Cloud Computing and Big Data Analytics
Spring, 2021

HW#3: Defect Synthesis

Submission Deadline:

2021/6/3 23:55

Submit to E3

Hard deadline, No extensions

Goals

- Generating anomaly images from gaussian noise
- Training GANs from scratch
- Evaluating generated defects

Dataset

- CODEBRIM: COncrete DEfect BRidge IMage Dataset
- https://drive.google.com/file/d/1j6MOyle05eJEcXI-bRDLZOnuutn1F7rt/view?usp=sharing
- Download **cropped** dataset and resize to 32X32.
- DO NOT USE EXTERNAL DATASET

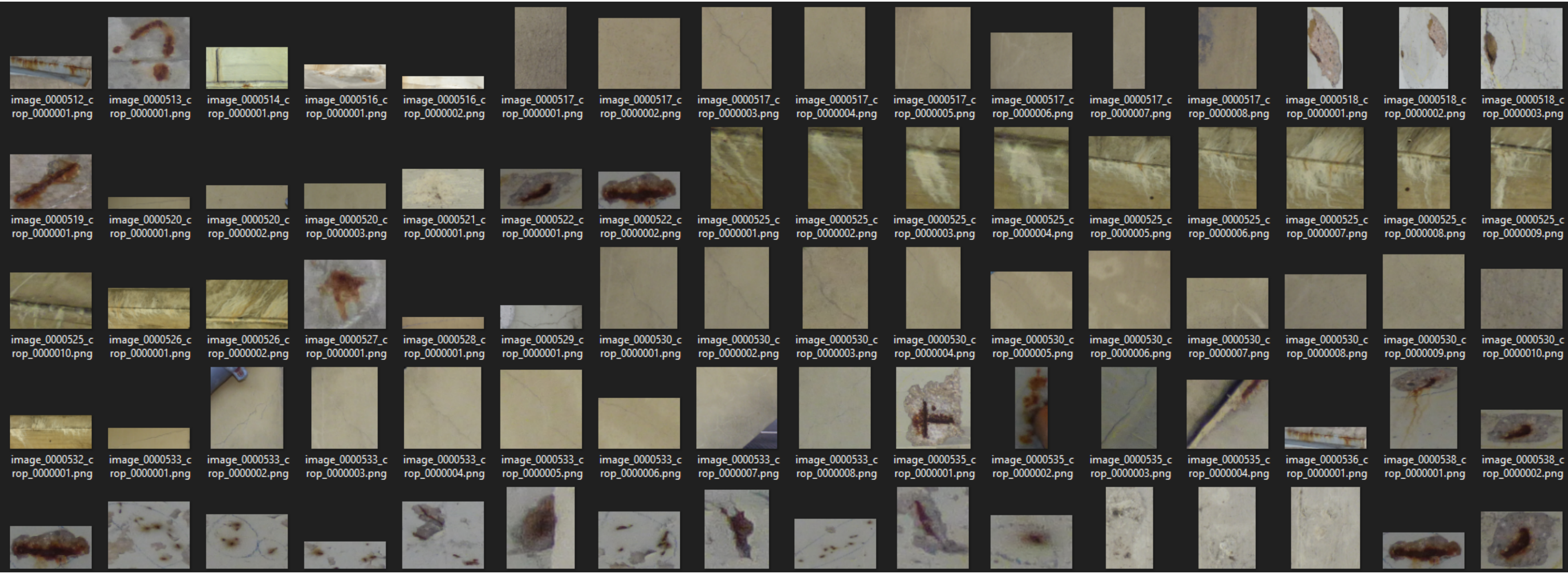


image_0000512_crop_0000001.png image_0000513_crop_0000001.png image_0000514_crop_0000001.png image_0000516_crop_0000001.png image_0000516_crop_0000002.png image_0000517_crop_0000001.png image_0000517_crop_0000002.png image_0000517_crop_0000003.png image_0000517_crop_0000004.png image_0000517_crop_0000005.png image_0000517_crop_0000006.png image_0000517_crop_0000007.png image_0000517_crop_0000008.png image_0000518_crop_0000001.png image_0000518_crop_0000002.png image_0000518_crop_0000003.png

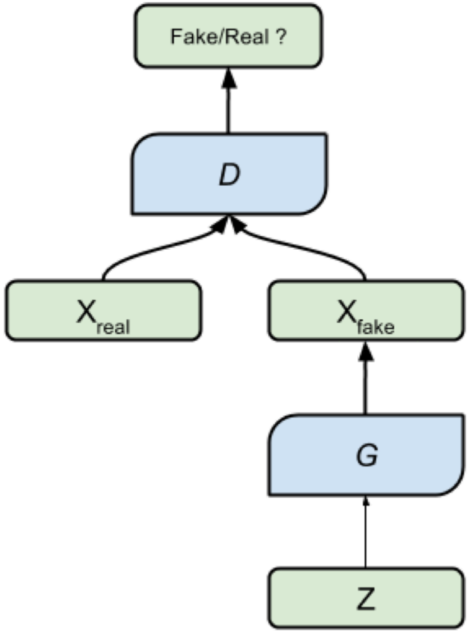
image_0000519_crop_0000001.png image_0000520_crop_0000001.png image_0000520_crop_0000002.png image_0000520_crop_0000003.png image_0000521_crop_0000001.png image_0000522_crop_0000001.png image_0000522_crop_0000002.png image_0000525_crop_0000001.png image_0000525_crop_0000002.png image_0000525_crop_0000003.png image_0000525_crop_0000004.png image_0000525_crop_0000005.png image_0000525_crop_0000006.png image_0000525_crop_0000007.png image_0000525_crop_0000008.png image_0000525_crop_0000009.png

image_0000525_crop_0000010.png image_0000526_crop_0000001.png image_0000526_crop_0000002.png image_0000527_crop_0000001.png image_0000528_crop_0000001.png image_0000529_crop_0000001.png image_0000530_crop_0000001.png image_0000530_crop_0000002.png image_0000530_crop_0000003.png image_0000530_crop_0000004.png image_0000530_crop_0000005.png image_0000530_crop_0000006.png image_0000530_crop_0000007.png image_0000530_crop_0000008.png image_0000530_crop_0000009.png image_0000530_crop_0000010.png

image_0000532_crop_0000001.png image_0000533_crop_0000001.png image_0000533_crop_0000002.png image_0000533_crop_0000003.png image_0000533_crop_0000004.png image_0000533_crop_0000005.png image_0000533_crop_0000006.png image_0000533_crop_0000007.png image_0000533_crop_0000008.png image_0000535_crop_0000001.png image_0000535_crop_0000002.png image_0000535_crop_0000003.png image_0000535_crop_0000004.png image_0000536_crop_0000001.png image_0000538_crop_0000001.png image_0000538_crop_0000002.png

Spec-1

- Unsupervised training
  - Only use the images in the dataset
  - Do **NOT** use any attributes
- Unconditional model
  - Only take Noises as the input
- Example: right figure



The **Discriminator** tried to distinguish between fake (generated) and real data

Input **data** either generated or from the real dataset

The **Generator** turns the input noise into fake data to try and fool the Discriminator

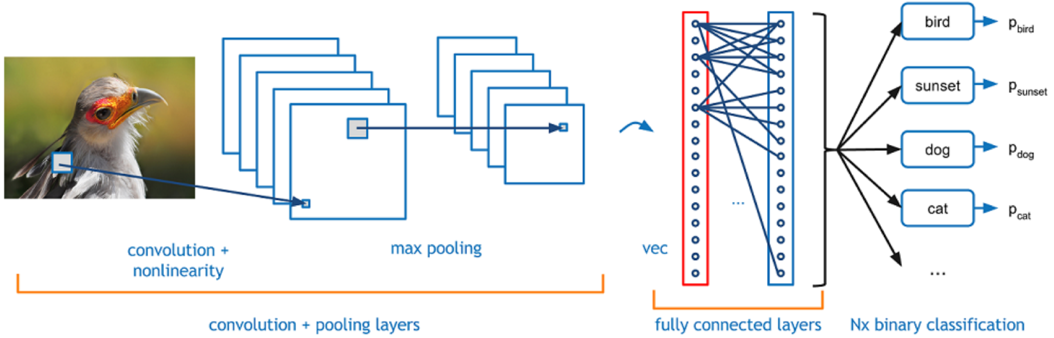
Input **Noise**

https://aws.amazon.com/cn/blogs/china/combining-deep-learning-networks-gan-and-siamese-to-generate-high-quality-life-like-images/

Spec-2

Evaluation: Inception Score

- A good GAN metric should contain the following properties
    - Responsive on some classes
    - Diversity
- Use Inception V3 to calculate the inception score

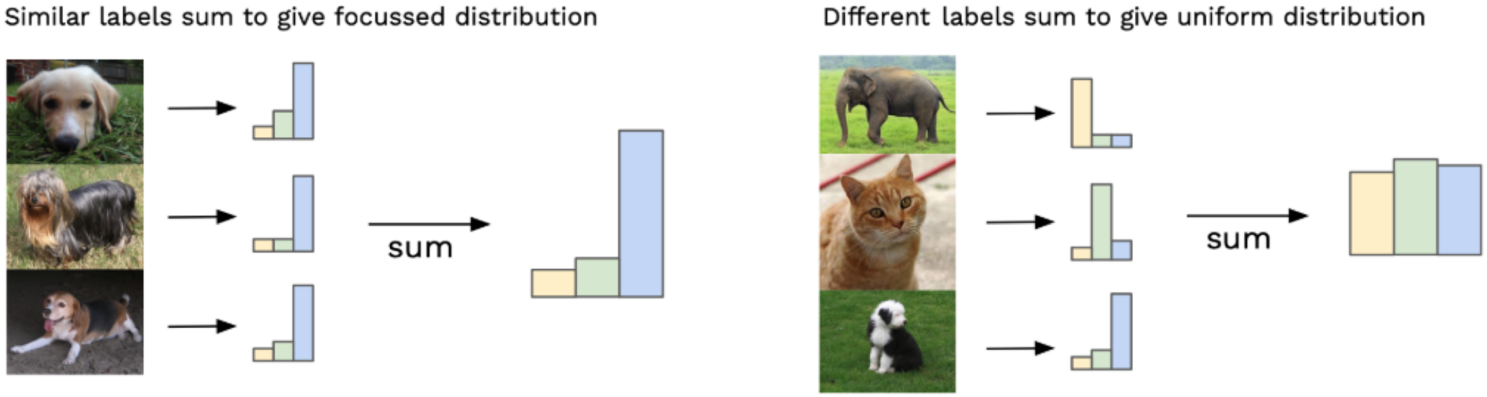


https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Spec-2

Evaluation: Inception Score

https://medium.com/octavian-ai/a-simple-explanation-of-the-inception-score-372dff6a8c7a


Similar labels sum to give focussed distribution
sum
Different labels sum to give uniform distribution
sum

$$\text{Inception Score}(G) = exp\left(\mathbb{E}_{x \sim G(z)}[KL(p(y|x) \| p(y))]\right)$$

More information https://medium.com/octavian-ai/a-simple-explanation-of-the-inception-score-372dff6a8c7a

Spec-2

Evaluation: Inception Score

- Codes in hw3.zip (Tensorflow or Pytorch)
- Ubuntu 16.04/18.04/19.04
  - Tensorflow
    - pip3 install -r tf_requirements.txt
    - Requires CUDA10.0 and cudnn > 7.4 (If the versions of CUDA or cudnn is not compatible, install the compatible tensorflow)
    - Windows probably works
    - Not support TensorFlow 2.0
  - Pytorch
    - pip3 install -r pytorch_requirements.txt
    - In windows, please install torch, torchvision, scipy, opencv

Spec-2

Evaluation: Inception Score

- Run
    - Tensorflow
        - `python tf_inception_score.py [image_directory]`
        - e.g., `python tf_inception_score.py imgs`
    - Pytorch
        - `python pytorch_inception_score.py [image_directory]`
        - e.g., `python pytorch_inception_score.py imgs`
    - `[image_directory]` file path, should be *.png
    - `splits=10` is used to partition the data into 10 splits for calculating the Inception score and taking the average

Spec-2

Evaluation: Inception Score

- The results by Tensorflow and Pytorch is slightly different. We use Tensorflow as the final score
- Reference github
  - Pytorch implementation: https://github.com/sbarratt/inception-score-pytorch
  - Tensorflow implementation: https://github.com/tsc2017/Inception-Score

Spec-3

- No restriction on the framework of deep learning. But please use the framework that supports python3.
- You can refer to the model online, but write your own codes.
- Existing GAN modules are forbidden, e.g., DCGAN in TF-GAN, DCGANGenerator in pytorch-gan.
- Do NOT generate images for your friends (0 point if we find any cases).

Grading policy

- Grading Scheme

Inception Score	**Points**
0 ~ 2	0
2 ~ 2.5	70
2.5 ~ 2.8	80
2.8~3	90

Grading policy

- Grading Scheme
  - If the number of students with scores > 3 <= (0.15* the number of students who make submissions)

  => All the students with scores >3 gets 100.

  - If the number of students with scores > 3 > (0.15* the number of students who make submissions)

  => Linear interpolation between 90～100.

Submission files-1

- Upload two files {studentID}_img.zip and {studentID}_src.zip to new E3

  e.g., 0850726_img.zip, 0850726_src.zip

- {studentID}_img.zip (~100MB)：

  When you use “right click and unzip here” or command “unzip {studentID}_img.zip”, we should find a folder named {studentID}_img.

  Inside the folder, there should have 50000 defects images with size 32x32.

  filename should be 1.png, 2.png, … , 50000.png,.

  System will judge the inception score based on the images

  Note that you need 50000 images to get the score. Less than 50000 will get 0 point since the score can not be compared with other students.

Submission files-2

- {studentID}_src.zip：

  When you use “right click and unzip here” or command “unzip {studentID}_src.zip”, we should find a folder named {studentID}_src, which contains your codes and readme files.

- For MacOS users, please exclude the folder of “ __MACOSX”.

Run environment

- OS: Ubuntu 18.04.3 LTS
- Python version: python3.6
- CPU: Intel(R) Core(TM) i7-8700K 3.70GHz 6 Cores 12 Threads
- RAM: 32G
- GPU: 2080 Ti (11G)

Hints

- Use GPU! The training time will last for hours.
- Start the training ASAP.
- When calculating the inception score, adjusting the batchsize can accelerate the process and avoid Memeoy Leak.
- If the hardware is not good, you do not have to record the IS during training. You can judge the pictures by your eyes or use a small number of images for estimating the Inception score to reduce the evaluation time.

Hints

- PyTorch DCGAN Tutorial：
  https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

If you have any question about HW#3, please email to Yi-Lun Wu or post on Facebook group.

w86763777@gmail.com