Droplet Pix2Pix Intro

Project Target

- Our aim is to generate droplet images by AI given specific physical conditions.
- We use an open-source model Pix2Pix (based on GAN), which is originally used in style transformation/prediction from picture to picture.
- In this project, we try to migrate this method: The main idea is to encode various physical conditions into 2d heatmap, and make tuples with image output to build our own dataset. And then re-train the original model.

Steps

Data Preprocessing

Original Experiment Dataset

- 1 single dataset = 1 group of physical conditions + 2000 pictures
- Total we have around 220 datasets

Extract Impacting Frames

- This is done in previous work: https://github.com/openhe-hub/droplet-img-process.git
- For single dataset, 2000 pictures can extract around 40 impacting pictures (2%)

Physical Conditions

- We have 6 physical conditions:
  i. surface type
  ii. liquid type
  iii. diameter
  iv. height
  v. fall point type & offset distance
  vi. time
- For example: `S1-W-20G-20cm-C-1_C001H001S0001`

Normalization Rule

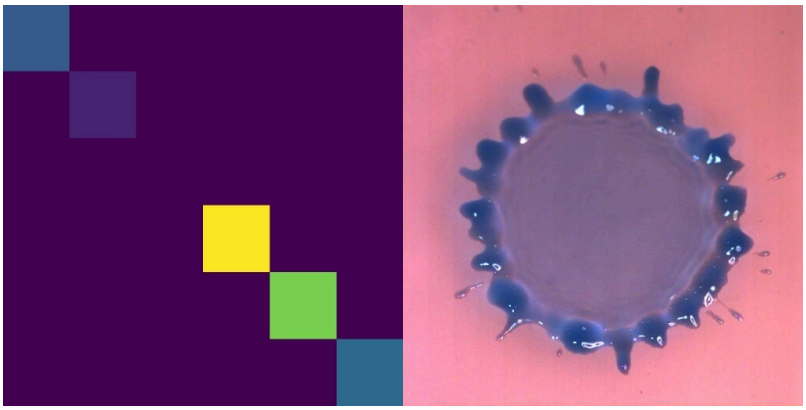
1. Enum type: surface type, liquid type, fall point type
   - We have n enums for each type, given i, we get $\frac{i}{n+1}$
   - For example, we have 6 surface types, given $S3$, output is 0.429
2. Continuous type: diameter, height, time
   - We have $[min, max]$ range, input $value$, we get $\frac{value-min}{max-min}$
   - For example, we accept height among $[10cm, 50cm]$, given $h = 20cm$, output is 0.25

Transform 1d Physical Condition Vector => 2d Heatmap

1. Suppose n physical conditions after normalization is $v \in \mathbb{R}^n$, we say $M \in \mathbb{R}^n \times \mathbb{R}^n, s.t. M[i][i] = v[i]$ and $M[i][j] = 0, i \neq j$, `n=6` here
2. We use `VIRIDIS` color map, transform M into M'
3. Suppose size of original picture `N` is `(p, q)`, re-scale M' into the same shape
4. Finally we get image tuple `(M, N)`, and we have around `6000` tuples for our dataset.

Building Own Dataset

- Combine Heatmap (Feature) and Original Image (Label)
  - We concat `(M, N)` horizontally, so the result size will be `(2*p, q)`
  - Example:

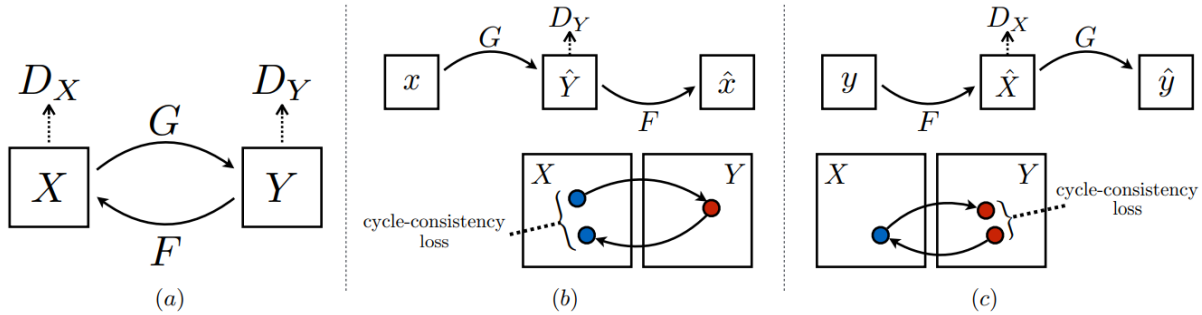


- Split Train/Valid/Test Set
  - Train = 80%
  - Valid = 10%
  - Test = 10%

Training model

- Network Intro

  Ref to https://arxiv.org/pdf/1703.10593



- Train Settings
  i. epoch = 200
  ii. learing rate: 0-100 epoch, const 0.0002; 100-200 epoch, linearly decay to 0
  iii. optimizer = Adam
  iv. checkpoint_freq = 5
  v. Environment: Pytorch 2.0, CUDA 11.7, Python 3.8 (time cost = 2h with RTX3080)

Prediction Result on Testset

> Share content: droplet_pix2pix_v3mini.zip
> Link: https://pan.sjtu.edu.cn/web/share/ee03a6c9ec5d8f2b8554ca9ecbca090a, Extraction code: wa3j

- Open `test_latest/index.html` on browser to view it

Next To Do

- [ ] Maybe we can modify the network structure to elevate performance
- [ ] Maybe we may want to add more experiment data, since 6 physical conditions are not enough