

卒業論文

머신러닝 기법을 활용한 도심유동 고속 예측

FAST PREDICTION  
of  
URBAN WIND FLOW  
using MACHINE LEARNING TECHNIQUE

2023年 12月

仁荷大學校 工科大学  
機械工學科

정현욱  
방성진

卒業論文

머신러닝 기법을 활용한 도심유동 고속 예측

FAST PREDICTION  
of  
URBAN WIND FLOW  
using MACHINE LEARNING TECHNIQUE

2023年 12月

指導教授 李 尙 承

이 論文을 卒業論文으로 提出함

仁荷大學校 工科大學

機械工學科

정현욱

방성진

이 論文을 다음 學生의 卒業論文으로 認定함

정현욱

방성진

2023年 12月

指導教授 李 尙 承 (印)

## 요약문

기후 변화로 인한 변동성 증가로 도심 환경의 여러 현상들에 대한 실시간 예측 값을 계산하는 것이 어려워졌다. 특히 도심 유동의 공기 역학적 해석의 경우 층류와 난류의 조합으로 이뤄져 있어, 계산하는 과정에 있어 고려해야할 조건들이 증가하여, 계산 과정 자체가 어렵고 복잡하게 된다. 일반적으로 공기 역학적 해석을 위해 전산유체역학 (Computational Fluid Dynamics, CFD) 프로그램들을 활용하게 된다. 앞서 언급한 조건들로 인한 계산값의 증가는 주로 LES(Large Eddy Simulation) 혹은 RANS(Reynolds Average Navier-Stokes Simulation)과 같은 난류 해석 모델들을 활용하여 경량화 과정을 거치지만, 도심과 같은 큰 영역을 해석함에 있어, 실시간 해석을 하기에는 어려움이 있다. 따라서 본 연구는 머신러닝 기법들을 활용하여 이러한 문제들을 극복하고자 했다. 실험은 도심 내 7개의 모형 건축물을 대상으로 10개의 각기 다른 initial 유동 ( $u_{-vel}$ )에 따른 유동장 내 압력( $p$ ), 속도( $u, v, w$ ) 값의 변화가 기록된 10개의 데이터 셋을 바탕으로 진행됐다. 선형회귀를 활용하여 2차원 평면 내에서의 유동 예측과 pseudo inverse matrix 방식을 활용한 선형회귀를 활용하여 3차원 평면 내에서 유동 예측을 수행하였다. 추가로 CNN(Convolutional neural network)의 적용 가능성을 검증하였다.

# ABSTRACT

The turmoil caused by climate change is leading to hardships in calculating and predicting various real-time phenomena in urban environments. Airflow analysis in such cases is much more challenging because the combination of laminar and turbulent flow generates numerous conditions that needs to be put under consideration. The state-of-art method in analyzing the effect of air in wind engineering is the use of computational fluid dynamics (CFD). In most cases, LES (Large Eddy Simulation) and RANS (Reynolds Average Navier-Stokes Simulation) are often used as turbulence models in wind flow analysis to reduce the amount of calculation that needs to be computed. Albeit the effort, it is still expensive when the calculation is applied in real-time analysis of larger flow-fields such as wind flow in urban areas. Thus, the paper is designed to overcome the given problems by applying machine-learning techniques in predicting urban wind flows. Flow over 7 simplified building models were analyzed. The flow was analyzed with 10 different initial u-direction velocities. The flow field data-sets include pressure (p), velocity (u,v,w) data in x,y,z coordinates. Initial machine-learning models include linear regression for 2D analysis, and pseudo inverse matrix-method linear regression for 3D analysis. Additional tests were done to validate the application of convolutional neural network (CNN).

## 목 차

요약문 .....	i
Abstract .....	ii
목차.....	iii
NOMENCLATURES.....	v
List of Figures.....	vi
제1장 서론 .....	1
1.1 연구동향 .....	1
1.2 연구목적 .....	2
1.3 연구내용 .....	2
제2장 이론적 배경 .....	4
2.1 도심유동 .....	4
2.1.1 벤츄리 효과 .....	4
2.2 기계학습 이론 .....	5
2.2.1 선형회귀 .....	5
2.2.2 Pseudo-Inverse Matrix .....	6
2.2.3 Convolutional Neural Network .....	7
2.2.4 Mean Absolute Error .....	8
제3장 실험 .....	10
3.1 개요 .....	10
3.2 2D 모델.....	11
3.2.1 데이터 전처리 .....	11

3.3 3D 모델 .....	13
3.3.1 데이터 전처리 .....	13
제4장 실험결과 및 고찰 .....	14
4.1 2D 모델 결과 .....	14
4.1.1 Linear Regression 모델 .....	14
4.1.2 2D CNN 모델 .....	17
4.2 3D 모델 결과 .....	20
4.3 2D와 3D 모델 결과의 고찰 .....	25
4.3.1 차트 해석 .....	25
4.3.2 2D Mean Absolute Error .....	26
제5장 결론 .....	28
참고문헌 .....	35

# NOMENCLATURES

$y_i$	Actual observed values
$\hat{y}_i$	Predicted values
$n$	Total number of data points
$x$	Input features
$X$	Matrix of input features
$b$	Bias of linear regression model
$w$	Weight assigned to the features of the linear regression model
$\tilde{x}$	Pseudo-Inverse solution

.....  
.....



## List of Figures

- Fig. 1-1 Initial Flow Field
- Fig. 1-2 System Roadmap
- Fig. 2 Flow Field at Initial U-Vel: 4.25 m/s
- Fig. 3-1 Comparison of GT & Prediction Values (LR) at Initial U-vel :3.75 m/s (P,U)
- Fig. 3-2 Comparison of GT & Prediction Values (LR) at Initial U-vel :3.75 m/s (V,W).
- Fig. 4-1 Hierarchical Structure diagram of the 2D CNN
- Fig. 4-2 Hierarchical Structure diagram of the 3D CNN
- Fig. 5-1 Comparison of GT & Prediction Values (CNN) at Initial U-vel :3.75 m/s (P) before Applying New Model
- Fig. 5-2 Comparison of GT & Prediction Values (CNN) at Initial U-vel :3.75 m/s (U).
- Fig. 5-3 Comparison of GT & Prediction Values (CNN) at Initial U-vel :3.75 m/s (P).
- Fig. 6-1 Comparison of Pressure Data of GT & Prediction Values (PINV) at Initial U-vel :5 m/s (GT).
- Fig. 6-2 Comparison of Pressure Data of GT & Prediction Values (PINV) at Initial U-vel :5 m/s (Pred)
- Fig. 7-1 Comparison of U-Velocity Data of GT & Prediction Values (PINV) at Initial U-vel :5 m/s (GT)
- Fig. 7-2 Comparison of U-Velocity Data of GT & Prediction Values (PINV) at Initial U-vel :5 m/s (Pred)
- Fig. 8-1 Comparison of V-Velocity Data of GT & Prediction

Values (PINV) at Initial U-vel :5 m/s (GT)

Fig. 8-2 Comparison of V-Velocity Data of GT & Prediction

Values (PINV) at Initial U-vel :5 m/s (Pred)

Fig. 9-1 Comparison of W-Velocity Data of GT & Prediction

Values (PINV) at Initial U-vel :5 m/s (GT)

Fig. 9-2 Comparison of W-Velocity Data of GT & Prediction

Values (PINV) at Initial U-vel :5 m/s (Pred)

Fig. 10-1 Comparison of MAE of Each Model (Chart)

Fig. 10-2 Comparison of MAE of Each Model (Graph)

.....  
.....

# 제1장 서론

## 1.1 연구동향

머신러닝 (Machine Learning, 이하 ML) 분야는 인공 두뇌학 (cybernetics) 와 전산학 (Computer Science) 와 연관된 분야로, 1900년대 부터 Frank Rosenblatt의 perceptron과 같이 꾸준히 연구돼 오던 분야이다. (Early History of Machine Learning, Alexander L Fradkov) 특히 여러 데이터의 경향성을 파악하고 예측하는 여러 회귀 모델들의 근간이 된 선형 회귀 (linear regression) 모델 같은 경우, Sir Francis Galton 에 의해 1894년에 처음 제안되기도 하였다. (Galton, Pearson, and the peas: A brief history of linear regression for statistics instructors) GPU의 등장과 같은 전산 기술의 발전은, 과거 제안만 됐던 여러 ML 이론들을 검증하고 활용할 수 있는 기회가 됐고, 여러 분야에서 이를 활용하는 계기가 됐다. 특히 유체역학 분야에서는 난류 해석이 요구하는 다량의 데이터를 전처리하거나, CFD 모델 경량화에 이용하는데 있어 ML 이 적용되는 연구들이 지속적으로 진행 중이다. D.Xiao, C.E. Heaney, L.Mottet와 7명의 연구진은 LES (Large Eddy Simulation) 으로 생성된 도심 유동 CFD 데이터를 ML 전처리 기법 중 하나인 SVD (Singular Value Decomposition)를 활용하여 POD (Proper Orthogonal Decomposition) 모델을 생성하고 또 다른 ML 기법인 GPR (Gaussian Process Regression)에 적용하여 NIROM (Non intrusive reduced order model) 을 완성시켰다. 이와 같이 도심 속 환경에서 바람의 공기역학적 해석을 위해 회귀 분석, 딥러닝 모델, 모드 분해 등의 ML 알고리즘을 통해 정확도를 유지하며 계산 비용을 효과적으로 줄

이기 위한 노력들이 계속되고 있다.

## 1.2 연구목적

본 연구는 도심 유동의 공기 역학적 해석과 예측에 있어, ML 알고리즘의 효용성을 검증하고자 하였다. 특히 도심 유동이 비선형적임에도 불구하고, 선형 회귀 모델 적용 가능 유무에 집중하였다. 이는 고속 예측함에 있어, 타 ML 모델들과 비교하였을 때, 선형 회귀 방식이 더 간단하고 명료하기 때문이다. 또한 예측 데이터 시각화를 통해 실측 값과 가시적인 비교를 하고자 하였다. 마지막으로, 한정된 데이터 셋 안에서 CNN 방식과 같은 추가적인 ML 알고리즘 적용을 검증하고자 하였다.

## 1.3 연구내용

본 연구는 CFD 를 통해 얻은 데이터를 기반으로, ML 알고리즘을 적용시켜 예측값을 얻고, 이를 CFD 예측값과 비교하는 방식으로 진행됐다. 실험에 사용된 유동장은  $x: [-40.95, 40.95]$ ,  $y: [-59.8, 59.8]$ ,  $z: [0, 29.25]$  로 설정됐고,  $u$  방향의 초기 풍속을 변경하였을 때 내부에 모델링 된 7개의 건물이 유동에 미치는 영향을 계산하였다. 우선적으로 2차원 유동장 해석을 위해, 높이가 가장 낮은 건물 기준으로 slicing을 하여 평면 데이터를 추출하여 선형회귀를 통한 weight 함수를 얻어 예측 값을 계산하였고, 추후 3차원 유동장 해석을 위해 pseudo-inverse matrix 방

식을 적용하여 3차원 데이터에 선형회귀를 적용시켰다. 시각화를 위해 2차원 데이터의 경우 matplotlib 을 활용하였으며, 3차원 데이터의 경우 Paraview를 활용하였다.

.....  
.....

## 제2장 이론적 배경

### 2.1 도심 유동

#### 2.1.1 벤츄리 효과

벤츄리 효과는 일반적으로 파이프 내부에 유동이 발생한다고 했을 때 단면적이 좁은 영역을 지나면서 유속은 증가하는 대신 유체의 압력이 감소하는 효과이다. 파이프 내부의 단면적이 좁은 영역을 통과하게 되어 넓은 통로로 이동하게 되면 이 과정에서 유체의 속도는 감소하게 되고 압력은 다시 증가하게 된다. 이처럼 벤츄리 효과는 유체의 속도와 압력 간의 상호작용으로 설명할 수 있다. 이는 연속 방정식과 베르누이 정리로 설명할 수 있다. 연속 방정식은 유체의 질량이 변하지 않는다는 원리를 기반으로 한다. 연속 방정식을 통해 유도한 단면적과 유속의 상관관계를 통해 베르누이 방정식과 정리하면 유속과 압력 간의 관계를 나타낼 수 있다.

$$\rho_1 A_1 V_1 = \rho_2 A_2 V_2 \cdots (1)$$

$$A_1 V_1 = A_2 V_2 \cdots (2)$$

$$P_1 + \frac{1}{2} \rho V_1^2 + \rho g h_1 = P_2 + \frac{1}{2} \rho V_2^2 + \rho g h_2 \cdots (3)$$

저층, 고층의 건물들로 인해 유동이 발생하는 도심 속에서, 두 건물 사이를 통과하는 바람이 가속되는 것이 바로 이 벤츄리 효과로 인해서이다. (A Review on the study of urban wind at the pedestrian level around buildings.)

## 2.2 기계학습 이론

### 2.2.1 선형회귀

선형 회귀 방식은 통계학 분야의 고전적 방식 중 하나이다. 선형회귀 방식은 간단하지만 매우 강력하다.  $X$  와  $y$  간의 관계를 매우 간단한 수준으로 회귀 시키고 가중치 행렬을 반환하기 때문에 다른 메소드들에 비해 계산과정이 간단하다. 또 이를 해석하는 과정도 단순한 행렬식을 이용하기 때문에, 결과값을 해석하고 인사이트를 정립하는데에 큰 도움을 준다. 벡터 표현식을 사용하면,  $d$  개의 변수가 있다 했을 때, 모델은 아래와 같이 표현된다.

$$\hat{y} = w_1x_1 + w_2x_2 + \dots + w_dx_d + b \cdots (4)$$

이때,

$$X = [x_1, x_2, x_3, \dots, x_d] \cdots (5)$$

$$w = [w_1, w_2, w_3, \dots, w_d] \cdots (6)$$

$$\hat{y} = wX + b \cdots (7)$$

로, 예측 값을 Input 데이터들과 가중치 (weight) 벡터의 곱에 bias  $b$ 를 합한 형태로 표현된다.

선형회귀의 단점으로는 단순한 수식을 사용하고 있고, 계산 layer가 하나로 이루어져 있는 회귀 모델이기 때문에 입력값과 결과값 간의 비선형성을 계산하는 것은 불가능하다. 현재 직면한 문제(도심 유동 예측)에서는 사용한 데이터가 간단한 1차원 형태의 array 이므로 어느정도 자유로울 수 있다. 하지만 입력행렬이 복잡해지면 복잡해질수록 모델이 Input data와 target data 간의 관계를 잘 설명할 수 없으므로 낮은 결정계수를 갖게되고 정확도 자체가 매우 떨어

질 가능성이 있다. 두 번째로는 모든 input data에 따른 차원축소에 한계가 있다는 단점이 있다. 선형회귀 문제는 기존의 Navier-Stokes equations 을 기반으로 한 CFD 방식에 비해 수식은 간단하지만 계산 횟수는 동일하여 결국 모든 데이터셋을 확인해야 한다. 결국 비교적 큰 차원의 행렬이 들어오게 되면 시간복잡도에 의해 기존 CFD 방식과 같은 한계점에 직면할 수 있다. 이 2가지 문제를 해결하기 위해 딥러닝 메소드 중 하나인 CNN(Convolution Neural Network) 방식을 이용한 비선형 회귀를 고려해 볼 수 있다.

## 2.2.2 Pseudo-Inverse Matrix

Pseudo-inverse matrix는 정사각행렬이 아닌 행렬에 대한 역행렬의 일종으로, 정사각행렬이 아닌 행렬 계산에 효과적으로 활용될 수 있다. 이를 이해하기 위해 우선적으로 이해해야하는 것이 특잇값 분해 (Singular Value Decomposition, SVD) 이다. SVD는 어떤 행렬이든 이를 세 개의 구성 행렬로 분해한다.

$$A = U\Sigma V^T \dots (8)$$

이때, U와 V는 직교 행렬이고,  $\Sigma$ 는 특잇값을 포함한 대각 행렬이다.

Pseudo-inverse matrix는 기본적으로 특잇값 분해의 구조를 활용한다. 이를 수식적으로 표현하면,

$$A^+ = V\Sigma^+ U^T \dots (9)$$

여기서,  $\Sigma^+$ 는 특잇값의 역수로 구성된 대각 행렬이다.

일반적으로, pseudo-inverse matrix 는  $Ax \approx b$  식의 문제에서 정확한 해 (exact solution)을 구하기 어려울 때 주로 사용된다. 특히, 선형회귀를



적용하고자 할 때 변수들 간의 높은 상관 관계를 갖을 경우 효과적으로 활용될 수 있다.

### 2.2.3 CNN (Convolutional Neural Network)

CNN(Convolution Neural Network) 방식의 경우 일반적으로는 Image detection, Image segmentation 및 영상처리 등의 컴퓨터 비전 분야에서 많이 쓰이는 기술이다. 일반적으로 Classification 문제를 해결하는데 이용되지만, Layer를 어떻게 구성하느냐에 따라 비선형회귀 문제를 푸는데에도 이용할 수 있다. 기존의 CNN 방식의 프로세스는 다음과 같다.

해당 문제를 회귀 문제로 연결시키기 위해서, CNN 아키텍처의 레이어 구성을 확인하게 되면,

1. Input Layer : 이미지 데이터를 입력으로 받아, 입력 이미지의 크기와 채널 수에 맞게 설정한다.
2. Convolution Layer : 입력 이미지와 필터 사이의 합성곱 연산을 수행한다. 필터는 작은 크기의 윈도우로 이루어져있고, 이미지 데이터를 슬라이딩하면서 합성곱을 계산한다. 필터 수만큼 슬라이딩을 마치게 되면 여러장의 데이터가 생성되며, 각 행렬데이터에는 특정 공간의 패턴과 공간적 특성이 담겨져 있다.
3. Activation Function Layer : 활성화 함수를 통해 연산하는 레이어이다. 활성화 함수에는 tanh, Relu, Leaky Relu, linear, sigmoid 등 다양한 함수들이 존재하며, 데이터에 비선형성을 부여해주기 위해 존재하는 레이어다.
4. Pooling Layer : 합성곱 레이어와 활성화 함수 레이어를 통과하게 되면 최대 풀링(Max pooling)이나 평균 풀링(Average Pooling) 을 통

해 각 공간을 설명하는 가장 대표적인 특성값을 출력한다. 이를 통해 공간적 해상도를 줄이고, 계산량을 감소시킬 수 있다.

5. Fully Connected Layer : 완전 연결레이어는 특징 맵을 1차원으로 Flat하게 펼친 후 정답 클래스의 모든 뉴런과 연결된다. 신경망의 모든 뉴런과 연결되기 때문에 전결합층(Dense Layer)라고도 불리며, 해당 행렬이 어느 클래스에 속하는지를 확률적으로 계산하여 출력 레이어에 전달한다.
6. Output Layer : 최종 예측 결과를 출력한다. 분류 작업에서는 클래스 수에 해당하는 뉴런 수와 활성화 함수로 소프트 맥스 함수가 사용된다.

해당 아키텍처를 통해서 해결할 수 있는 문제는 상기한 Linear Regression의 한계점과 연결된다. 모델에 비선형성을 부여하는 Activation Function Layer를 통해 Input matrix와 output matrix의 관계가 복잡해지더라도 정확하게 회귀시킬 수 있다. 또 Convolution Layer가 행렬 특성을 공간적으로 축소하기 때문에 데이터의 차원을 축소할 수 있다.

## 2.2.4 MAE (Mean Absolute Error)

평균 절대 오차(Mean Absolute Error, MAE)는 회귀 분석에서 예측된 값과 실제 값 사이의 오차를 측정하는 지표로서, 다음과 같이 정의된다.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad \cdots (10)$$

여기서  $n$  은 데이터 포인트의 총 수,  $y_i$  는  $i$ 번째 데이터의 실제 값,

$\hat{y}_i$  는 I번째 데이터의 예측 값이다. MAE는 예측 오차를 각각의 데이터 포인트에서 발생한 절대적인 크기로 측정함으로써, 모델의 성능을 정량적으로 평가하는 데 사용된다. MAE는 모든 예측 오차를 동등하게 다루기 때문에, 이상치나 큰 오차가 전체적인 평가에 미치는 영향을 균형 있게 반영한다는 특징이 있다.

.....  
.....

## 제 3 장 실험

### 3.1 개요

본 실험에서 활용된 유동장 데이터는 (주)이에이트에서 제공을 하였으며, (주)이에이트는 자체 개발한 Lattice Boltzmann 시뮬레이션 솔버인 NFLOW로 유동장 데이터를 구축하였다.

실험에 활용된 데이터는 training 과 validation set 으로 구성되어 있다. 각 data set 은 각기 다른 initial inlet velocity ( $u$ ) 에 따라 initial 과 mean 유동 데이터를 ( $x, y, z, u, v, w, p$ ) 형태로 제공됐다. 전체 유동장은  $x$   $[-40.95, 40.95]$ ,  $y$   $[-59.8, 59.8]$ ,  $z$   $[0, 29.25]$ 의 그리드로 구성됐고, 총 7개의 간단한 건물 형태의 구조물이 유동장 내에 존재한다.

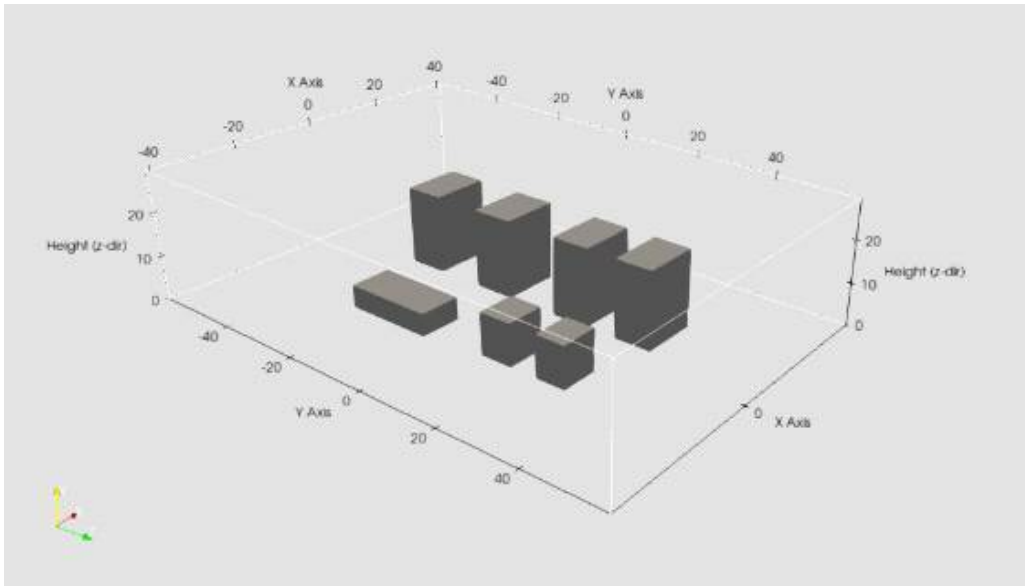


Fig. 1-1 초기 유동장 형태

본 실험은 다음과 같은 사항들을 목적으로 구성되었다.

- 적은 계산 비용으로 예측값을 얻을 수 있도록 데이터 기반 알고리즘을 적용해본다.
- 예측값을 시각화함으로써 정확도를 비교한다.
- 도심 유동을 예측함에 있어 계산 비용이 높은 CFD를 데이터 기반 알고리즘을 적용한 방식들이 대체가 가능한지를 확인한다.

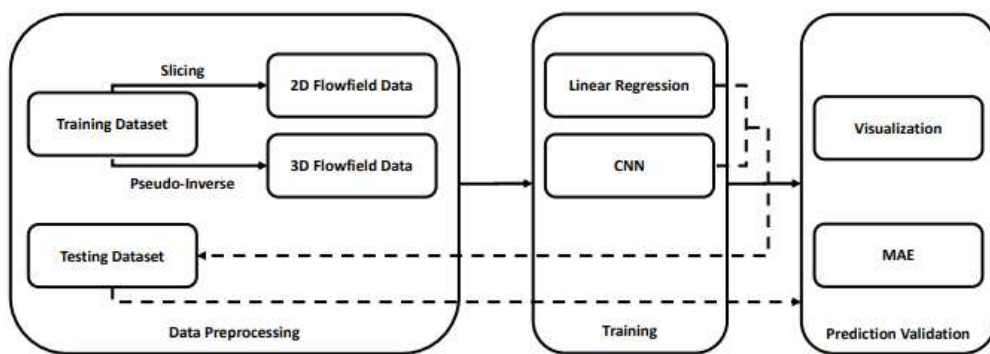


Fig. 1-2 시스템 로드맵

## 3.2 2D 모델

### 3.2.1 데이터 전처리

유동장을 slicing 함으로써 3D 모델을 2D 모델로 근사시켰다. 슬라이스가 되는 방향은 높이 방향으로 선정하였는데, 이는 유동장을 top view로 확인할 수 있기 때문이다. 해당 방향에 적용되는 slicing 인덱스는 사람 키를 근거로 2m로 선택하였다. 이러한 과정을 거치면 다루게 되는 데이터 수가 획기적으로 줄기 때문에, computing cost가 낮아진다는 것과, 전체적인 유동장에서의 유체 거동을 한눈에 볼 수 있다는 장점이 있다.

Train data set은 총 두가지 initial, mean 데이터 셋이 제공 된다. Initial 데이터의 경우 제공되는 (x,y,z,u,v,w,p) 중 초기 u 값만 0.5, 0.75, 1, 1.25, 2, 2.25, 3, 3.25, 4, 4.25 [m/s] 로 변화된 10개의 데이터 셋으로 구성되어있다. Mean 데이터 셋의 경우 초기 유동장에 u 속도를 가했을 때의 변화된 값의 평균을 제공한다. 해당 2D Regression Model은 이 Train Data 셋에서 Initial 과 Mean 데이터 셋의 연관성을 계산하여 weight를 구하고, 이를 Test 데이터 셋에 적용하여 예측 값을 구하는 방식을 택하였다.

data normalization의 경우,

1. dataset 간의 편차가 크지 않다.
2. 실험 데이터에 결측치나 이상치가 존재하지 않는다.
3. 표준 편차를 고려할 필요가 없다.
4. training data x와 y 간의선형적 관계가 입증되어 있으므로,, data scale을 같은 간격으로 유지시키며 변형해야한다.

이러한 이유로 min-max normalization을 사용했고, scaling 범위는  $[-1, 1]$  이다.

결과값과 비교하기 위해 Regression 모델을 적용시키기에 앞서, Train Data 로 활용될 mean 데이터 셋의 유동장을 먼저 시각화화하였다.

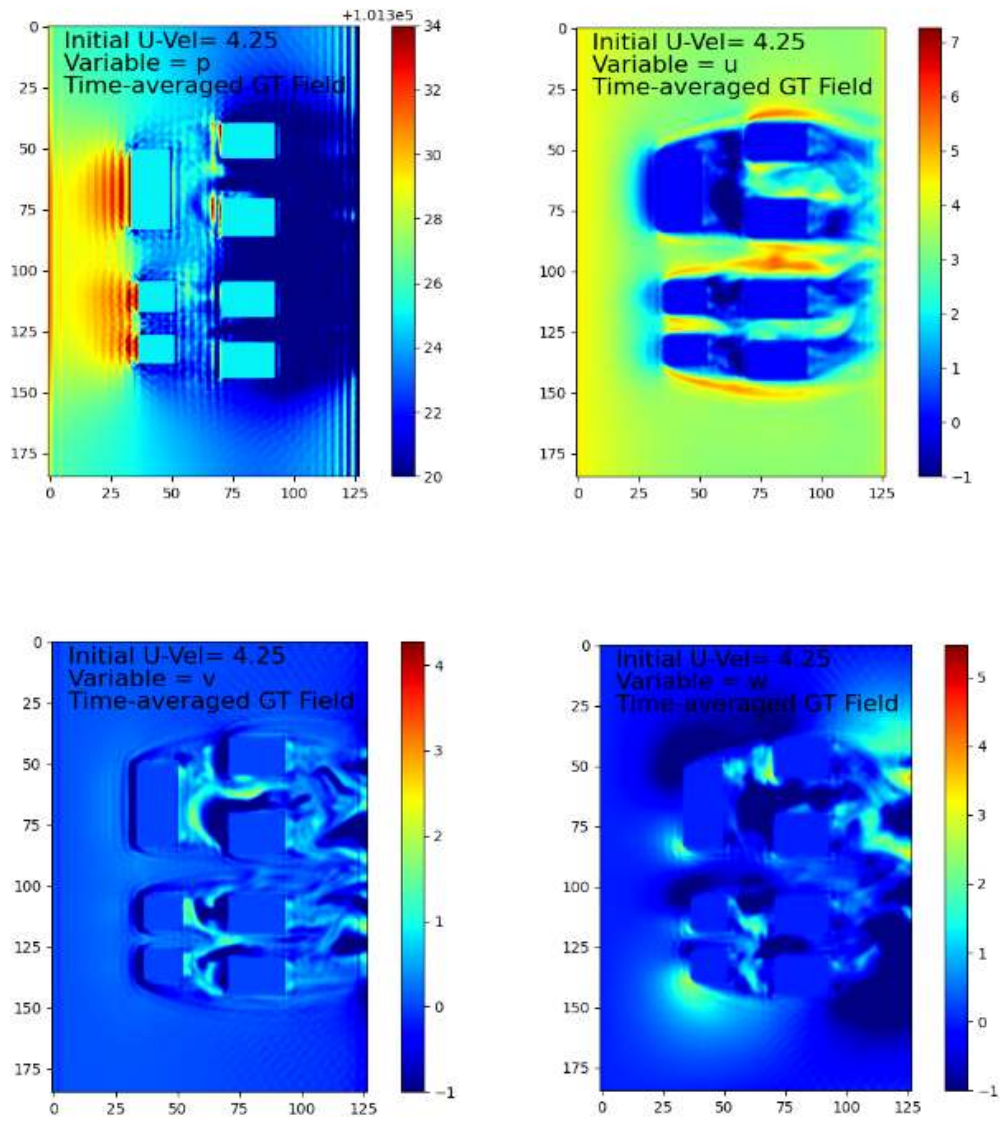


Fig. 2 Initial U-vel : 4.25 m/s 일때의 유동장

초기 유동장에 가해지는 유동은 x 축 방향으로 고정된 상태에서, 초기 유동이 먼저 건물 3개와 충돌을 하게 된다. 이후 유동이 flow over external bodies 처럼 갈라지는 현상을 확인할 수 있다. 먼저 압력 분포를 확인하면, 초기 유동의 1차 충돌을 견뎌야하는 앞 건물들이 유동으로부터 가장 높은 압력을 전달 받는다. 이후 2차 건축물들은 앞 건물에 따라 유동으로부터 다른 압력을 받는 것을 확인할 수 있다. 속도 성분의 경우,  $V=(u,v,w)$  로 나누어 시각화 하였다. u 방향의 속도 성분의 경우, 건물들 사이에 유동이 흘러갈 때, 벤츄리 효과로 인해 가속이 되는 것을 확인할 수 있다.

### 3.3 3D 모델

#### 3.3.1 데이터 전처리

Pseudo-inverse matrix를 적용시키기 위해, 유동장 데이터를 xy평면 기준으로 나누었다.

.....  
 .....



## 제 4 장 실험결과 및 고찰

### 4.1 2D 모델 결과

#### 4.1.1 Linear Regression 모델

Regression 모델 적용 결과는 다음과 같다.

.....  
.....

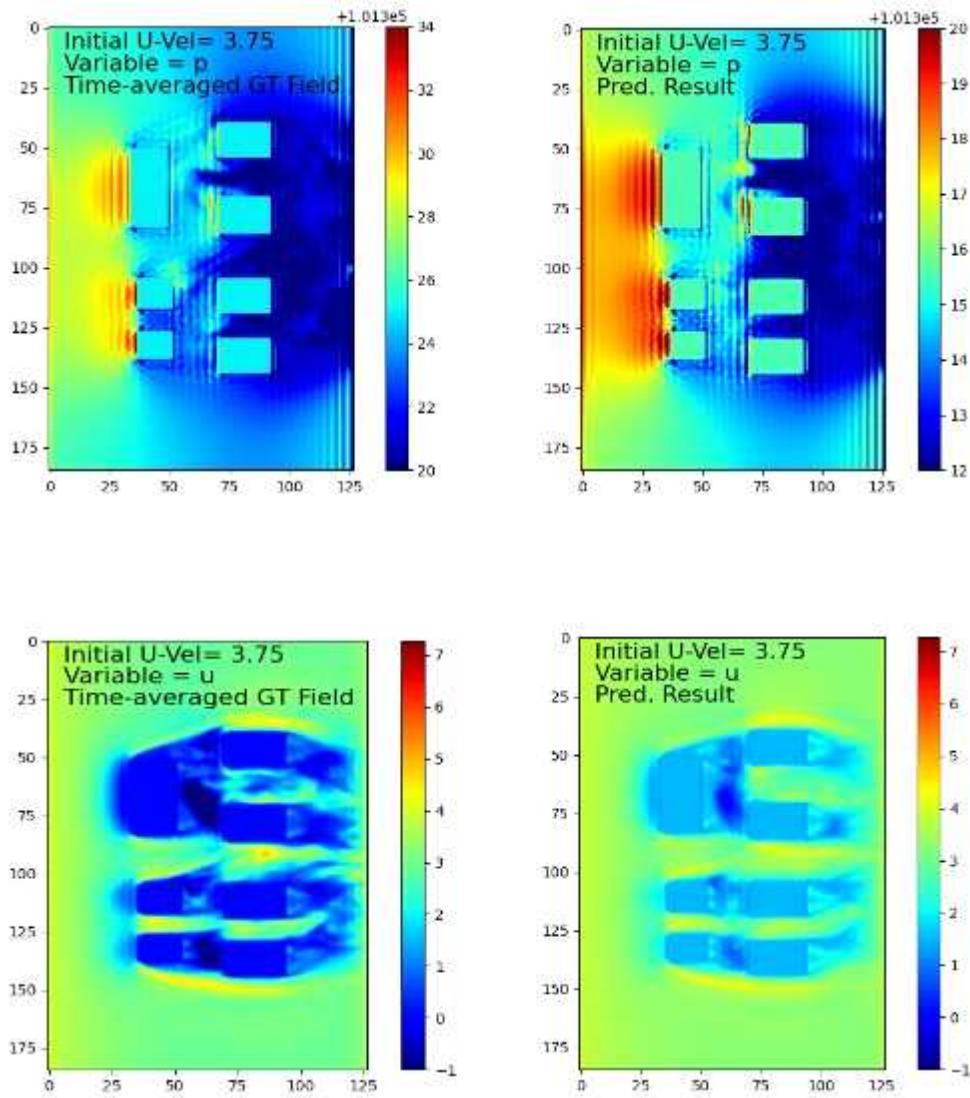


Fig. 3-1 Initial U-vel : 3.75 m/s 일때의 GT와 Prediction 값 비교 (p,u)

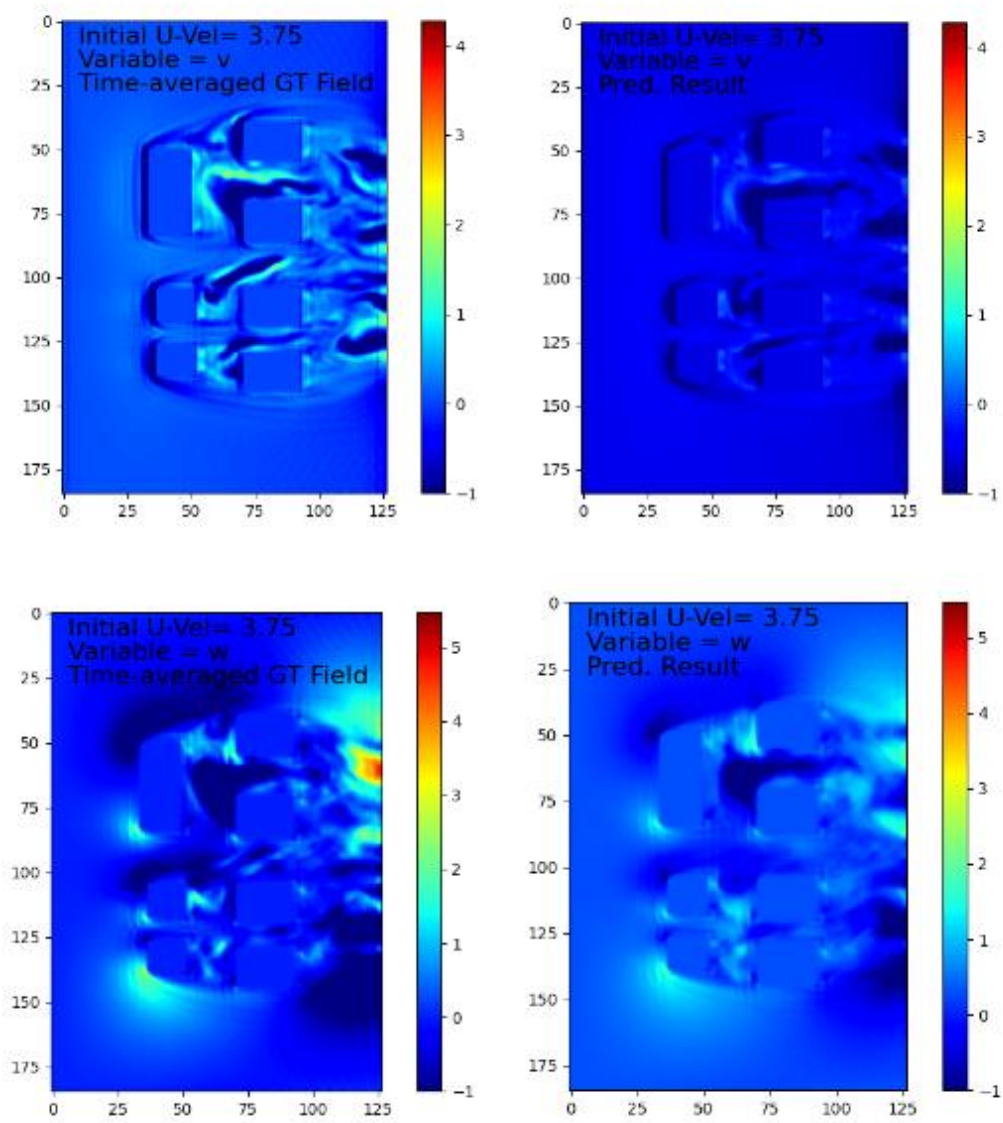


Fig. 3-2 Initial U-vel : 3.75 m/s 일때의 GT와 Prediction 값 비교 ( $v, w$ )

초기 u 방향 유속 [0.25, 1.5, 1.75, 2.5, 2.75, 3.5, 3.75, 4.5, 4.75, 5.0] 에 따른 압력과 속도 예측값을 시각화한 후 이를 Ground Truth 값과 비교를 하였다. 압력 분포와 속도에 따른 유동장의 형태는 예측값과 Ground Truth 값과 유사함을 보였지만, 예측 수치 자체는 일정 수준 오차가 있음을 시각적으로 확인할 수 있었다.

#### 4.1.2 2D CNN 모델

부족한 데이터셋으로 인해 초기 모델의 성능은 굉장히 낮았다. 이에 따라 훈련세트와 테스트 세트의 비율이 5:5 였으나, 테스트 세트의 target data가 있음을 이용해 이를 8:2 정도로 보정하여 훈련세트와 검증 세트로 사용하였다. 이 과정을 통해 부족한 데이터셋을 보완하고, 영역에 따른 특성값 추출 성능을 높였다. 성능 평가 시 일반적으로 채택하는 8:2( train:validation ) 정도의 비율로 분리하였으며, 모델의 구조는 input layer, convolution 연산을 수행하는 3개의 hidden layer, input shape 과 같은 크기의 출력값을 갖는 output layer 로 구성되어 있다. convolution 과정에서 데이터가 padding 등에 의해 데이터가 소실되는 것을 막기 위해 input shape을 유지시키며 레이어를 통과하도록 수정하였다. 또한 선형 회귀모델에 의해 계산된 prediction 값의 loss가 크지 않음을 고려하여 영역들이 상대적으로 큰 선형성을 가지고 있다고 판단하였다. 따라서 이러한 특성을 고려하기 위해 hidden layer의 커널 사이즈는 (3,3) 이내, 데이터의 형식과 전처리 방식등에 근거하여 활성화 함수의 경우 leaky relu를 사용하였다.

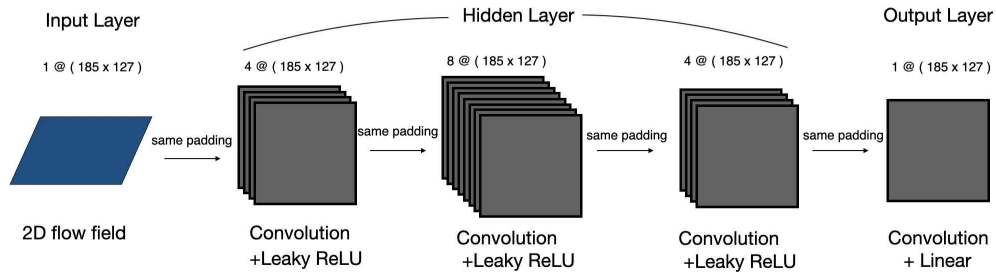


Fig. 4-1 2D CNN의 계층적 구조

다만, 예측값을 시각화해 본 결과 속도 성분의 경우 2D CNN 활용 시, 그 분포가 적절함을 보였지만, 압력 분포에 대해서는 그렇지 않음을 확인했다. (Fig. 5-1, 5-2)

기본적으로 머신러닝 및 딥러닝 기반의 학습 시에는 데이터 셋의 개수가 모델의 성능을 결정하는 데에 큰 영향을 미치게 되는데, 현재의 문제 상황을 해결하는 데에 있어 Fig 4-1의 모델은 데이터 셋이 충분하지 않아 지역적인 특성을 충분히 학습하지 못해 채널별로 분리된 듯한 결과값을 얻었다고 판단하였다. 이에 대한 보완책으로 layer 별 유속과 압력 간의 상관관계를 고려하여 데이터 셋을 증강시키는 방향으로 접근하여 다음과 같은 아키텍처를 구성하였다. 3차원 유동장 내의 데이터를 활용하여 CNN 모델을 구성한 후, Output layer에서 도출되는 결과 중 높이 2m에서의 값을 활용하여 시각화 하였다 (Fig 5-3).

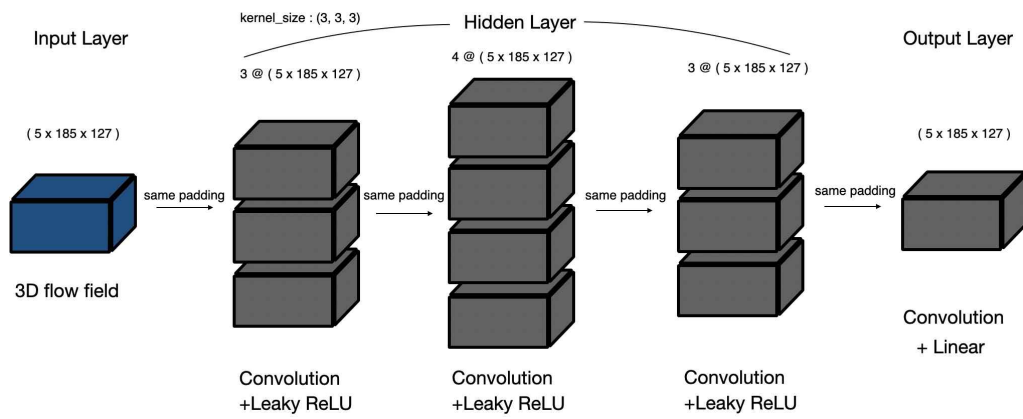


Fig. 4-2 3D CNN의 계층적 구조

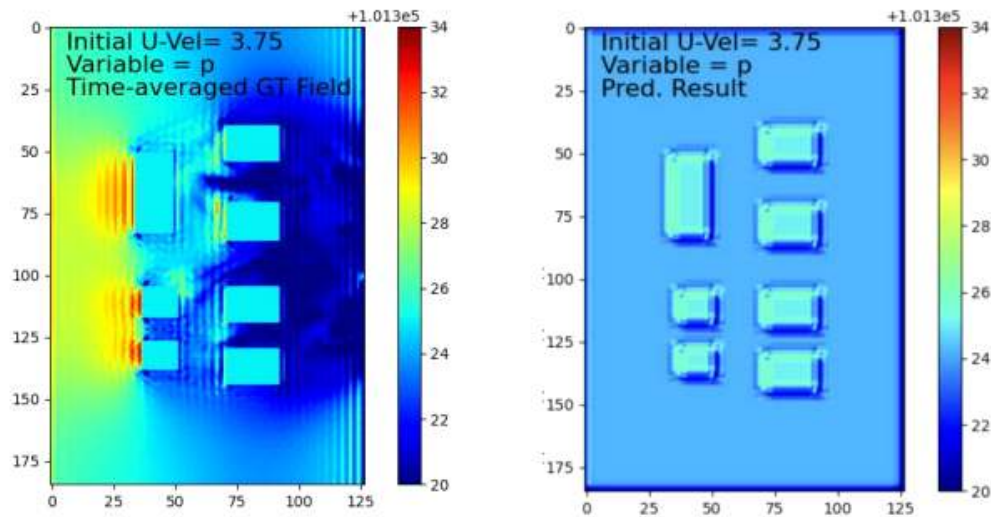


Fig. 5-1 새로운 모델 적용 전 Initial U-vel : 3.75 m/s 일때의  
GT와 2D CNN Prediction 값 비교 (u)

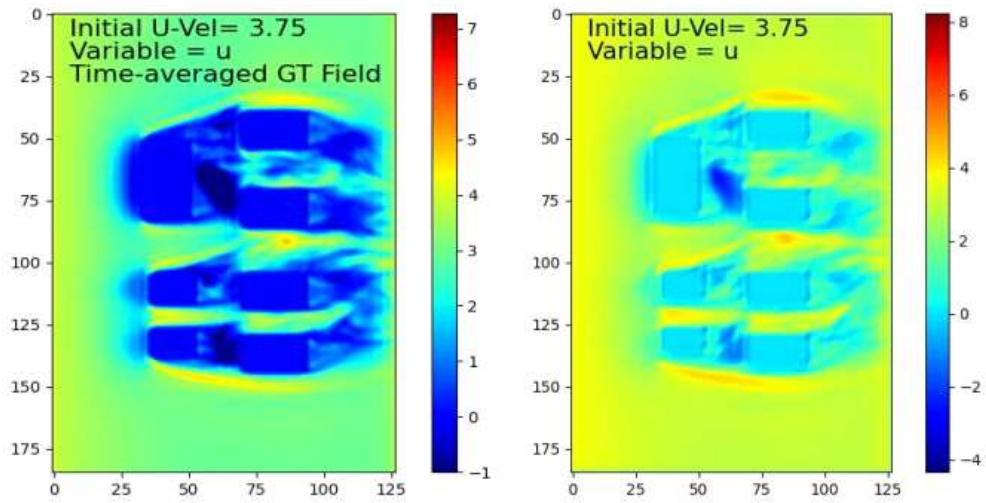


Fig. 5-2 Initial U-vel : 3.75 m/s 일때의 GT와 2D CNN Prediction 값 비교 (u)

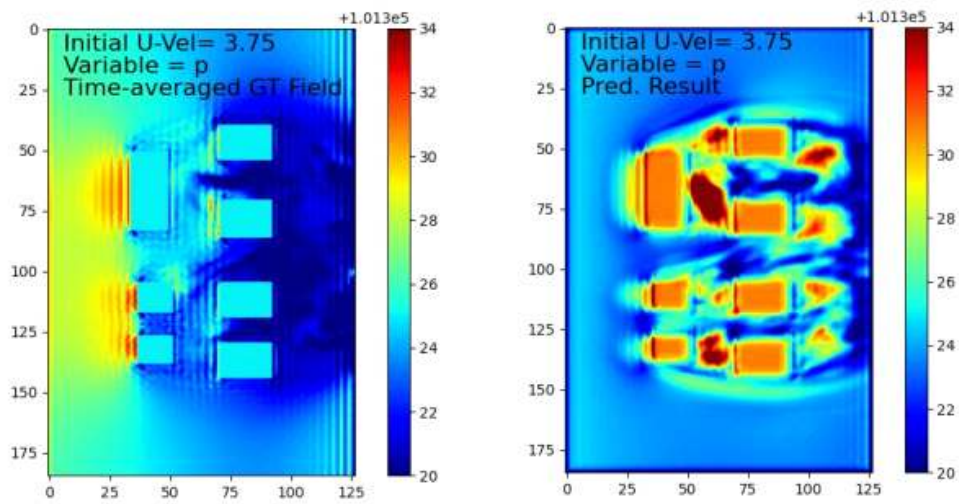


Fig. 5-3 Initial U-vel : 3.75 m/s 일때의 GT와 2D CNN Prediction 값 비교 (P)

Convolution layer를 사용한 모델의 특성 상 데이터량이 많아야지만 지역적인 특성을 높은 정확도로 학습할 수 있다. Fig 5-3를 통해 압력 분포 예측값을 확인하면 건물 전면부에서의 압력 증가가 잘 반영 됐음을 확인할 수 있다. 다만 삼차원 유동장 내의 데이터를 활용해서 도출해낸 값을 확인했기 때문에 속도와 압력 간의 연관성 또한 예측값에 반영된 것을 확인할 수 있다. 특히 전면 건물들과 후면 건물들 사이 발생하는 압력값의 증가는 베르누이 효과로 인한 것으로 해석할 수 있다. 해당 영역에서 유속의 감속이 발생하는데 이로 인해서 압력 증가가 발생한 것으로 볼 수 있기 때문이다. (식 2, Fig 3-2) 다만, Fig 5-2과 5-3에서 볼 수 있 듯, GT 값의 스케일과 prediction 값의 스케일이 상이함을 확인할 수 있다. 이 부분은 추가적인 유동장 데이터를 활용할 수 있다면, 그 오차가 줄어들 것이라 판단된다.

계산 비용은 모델의 복잡도에 따라 다르겠지만, 선형회귀 모델에 비해 360%, Pseudo inverse 모델에 비해 10000 % 정도 큰 계산비용이 발생한다. 그러나 유동장의 구조가 복잡해지거나, 유동장의 크기가 커지는 경우, 또 유동장에서의 고려해야하는 특성값(feature)이 늘어나게 되는 등 다양한 문제상황에 의해 비선형성이 강해지고, 모델을 고도화 시켜야하는 경우 CNN이 앞선 두개의 모델에 비해 높은 정확도를 보일 수 있다. 또한 튜닝에 의해서 문제상황에 모델을 맞추어 편집하는 것도 가능하기 때문에 더욱 다양한 상황에서 적용가능하다.

.....  
 .....



## 4.2 3D 모델 결과

3D 모델을 얻기 위해 데이터 셋을 직접적으로 활용하여 해를 구하려고 하면 과도한 계산값이 요구된다. 이는 시스템이 과결정계에 해당하기 때문에 해를 구하는 것이 어렵기 때문인데, Pseudo-inverse matrix 를 적용하면 근사해를 구하는 것이 가능해진다.

$$\vec{Ax} = \vec{b} \cdots (11)$$

$$A_{n \times m} = \begin{cases} A_{m \times n}^+ = (A^T A)^{-1} A^T, & A^+ A = I_{m \times m} \text{ if } n \geq m \\ A_{m \times n}^+ = A^T (A A^T)^{-1}, & A A^+ = I_{n \times n} \text{ if } n < m \end{cases} \cdots (12)$$

$$\tilde{x} = A^+ b \cdots (13)$$

이렇게 얻어진 유사역행렬 근사해에 해당하는  $\tilde{x}$ 를 기반으로 3D 모델을 얻을 수 있다. 이를 위해 Numpy 내장 함수를 활용하여 초기 데이터 값의 pseudo-inverse matrix를 구한 후 이를 train 데이터 셋과의 행렬 곱셈을 통해 모델 행렬을 구하였다.

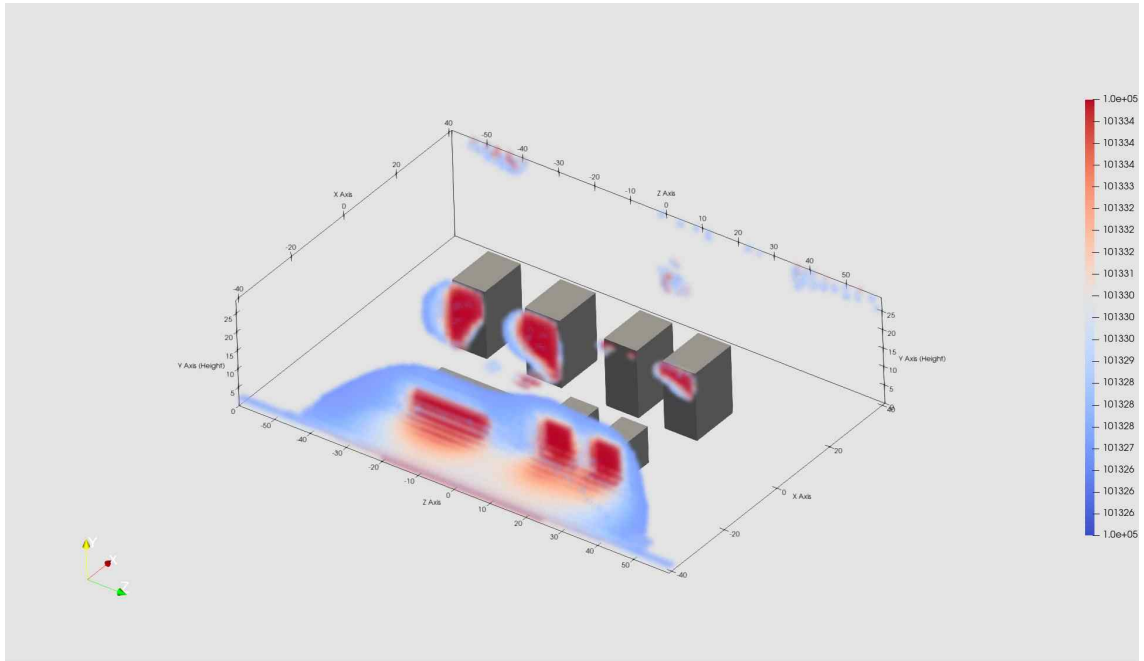


Fig. 6-1 Initial U-vel : 5 m/s 일때의 GT와 Prediction P값 비교 (GT)

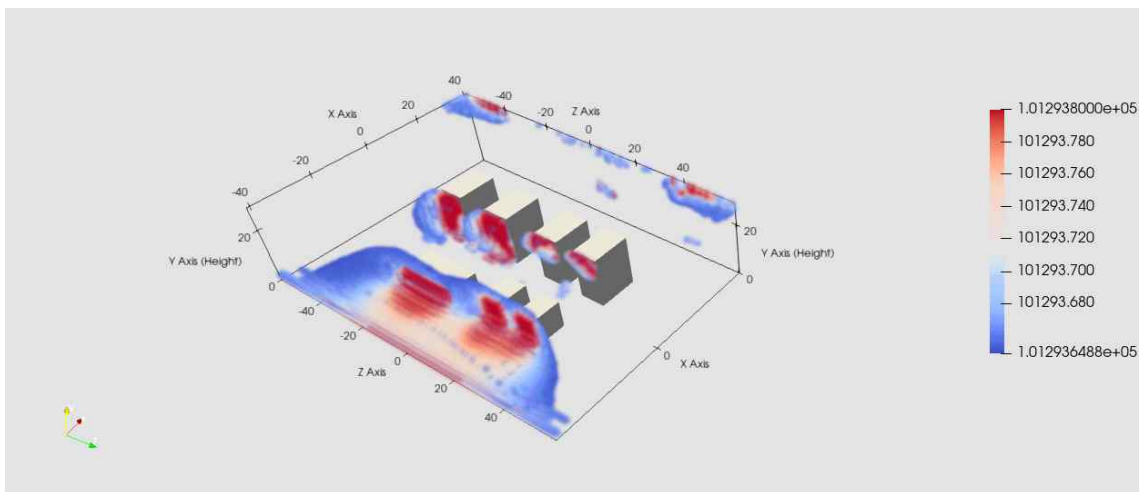


Fig. 6-2 Initial U-vel : 5 m/s 일때의 GT와 Prediction P값 비교 (Prediction)

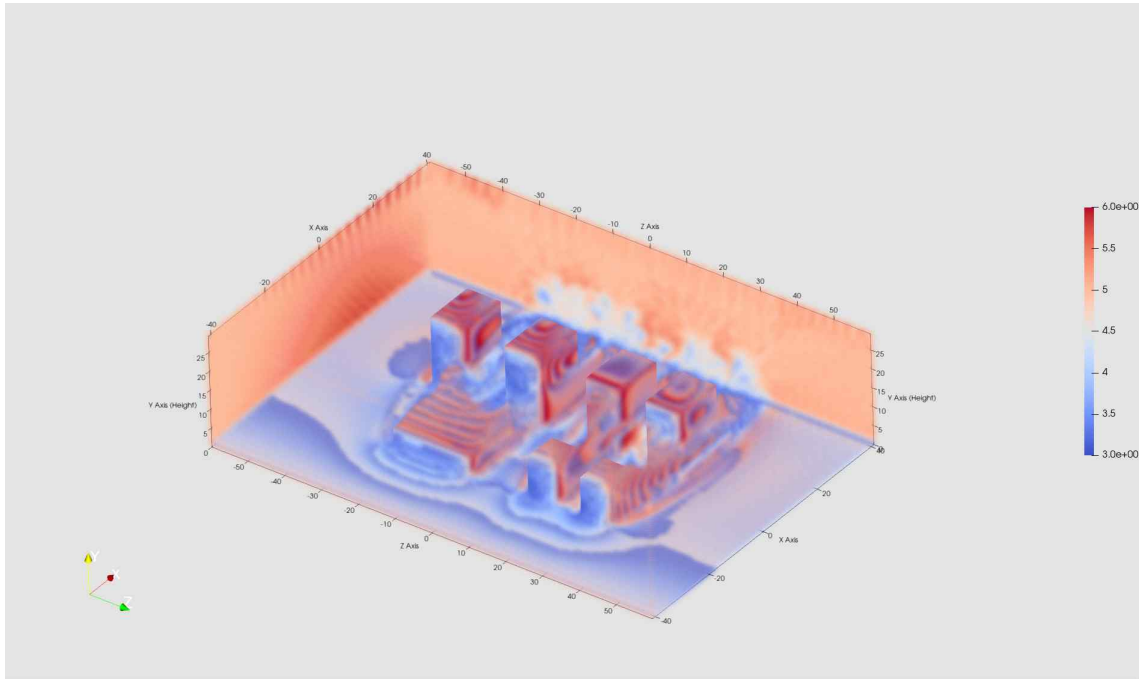


Fig. 7-1 Initial U-vel : 5 m/s 일때의 GT와 Prediction  $u_{\text{값}}$  비교 (GT)

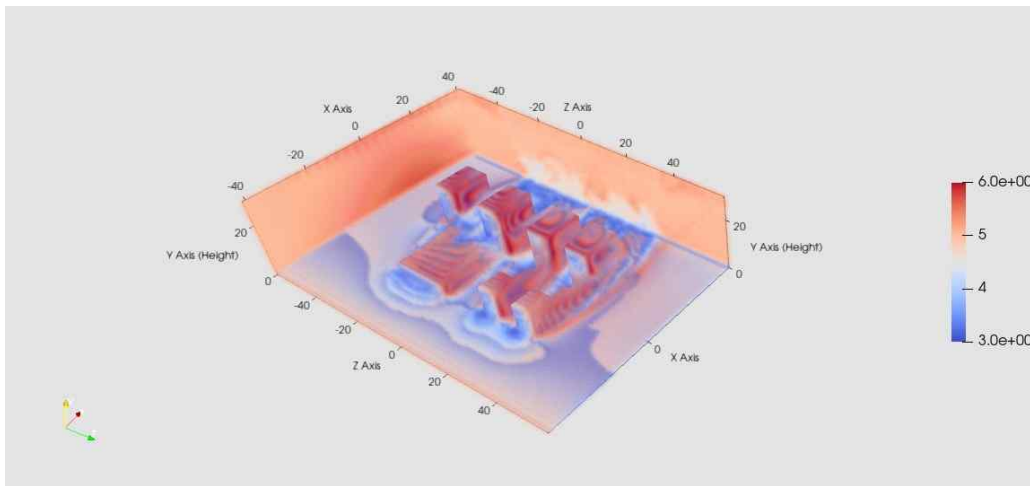


Fig. 7-2 Initial U-vel : 5 m/s 일때의 GT와 Prediction  $u_{\text{값}}$  비교 (Prediction)

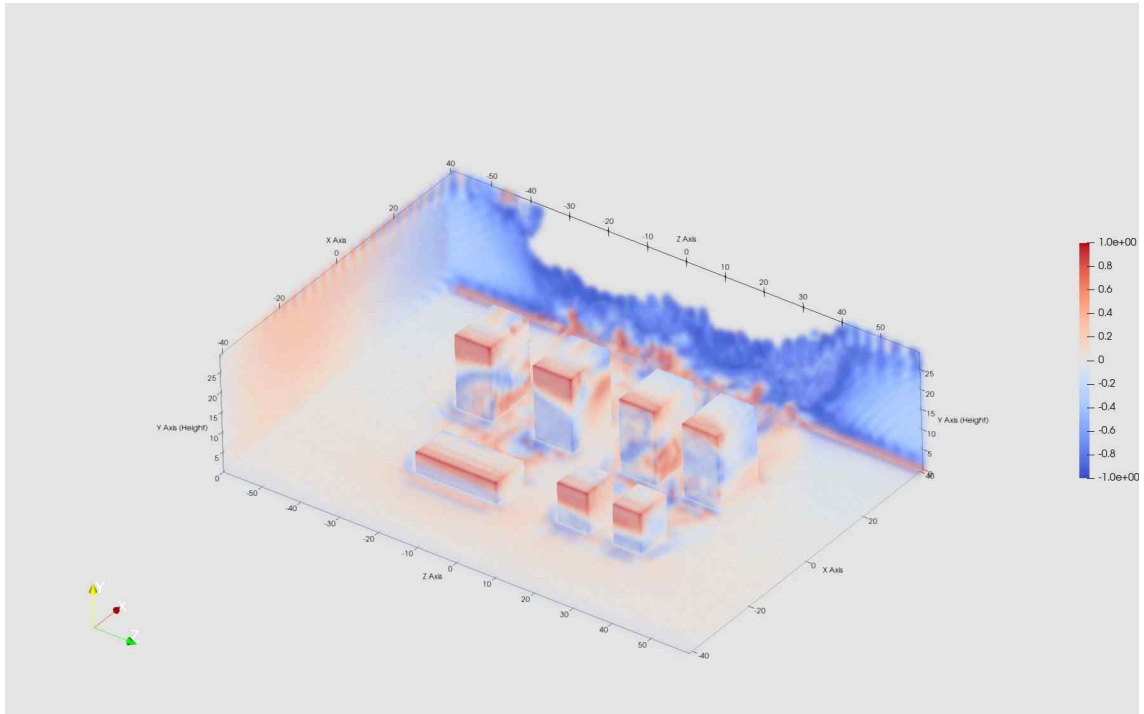


Fig. 8-1 Initial U-vel : 5 m/s 일때의 GT와 Prediction  $v$  값 비교 (GT)

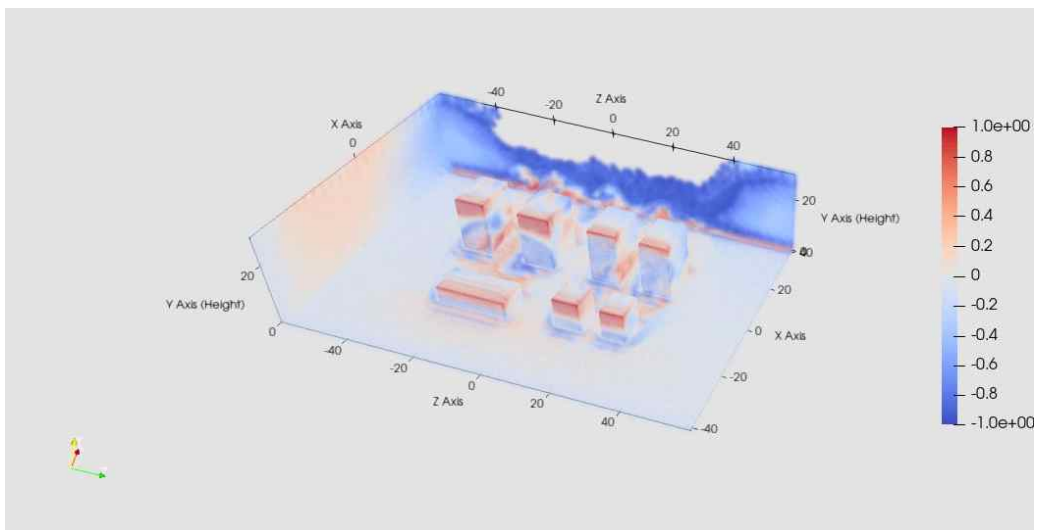


Fig. 8-2 Initial U-vel : 5 m/s 일때의 GT와 Prediction  $v$  값 비교 (Prediction)

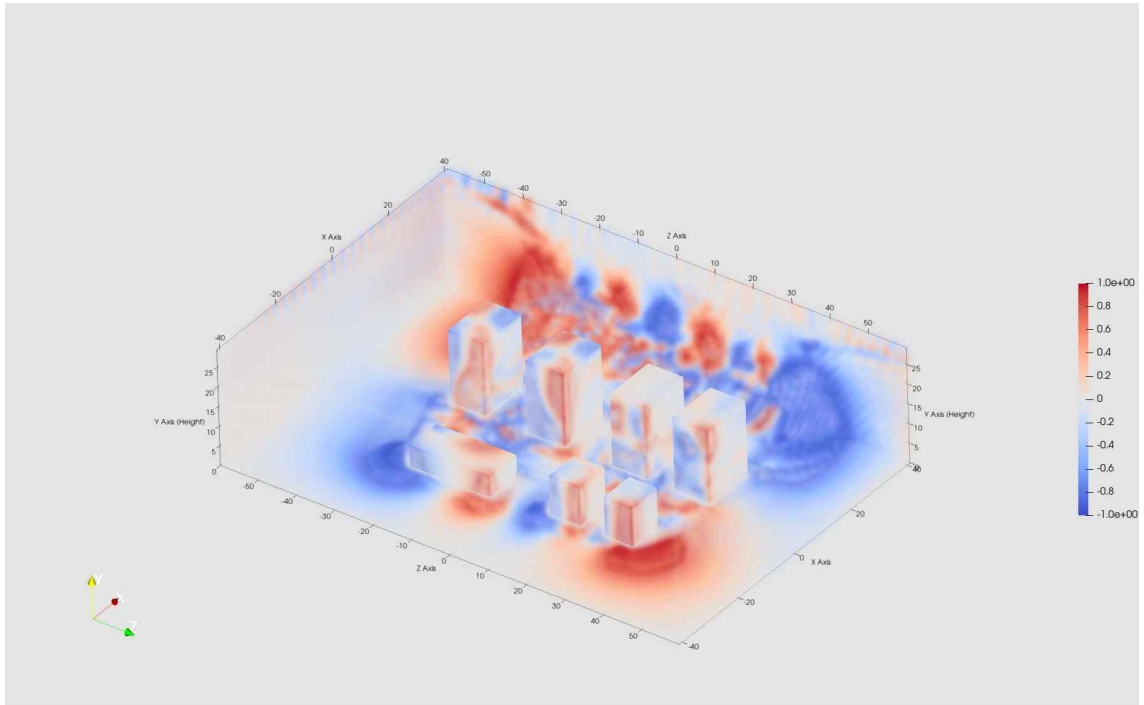


Fig. 9-1 Initial U-vel : 5 m/s 일때의 GT와 Prediction w값 비교 (GT)

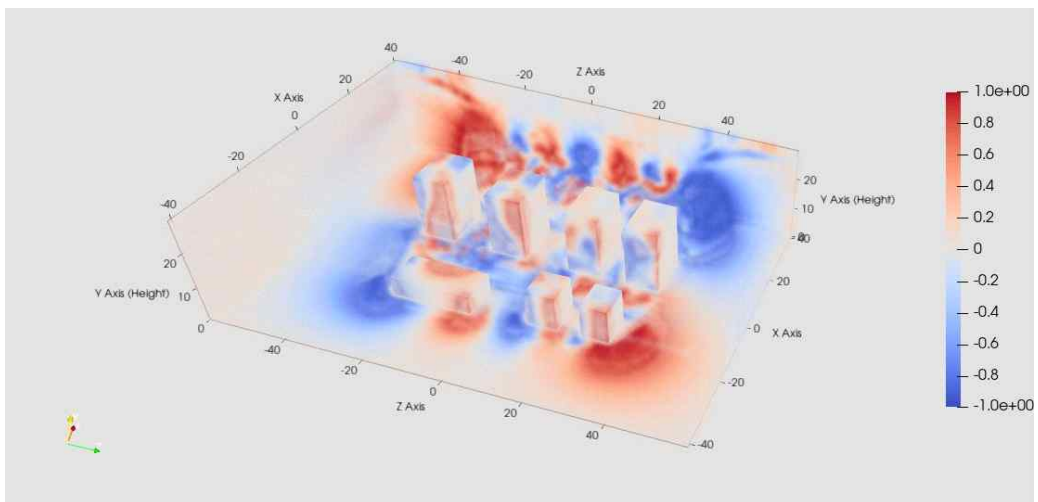


Fig. 9-2 Initial U-vel : 5 m/s 일때의 GT와 Prediction w값 비교 (Prediction)

2D 모델에서와 동일하게 초기  $u$  방향 유속 [0.25, 1.5, 1.75, 2.5, 2.75, 3.5, 3.75, 4.5, 4.75, 5.0] 에 따른 압력과 속도 예측값을 시각화한 후 이를 Ground Truth 값과 비교를 하였다. 예측값과 Ground Truth 값을 시각화한 자료를 비교해본 결과 압력과 속도 분포가 매우 유사함을 확인할 수 있었다. 다만, Ground Truth 값과 비교했을 때 예측값의 수치와 일정 수준 오차가 2D 모델처럼 존재함을 보였다.

## 4.3 2D와 3D 모델 결과의 고찰

### 4.3.1 차트 해석

2D 모델의 경우 결정계수가 약 0.9879, 3D 모델의 경우 결정계수가 약 0.8179가 나왔다. 이에 대한 이유론 데이터의 특성과 다중공선성을 이유로 들 수 있다. 먼저 2D 모델의 경우,  $y$ 축 (높이) 값을 고정시키고 weight 값을 계산했기 때문에, 활용된 데이터의 수가 3D 모델과 비교했을 때 적다. 즉 계산에 활용된 독립 변수가 감소한 효과를 얻기 때문에, 더 높은 결정 계수를 얻었다고 할 수 있을 것이다. 또 다른 이유론 다중공선성을 들 수 있다. 다중공선성 (multicollinearity)은 회귀 분석에서 독립 변수들 간에 높은 상관관계가 있는 경우를 말하는데, 기본적으로 실험에 활용된 유동장 내의 압력과 속도 ( $p, u, v, w$ ) 는 역학적으로 상관관계를 갖을 수 밖에 없다. 그렇기 때문에 3D 모델의 경우 2D 모델과 비교했을 때, 더 많은 양의 데이터를 계산에 활용하기 때문에 회귀 분석 결과 독립 변수들의 영향을 정확하게 파악하기 어려워졌을 가능성이 높다.

### 4.3.2 MAE

모델별 MAE 값은 다음과 같다.

MAE	2D LR	2D CNN	3D P-Inv
P	0.1269257112270313	0.16937118195556583	0.44815279018820947
U	0.10535058717264359	0.02943172661789585	0.28501266400037406
V	0.10128979578718336	0.03710104170816678	0.4571390453492314
W	1.8247506752293146	0.0774175106653103	0.266211624540983

Fig. 10-1 각 모델들의 MAE 비교

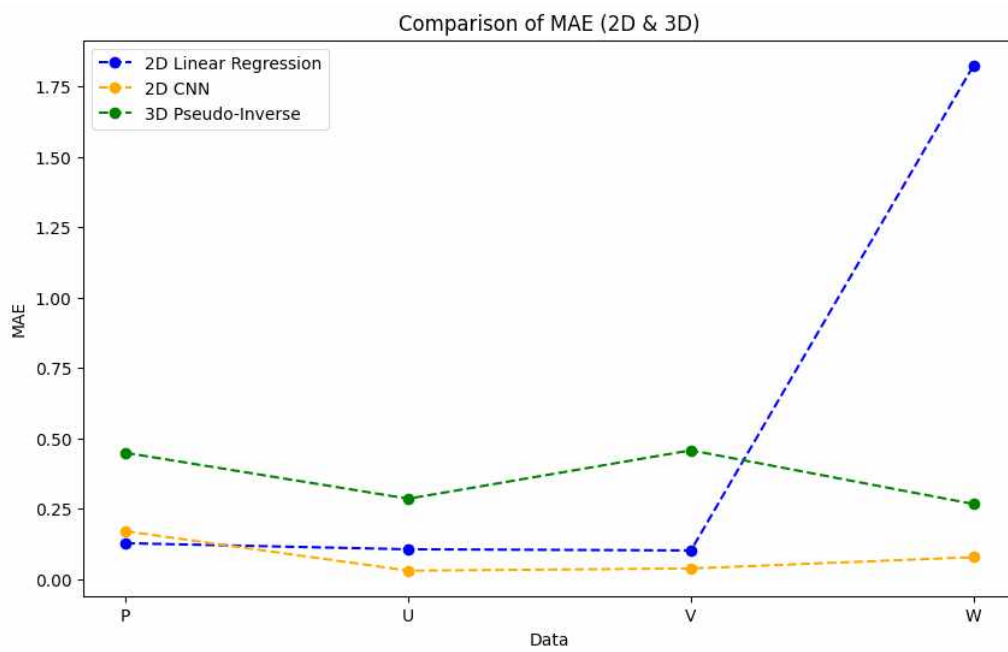


Fig. 10-2 각 모델들의 MAE 비교

2D 모델의 경우 MAE는 2D CNN이 2D 선형회귀 방식과 비교했을 때 압력 값 (P)를 제외한 나머지 변수들에서 모두 낮게 나왔다. 이는 일반적으로 2D 모델을 예측함에 있어 CNN 방식이 더 우수하다는 해석을 할 수 있다.

.....  
.....



## 제 5 장 결론

기존 CFD 를 통한 유동장 예측은 계산비용이 지나치게 큰 양상을 보였다. 따라서 위 논문에서는 data-driven방식을 이용하여 도심 유동장을 예측해보았다. 고전적인 통계방식인 선형회귀 모델, 차원축소로 획기적으로 비용을 낮출 수 있는 아이디어인 Pseudo inverse matrix 에 따라 해를 근사시키는 방식, 마지막으로 영역적 특성을 잘 고려할 수 있는 Deep learning architecture 인 CNN(Convolution Neural Network)를 적용하여 계산 비용을 획기적으로 낮추면서도, 낮은 loss 를 갖는 방식을 연구하였다.

1. 2차원 형태의 유동장을 예측하는 선형회귀모델의 경우 해를 계산하는데 36초가 소요되었고  $u$ ,  $v$ ,  $w$ ,  $p$  의 평균절대오차(MAE)는 각각 0.1053, 0.1012, 1.8247, 0.1269 정도로  $w$  벡터를 제외하면 절대오차가 0.1 이내로 작게 나타났다. 그러나 3차원 형태의 유동장을 예측하는 선형회귀모델의 경우 계산비용이 굉장히 크게 책정되어 입력 데이터 사이즈를 급격히 줄일 수 있는 유사역행렬변환을 통한 회귀모델을 고안하였다.

2. 3차원 형태의 유동장을 예측하는 유사역행렬변환을 통한 회귀모델은 해를 계산하는데 0.4초가 소요되었지만  $u$ ,  $v$ ,  $w$ ,  $p$  의 평균절대오차(MAE)는 각각 0.2850, 0.4571, 0.2662, 0.4481 정도로  $w$  벡터를 제외하면 절대오차가 비교적 크게 측정되었다. 입력 데이터가 단순해짐에 따라 회귀 시에 한계점이 발생한 것으로 추측된다. 그러나 충분한 데이터셋이 주어지게 되면 성능이 개선될 수 있다.

3. 앞선 두가지 방식들을 통해 계산된 Loss 값을 통해 데이터셋 자체의 비선형성이 고려되어야함을 알 수 있었다. 이에 따라 Convolution 을 이용한 딥러닝 기반의 회귀 모델을 고안하였다.

4. 2차원 형태의 유동장을 예측하는 CNN을 통한 회귀모델의 경우 20분 정도가 소요되었고,  $u$ ,  $v$ ,  $w$ ,  $p$  의 평균절대오차(MAE)는 각각 0.0294, 0.0371, 0.0774, 0.1694 정도로 정확도 측면에서 이전 방식들에 비해 상당히 개선된 성능을 확인할 수 있었다. 계산비용이 매우 크게 책정되었으나, 유동장을 예측하는데 있어, 적용할 수 있는 문제상황이 광범위하며, 튜닝을 통해 문제상황에 fit하게 적용시킬 수 있다.

.....  
.....

## 참고문헌

- [1] Ahsan Kareem, Emerging frontiers in wind engineering: Computing, stochastics, machine learning and beyond, Journal of Wind Engineering and Industrial Aerodynamics, Volume 206, 2020,
- [2] H. Tanaka et al., "Optimization Calculations and Machine Learning Aimed at Reduction of Wind Forces Acting on Tall Buildings and Mitigation of Wind Environment," International Journal of High-Rise Buildings, vol. 8, no. 4, pp. 291-302, Dec. 2019.
- [3] Giovanni Calzolari, Wei Liu, Deep learning to replace, improve, or aid CFD analysis in built environment applications: A review, Building and Environment, Volume 206, 2021,