Final Project Instructions

Deadline: April 28, 12:00 PM (noon)
Submission Format: One submission per group (no duplicate submissions)

Project Overview

Each group will complete **two major tasks**:

Part A: Image Generation, Object Detection, and Image Captioning

You will build a pipeline that performs:

1. Image generation using GAN or VAE models
2. Object detection on the generated images
3. Image captioning

You are required to use the MS COCO dataset for training these models. The dataset can be accessed and downloaded from the official COCO website: https://cocodataset.org/#download. Please randomly select 20,000 images from the MS COCO dataset to build and train your models.

Part B: Reinforcement Learning Strategy Design

Design an optimal strategy for a computer (AI) to play a video game using **Reinforcement Learning**.
Note: No coding is required for this part. Focus on hypothesis and design.

Submission Requirements

Each group must submit the following **three files separately** as part of a **single group submission**:

1. **Jupyter Notebook (.ipynb)**
    - All relevant code for Part A
    - Make sure it runs correctly and is clearly documented
2. **Project Report (minimum 7 pages)**
    - Organized by sections described below
3. **Presentation Slides (approx. 20 slides)**
    - Based on the report content, but more concise
    - To be used during in-class presentations

Presentation Guidelines

- Each student must participate in the group presentation.
- Each group will have **20 minutes** total.
- Once your presentation is over, please leave the classroom quietly to avoid interrupting others.
- Presentation will be graded based on:
  (Presenting = 20 points)
    - Clarity and professionalism of delivery
    - Ability to answer questions effectively

Detailed Project Components

Part A: Image-Based Deep Learning Tasks (55 points total)

Python Coding Tasks

1. Generate 5 images using a GAN or VAE model
2. Perform object detection on these images
3. Generate captions for the images

Report Requirements for Part A

- Task overview and dataset summary
- Detailed explanation of your models, architectures, and rationale
- Visual outputs:
    - Display the 5 generated images
    - Show detected objects
    - Include generated captions

Part B: Reinforcement Learning Game Strategy (25 points)

(No coding required)

Your design should include:

1. **Game Description**
    - Choose any video game (e.g., a simplified version of *Plants vs. Zombies*)
    - Describe actions, states, rewards, and penalties
2. **Modeling Hypothesis**
    - Outline your action rules and reward/state distribution
    - Specify what kind of neural network structure (layers, activation functions) you plan to use
3. **Reward Strategy**
    - Define your final reward structure (e.g., binary win/loss, cumulative rewards, risk-adjusted rewards)
4. **Analysis Framework**
    - Write down your proposed analytical procedure
    - Assume someone else will implement the code based on your description

---

Formatting and Minor Requirements

- Use your **business question or project title** as the main title for both the report and slides.
- List your **group number and members (sorted by last name)** on the first page of both the report and slides.
- **Avoid using raw Python code screenshots** in your slides and report.
  - Instead, present results in **tables**, **diagrams**, and **visualizations** where possible.

Update:

Hello everyone,

If you have troubles to train the models, please try to follow guidelines when running your projects:

1. **Use GPU in Colab**: We strongly recommend utilizing the GPU runtime in Google Colab rather than using the CPU version in a local Jupyter notebook. This will significantly speed up your computations and improve model training efficiency.
2. **Image Resolution**: Begin your experiments with small-scale resolutions such as `128 x 128` or `64 x 64`. Once successful, you may incrementally increase resolutions based on your computational capacity.
3. **Dataset Size Increment**: Initially test your model using a smaller subset, for example, `1000 images`. After verifying the model runs correctly, incrementally increase your dataset to `5000 images` and beyond.
4. **Downloading Data in Colab**: When downloading large files such as `train2017.zip` from the COCO dataset, please use the following

chunk-based approach to avoid issues:

```
print("Downloading train2017.zip (this may take a while)...")
url = "http://images.cocodataset.org/zips/train2017.zip"
response = requests.get(url, stream=True)
total_size = int(response.headers.get('content-length', 0))
chunk_size = 1024 * 1024 # 1 MB
with open("train2017.zip", "wb") as f:
for data in tqdm(response.iter_content(chunk_size), total=total_size // chunk_size, unit='MB'):
f.write(data)
```

Update:

Hello everyone,

For groups that were able to generate acceptable images using GANs or VAE models, please continue using your 5 generated images for the object detection and image captioning tasks.

For groups that could not produce satisfactory images from the GANs or VAE models, please randomly select 5 images from the following website:
https://storage.googleapis.com/openimages/web/download_v7.html
Then proceed with object detection and image captioning using those images.

Please do your best. If you are still unable to produce strong results before your presentation, simply present what you have and clearly explain the different strategies you tried. I will still give you a good score for your effort and thoughtful work.