Assignment 2

Instructions:

1. **Submission Requirements:**
    - Upload the Jupyter notebook in HTML or PDF format with outputs and its corresponding PDF.
    - You may also choose to use Google Colab for working on the project.
    - Data files do not need to be uploaded with the submission.
    - Ensure the updated Jupyter notebook is submitted with properly formatted and aligned outputs. Incomplete outputs, misalignments, or poorly written comments will result in a deduction of marks.
    - Partial code and partial output will be evaluated, and marks will be awarded based on the PDF file.
    - Only the latest submission would be considered for marking.
2. **File Naming Convention:**
    - Name the file as: CV_assignment2_group_problemstatmentnumber
3. **Plagiarism Policy:**
    - Any form of plagiarism will be taken very seriously, resulting in zero (0) marks.
    - All submissions must be the result of your original effort.
    - Copying from any sources, whether online or from peers, is strictly prohibited.
    - Unauthorized collaboration to gain an unfair advantage is prohibited.
    - Both the person sharing resources and the one receiving them will face consequences for plagiarism.
    - Identical or significantly similar submissions will be investigated, and severe punishments will be imposed on those found guilty.
4. **Late Submission Policy:**
    - Late submissions will incur a penalty of -2 marks.
5. **Queries:**
    - For any questions regarding the assignment, use the discussion forum.
6. **Programming Instructions:**

- Use appropriate Python programming techniques to read and process the dataset.
- Convert the notebook into HTML or PDF format along with outputs and upload the file.

- Avoid submitting excessively long notebooks by eliminating irrelevant lengthy prints.

9. **Problem Statement Selection:**

- Choose **ANY ONE** problem statement from the three provided below.
- Enter the problem statement number in the following sheet (Do not edit others' responses): https://docs.google.com/spreadsheets/d/1xetRC_POpf8bbztxZ_b_mNiipzHUKpG20YFoPUspfGY/edit?usp=drive_link

==========

Assignment 2 Rubric

Criteria	Description	Pts
Data Preprocessing	Implement necessary preprocessing steps such as normalization, resizing, and semantic segmentation to prepare data for model input.	2.5 pts
Model Development	Implement the model (e.g., Faster R-CNN) and integrate relevant techniques (contextual awareness, multi-task learning, etc.) for improved performance.	5 pts
Evaluation Metrics	Evaluate model performance using appropriate metrics (precision, recall, F1-score, speed, etc.) and justify their relevance to the task.	2.5 pts
Justification	Analyze and explain the results, including reasons for the model's success or poor performance (e.g., overfitting, underfitting, model choices).	2.5 pts
Documentation, Study Presentation, and Code Quality	Ensure clear, readable code and well-organized documentation. Present the study logically, summarizing the problem and key findings.	2.5 pts

Total:15pts

==========

1. Develop a deep learning fusion model that integrates both CT and MRI data using cross-modal attention mechanisms to perform feature-level fusion, enhancing multimodal representation learning.

Dataset : https://www.kaggle.com/datasets/darren2020/ct-to-mri-cgan
Links to an external site.

Objectives:

1. **Data Preprocessing & Augmentation:**
    - Perform image alignment, normalization, and augmentation for both CT and MRI images to improve model robustness.
    - Implement appropriate preprocessing techniques for optimal feature extraction.
2. **Cross-Modal Feature Extraction & Fusion:**
    - Design a **dual-branch CNN or Vision Transformer** to extract features separately from CT and MRI images.
    - Implement **cross-modal attention** to fuse features from both modalities, allowing CT and MRI to enhance each other's representations.
    - **Integrate the fused features** into a unified representation that captures complementary information from both modalities.
3. **Evaluation & Benchmarking:**
    - Compare the proposed **fusion-based** model against single-modality models (CT-only, MRI-only).
    - Use appropriate evaluation metrics to assess **the quality of the fused feature representations** and model effectiveness.

2. For the given dataset https://www.kaggle.com/datasets/darren2020/ct-to-mri-cgan

Links to an external site.

Problem Statement: Develop an AI model to detect anomalies by learning normal patterns and identifying deviations. Also include the following in the implementation:

1. **Preprocessing & Augmentation** – Perform necessary preprocessing steps such as normalization, resizing, and data augmentation to improve model performance.
2. **Feature Extraction** – Use deep learning models (e.g., CNNs, autoencoders, or transformers) to extract meaningful features from CT and MRI images.
3. **Unsupervised Anomaly Detection** – Implement an anomaly detection method, such as:

- **Autoencoders** (learning normal patterns and detecting deviations)
- **One-Class SVMs** (traditional anomaly detection methods)

4. **Comparison & Evaluation** – Compare different methods using evaluation metrics like AUC-ROC, reconstruction error, or anomaly scores to determine the best-performing approach.

3. Advanced Object Tracking and Detection in Video Streams

Problem Statement: Develop an advanced object tracking and detection system that utilizes the Faster R-CNN model to accurately identify and track multiple objects in video streams. The system should incorporate novel techniques such as temporal consistency checks and adaptive tracking to enhance performance in dynamic environments.

- **Objectives**:
    - a. Extract frames from video sequences and perform normalization to standardize input data.
    - b. Implement data augmentation techniques such as random cropping, flipping, and color jittering to improve model robustness.
    - c. Design a Faster R-CNN model for object detection, fine-tuning it on the selected dataset.
    - d. Integrate a temporal consistency check mechanism to ensure that detected objects maintain consistent identities across frames.
    - e. Implement adaptive tracking algorithms (e.g., Kalman filter or SORT) that adjust tracking parameters based on object speed and direction.
    - f. Evaluate the model's performance using metrics such as mean Average Precision (mAP), tracking accuracy, and identity switch rate.
    - g. Compare the performance of the proposed system against baseline models and other state-of-the-art tracking algorithms. (Optional)
    - **Data Preprocessing**:

    - **Model Development**:
    - **Evaluation**:
- **Dataset**: https://motchallenge.net/data/
- **Sample Papers**:
    - "Real-Time Object Detection and Tracking Using Faster R-CNN" - https://arxiv.org/abs/2006.04567
    - "A Survey on Object Detection and Tracking" - https://www.sciencedirect.com/science/article/pii/S0031320321001234