

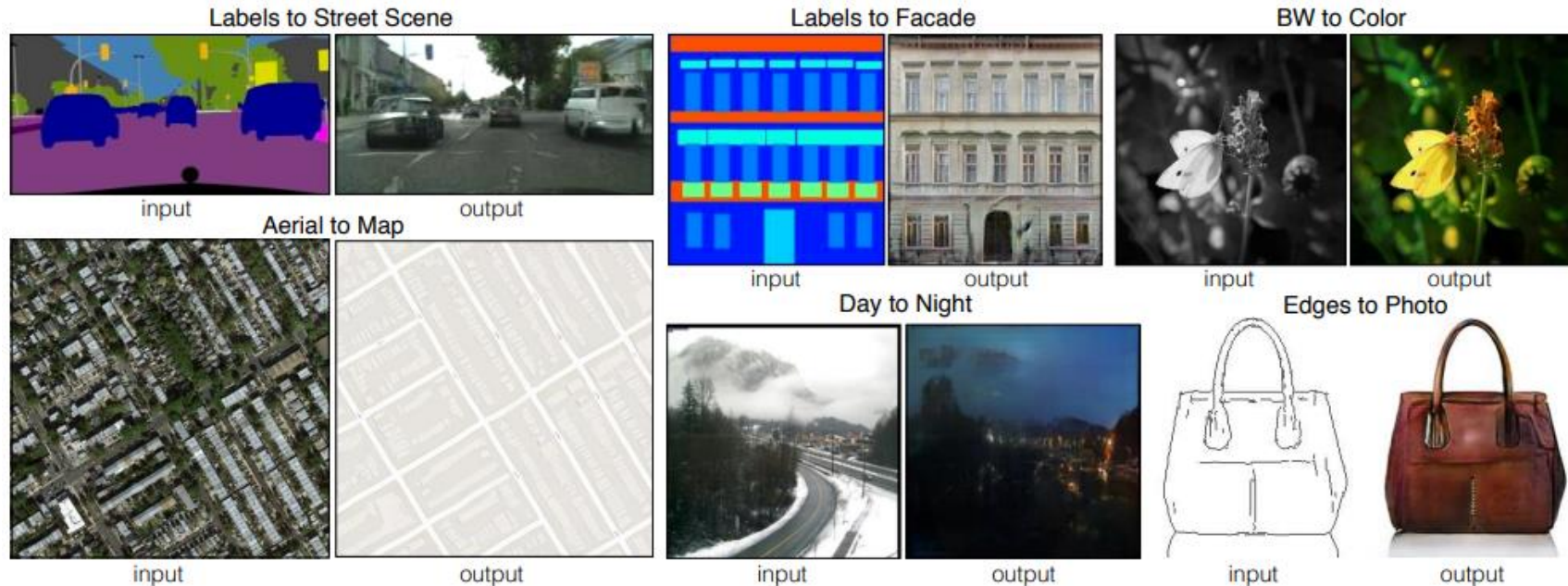
# Pix2Pix

The pix2pix architecture is similar to GAN, but the purpose is not to generate simulated images, but to use supervised learning to output the image style learned by the original image. For example: conversion of grayscale to color, edge map conversion to photo, map conversion to satellite map and other applications.

Version 20230103

# Applications

- It can be applied to various applications such as conversion of grayscale to color, edge map conversion to photo, map conversion to satellite map and other applications.



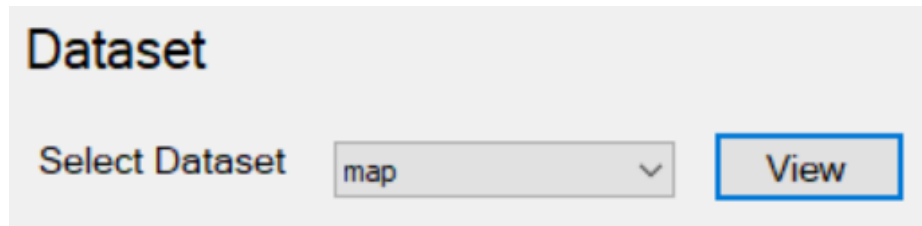
# Process

- Data preparation
  - 1. combine\_A\_and\_B
- Train
  - 2. visdom server
  - 3. visdom server browser
  - 4. train
- Inference, Inference API
  - 5. inference folder
  - 6. inference\_folder\_image\_pairs
  - 7. inference api
  - 8. inference api browser



# Select Dataset

- Select Dataset: Select the data set for AI learning. If you want to train your own images, please click View to view the data folder. Please copy a default map folder and change it to your own data name. At this time, please do not delete any sub-folders and file.

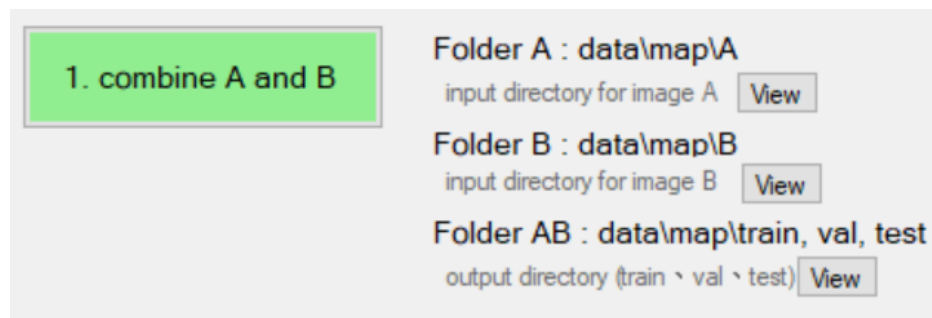


**Dataset**

Select Dataset

# 1. combine\_A\_and\_B (1/2)

- Prepare the images that you want to convert. The conversion needs the same attributes and the same pair of image file names must be the same (the attached file name must be .jpg).
- Click Folder A view and Folder B view.
  - A is the image after conversion, B is the image before conversion.
- Click the 1. combine A and B button to merge the corresponding images into one image, which can be viewed by clicking Folder AB view.
- Note :
  - The image size is recommended to be the same, square, if not the same, you can zoom or crop.
  - The image file must be .jpg.
  - Images need to be matched in pairs.



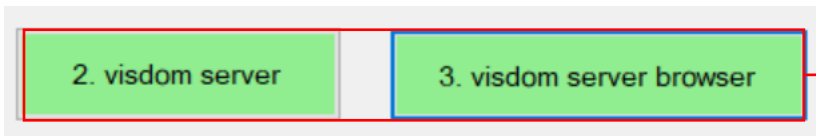
# 1. combine\_A\_and\_B (2/2)

- For example: map to satellite image.
  - data/A/train: training map image. data/B/train: training satellite image.
    - => data/train: combine the training images of A and B.
  - data/A/val: map image for verification. data/B/val: satellite imagery for verification.
    - => data/val: A single image combining A and B for verification.
  - data/A/test: map image for test. data/B/test: satellite imagery for test.
    - => data/test: a single image combining A and B for verification.
- The same pair of image files must have the same name.
  - 1.png map can be found in the train folder in A.
  - the corresponding 1.png satellite image should be found in the train folder in B.
  - the merged image 1.png can also be found in the train folder.

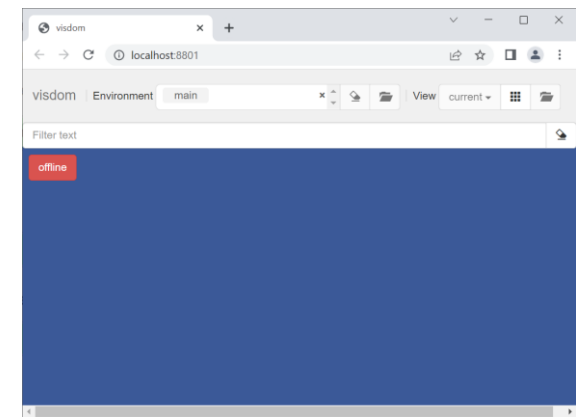
## 2. visdom server

## 3. visdom server browser

- 2. visdom server and 3. visdom server browser in order before starting training, as shown in the figure below, please do not press x to close it. Please keep it open during training.
- After starting the training, the training loss curve and other information will be post to the visdom server and displayed on the visdom server browser.



```
C:\Users\ah\Desktop\App4\App\Pix2Pix-6\bin\1_visdom_server\1_visdom_server.exe
8801 is unused
server.py:40: DeprecationWarning: zmq.eventloop.ioloop is deprecated in pyzmq 17. pyzmq
now works with default tornado and asyncio eventloops.
checking for scripts.
It's Alive!
INFO:root:Application Started
You can navigate to http://localhost:8801
INFO:tornado.access:200 GET / (:::1) 1073.23ms
INFO:tornado.access:200 GET /static/css/bootstrap.min.css?v=e8c5daf01eae68ed7e1e277a6e
544c7ad108a01a8771b531d6d9f2210769b7da88e4e002c7b0be3b72154ebf7cbf01a795c8342ce2dad368
bd6351e956195f8b (:::1) 246.00ms
INFO:tornado.access:200 GET /static/css/react-resizable-styles.css?v=ca5ba57167c45b93cc
f0e85aee67736a9267ca450452524278aa2538934caa2d425b3ad9de53560815983729f1251b9edc98b79
6cd639d6e4633e5d688c59392 (:::1) 1.00ms
INFO:tornado.access:200 GET /static/js/bootstrap.min.js?v=8b3b66a1bb2f9e329f02d4cd47479
065630184c8ae942ee61a9ff9e1cc34c28c0e0cb854458977815cf3704a8697fa8a5d096d2761f032b74b
70d51da3e37f45 (:::1) 2.00ms
INFO:tornado.access:200 GET /static/css/react-grid-layout-styles.css?v=088afb33aca5476
eb6b9bac95c72316c59771cea8c53b9f4c6429bb3b0ae728e6518b45e83855af2e21319e734959da931f04
21421e62acc80b6e0bcb701f1 (:::1) 3.00ms
INFO:tornado.access:200 GET /static/js/react.react.min.js?v=52fa608d7e6bb1982d7ba2364e
0dd36321915733177ce2794330878e79ba97bc15d01a09f23183ff948745e92da50d6d8c8daF406832720
f3912bc85fe232f3 (:::1) 4.00ms
INFO:tornado.access:200 GET /static/js/jquery.min.js?v=53a2b560b205516721bb06e72632d4
f11c7e2dd2ecf737eaaab179cb88be7c87e9d803ce7765706bc71cbc1993c34587cd1237de5a279aca199
116e9057865 (:::1) 7.00ms
INFO:tornado.access:200 GET /static/css/style.css?v=f043dc30e08d15702b92312329e9e8101e
381a0b45b522c7dd26449e3124533398fc8cae6f7d8095f3f701bde44f9561526c8d653e82e8241550476
Saa06034 (:::1) 6.00ms
```



## 4. train (1/2)

- Click 4. train to start training. During the training process, you can use the 3. visdom server browser opened in the previous step to watch the loss curve and the training effect.

**Train**

[4. train \(GPU\)](#)

[4. train \(CPU\)](#)

Continue training : load the latest model

Dataroot : data\map  
path to images (should have subfolders train, val, test) [View](#)

Checkpoints dir : data\map\model  
models are saved here [View](#)

GPU ID :   
0, 1, 2, ... for Nvidia GPU

Batch size :

N epochs :

Save epoch freq :



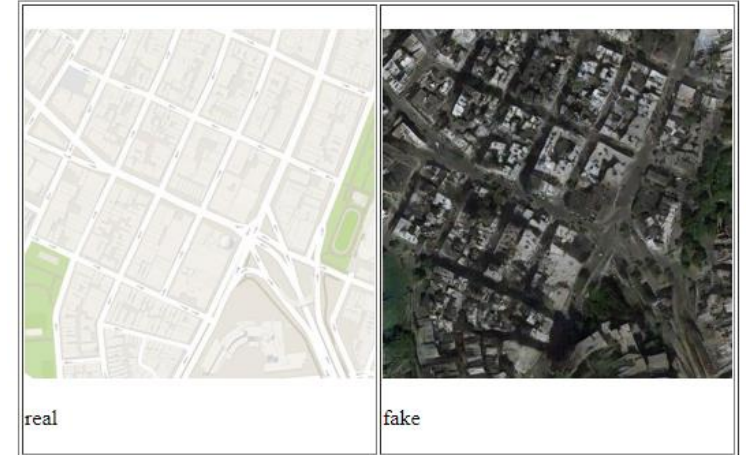
## 4. train (2/2)

- Parameter :
  - Continue training : if you want to continue training, please tick it, and the training will automatically load the last model and continue training (latest\_net\_G.pth \ latest\_net\_D.pth). Otherwise, don't tick it.
  - Dataroot : the file path of the training image. Press view to view the train and val folders, which contain training images.
  - Checkpoints dir : The folder of the training model, click view to see the folder.
  - GPU ID : If your device supports NVIDIA GPU accelerated computing, please set the GPU ID, if you use the 0th GPU, set it to 0, and use the 0th and 1st GPU to set it to 0, 1. If NVIDIA GPU accelerated computing is not supported, set to -1.
  - Batch size : The number of training examples in one forward/backward pass.
  - N epochs : Set the number of epochs for training. The total number of epochs for training will add an additional 100 epochs, using a learning rate that decays linearly to 0.
  - Save epoch freq : The number of frequency epochs to store the model.

# 5. inference folder

- Press 5. inference folder, select the model, and start inferring all images in the B/test folder.
- After the inference is completed, jump out of the browser to view the inference results.

1072



1073



# 6. inference\_folder\_image\_pairs

- Infer the combined image (1. combine A and B's output image), output three images, before, after, and the target image.
- After the inference is completed, jump out of the browser to view the inference results.

1072



1073



# Reference

- Please refer to the readme.txt in the APP folder.
- LEADERG AppsForAI : <https://www.leaderg.com/appsforai-windows>
- Copyright © LEADERG INC. All rights reserved.