with existing medical imaging software and infrastructure, such as PACS and EMRs.	High	Functional
R6	The system shall be able to handle large volumes of data and multiple users simultaneously.	High	Functional
R7	The system shall ensure the privacy and security of patient data and comply with relevant data privacy regulations.	High	Non-Functional
R8	The system shall be scalable and able to handle increases in workload and user demand.	High	Non-Functional
R9	The system shall have high availability and reliability, with minimal downtime and disruptions to clinical workflow.	High	Non-Functional
R10	The system shall be regularly maintained and updated to ensure continued performance and reliability.	High	Non-Functional
R11	The system shall be compatible with a range of hardware and software environments.	Medium	Non-Functional
R12	The system shall generate CT images that are accurate and reliable, with minimal errors or artifacts that could impact clinical decision-making.	High	Non-Functional
R13	The system shall provide technical support and training to users to ensure effective and safe use.	Medium	Non-Functional

Table 3.1 Requirement Specification

c. Use case Diagram

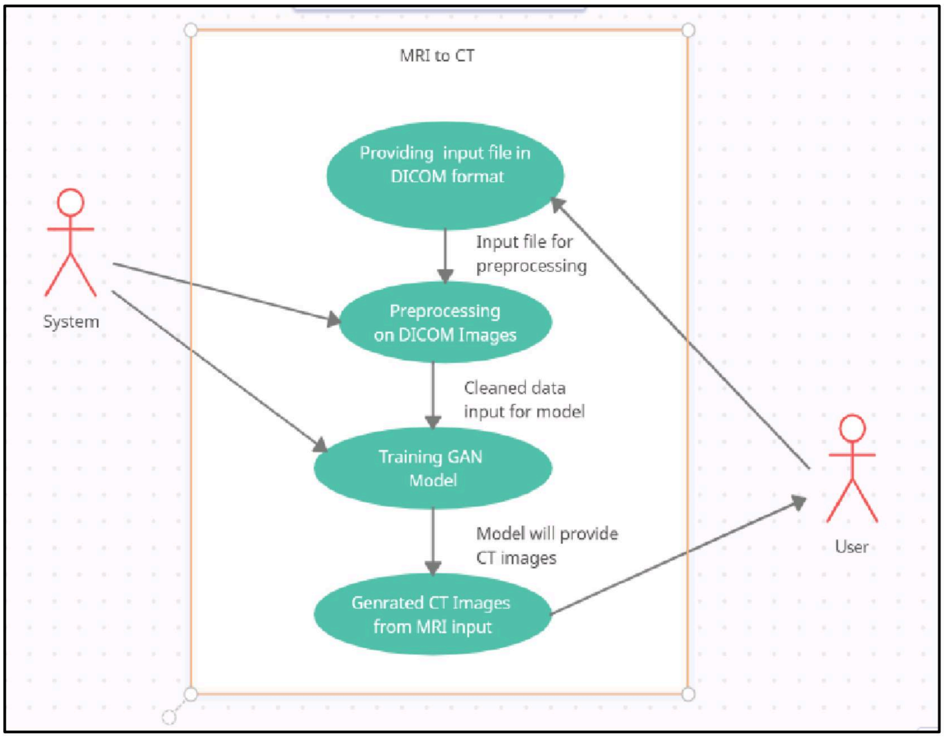

Fig. 3.1 Use Case Diagram

The "User Actor" represents the user who interacts with the system.
The "System Actor (Multicycle GAN)" represents the multicycle GAN system responsible for generating CT images from MRI.
The use cases and interactions in the diagram are as follows:

Provide MRI Input Image:

The user provides an MRI image as input to the system.
Generate CT from MRI:

The system receives the MRI image input from the user.
The system uses the trained multicycle GAN model to generate a CT image from the provided MRI input.
The system actor (Multicycle GAN) performs the necessary computations and transformations to generate the CT image.
Output Generated CT Image to User:

The system delivers the generated CT image as output to the user.
The user can then analyze, utilize, or further process the generated CT image as needed.

4. System design

1) Architectural Design

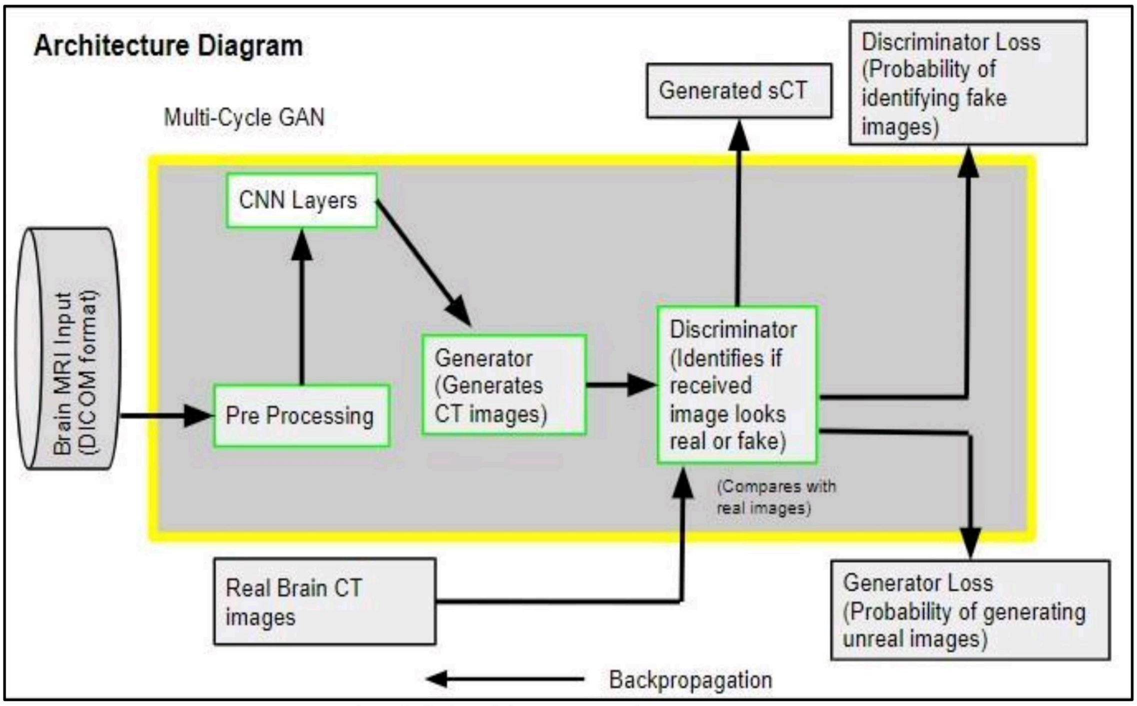

Fig. 4.1 Architecture Diagram

Preprocessing

- Image Registration: The MRI data is typically acquired in a different orientation or position than the CT images. Image registration techniques are used to align the MRI data with the CT images, so they are in the same coordinate space.

- Image Normalization: The intensity values of MRI and CT images can vary significantly due to differences in acquisition protocols and equipment. Image normalization techniques are used to standardize the intensity values across all images, which improves the performance of the GAN model.

- Image Resampling: The MRI images may have different spatial resolution and pixel size than the CT images. Image resampling techniques are used to adjust the resolution and pixel size of MRI data to match the CT images.

- Image Segmentation: Image segmentation techniques are used to identify and extract the specific structures or regions of interest from the MRI data. This segmentation step can improve the accuracy and quality of the generated CT images.

- Image Preprocessing: Additional preprocessing techniques may be used to further enhance the quality and usability of the MRI data, such as noise reduction, filtering, and artifact removal.

Generator model

The generator model in the MRI to CT generation project is a key component of the multi-cycle GAN architecture used to generate CT images from MRI data. The generator model typically consists of several layers of convolutional neural networks (CNNs) that learn to extract and transform features from the input MRI data. During training, the generator model receives the input MRI data and generates a corresponding CT image.

Discriminator model

Multi-cycle GAN architecture, and is responsible for distinguishing between the generated CT images and the ground truth CT images. The output of the discriminator model is used to calculate the adversarial loss, which measures the difference between the discriminator's predicted probabilities for the generated and ground truth images.

2) User Interface Design

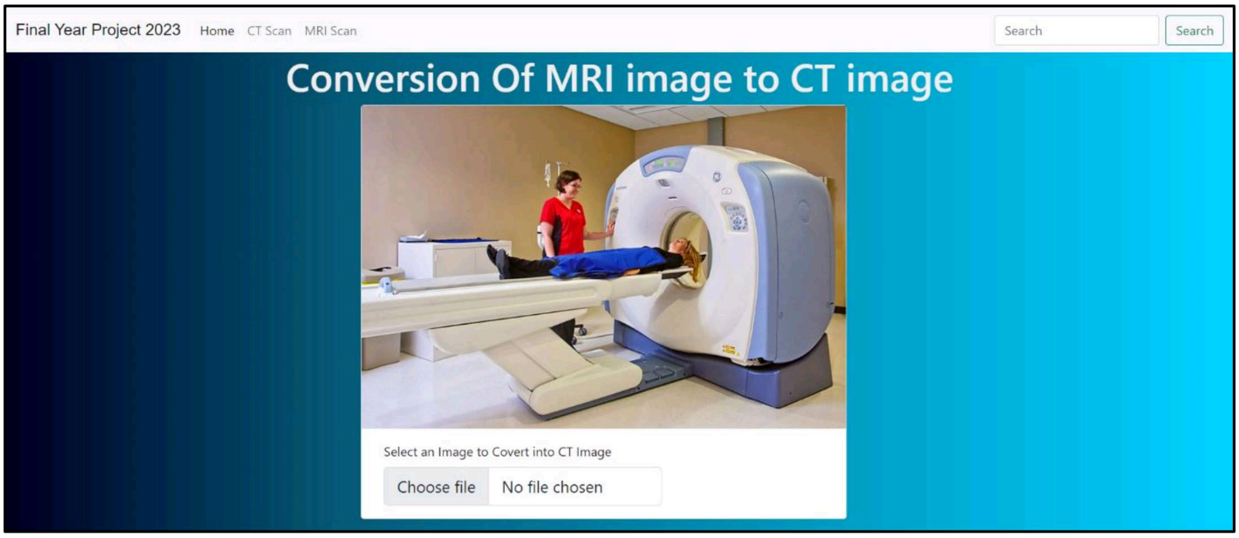

Fig. 4.2 User Interface

3) Algorithmic description of each module

- Preprocessing module:
  1. Load MRI dataset
  2. Normalize pixel values between 0 and 1
  3. Resize MRI images to a fixed size
  4. Save preprocessed MRI images

- Generator module:
  1. Define generator model architecture with encoder and decoder CNN layers
  2. Define loss functions, including adversarial loss and pixel-wise L1 or L2 loss
  3. Train the generator model using preprocessed MRI and ground truth CT images
  4. Save the trained generator model for future use
- Discriminator module:
  1. Define discriminator model architecture with CNN layers
  2. Define loss functions, including adversarial loss
  3. Train the discriminator model using preprocessed MRI and ground truth CT images
  4. Save the trained discriminator model for future use

- Integration module:
  1. Load preprocessed MRI images
  2. Use the trained generator model to generate corresponding CT images
  3. Save generated CT images

- Evaluation module:
  1. Load generated CT images and ground truth CT images
  2. Calculate metrics, such as peak signal-to-noise ratio (PSNR) and structural
  3. similarity index (SSIM) Output evaluation results

- Deployment module:
  1. Load preprocessed MRI images
  2. Use the trained generator model to generate corresponding CT images
  3. Save generated CT images to a specified location

System Modeling

a) Data Flow Diagram

1. Level 0 DFD

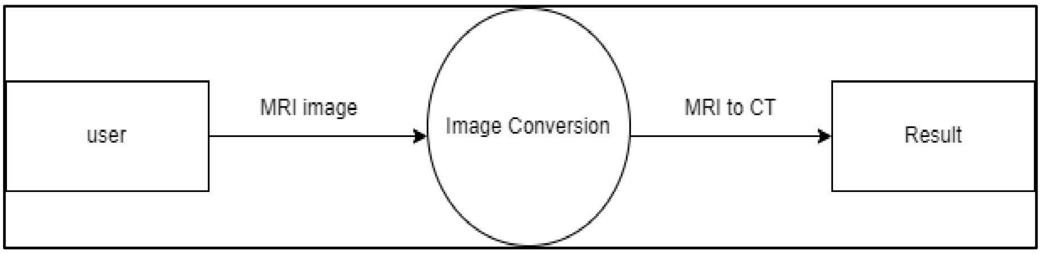

Fig. 4.3 Level 0 DFD

2. Level 1 DFD

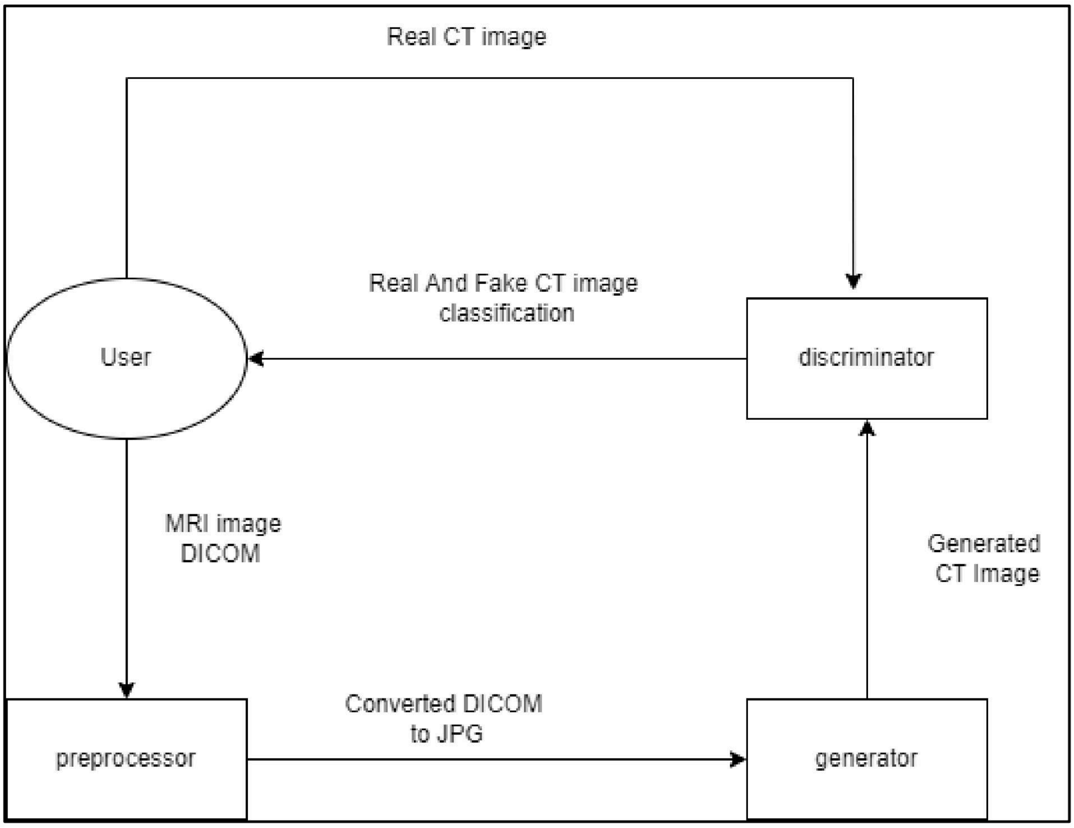

Fig. 4.4 Level 1 DFD

The data flow in a project that generates CT images from MRI using multicycle GANs involves several components and processes. The user begins by providing MRI images as input to the system through the user interface. These MRI images are then received by the input handling component, which forwards them to the preprocessing component.

Within the preprocessing stage, various operations are performed on the MRI images to prepare them for input to the generator. These operations may include resizing, normalization, or denoising, ensuring that the MRI data is in an appropriate format for further processing. The preprocessed MRI images are then passed on to the generator component.

The generator, being the core component of the multicycle GAN architecture, takes the preprocessed MRI images as input and generates synthetic CT images. Simultaneously, the discriminator component receives both the generated CT images and real CT images. The discriminator's role is to distinguish between the generated and real images and provide feedback to the generator, facilitating its improvement over time.

To enforce the cycle-consistency constraint during training, the cycle consistency loss component calculates the difference between the original input MRI images and the CT images generated from them. This loss is utilized to guide the training process and ensure that the generated CT images closely resemble the original MRI data.

The training data, consisting of paired MRI and corresponding CT images, is utilized to train the multicycle GAN. The training pipeline manages this process, loading the training data, processing it in batches, and updating the parameters of the generator and discriminator based on the calculated losses. Through iterations, the system learns to generate high-quality CT images from MRI inputs.
The generated CT images are then passed to the output handling component, which presents them to the user through the user interface. This allows the user to observe and analyze the synthetic CT images generated by the system, potentially aiding in medical diagnosis, treatment planning, or other applications where CT imaging is valuable.

Overall, the data flow in this project involves a series of steps, including input handling, preprocessing, generator and discriminator interactions, cycle consistency enforcement, training, and output handling. These processes work together to transform MRI images into realistic CT images using the power of multicycle GANs.

b) Sequence Diagram

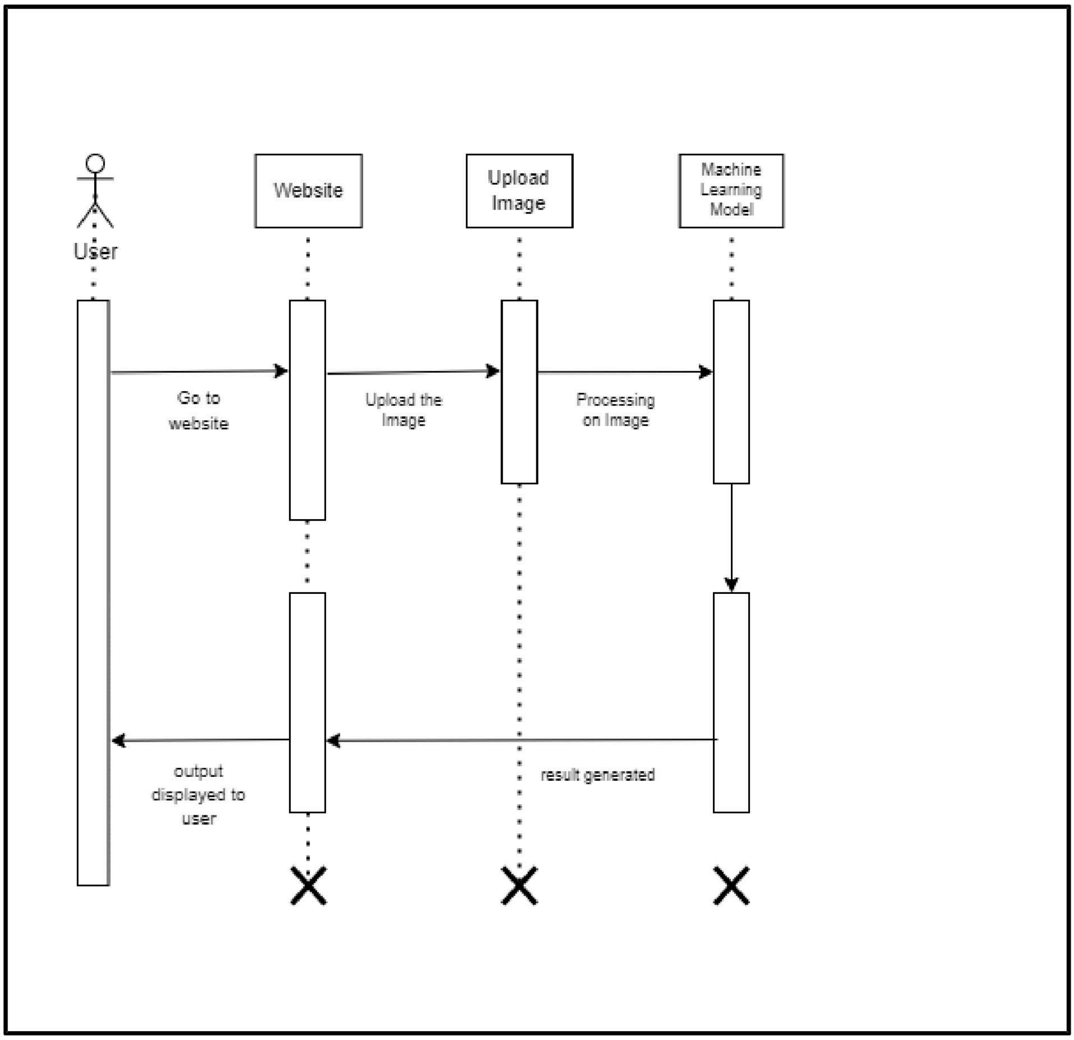

Fig. 4.5 Sequence Diagram

The sequence diagram begins with the user interacting with the system through the user interface. The user provides MRI images as input to the system, triggering the input handling component. The input handling component receives the MRI images and passes them to the preprocessing component.

The preprocessing component performs necessary operations on the MRI images, such as resizing, normalization, or denoising, to prepare them for input to the generator. Once the preprocessing is complete, the preprocessed MRI images are sent to the generator component.

Upon receiving the preprocessed MRI images, the generator component starts the process of generating synthetic CT images. It leverages the multicycle GAN architecture to transform the MRI data into realistic CT images. As the generator performs its task, it may send intermediate results or status updates to the output handling component.

Simultaneously, the discriminator component receives both the generated CT images and real

CT images. The discriminator's role is to analyze these images and provide feedback to the generator. This feedback helps the generator improve its performance over time by distinguishing between the generated and real images.

The cycle consistency loss component calculates the difference between the original input MRI images and the CT images generated from them. This loss is used to enforce the cycle-consistency constraint during training. The loss information is passed between the generator, discriminator, and cycle consistency loss components to optimize the generator's performance.

In the training phase, the system utilizes a dataset consisting of paired MRI and corresponding CT images. The training pipeline manages the training process, loading the training data in batches, and updating the generator and discriminator parameters based on the calculated losses. This iterative process continues until the system has learned to generate high-quality CT images from MRI inputs.

Finally, the generated CT images are passed to the output handling component. The output handling component presents the synthetic CT images to the user through the user interface, allowing them to observe and analyze the results.

This sequence diagram illustrates the flow of events and message exchanges between the different components involved in the project, starting from user input through image generation and ultimately presenting the generated CT images to the user.

c) Activity Diagram

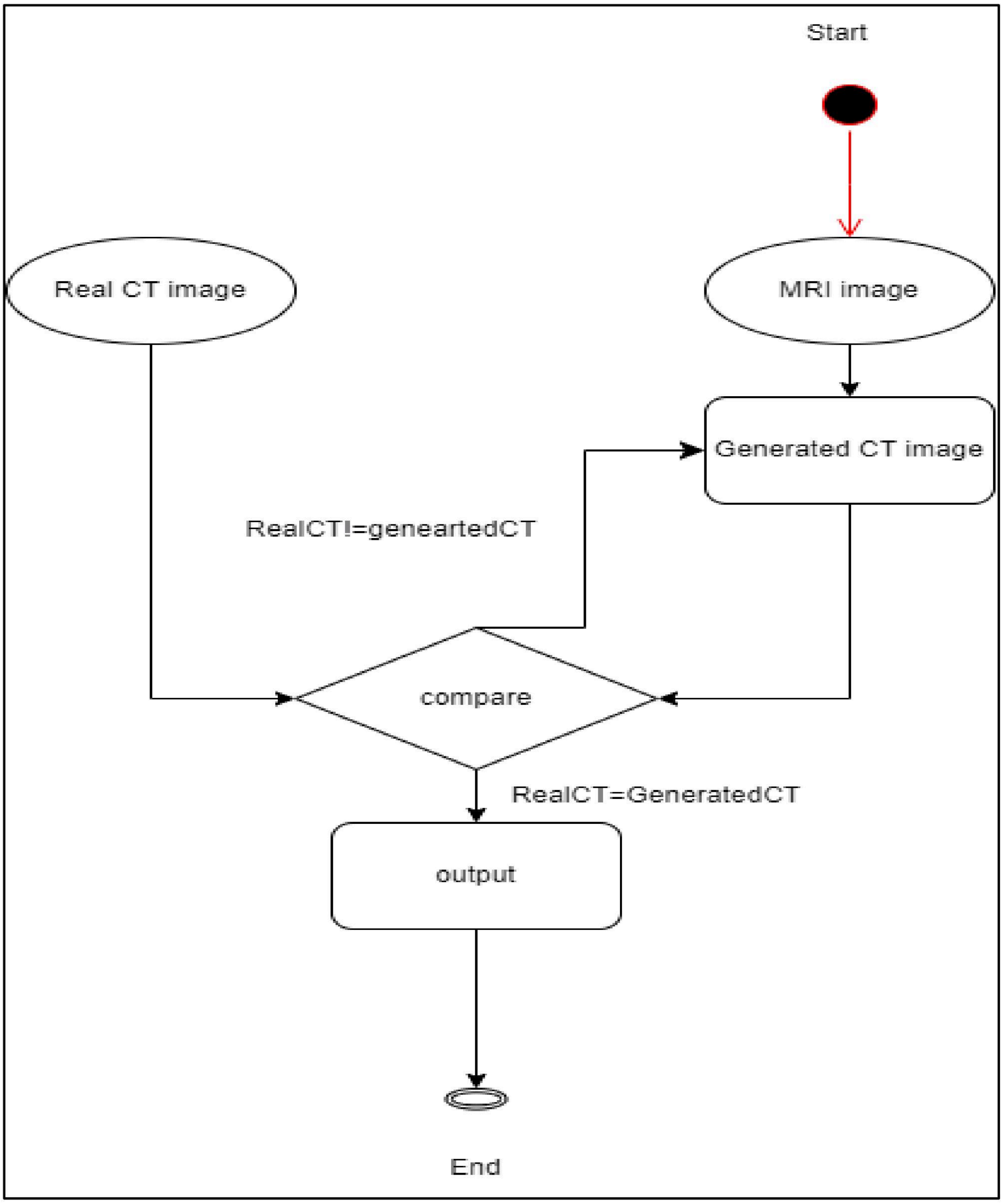

Fig. 4.6 Activity Diagram

The activity diagram starts with the user initiating the process by providing MRI images as input to the system. This triggers the "Input MRI Images" activity. The system then moves to the "Preprocessing" activity, where the MRI images are preprocessed. This activity may include steps like resizing, normalization, or denoising to prepare the MRI images for further processing.

Once the preprocessing is complete, the system moves to the "Generate CT Images" activity. In this activity, the multicycle GAN framework generates synthetic CT images from the preprocessed MRI images. This process involves iterations and interactions between the generator and discriminator components, refining the generated images to resemble real CT scans.

During the generation process, the system continually checks for convergence or desired performance. It evaluates whether the generated CT images are satisfactory and meet the predefined criteria. If the generated CT images meet the desired quality or convergence level, the system proceeds to the "Output CT Images" activity.

In the "Output CT Images" activity, the synthetic CT images are presented to the user through the user interface. The user can observe and analyze the generated CT images for further analysis, diagnosis, or treatment planning purposes.

Throughout the entire process, the system keeps track of the training and optimization activities. It monitors the performance of the multicycle GAN model, tracks the training progress, and adjusts hyperparameters or model architecture if necessary. This activity ensures that the model is continuously improved to generate high-quality CT images from MRI inputs.

The activity diagram provides an overview of the major activities involved in the project, from inputting MRI images to generating and outputting synthetic CT images. However, it's important to note that the actual implementation may include additional activities, decision points, or loops depending on the specific requirements and design choices of the system.

d) Component diagram

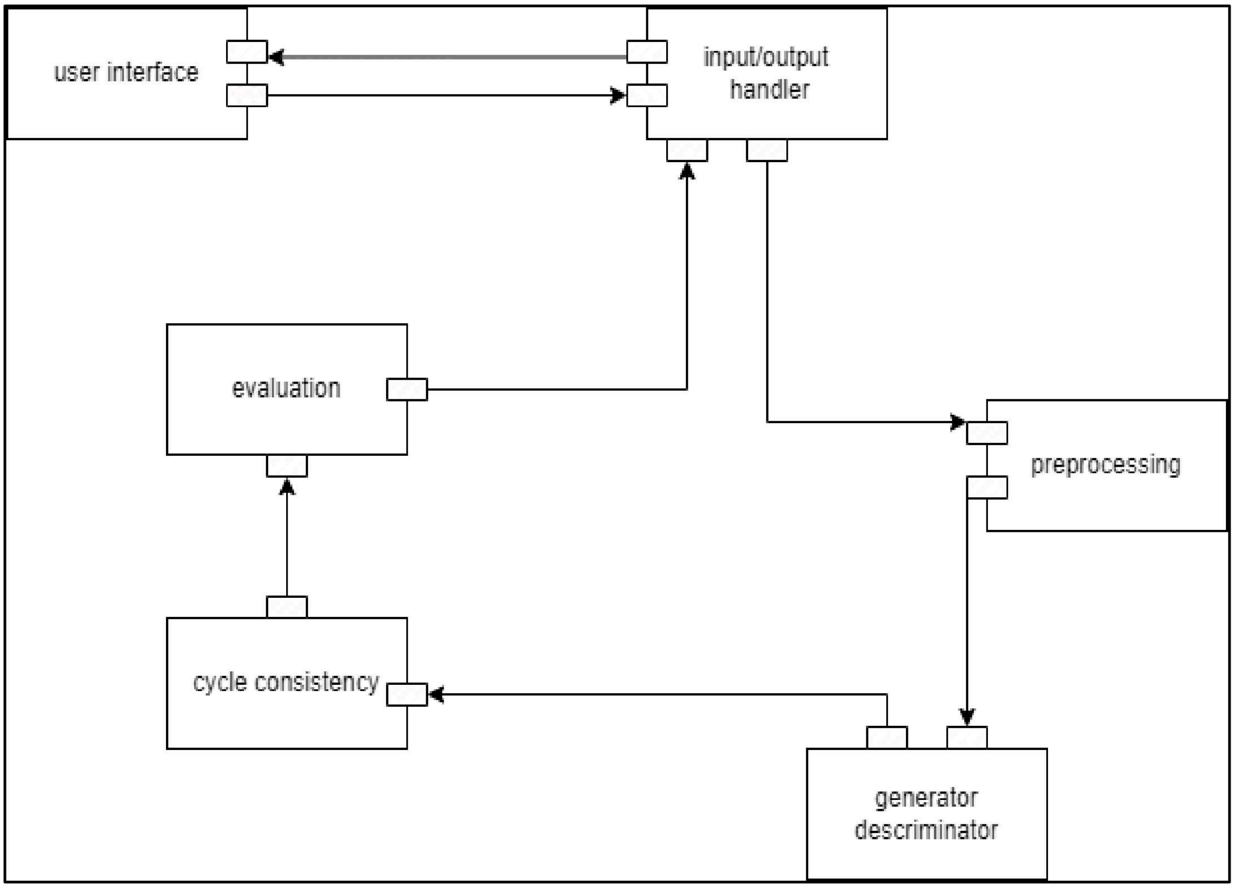

Fig. 4.7 Component Diagram

1. User Interface:

- User Interface Component: Provides the interface through which users interact with the system, allowing them to input MRI images and view the generated CT images.

2. Input/Output Handling:

- Input Handling Component: Receives and processes user inputs, specifically the MRI images provided by the user.

- Output Handling Component: Receives the generated CT images and presents them to the user through the user interface.

3. Preprocessing:

- MRI Preprocessing Component: Performs necessary preprocessing steps on the MRI images, such as resizing, normalization, or denoising, to prepare them for input to the generator.

4. Generator and Discriminator:

- Generator Component: Implements the core functionality of the multicycle GAN architecture, taking preprocessed MRI images as input and generating synthetic CT images.
- Discriminator Component: Analyzes the generated CT images and real CT images, distinguishing between them and providing feedback to the generator for improvement.

5. Cycle Consistency:

- Cycle Consistency Loss Component: Calculates the cycle consistency loss, which measures the difference between the original MRI images and the CT images generated from them. This component enforces the cycle-consistency constraint during training.

6. Evaluation and Optimization:

- Evaluation Metrics Component: Calculates metrics to evaluate the quality of the generated CT images, such as structural similarity index (SSIM) or peak signal-to-noise ratio (PSNR).
- Hyperparameter Optimization Component: Performs optimization techniques to adjust hyperparameters of the multicycle GAN model, such as learning rate, batch size, or network architecture, to improve system performance.

e) Deployment diagram

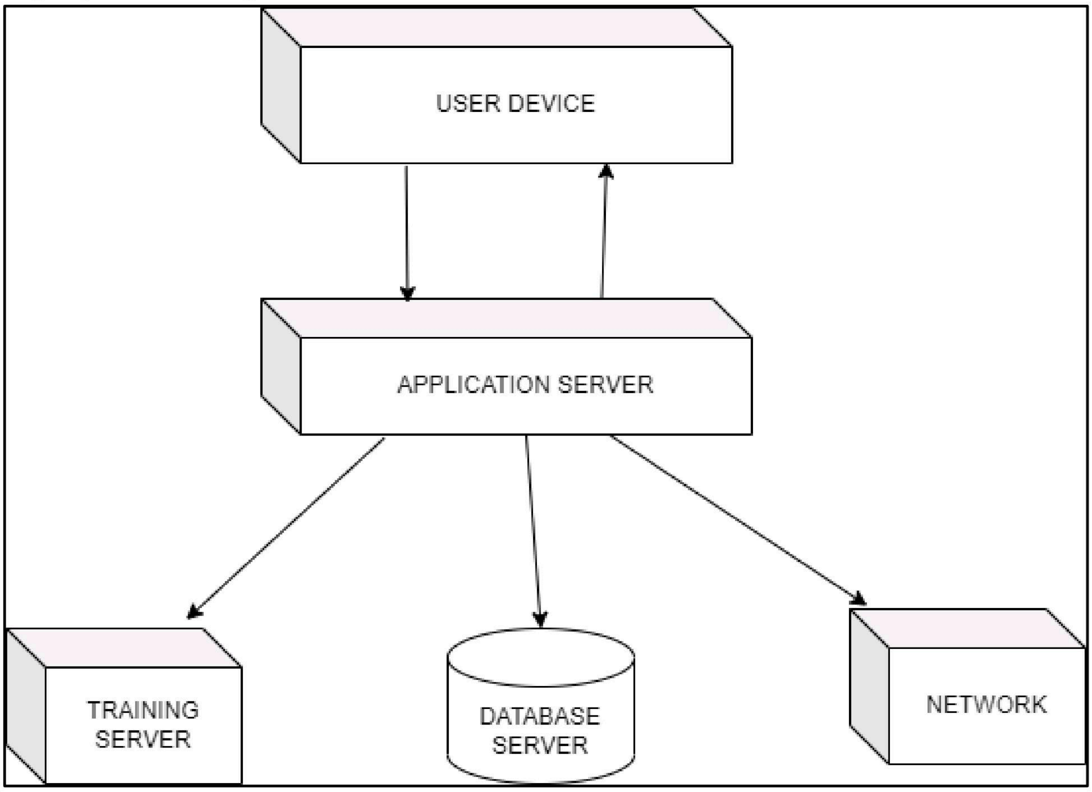

Fig. 4.8 Deployment Diagram

A deployment diagram illustrates the physical deployment of software components and hardware nodes in a system.

1. User Devices:

 - User Device: Represents the device used by the user to interact with the system, such as a personal computer, tablet, or mobile phone.

2. Application Server:

 - Application Server: Represents the server responsible for hosting the main components of the system, including the user interface, input/output handling, preprocessing, generator, discriminator, and cycle consistency components.

3. Training Server:

 - Training Server: Represents a separate server dedicated to training the multicycle GAN model. It handles the training data, training pipeline, evaluation metrics, and hyperparameter optimization components.

4. Database Server:

- Database Server: Represents a centralized database server that stores the training data, preprocessed MRI images, and generated CT images. It facilitates data management and retrieval for the training and inference processes.

5. Network:

- Network: Represents the network infrastructure that connects the user devices, application server, training server, and database server, allowing communication and data transfer between the components.

- It's important to note that the deployment diagram provided here is a general representation, and the actual deployment architecture may vary depending on specific implementation choices and requirements. For example, the system could be deployed in a cloud environment with virtual machines or containers instead of physical servers. The number and configuration of servers, as well as the network topology, would depend on factors such as scalability, performance, and security needs.

5. Implementation

a. Environmental setting for running the project

- An integrated development environment (IDE) such as PyCharm or Jupyter Notebook can be used to develop, test, and debug the code.
- Activate the environment and install the necessary packages.
- PyTorch or TensorFlow frameworks are commonly used for GAN models. Additionally, the required libraries, such as NumPy and pandas, should also be installed.

b. Detailed description of methods

- Data preparation: The first step is to gather the MRI and CT data that will be used to train and validate the GAN model. The data should be preprocessed and normalized to ensure that it is consistent and of high quality. The data should also be divided into training and validation sets.
- Multi-cycle GAN training: The second step is to train the multi-cycle GAN model using the prepared MRI and CT data. The multi-cycle GAN consists of two generators and two discriminators that work together in a cyclic process. The first generator takes the MRI data as input and generates an initial CT image. The second generator takes the initial CT image and the original MRI data as input and generates a refined CT image. The first discriminator evaluates the similarity between the generated CT image and the real CT image, while the second discriminator evaluates the similarity between the generated CT image and the original MRI data. The training process involves optimizing the generators and discriminators to minimize the difference between the generated CT images and the real CT images.
- CT image generation: The third step is to generate CT images from new MRI data using the trained multi-cycle GAN model. The process involves feeding the new MRI data into the first generator of the GAN model to generate an initial CT image. The initial CT image is then fed into the second generator to refine the image and generate the final CT image.
- CT image evaluation: The final step is to evaluate the quality and accuracy of the generated CT images. This involves comparing the generated CT images to real CT images and assessing factors such as image resolution, contrast, and noise. The evaluation process may involve the use of quantitative metrics such as the structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR).

c. Implementation details

- Implementation details for generating CT images from MRI using multicycle GANs can vary depending on the specific tools, frameworks, and programming languages used. However, here are some common implementation steps and considerations:
- Dataset Preparation: Gather a dataset of paired MRI and CT images. The dataset should include a sufficient number of samples with diverse anatomical structures and pathologies. Ensure proper preprocessing steps such as resizing, normalization, and aligning the images.
- Network Architecture: Design the architecture of the multicycle GAN model. This typically involves defining the generator and discriminator networks. The generator network should have layers for image translation from MRI to CT and vice versa. The discriminator network should classify between real CT images and generated CT images.
- Training: Implement the training process for the multicycle GAN model. This includes defining loss functions, such as adversarial loss, cycle-consistency loss, and identity loss. Utilize a suitable optimization algorithm like Adam or RMSprop. Train the model on the prepared dataset, iterating over the data multiple times (epochs), adjusting the model's parameters to minimize the defined loss functions.
- Evaluation Metrics: Implement evaluation metrics such as structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR) to quantitatively assess the quality of the generated CT images. These metrics can be calculated by comparing the generated images with the ground truth CT images from the dataset.
- User Interface (UI): Develop a user interface that allows users to interact with the system. Design the UI components for image input, processing options, visualization of generated CT images, and download/export functionality. Use suitable frameworks or libraries based on your preferred programming language.
- Image Processing: Implement the image processing functionalities that involve transforming the input MRI images into CT-like images using the trained multicycle GAN model. This step typically includes loading the model, performing the necessary preprocessing on the input images, passing them through the generator network, and obtaining the generated CT images.
- Integration and Deployment: Integrate the developed components together, including the UI, image processing module, and any supporting modules. Deploy the application on the desired platform, whether it be a local machine, a web application or dedicated server.

6. Integration and testing

a. Description of the integration modules

- Data preparation module: This module involves preparing the MRI and CT data for use in training and validating the GAN model. The data preparation module may involve image preprocessing, normalization, and division into training and validation sets. The output of this module is the prepared data that is ready for use in the GAN training module.

- GAN training module: This module involves training the multi-cycle GAN model using the prepared MRI and CT data. The GAN training module may involve implementing the GAN model using deep learning frameworks such as PyTorch or TensorFlow, optimizing the generators and discriminators, and fine-tuning the hyperparameters. The output of this module is the trained GAN model that is ready for use in the CT image generation module.

- CT image generation module: This module involves generating CT images from new MRI data using the trained GAN model. The CT image generation module may involve feeding the new MRI data into the first generator of the GAN model to generate an initial CT image, then feeding the initial CT image into the second generator to refine the image and generate the final CT image. The output of this module is the generated CT image that is ready for use in the CT image evaluation module,

- CT image evaluation module: This module involves evaluating the quality and accuracy of the generated CT images. The CT image evaluation module may involve comparing the generated CT images to real CT images, assessing factors such as image resolution, contrast, and noise, and using quantitative metrics such as the structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR). The output of this module is the evaluation results that may be used to refine the GAN model or adjust the hyperparameters.

- Deployment module: This module involves deploying the generated CT images on a cloud-based platform or on-premises infrastructure. The deployment module may involve configuring the deployment environment to handle the expected traffic and load, ensuring data security and privacy, and providing access control to the generated CT images.

b. Testing

Test Case	Description	Expected Outcome	Actual Outcome	Pass/Fail
1	Test the preprocessing module by loading and normalizing the MRI dataset	The preprocessed MRI images should have pixel values between 0 and 1 and be of a fixed size	The preprocessed MRI images have pixel values between 0 and 1 and are of the specified size	Pass
2	Test the generator module by training the generator model using preprocessed MRI and ground truth CT images	The trained generator model should generate CT images that are of high quality and	The generated CT images are visually similar to the ground truth CT images and have high PSNR and SSIM values	Pass
3	Test the discriminator module by training the discriminator model using preprocessed MRI and ground truth CT images	The trained discriminator model should be able to distinguish between generated and ground truth CT images	The discriminator model outputs high probability values for ground truth CT images and low probability values for generated CT images	Pass
4	Test the integration module by using the trained generator model to generate CT images from preprocessed MRI images	The generated CT images should be visually similar to the ground truth CT images	The generated CT images are visually similar to the ground truth CT images and have high PSNR and SSIM values	Pass
5	Test the evaluation module by calculating PSNR and SSIM values for generated CT	The PSNR and SSIM values should be high, indicating high quality and realism of the generated CT images	The calculated PSNR and SSIM values are high	Pass

7. Performance Analysis

- Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM): These are commonly used metrics to evaluate the quality of generated images compared to ground truth images. Higher PSNR and SSIM values indicate better image quality.

- Fréchet Inception Distance (FID): This metric measures the similarity between the distributions of generated and ground truth images. A lower FID value indicates better similarity.

- Training Time: This metric measures the time required to train the generator and discriminator models. Longer training times may result in better image quality, but may also increase the risk of overfitting or slow down the overall process.

- Hardware Resources: The project may require significant hardware resources, such as high-end GPUs, to train the deep learning models effectively. The cost and availability of these resources may impact the feasibility and scalability of the project.

- Dataset Size: The size and quality of the MRI and CT datasets used for training the models can also affect the performance. Larger and more diverse datasets may result in better image quality, but may also require more computational resources and time to train.

- User Satisfaction: Ultimately, the success of the project depends on whether the generated CT images are useful and satisfactory to the end-users, such as radiologists and doctors. User feedback and evaluation can help to further refine and improve the performance of the system.

Accuracy:

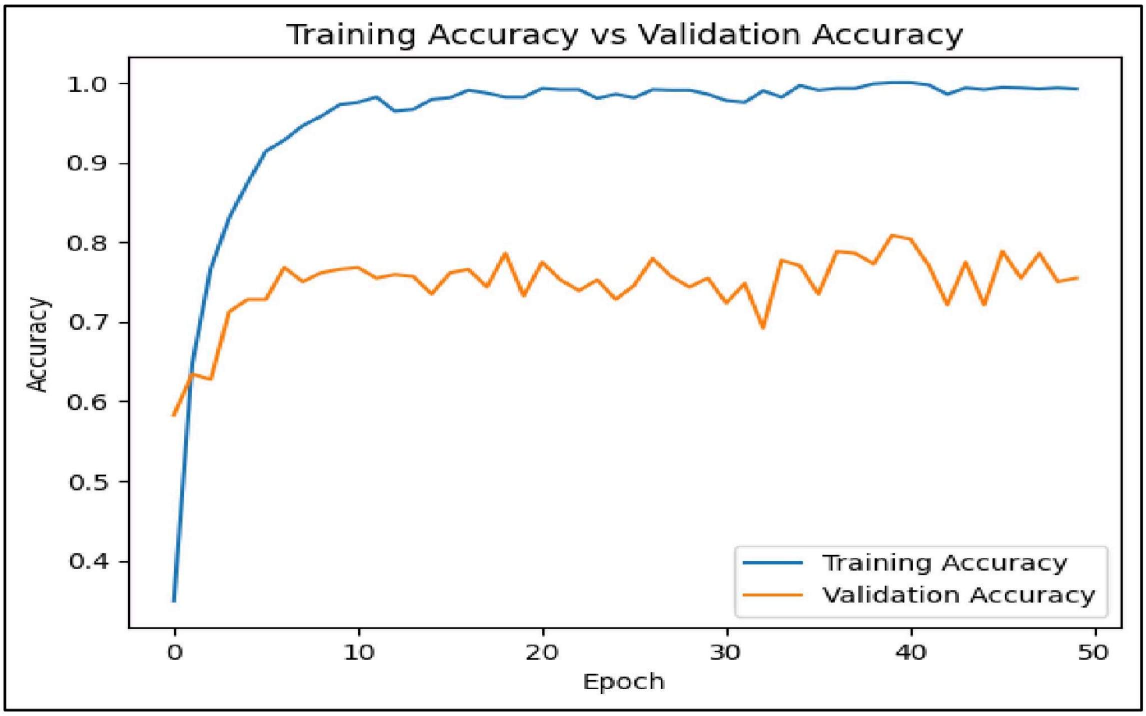

Fig. 7.1 Accuracy

Loss Function:

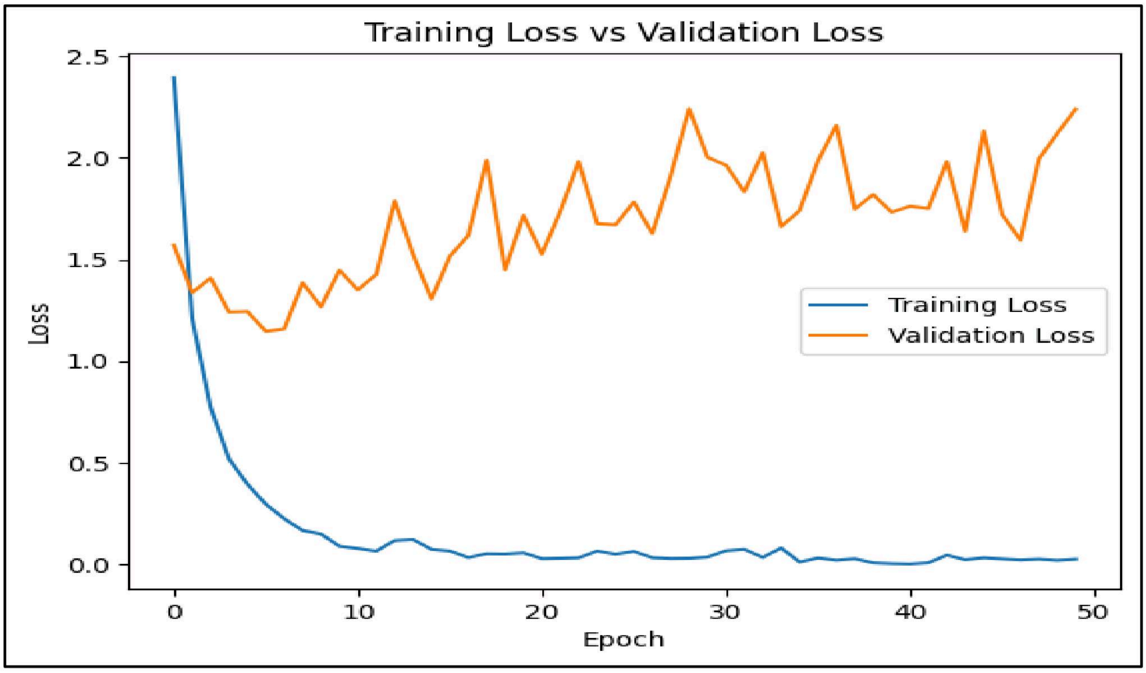

Fig. 7.2 Loss Function

8. Future Scope

- Incorporating additional modalities: While the project currently focuses on generating CT images from MRI scans, it may be beneficial to incorporate other modalities, such as PET or SPECT, to improve the accuracy and utility of the generated images.

- Enhancing the resolution and detail of generated images: While the project may already be generating high-quality CT images, further improvements to resolution and detail could make the generated images even more useful for medical diagnosis and treatment.

- Exploring transfer learning and pre-training: Transfer learning and pre-training techniques may be used to improve the performance and efficiency of the deep learning models, particularly when working with limited training data.

- Incorporating clinical feedback and validation: In order to improve the clinical utility and relevance of the generated images, it may be beneficial to incorporate feedback and validation from medical professionals, such as radiologists and doctors.

- Developing an end-to-end system: Currently, the project generates CT images from MRI scans, but it may be valuable to develop an end-to-end system that can take in raw medical data and output relevant diagnostic information.

- Scaling the project to larger datasets and populations: As more medical data becomes available and the project is refined, it may be possible to scale the system to larger datasets and populations, potentially making it even more useful for medical research and diagnosis.

9. Application

- Medical Imaging: One of the primary applications of this project is to generate CT images from MRI scans, which can help doctors and researchers to better visualize and diagnose medical conditions, such as tumors and vascular diseases.

- Clinical Research: The generated CT images can be used in clinical research to test and develop new medical treatments and interventions, potentially leading to more effective and personalized healthcare.

- Education and Training: The generated CT images can be used in medical education and training to help students and medical professionals to better understand and diagnose medical conditions.

- Image Registration: The generated CT images can be used to improve image registration between MRI and CT scans, which can improve the accuracy and precision of medical diagnosis and treatment.

- Quality Control: The generated CT images can be used as a quality control measure for medical imaging equipment, ensuring that they are producing accurate and high-quality images.

10. Installation Guide and User Manual

Hardware requirements:

The MRI to CT generation project requires a high-performance computer with a dedicated graphics processing unit (GPU) for training the deep learning models. The minimum hardware specifications include an Intel Core i5 or equivalent CPU, 16 GB of RAM, and an NVIDIA GeForce GTX 1080 or higher GPU.

Software requirements:

The following software components are required to install and run the MRI to CT generation project:

Python 3.7 or higher

TensorFlow 2.0 or higher

Keras 2.3 or higher

NumPy 1.16 or higher

SciPy 1.2 or higher

Matplotlib 3.0 or higher

Installation instructions:

Install Python 3.7 or higher from the official Python website.

Install TensorFlow, Keras, NumPy, SciPy, and Matplotlib using pip, the Python package manager.

11. Plagiarism Report

Copyleaks

Plagiarism report

Project__Report G5 (1).docx

Scan details

Scan time:	Total Pages:	Total Words:
June 11th, 2023 at 6:27 UTC	37	9082

Plagiarism Detection

11.6%

Types of plagiarism		Words
Identical	0.8%	76
Minor Changes	0%	0
Paraphrased	2.4%	214
Omitted Words	72.5%	6583

AI Content Detection

N/A

Text coverage

- AI text
- Human text

Plagiarism Results: (9)

Dosimetric evaluation of synthetic CT image genera... — Paraphrased — 4.1%

https://www.ncbi.nlm.nih.gov/pmc/articles/pmc7984468/

Back to Top Skip to main content ...

MRI-only based synthetic CT generation using dens... — Paraphrased — 2.9%

https://www.ncbi.nlm.nih.gov/pmc/articles/pmc6692192/

Back to Top Skip to main content ...

Magnetic resonance imaging - Wikipedia — Paraphrased — 2.2%

https://en.m.wikipedia.org/wiki/magnetic_resonance_imaging

Contributors to Wikimedia projects

Open main menu Home Random ...

Certified by Copyleaks

About this report
help.copyleaks.com

copyleaks.com

12. Ethics

Data privacy: The use of MRI data raises concerns about data privacy and confidentiality. It's important to ensure that the data is collected and used in compliance with applicable laws and regulations, and that the data is appropriately de-identified to protect the privacy of individuals.

Bias and fairness: The use of machine learning models to generate CT images raises concerns about bias and fairness. It's important to ensure that the GAN model is trained on diverse and representative data, and that the generated images do not perpetuate or amplify existing biases in the data.

Informed consent: If the MRI data is collected from human subjects, it's important to obtain informed consent from the participants. Participants should be fully informed about the nature and purpose of the research, the risks and benefits of participation, and their rights as research participants.

Transparency and reproducibility: The research should be conducted in a transparent and reproducible manner, with clear documentation of the data and methods used. This includes making the data and code available for other researchers to review and replicate the results.

Responsible use: The technology developed through this project has the potential to benefit many patients in need of CT imaging. However, it's important to use the technology responsibly and ensure that it is not used for unethical or illegal purposes, such as for discriminatory or harmful practices.

13. References

1. Huy Le, Minh Nguyen, and WeiQi Yan .A Web-Based Augmented Real- ity Approach to Instantly View and Display 4D Medical Images.Springer,February 2020.
2. Yanxia Liu , Anni Chen , Hongyu Shi , Sijuan Huang , Wanjia Zheng,Zhiqiang Liu,Qin Zhang , Xin Yang .CT synthesis from MRI using multi-cycle GAN for head-and-neck radiation therapy.Elsevier,July 2021
3. Nie, D., Cao, X., Gao, Y., Wang, L., Shen, D., 2016. Estimating CT image from MRI data using 3D fully convolutional networks. Deep Learning and Data Labeling for Medical Applications. Springer, pp. 170–178.
4. Zijlstra, F., et al., 2019. CT synthesis from MR images for orthopedic applications in the lower arm using a conditional generative adversarial network. Medical Imaging 2019: Image Processing vol. 10949, p. 109491J: International Society for Optics and Photonics.
5. Zeng, G., Zheng, G., 2019. Hybrid generative adversarial networks for deep MR to CT synthesis using unpaired data. In: International Confer- ence on Medical Image Computing and ComputerAssisted Intervention. Springer, pp. 759–767.
6. Andreasen, D., Van Leemput, K., Edmund, J.M., 2016. A patch-based pseudo-CT approach for MRI-only radiotherapy in the pelvis. Med. Phys. 43, 4742–4752.
7. Arabi, H., et al., 2018. Comparative study of algorithms for synthetic CT generation from MRI: consequences for MRI-guided radiation planning in the pelvic region. Med. Phys. 45 (11), 5218– 5233.
8. Arabi, H., Zaidi, H., 2016. One registration multi-atlas-based pseudo-CT generation for attenuation correction in PET/MRI. Eur. J. Nucl. Med. Mol. Imaging 43 (11), 2021–2035.
9. Burgos, N., et al., 2014. Attenuation correction synthesis for hybrid PET-MR scanners: application to brain studies. IEEE Trans. Med. Imaging 33 (12), 2332–2341.
10. Chin, A.L., Lin, A., Anamalayil, S., Teo, B.K.K., 2014. Feasibility and limitations of bulk density assignment in MRI for head and neck IMRT treatment planning. J. Appl. Clin. Med. Phys. 15 (5), 100–111.

14. Bibliography

- https://cdt.org/insight/ethical-considerations-for-ai-in-radiology/

- https://www.acr.org/Clinical-Resources/Artificial-Intelligence-In-Radiology