The Chinese University of Hong Kong, Shenzhen
School of Data Science

DDA4220/MDS5122/AIR5011/AIR6011/MBI6011 Final Project

Due by: 23:59, May 5th, 2025

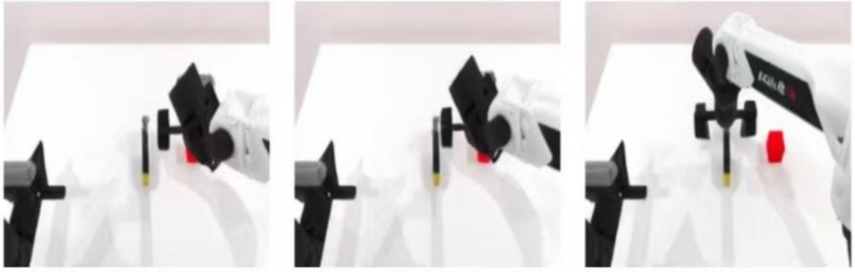
Instructions:

1. You must submit your files on Blackboard. Please upload a PDF file along with your code (excluding large files such as the dataset).
2. Your submission must be well-presented to receive full credit. Ensure that your solutions are legible and written in English.
3. The programming language is Python, and your code should include necessary comments so that others can easily follow its logic. You are required to use PyTorch.
4. Your report should reflect your individuality and originality. If your answers are inspired by other sources (*e.g.*, the Internet or AI), please maintain academic integrity by including a reference or acknowledgment section in your submitted paper and indicate how these sources inspired you.
5. Late submissions or instances of plagiarism will not be graded.

A. AI Model for Robotic Action Frame Prediction (100 points)

Predicting robotic actions is an important task because making accurate predictions ensures that the robot can operate as expected and safely. In this final project, you are asked to make a visual prediction about a robot's behavior in a virtual simulation environment.

Specifically, you will train an image generation model to predict a future frame of a robotic action. Given an observed image of what the robot is currently seeing (*e.g.*, a table with multiple items on it) and a textual action instruction (*e.g.*, "hit the block with the hammer"), the model should predict what the robot will see 50 frames later. The image size must be at least 128x128.



Your report **MUST** be written using the CVPR 2025 Author Kit in a camera-ready format, with your names, student IDs, and affiliations clearly displayed, and must adhere strictly to CVPR's requirements as if you were submitting to this conference, *e.g.*, a maximum of eight pages, although you may not actually submit it. You must also release and upload your code as a supplement to the Blackboard system, along with a Markdown document that clearly instructs others on how to set up the environment and reproduce your results. If you are confident in your project, you may optionally submit it to any other conference you like.

Apart from the requirements stated above, there are no concrete restrictions on how your method should be implemented. If you cannot come up with your own ideas, you can follow the suggestions below to complete this final project.

Suggestions (Not Mandatory):

1. You can leverage a pretrained InstructPix2Pix model as a base and fine-tune it for this task. Fine-tune the model to accept the current frame as well as a textual action instruction as inputs and predict a future frame.
2. You can use RoboTwin to help generate the dataset for training and testing. You may only focus on three tasks: `block_hammer_beat`, `block_handover`, `blocks_stack_easy`. For each task, please generate 100 observations; the model can be fine-tuned using these 300 observations with the text descriptions "beat the block with the hammer", "handover the blocks", and "stack blocks", respectively.
3. Evaluate using the SSIM (Structural Similarity Index) and PSNR (Peak Signal-to-Noise Ratio) metrics.
4. Additionally, you may find GR-MG useful.
5. If you don't know what a typical CVPR paper looks like, you just need to refer to and emulate their writing style and the way they present their methods. For example, the ResNet paper is an excellent reference.

Scoring Criteria:

Your report/paper must contain the following sections:

1. Introduction **(5 points)**
2. Related Work **(2 points)** Contain at least 5 relevant papers and cite them
3. Method **(35 points)**
4. Experiment **(40 points)** Comparison/ablation experiments are not mandatory
5. Conclusion and Future Work **(2 points)**
6. Distribution of the Workload **(1 points)** Detail who conducts which part of the project

Quality, clarity, reproducibility, and rationality of the code, paper, and design **(15 points)**

We will review and score each part of your paper as if you had submitted it to the CVPR conference. Using your own ideas, conducting ablation studies, performing comparative experiments, and so on means that you have a good chance of earning higher credits.

Some Tips on Experiments and Computational Resources:

1. **Adjusting Hyperparameters:** Consider adjusting hyperparameters, *e.g.*, model size, batch size, to ensure your experiments can be conducted within the anticipated time budget. You may include various techniques to reduce memory usage, such as half-precision training and quantized training. It is advisable to train using a GPU, with the experiment typically requiring a minimum of 12GB of GPU memory.
2. **NVIDIA GPUs:** If you have an NVIDIA GPU on your local machine, I recommend installing the CUDA version of PyTorch. This will allow the training process to utilize the GPU for acceleration.
3. **Online Free GPUs:** If you only have access to a CPU, you can explore online computational resources. Google Colab is an excellent platform for running deep learning experiments online. It is completely free to use, just sign in with a Google account. Similarly, you might consider Kaggle and Tianchi, both of which offer free GPU resources.
4. **University Resources:** Additionally, our school's high-performance computing platform is available, although registration is required and it is not free.
5. **Third-Party GPU Providers:** There are also several third-party GPU cloud providers, such as GpuShare and AutoDL, which offer affordable options, though they are not free.

If you have any issues, please don't hesitate to ask in our Q&A WeChat group.