# Project: Image2Image Translation & Image Detection

*COMP4423 Computer Vision 2024/25 Semester 2*

Deadline: 2025, May 10th (Saturday), 23:59

## Important Notes

- Form your groups, each consists of no more than three students, on or before **6 April 2025**. Afterward, students with no group will be randomly grouped, and requests for group change are only approved if written agreements from all members of the involved groups are provided before or on **13 April 2025**.
- Compress the source code and the final report in PDF into a single ZIP archive and submit it on Blackboard before or on **10th May 2025**.
- Each member in the group need to submit an individual report. The report should be prepared **from your own perspective and reflect your understanding of the project**. It is mandatory for the individual report to demonstrate a distinct difference of at least 30% from the reports of other group members.

## 1 Task Description

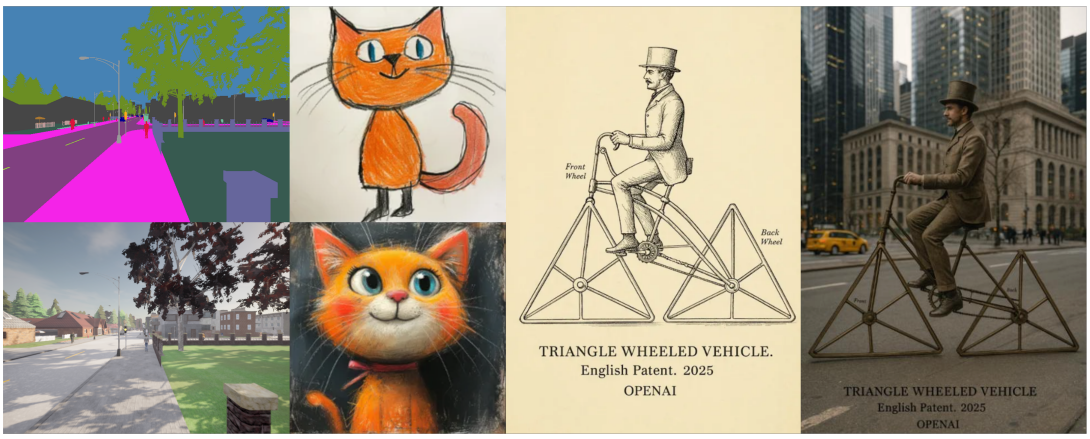


Figure 1: Examples of image-to-image translation pairs

Many problems in image processing, computer graphics, and computer vision can be framed as the task of translating an input image into a corresponding output image. Drawing an analogy to automatic language translation, we define automatic image-to-image translation as the task of transforming one representation of a scene into another.

This domain has witnessed significant advancements through various methodologies. Recent powerful models like GPT-4o and stable diffusion have demonstrated remarkable capabilities in

solving such tasks, often exhibiting generalized abilities. Prior to these advancements, Generative Adversarial Networks (GANs) played a pivotal role in achieving image-to-image translation. Notable examples include Pix2Pix and CycleGAN. These approaches primarily focused on training a domain-specific model using a dedicated training dataset. Typically, these models employed a U-Net generator and a classifier discriminator, trained iteratively. In this project, you are required to build and train such a network to gain a deeper understanding of pixel-to-pixel image transformation. This experience with domain-specific model training can serve as a foundation for exploring more advanced image translation tasks, including image-to-image (img2img) and even text-to-image (text2img) generation.

To achieve this, an effective domain-specific image-to-image translation system should focus on the following key aspects:

- **Domain-Specific Dataset Construction**: As this project focuses on a domain-specific model, the collection of a specific dataset is a crucial initial step.
- **Generator Training**: Train a generator network to perform pixel-to-pixel transformation. A fundamental architecture for this purpose is the U-Net.
- **Discriminator Training**: Train a discriminator (classifier) network that takes two images as input and evaluates whether the generated image meets the required quality standards, distinguishing between "real" and "fake" (generated) images.
- **Iterative Training**: Iteratively train both the generator and discriminator networks over several epochs to achieve convergence.

It's important to remember that this project is about learning by doing. **Don't worry too much about getting perfect results.** Furthermore, you are required to submit a comprehensive report outlining your step-by-step process for completing the assigned tasks. This report should encompass your ideas, algorithm design, any challenges encountered, corresponding solutions implemented, experimental results, and significant findings made throughout the project.

Please note that this project involves the following key stages: **dataset collection**, defining the **network architecture** (specifically the generator and discriminator), designing the **training algorithm** (including loss functions and optimization), and **validating** the performance of the trained model by evaluating the quality of the generated images. Furthermore, you are encouraged to test your trained model on new, unseen data within your chosen domain.

**For data collection, you have the flexibility to gather data in various ways.** A reference image dataset (**reference.zip**) of semantic-real image pairs is provided. You can also collect images from public open-source datasets suitable for this dataset. The specific domain can involve very fine-grained pixel-to-pixel mapping with strictly paired images, or it can be style transfer, requiring only the style.

For network construction and training, you can utilize a framework such as PyTorch. This project requires the implementation of two networks: a generator and a discriminator. **However, the specific architecture for each network is flexible and open to your choice.** Similarly, **the selection of model parameters and training parameters is also flexible.** **Please note that utilizing advanced generative AI tools like Midjourney for the generator is not permitted for this project.** Images generated with such AI tools can only serve as a reference for comparison with your model.

## 2 YOLO Detection Task

### Task Overview

In addition to the image-to-image translation task, this project includes performing object detection using a pretrained YOLO (You Only Look Once) model. Object detection involves locating and classifying objects within an image, and YOLO achieves this by dividing the image into a grid and predicting bounding boxes and class probabilities for each grid cell.

The goal is to apply a pretrained YOLO model on the images generated from your image-to-image translation model. Since this task uses a pretrained model, the primary objective is not to retrain the YOLO network but rather to fine-tune or adapt the pretrained model to improve its performance on the generated data.


Figure 2: Examples of YOLO detection results

### Key Objectives

- **Applying Pretrained YOLO Model:** Use an existing pretrained YOLO model (e.g., YOLOv3, YOLOv4, or YOLOv5) to perform object detection on the images generated by your translation model.
- **Model Fine-Tuning or Adaptation:** Instead of training a new YOLO model from scratch, explore strategies to enhance the pretrained model's performance on the generated images. Potential approaches include:
  - **Test-Time Adaptation (TTA):** Dynamically adjust model parameters during inference to align with the distribution of the generated images.
  - **Data Augmentation and Fine-Tuning:** Fine-tune the pretrained YOLO model by adding a small subset of labeled samples from your generated images to the original dataset.
- **Evaluating Model Performance:** Evaluate the performance of the adapted YOLO model using key metrics such as:
  - **mAP (Mean Average Precision):** Assessing the model's detection accuracy.
  - **Precision and Recall:** Measuring the balance between correct and missed detections.

– **FPS (Frames Per Second):** Validating the real-time performance of the model.

### Important Considerations

- **No Retraining from Scratch:** You are not required to retrain a YOLO model from scratch. Instead, the focus is on applying a pretrained model and improving its performance on the generated images.
- **Data Augmentation and Fine-Tuning:** If you choose to fine-tune the model, ensure that the new training data is properly labeled, and only a limited amount of fine-tuning is performed to prevent overfitting.
- **Evaluation Metrics:** Report mAP, Precision, Recall, and FPS before and after adaptation to analyze the effectiveness of the fine-tuning or adaptation process.

## 3 Tasks & Assessment

**Task 1:** (10 marks) Build the necessary **datasets**.

**Task 2:** (10 marks) Design and implement **the Generator network and the Discriminator Network.**

**Task 3:** (20 marks) Implement the training pipeline for the Generator and Discriminator networks, including the loss functions and optimization process. **Train** the complete image-to-image translation model iteratively.

**Task 4:** (10 marks) **Evaluate** the trained generator on a test set of images that were not used during training. Analyze the generator's performance and identify potential areas for improvement.

**Task 5:** (10 marks) Apply **model adaptation or fine-tuning techniques** to improve the performance of a pretrained YOLO model on the images generated by your translation model. You are required to explore and implement at least **two techniques** to adapt or fine-tune the pretrained model and enhance detection performance.

**Task 6:** (10 marks) **Evaluate the performance of the adapted or fine-tuned YOLO model** on the images generated by your translation model. **An essential part of this task is determining how to generate labels for the unlabeled data effectively.**

**Task 7:** (30 marks) Prepare **a comprehensive report** detailing your approach, methodology, results, and insights gained throughout the project, as well as **your contributions** for the aforementioned tasks.

***Bonus:*** *Extra credit will be awarded for exceptional and well-executed implementations that demonstrate high quality and reasonable additions beyond the tasks outlined above. This includes insightful approaches, additional functionalities, or any other noteworthy contributions that go beyond the initial project requirements. The final grade of this project will be calculated as min(100, normal_grade+bonus).*

## 4 Group formulation

This project requires group collaboration, with **each group limited to a maximum of 3 members**. Please organize yourselves into groups by 6 April 2025. In case any students are without a group, they will be randomly assigned to a group. Requests for group changes will only be considered if all members of the involved groups provide written agreements before or on 13 April 2025. After this date, no further group changes will be permitted. To find the groups you have formed or wish to join, please access the "Group" section on the BlackBoard platform. Kindly refrain from joining a group without obtaining consent from all other members involved.

## 5 Individual Report

***It is crucial for each member to submit an individual report.*** The report should be prepared from your own perspective and reflect your understanding of the project. It is mandatory for the individual report to demonstrate a distinct difference of at least 30% from the reports of other group members.

Additionally, your report should include **a dedicated subsection titled "My Contribution and Role Distinction."** In this subsection, you should provide a detailed account of your individual contributions to the project and highlight the unique aspects of your role in comparison to other group members. The content of all other sections should align with the claims made in this subsection.

## 6 Submission

Follow the steps below:

1. Create a main directory with your project's name, e.g., `G001`.
2. Within the main directory, create subdirectories for each component of the submission, such as `Source Code`, `Dataset`, and `Report`. If the dataset is too large, you can share a Google Drive link.
3. Place the relevant files in their respective directories. For example:
   - The source code files should be placed inside the `Source Code` directory.
   - The collected data should be placed inside the `Dataset` directory.
   - The report document should be placed inside the `Report` directory. The report should be named as `<your_ID>_<your_name>.pdf`, e.g., `12345678d_DawenChan.pdf`.
4. Ensure that all files are appropriately named and in the required file formats.
5. Double-check that the files and directories are organized correctly and follow the specified structure.
6. Compress the main directory and its components into a `ZIP` file.
7. Submit the ZIP file on the blackboard. **Each group only need to submit once.**

Example Directory Structure:

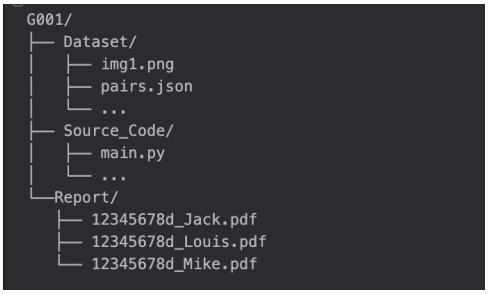


Figure 3: Example Directory Structure

**Warning:** If you are unable to complete the whole program, try to accomplish part of the tasks and make sure it can run successfully.

Any wrong file naming and submission will be given a ZERO mark in this project.

The deadline for this assignment is 2025, May 10th (Saturday), 23:59.

**Late submission penalty** 10% is deducted for each day that the work is late. The penalty will be applied up to a maximum number of three days after and including the submission deadline day. After three days the work will be marked at zero.

**Plagiarism is a serious offense. Copying code from web resources is prohibited as well. Any plagiarism case (for both the copier and the copiee) will be given a ZERO mark in this project.**