11110COM 526000 Deep Learning

Homework 2: Variational Autoencoder

Announced: 2022 / 11 / 14 (Monday)
Deadline: 2022 / 12 / 9 (Friday) at 23:59 PM (eeclass)
TA:

許廷睿 (qwertyuiop5055@gmail.com)

Introduction:

A variational autoencoder is used to learn recoverable data in an unsupervised manner. In deep learning homework two, you need to implement both autoencoder and variational autoencoder trained on the eye dataset using PyTorch.

Rules:

1. **Built-in machine learning libraries (ex. Sklearn, PyTorch …) are allowed to use.**
2. Please properly comment your code to let us understand your train of thought.
3. Discussions are encouraged, but plagiarism is strictly prohibited and punishable!

Submission:

You should compress the files below into a ZIP file and submit it on **eeclass** before the due date. Please make sure we can reproduce the results in your report, otherwise, you would not get the scores.

- Zip file (hw2_studentID.zip)
  1. Code (*.py)
  2. Readme.txt (Explain how to run your code.)
  3. Model weight (*.pth, **Both AE and VAE**)
  4. Report (hw2report_studentID.pdf)
  5. Generated images & labels (gen_data/gen_label.npy, **Only VAE**)

Data:

There are two classes in the given dataset (data.npy), containing **female eyes (0)** and **male eyes (1).** The total number of data is 1476 and the shape of each image is (50,50,3).

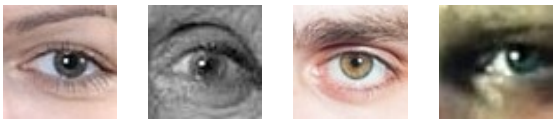

Fig. 1: Examples of the given eye dataset.

Implementation (PyTorch) [50%]

1. Design your own autoencoder (AE) model. **(10%)**
2. Design your own variational autoencoder (VAE) model. **(10%)**
3. Train the designed AE and VAE on all 1476 images and check the reconstructed results. (**Training processing and loss function 10%**)
4. Evaluate averaged PSNR score **(VAE)** on all 1476 images. **(10%)**

Reach 22 →10 points, 20→5 points, lower than 20→0 points

5. Evaluate averaged SSIM score **(VAE)** on all 1476 images. **(10%)**

Reach 0.64 →10 points, 0.6→5 points, lower than 0.6→0 points

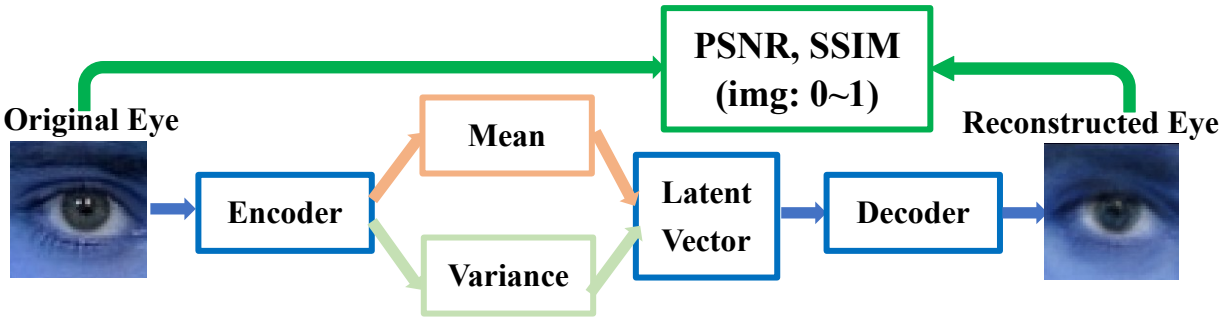

Fig. 2-a: Example structure for training VAE.

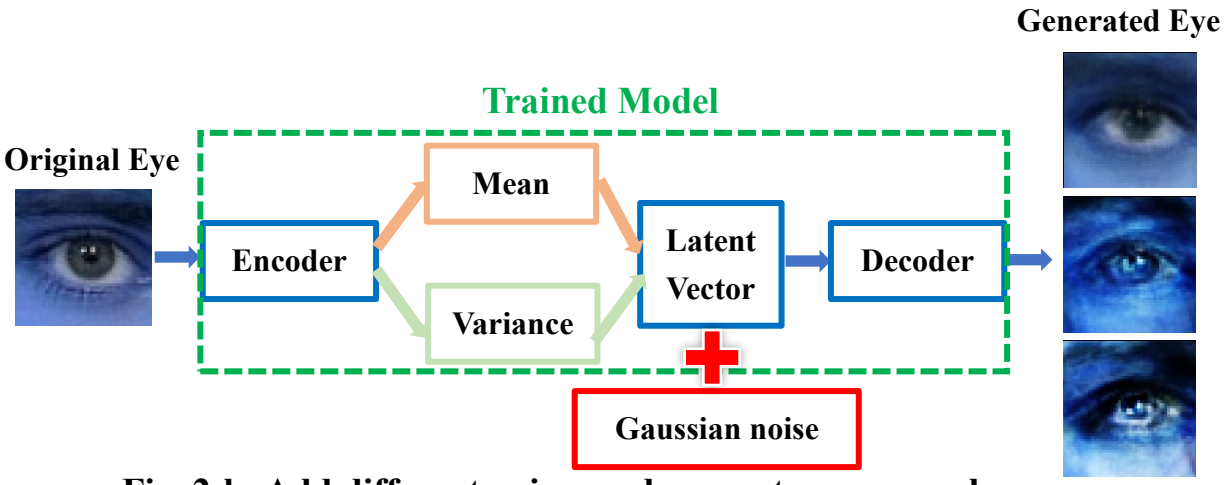

Fig. 2-b: Add different noises and generate new samples.

Report [50%]

The format is not limited, but the following matters must be discussed in your report: (The more details you mentioned, the more scores you will get.)

1. Show the model architecture, implementation detail, **PSNR,** and **SSIM** scores between original images and reconstructed images. Please describe the methods to achieve the final result in detail, e.g., epochs, batch size, learning rate, reconstruction loss, etc. At last, plot training loss curve of your **best model**. (Both **AE** and **VAE.**) **(10%)**

2. Generate another **100 new** samples from the original **20** samples by adding Gaussian noise into the latent code (Generate 5 samples from one sample) and store them into gen_data.npy and save the corresponding label to gen_label.npy. For the fairness, we specified the image id as **[1-5, 226-230, 841-845, 1471-1475]**. Eventually, please demonstrate the generated images of **[3, 227, 841, 1475]** with labels in the report. (Totally **20** images, and you only need to show the generated results of **VAE** in this part!) **(10%)**

3. In the training process, you would adjust hyperparameters (epochs, batch size, …) to achieve higher **PSNR** and **SSIM** results. How do you choose these hyperparameters? Please conduct experiments (in **Table** form) to validate that these hyper-parameters are suitable to achieve your best result. (Both **AE** and **VAE.**) **(10%)**

4. Please use the same gaussian noise to generate an image by **VAE** and **AE**. Discuss the difference between **VAE** and **AE** and generate at least **two** sets of images to explain. **(10%)**

5. In the reconstructed images, you might find that these images are not as sharp as the original images. Please explain the reasons for it. Second, what could we do to preserve the high-frequency information in the original images in your **AE** or **VAE** models? Please do some experiments to solve this problem. (Hint: Loss) **(10%)**