Applied Machine Learning/Deep Learning on Predicting Fluid Dynamics

Zhi Li

Email: zli008@citymail.cuny.edu; lz.owen1991@gmail.com

Advisor: Michael Grossberg & Hansong Tang

Introduction

Machine learning is bridging the computational paradigms and building on hundreds of years of modeling history and progress.

Traditional computational paradigms are based on physical laws. To make a prediction, the typical process includes obtaining a certain amount of physical-related data, selecting physical parameters, and conducting iteration until convergence. Although with high accuracy, it requires iteration in each time prediction. It is computationally expensive. However, machine learning as a data-driven approach is much faster in the prediction process, since parameters are fixed after training and no more iteration required. In the case that obtaining the fluid data costs a large amount of time and human efforts, and that the model wants to be more interpretable, this hybrid modeling approach should be taken into consideration.

To this end, this project is to conduct hybrid modeling on fluid flows, which combines physics-based simulation and data-driven machine learning modeling. Specifically, we will use inpainting techniques to predict completed Computational Fluid Dynamics (CFD) maps from a certain small part of the CFD map.

State-of-the-Art Inpainting Techniques

Current image inpainting research mainly includes tasks such as repairing rectangular block mask, irregular mask, target removal, denoising, removing watermark, removing text, removing scratches, and coloring of old photos.

Traditional image inpainting

Traditional image inpainting, mainly divided into diffusion-based methods and patch-based methods.

Diffusion-based image inpainting mainly spreads the pixel information around the damaged hole in the image gradually and synthesizes new textures to fill the hole. Reconstruction is usually restricted by the information around the hole, it is difficult to reasonably learn from distant information, and lack high-level semantic understanding of the image, making it difficult to restore meaningful texture structure in the missing area. And as the diffusion distance of pixel information around the hole increases, the larger the hole is, the less effective pixel information will be obtained in the center. So the traditional diffusion-based method is more suitable for structure texture background inpainting and removal of small objects in the image, but the effectiveness of large-hole restoration for natural scene objects with complex textures in real life is limited.

The patch-based image inpainting assumes that the damaged area and visible area of the image have similar content. It searches for the best matching similar patch in the visible area of the image, and then copies the information to fill the missing area at pixel level. The traditional method usually requires an enormous amount of computing power to calculate the similarity score between patches.

Image inpainting based on generative network

With the emergence of two generative models such as VAE and GAN , the image inpainting method based on generative model can fully learn the high and low frequency feature information of the damaged image visible area and learn the consistency of image structure and texture at the high-level semantic level by adding different constraints to generate novel and reasonable features to complement the damaged area. Therefore, in recent years, various deep learning image inpainting models based on generative network have been the hot direction for many researchers to make further improvement.

Another novelty of this research is that RGB channels are replaced by velocities of x, y directions and pressure. Since the physical components are calculated based on partial differential equation(PDE), and that CNN is based on discrete convolution, it is more likely to provide a foundation for propose a more interpretable hybrid modelling approach in the future.

Project Framework

The below framework is proposed for the project:

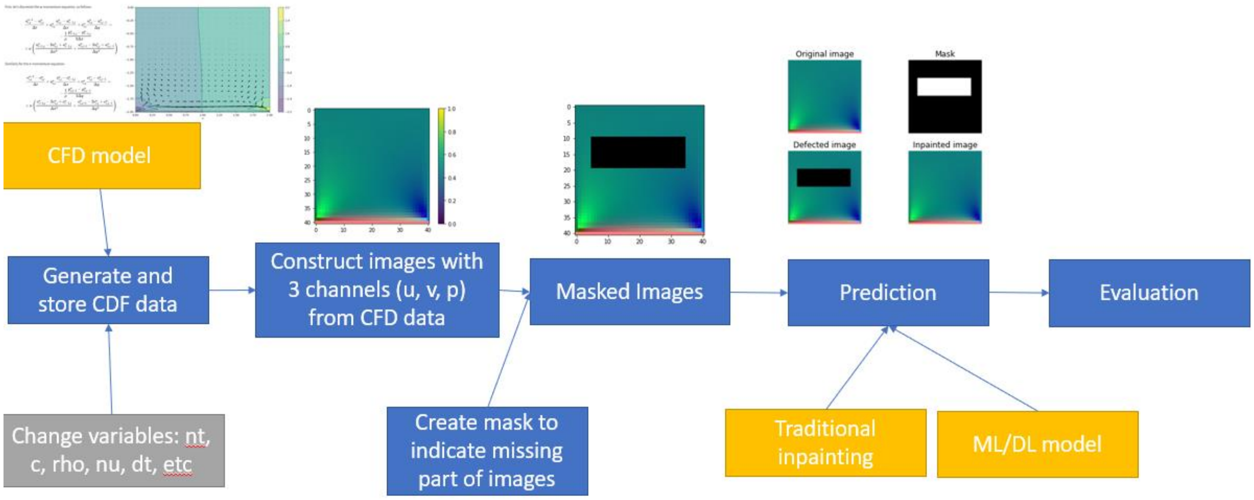
1, we generate CFD data, changing variables, and store the data into .npy format. Each data is 41X41X3 arrays. The data has 3600 data points in total.

2, Next, we constructed data into images, and consider u v p are the 3 channels, which denotes velocities in x,y directions and pressure.

3. Then, we created masks and put them on the images. We used masked img as the input, the missing parts as labels.

4. We made a prediction, using different inpainting techniques, respectively

5. Finally, we compare and evaluate prediction results



Data Generation

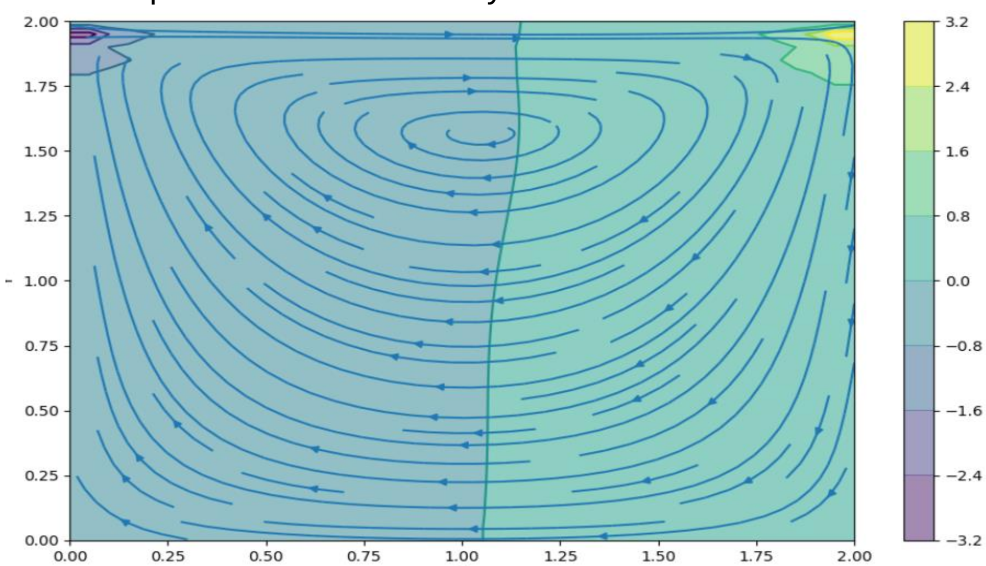
Building Computational Fluid Dynamics (CFD) model to generate data

We will use [CFDpython] to generate cavity flow data (u,v,w).

Here is the system of differential equations: two equations for the velocity components u,v and one equation for pressure w:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$
$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$
$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = -\rho\left(\frac{\partial u}{\partial x}\frac{\partial u}{\partial x} + 2\frac{\partial u}{\partial y}\frac{\partial v}{\partial x} + \frac{\partial v}{\partial y}\frac{\partial v}{\partial y}\right)$$

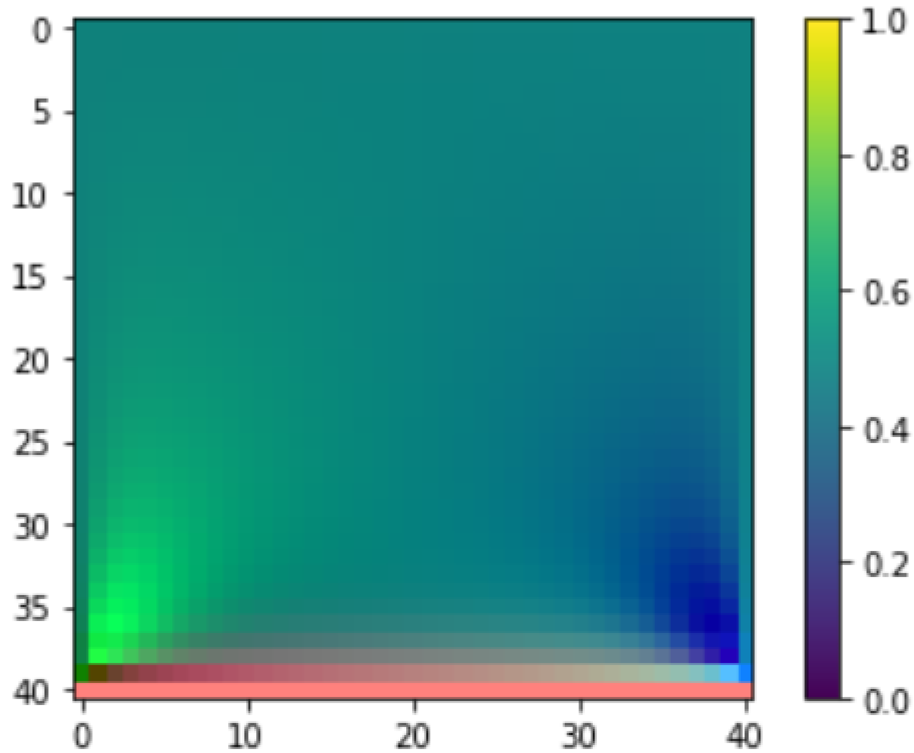
Each data point is a 41x41x3 array and could be visualized as below:



The above figure is easy to understand by human. However, in order to deal with 3 dimensional arrays as images, we take components u,v,p as 3 channels instead of RGB.

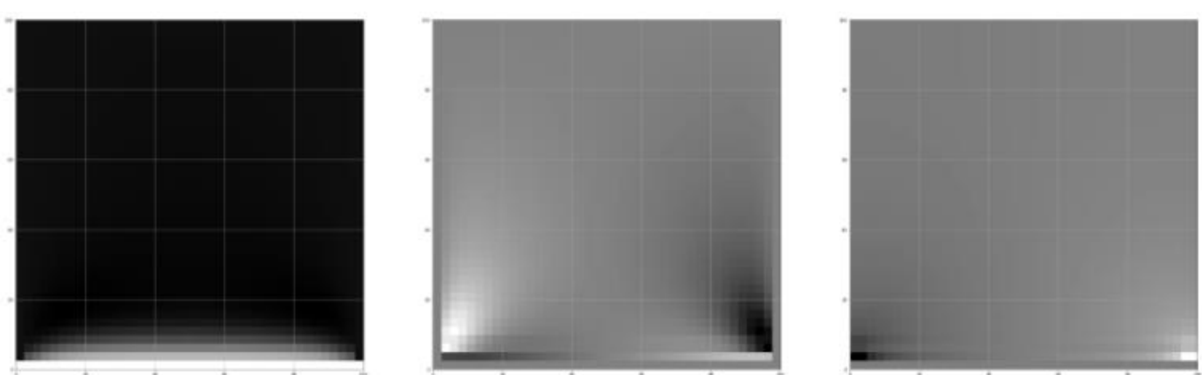
Normalization and Scale

In order to display as images, each channel, namely, u, v, p, should be normalized and scaled to the range of 0 to 1. The CFD image example is shown below.



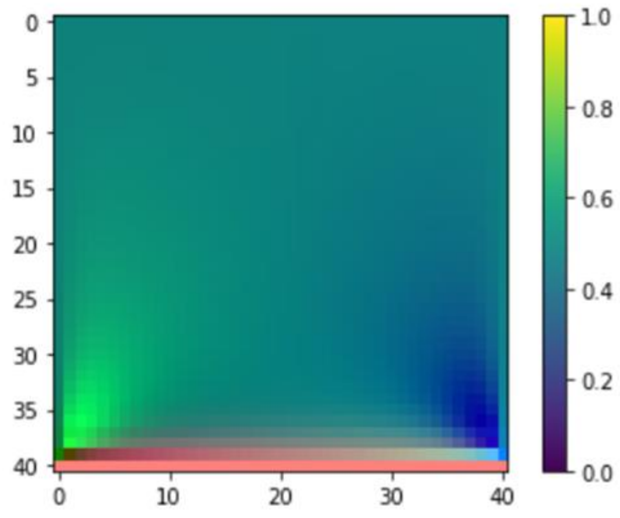
Sanity Check

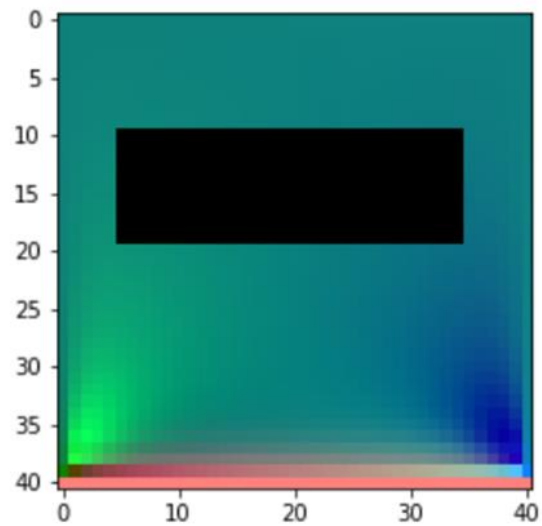
The below is the grayscale images for u,v,p components separately.



Mask Preparation

After generating CFD data, we stored the data as 41X41X3 array and put masks on them. As shown below, it is the example of the original image and masked image.





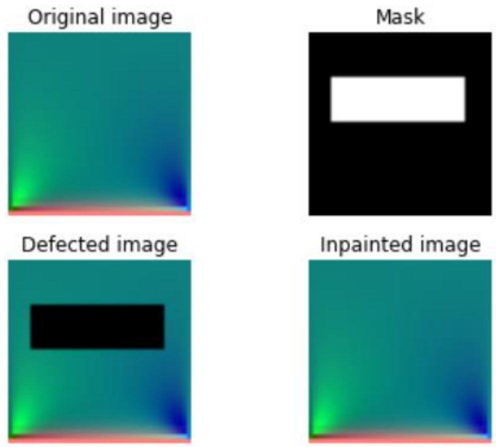
CFD image inpainting

Traditional approach

Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting” by Bertalmio, Marcelo, Andrea L. Bertozzi, and Guillermo Sapiro in 2001.

This algorithm is based on fluid dynamics and utilizes partial differential equations. Basic principle is heurisitic. It first travels along the edges from known regions to unknown regions (because edges are meant to be continuous). It continues isophotes (lines joining points with same intensity, just like contours joins points with same elevation) while matching gradient vectors at the boundary of the inpainting region. For this, some methods from fluid dynamics are used. Once they are obtained, color is filled to reduce minimum variance in that area. This algorithm is enabled by using the flag, cv2.INPAINT_NS

Biharmonic equation is also used to do inpainting. Below is the result of traditional inpainting techniques.



Machine Learning approach

In addition to traditional methods, varying machine learning and deep learning algorithms are employed to do the inpainting. The input is the images with masks, and the output is the missing part which is generated by the algorithms.

Random Forest

For random forest, we take max depth as 2 and random state as 0. And the MSEs and determination coefficients of training and test are shown as below:

```
MSE train: 0.0000361989, test: 0.0000382208
R^2 train: 0.9454166944, test: 0.9472256837
```

General Deep Learning

For deep learning model, 1 input flatten layer and 1 dense layer, together with 2 hidden layers, are added in the model.

```
Model: "sequential_52"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_33 (Flatten)         (None, 3936)              0         
_________________________________________________________________
dense_150 (Dense)            (None, 100)               393700    
_________________________________________________________________
dense_151 (Dense)            (None, 100)               10100     
_________________________________________________________________
dense_152 (Dense)            (None, 1230)              124230    
=================================================================
Total params: 528,030
Trainable params: 528,030
Non-trainable params: 0
```

CNN model

For CNN model, it includes 5 Conv2D layers.

```
Model: "sequential_67"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_116 (Conv2D)          (None, 41, 41, 3)         84
_________________________________________________________________
conv2d_117 (Conv2D)          (None, 41, 41, 64)        1792
_________________________________________________________________
conv2d_118 (Conv2D)          (None, 41, 41, 128)       73856
_________________________________________________________________
conv2d_119 (Conv2D)          (None, 41, 41, 64)        73792
_________________________________________________________________
conv2d_120 (Conv2D)          (None, 41, 41, 3)         1731
=================================================================
Total params: 151,255
Trainable params: 151,255
Non-trainable params: 0
```

As shown below, based on current results, generally all models have good performance, no significant overfitting or underfitting.

	Random Forest	General DL	CNN
Train MSE	3.6e-5	3.86e-6	2.22e-5
Test MSE	3.8e-5	1.87e-6	2.14e-5

Conclusion and future work

In this capstone, ML algorithm, including RF and DL, had a good performance on inpainting CFD images.
It can be explained as followings:
1) Compared to natural images, CFD data is generated based on physics laws (PDE). So ML algorithm can learn image more easily.
2) Image and mask sizes may be not large enough, which also reduced the complexity to make a prediction.
Future work:
1) Explore more interpretable ML algorithms for CFD models.
2) Increase Image and mask sizes

Reference

- H.S. Tang, S. Casey Jones, Fotis Sotiropoulos, An overset-grid method for 3D unsteady incompressible flows,Journal of Computational Physics, Volume 191, Issue 2, 2003, 567-600, https://doi.org/10.1016/S0021-9991(03)00331-0.
- H.S. Tang, L. Li, M. Grossberg, Y.J. Liu, Y.M. Jia, S.S. Li, W.B. Dong, An exploratory study on machine learning to couple numerical solutions of partial differential equations, Communications in Nonlinear Science and Numerical Simulation,Volume 97,2021,105729, https://doi.org/10.1016/j.cnsns.2021.105729.

- Qu, K., Yao, W., Tang, H. S., Agrawal, A., Shields, G., Chien, S. I., Gurung, S., Imam, Y. and Chiodi, I., (2021), Extreme storm surges and waves and vulnerability of coastal bridges in New York City metropolitan region: an assessment based on Hurricane Sandy, Natural Hazards: Journal of the International Society for the Prevention and Mitigation of Natural Hazards, 105, issue 3, p. 2697-2734.
- Navid Ataei, Jamie E. Padgett, Fragility surrogate models for coastal bridges in hurricane prone zones, Engineering Structures,Volume 103, 2015, 203-213, https://doi.org/10.1016/j.engstruct.2015.07.002.
- Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, Thomas S. Huang, Generative image inpainting with contextual attention, in: Proc. CVPR, 2018. 1, 2,3, 4, 8, 5505–5514.
- Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, Thomas S. Huang, Free-form image inpainting with gated convolution, in: Proc. ICCV, 2019. 1, 3, 6, 7, 4471–4480.
- Hongyu Liu, Bin Jiang, Yi Xiao, Chao Yang, Coherent semantic attention for image inpainting, in: Proc. ICCV, 2019. 3, 7, 4170–4179.
- Min-cheol Sagong, Yong-goo Shin, Seung-wook Kim, Seung Park, Sung-jea Ko, Pepsi: Fast image inpainting with parallel decoding network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 1,11360–11368.
- Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, Huchuan Lu, Highresolution image inpainting with iterative confidence feedback and guided upsampling, 2020.
- Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, Zhan Xu, Contextual residual aggregation for ultra high-resolution image inpainting, 2020. 10.1109/CVPR42600.2020.00753.
- Wenchao Du, Hu Chen, Hongyu Yang, Learning invariant representation for unsupervised image restoration, 2020, 14471–14480. 10.1109/CVPR42600.2020.01449.
- Ziyu Wan, Bo Zhang, Dongdong Chen, Pan Zhang, Dong Chen, Jing Liao, Fang Wen, Bringing old photos back to life, 2020, 2744–2754. 10.1109/CVPR42600.2020.00282.
- D. Ulyanov, A. Vedaldi, V. Lempitsky, Deep Image Prior, Int. J. Comput. Vision 128 (2020), https://doi.org/10.1007/s11263-020-01303-4.
- C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, J. Verdera, Filling-in by joint interpolation of vector fields and gray levels, IEEE Trans. Image Process. 10 (8) (2001) 1200–1211.
- Anat Levin, Assaf Zomet, Yair Weiss, Learning how to inpaint from global image statistics, in: null, 305. IEEE, 2003