IIAI30018

Generative AI

HW2: GAN: Pix2Pix

Instructor: YuanFu Yang

yfyangd@nycu.edu.tw

Pix2Pix

- Homework due: 12/6
- Late submissions will incur a penalty of one point for each day overdue.
- The assignment allows a maximum extension of 3 days (it will not be accepted if submitted later than 3 days).
- Submit files: code and report (4 questions), and submit them in both **.ipynb** and **PDF** file formats respectively.
- This assignment can be carried out using Colab or completed on your PC.

Pix2Pix

Objective: Pix2Pix Model Implementation on CIFAR-10

- Gain practical experience in implementing and training a Pix2Pix model using PyTorch on a subset of the CIFAR-10 dataset. Evaluate the model with both quantitative metrics (PSNR, SSIM, LPIPS) and qualitative visualizations.

Assignment Details:

- Data Preparation:
  - Load CIFAR-10 and select 3 classes.
  - Use 100 images per class, totaling 300 training images (or more). Normalize and resize images appropriately.
- Model Implementation:
  - Build the Pix2Pix Generator and Discriminator networks.
  - Train the model for 100 epochs (or more) and plot the Generator and Discriminator loss curves.
- Evaluation:
  - Quantitatively assess the model using PSNR, SSIM, and LPIPS on 100 evaluation images.
  - Perform inference to generate 4 images per category and display them with appropriate titles.
- Submission:
  - Submit your Python script, loss curves, evaluation metrics, and generated image results.
  - Include a brief discussion on model performance and any challenges faced.

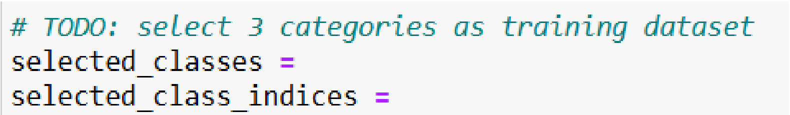
Pix2Pix

Learning Goals

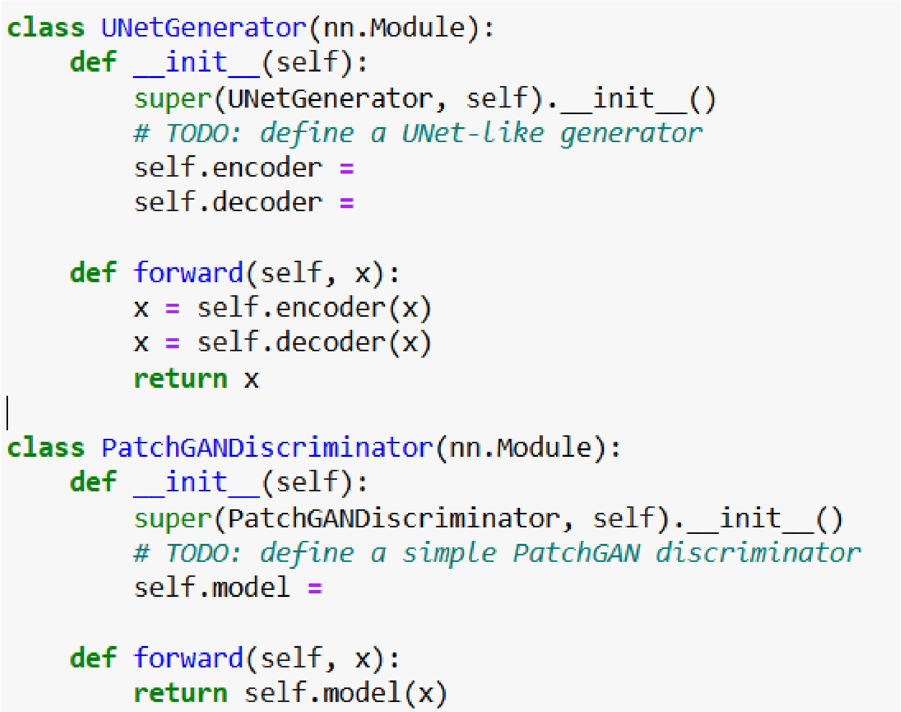
- Understand GAN architectures and training dynamics.
- Practice data preprocessing and PyTorch model implementation.
- Evaluate models using quantitative and qualitative methods.

Pix2Pix

- You need to complete all the TODOs in **pix2pix.ipynb**, follow the instructions provided in the TODOs for coding, and carry out model training and testing.



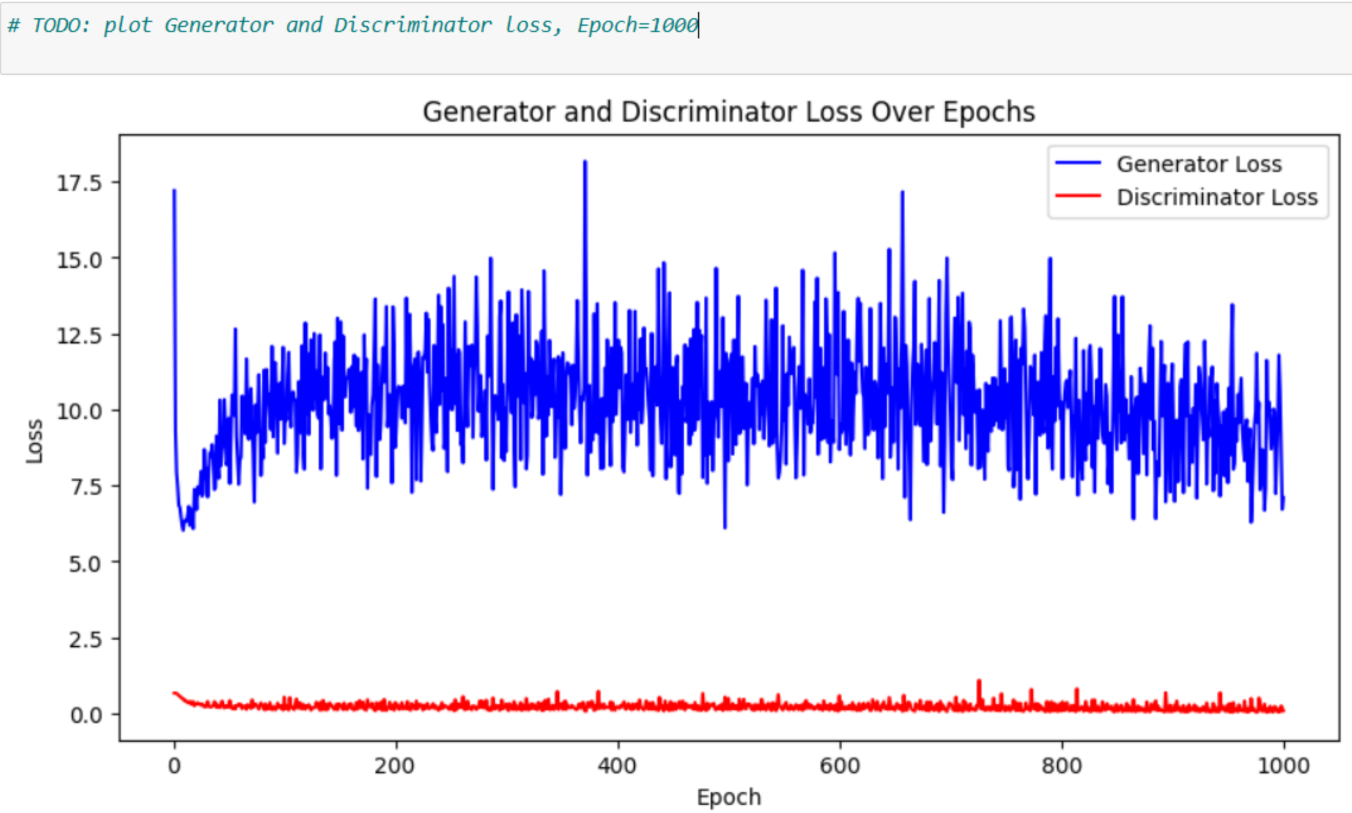
```
# TODO: select 3 categories as training dataset
selected_classes =
selected_class_indices =
```



```
class UNetGenerator(nn.Module):
    def __init__(self):
        super(UNetGenerator, self).__init__()
        # TODO: define a UNet-like generator
        self.encoder =
        self.decoder =

    def forward(self, x):
        x = self.encoder(x)
        x = self.decoder(x)
        return x

class PatchGANDiscriminator(nn.Module):
    def __init__(self):
        super(PatchGANDiscriminator, self).__init__()
        # TODO: define a simple PatchGAN discriminator
        self.model =

    def forward(self, x):
        return self.model(x)
```

Pix2Pix

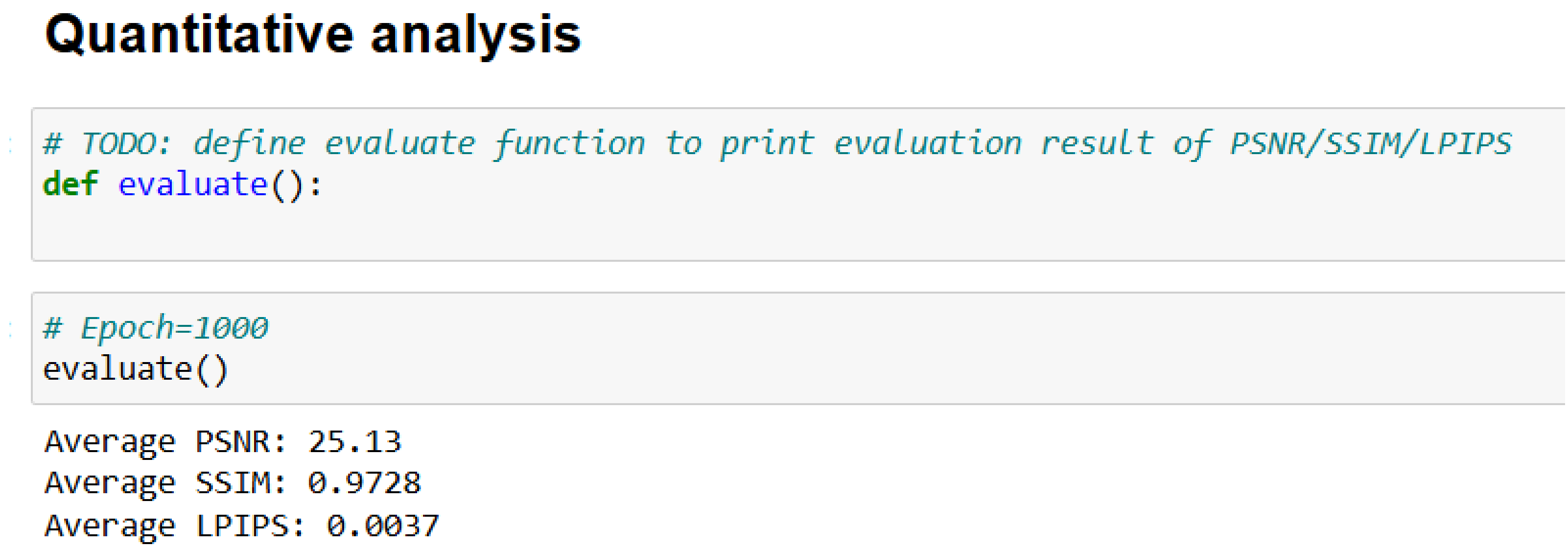
- HW2.1: Print the generator and discriminator training cruve (epoch vs loss).



```
# TODO: plot Generator and Discriminator loss, Epoch=1000
```

Pix2Pix

- HW2.2: Quantitative test result. Print PSNR SSIM, and LPIPS, and explain the representative meaning of PSNR SSIM, and LPIPS.



Quantitative analysis

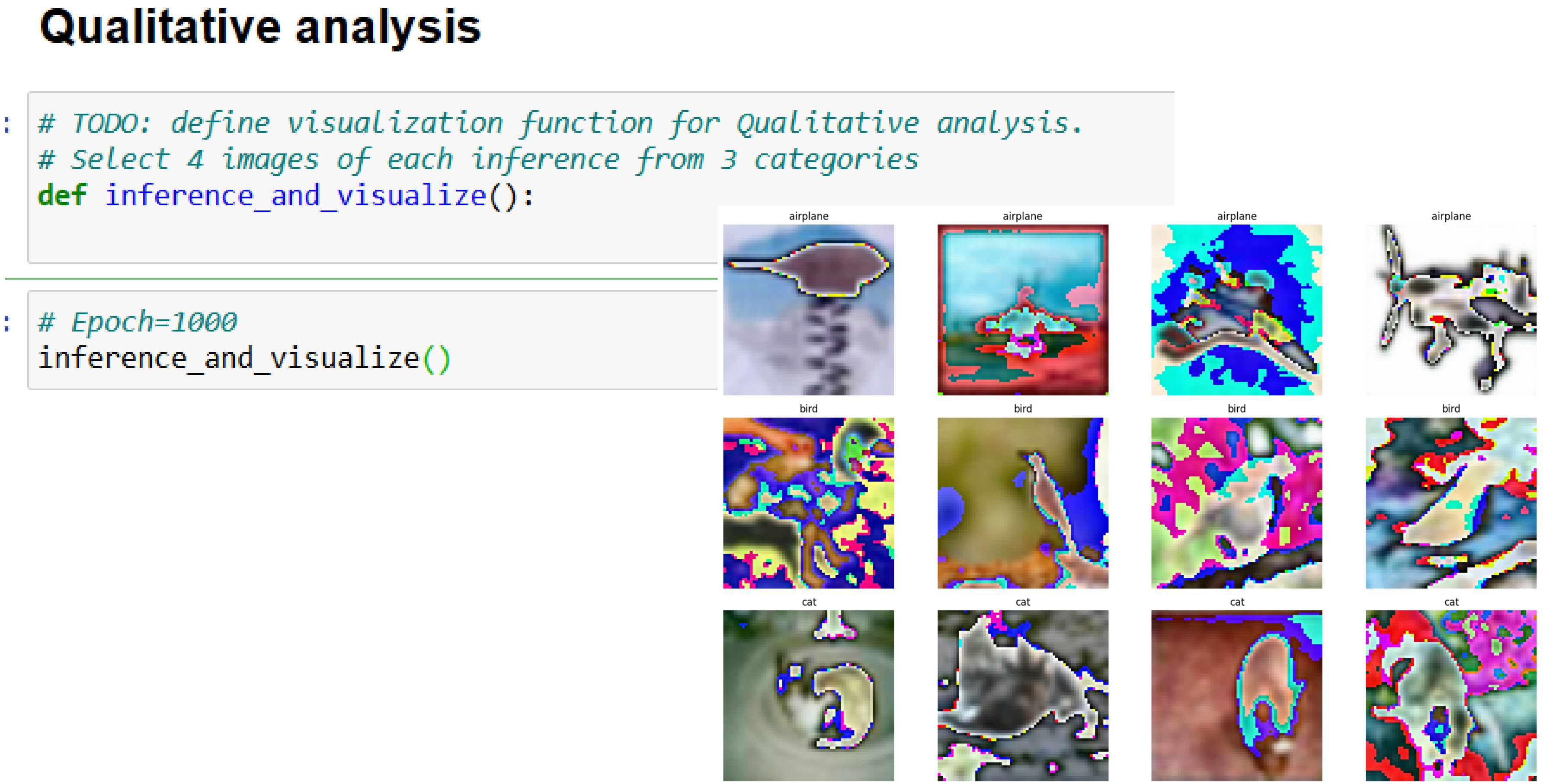
```
# TODO: define evaluate function to print evaluation result of PSNR/SSIM/LPIPS
def evaluate():
```

```
# Epoch=1000
evaluate()
```

```
Average PSNR: 25.13
Average SSIM: 0.9728
Average LPIPS: 0.0037
```

Pix2Pix

- HW2.3: Qualitative result. Print 4 images of each inference from 3 categories.



Qualitative analysis

```
# TODO: define visualization function for Qualitative analysis.
# Select 4 images of each inference from 3 categories
def inference_and_visualize():
```

```
# Epoch=1000
inference_and_visualize()
```

Pix2Pix

- HW2.4: Discuss the pros and cons of Pix2Pix compared with older methods.

IIAI30018

Generative AI

HW2: GAN: Pix2Pix

Q & A

Instructor: YuanFu Yang

yfyangd@nycu.edu.tw