

Artificial Intelligence in Ship Design

Pusan National University

Dept. of Naval Architecture & Ocean Engineering

Ship Design Production Laboratory

Sung-chul Shin

SDPL

선박설계생산연구실 Ship Design & Production Lab.

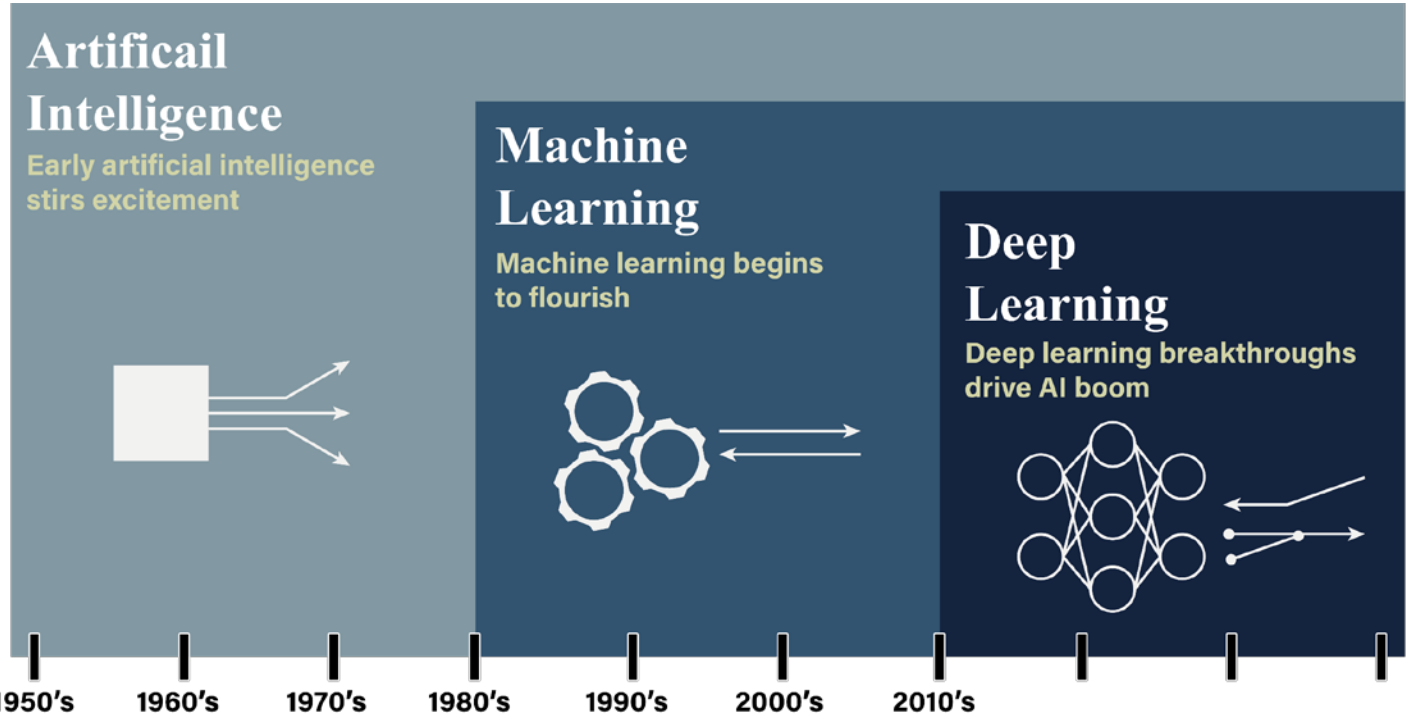
Contents

- Introduction
- Development of Design Optimisation Model through Component Analysis
- Automate Production through Image Processing
- Reinforcement learning

Introduction



Artificial Intelligence



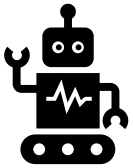
Since an early flush of optimism in the 1950's, smaller subsets of artificial intelligence - first machine learning, then deep learning, a subset of machine learning - have created ever larger disruptions.



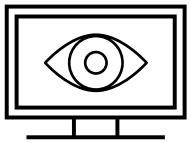
Natural Language Processing



Speech Interpretation



Robotics

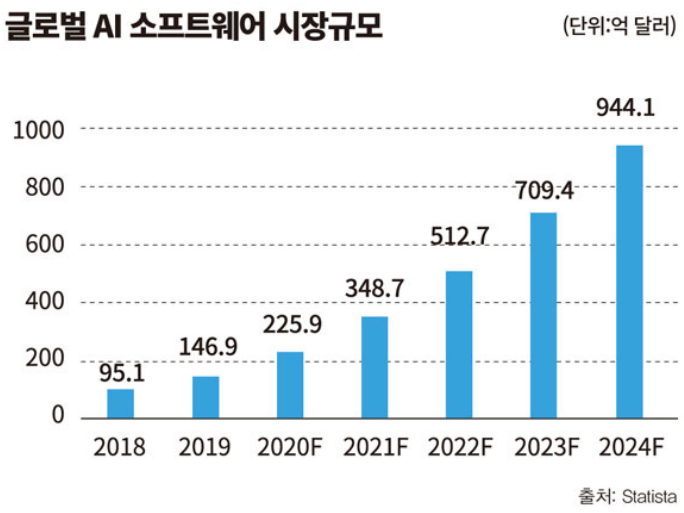


Computer vision

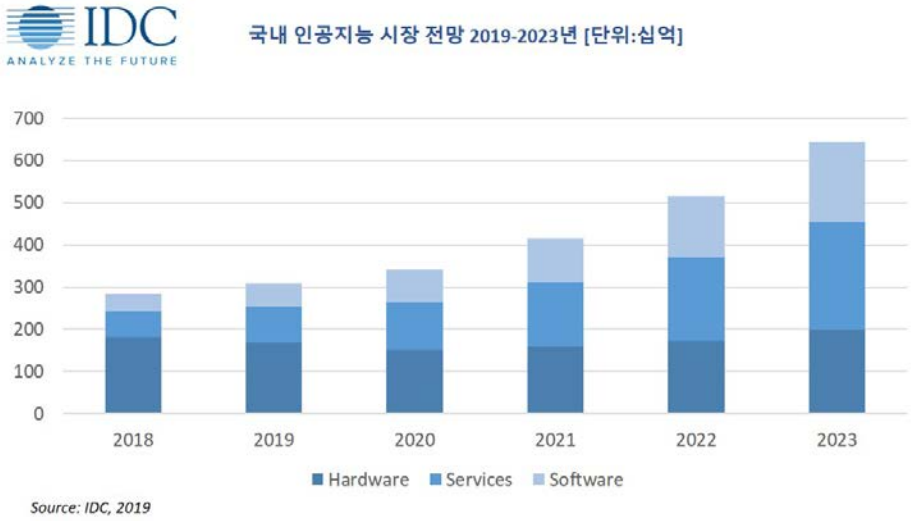
AI Market Size

❖ The Growing AI Market

- **The global AI market is expected to grow 43.4% annually from $9.5 billion in 2018 to $94.4 billion by 2024.**
- **The domestic artificial intelligence(AI) market grow an average of 17.8% annually over the next five years to form more than 640 billion won in 2023.**



Prospects for the global artificial intelligence market
[unit: 100 million dollars]



Prospects for the domestic artificial intelligence market
[unit: 1 billion won]

AI in the Shipbuilding Industry



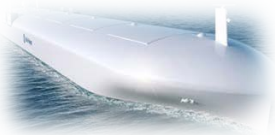
Smart ship

- Smart vessel in a comprehensive sense that applies advanced equipment, ICT, etc.



Autonomous ship

- Operated with only a minimum crew member by replacing the role played by the crew by the system



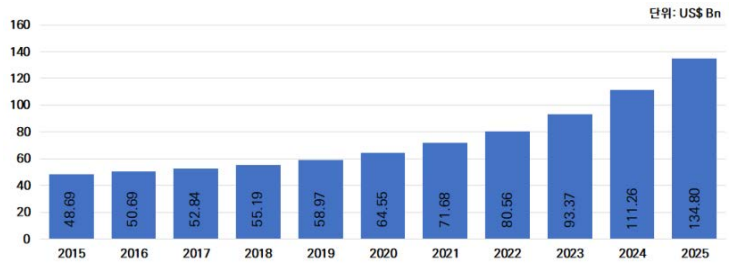
Unmanned ship

- A fully autonomous ship that is possible without human intervention

Trends in Autonomous Ship Technology by Country

Korea	• Promoting the development of unmanned ships through 'multi-purpose intelligent unmanned vessel localization project'
Japan	• Japan is jointly developing 250 autonomous cargo ships by 2025
China	• After building the world's first smart ship, the Green Smart, it entered the phase of operation.
Europe	• It aims to develop a fully autonomous ocean-going ship within 2030.

❖ *The size of the global autonomous ship market*



Prospects for the global Autonomous ship market
[unit: 1 billion dollars]

- **The global market for commercial autonomous ships was valued at $50.6 billion as of 2016 and is expected to grow 12.4% annually to $134.8 billion by 2025 over the forecast period from 2017 to 2025**

Recognition



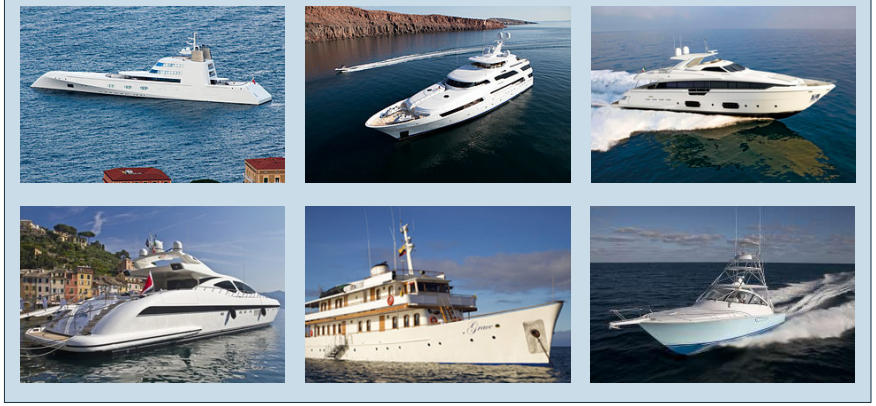




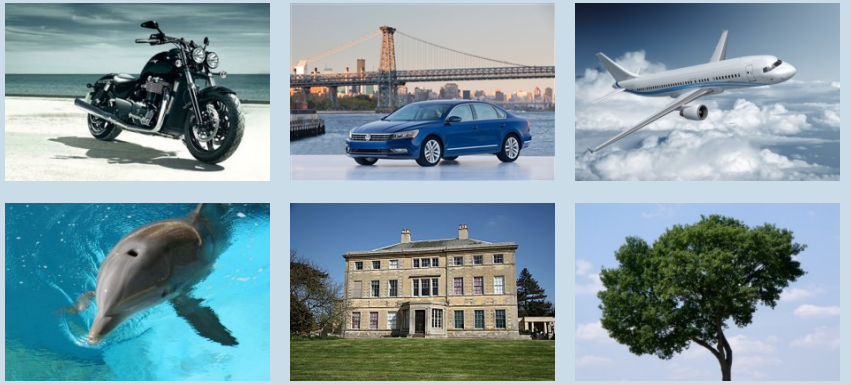
194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

Computer vision: yacht detection

Yachts



Not a yacht



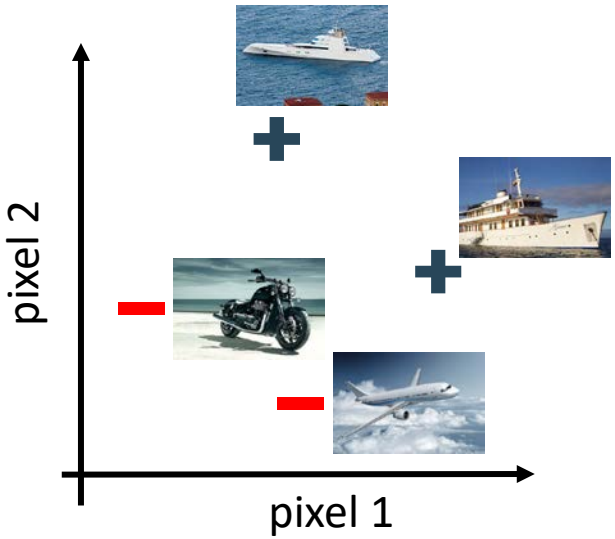
Testing:



What is this?

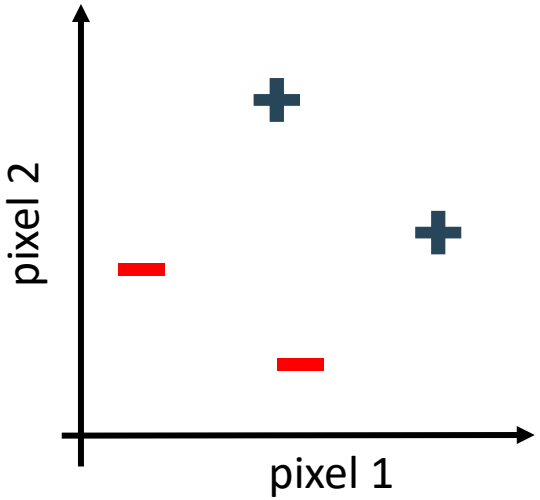


Learning Algorithm



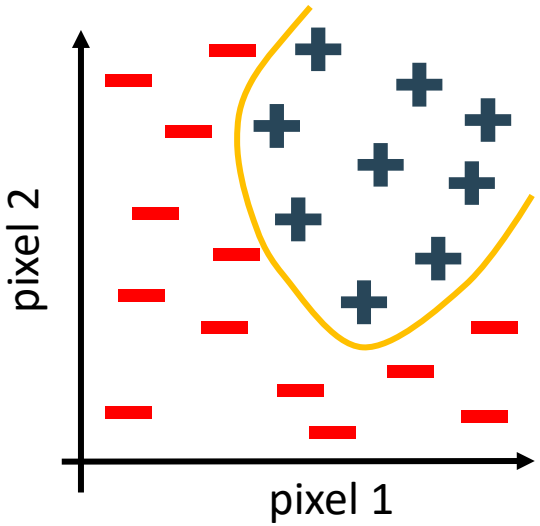


Learning Algorithm





Learning Algorithm



50×50 pixel images: 2500 pixels

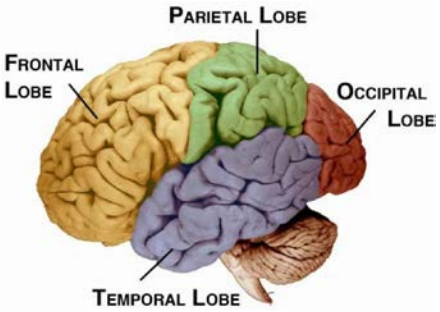
$n = 2500$ (black & white)

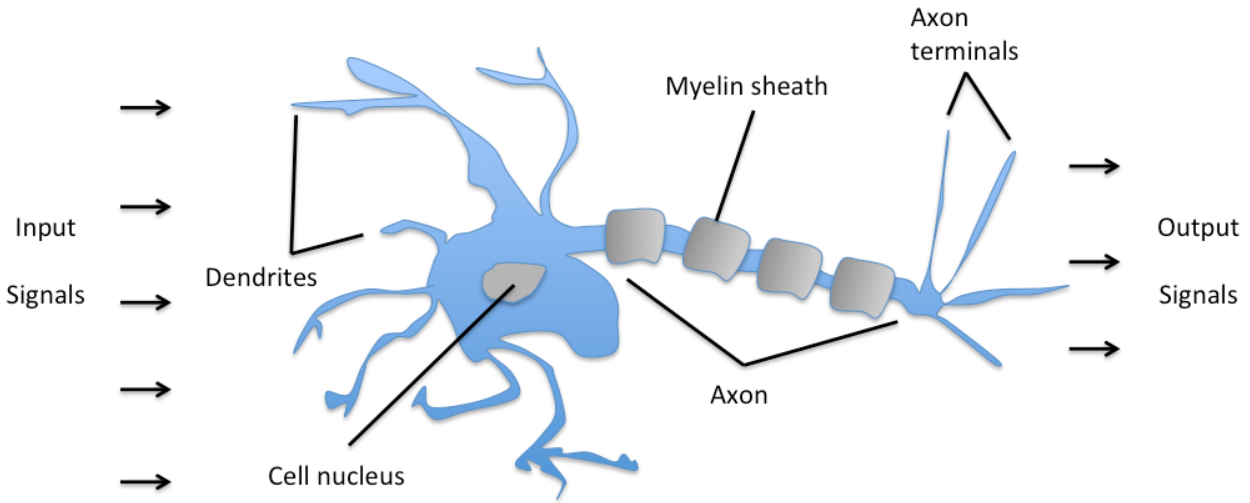
$n = 7500$ (color, RGB)

$$\vec{x} = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2 intensity} \\ \vdots \\ \text{pixel 2500 intensity} \end{bmatrix}$$

Quadratic features $(x_i \times x_j)$: 3 million features

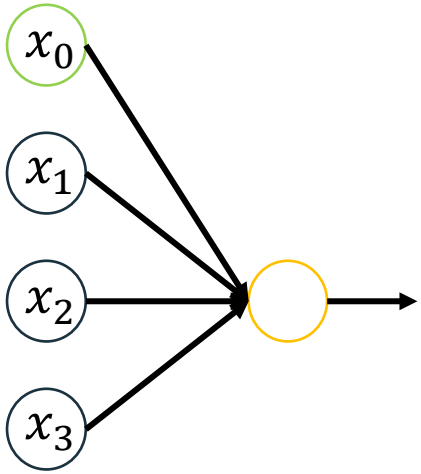
Neurons in the brain





Neuron model: logistic unit

bias unit
$x_0 = 1$



$h_{\vec{\theta}}(\vec{x})$ output

$$h_{\vec{\theta}}(\vec{x}) = \frac{1}{1 + e^{-\vec{x}^T\vec{\theta}}}$$

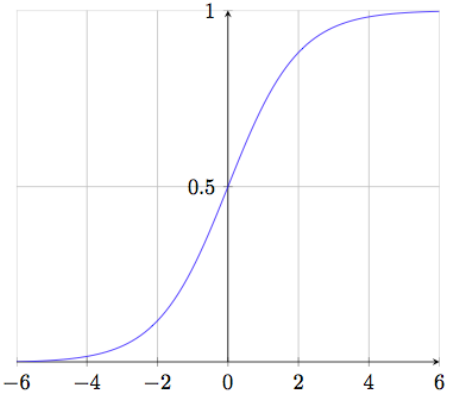
$$\vec{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \vec{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

input features

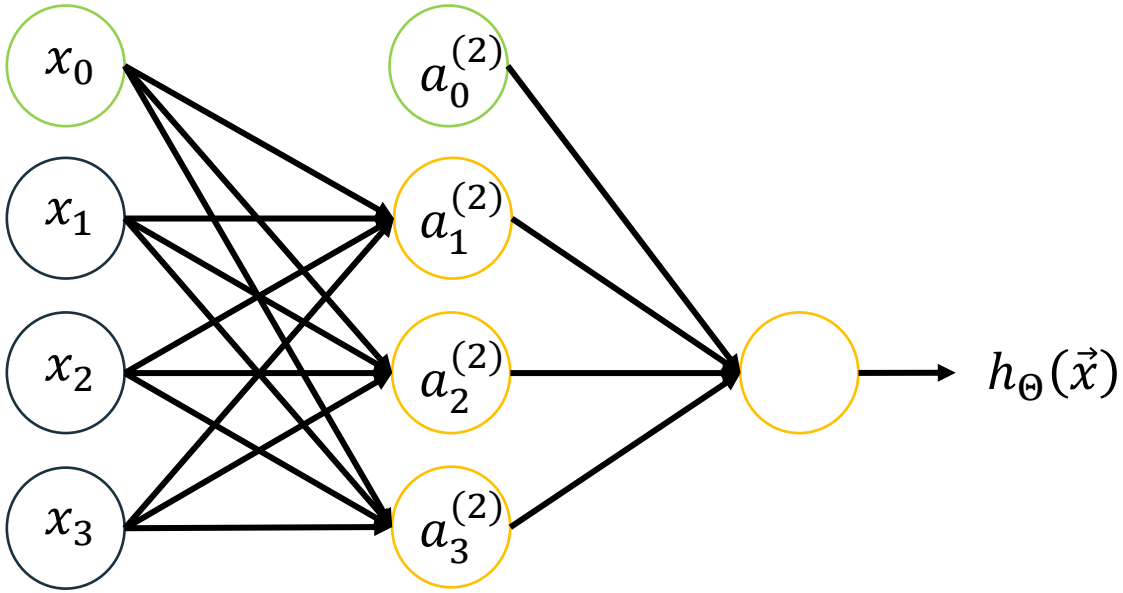
weights
parameters

Sigmoid (logistic) activation function

$$g(z) = \frac{1}{1 + e^{-z}}$$



Neural network



Layer 1
Input layer

Layer 2
Hidden layer

Layer 3
Output layer

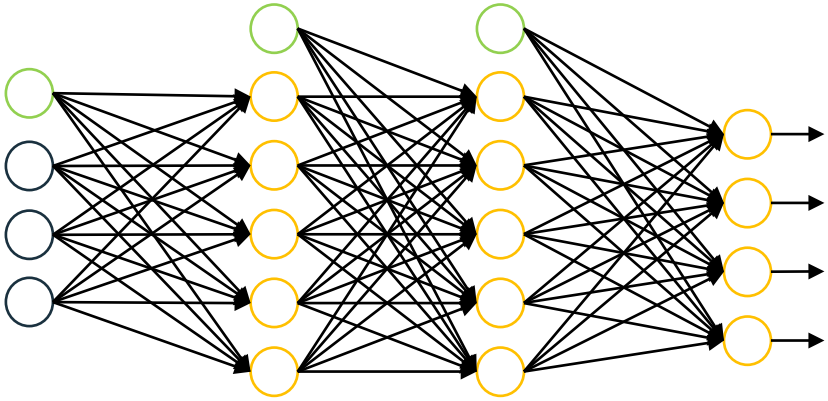
Multiple output units


Yacht


Dolphin


Tree


Airplane



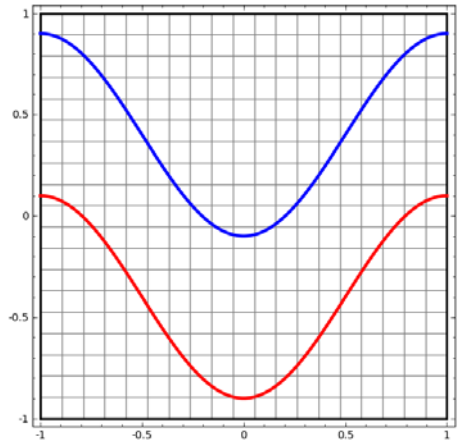
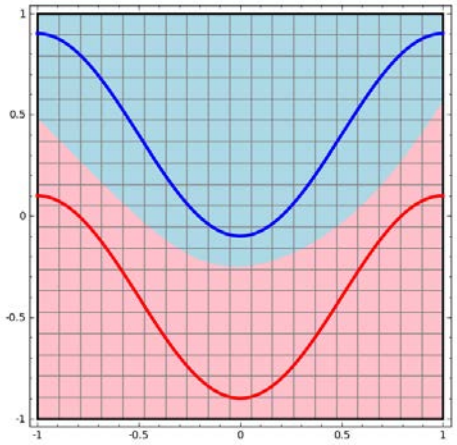
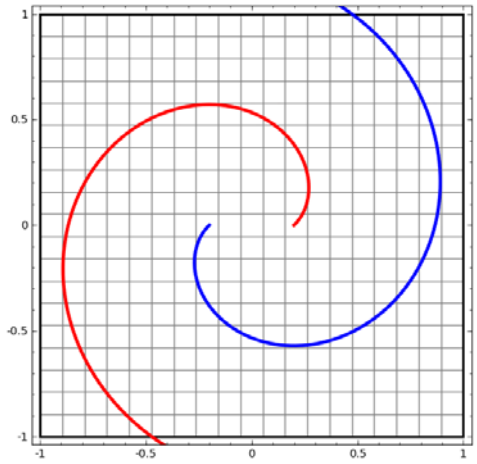
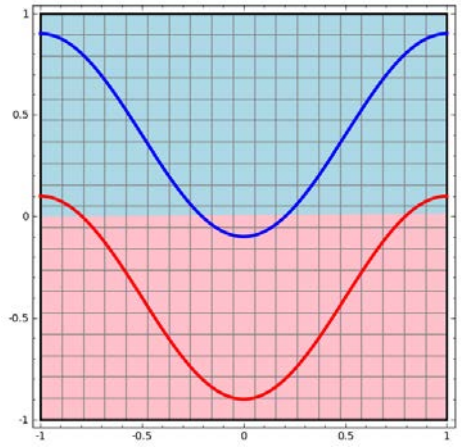
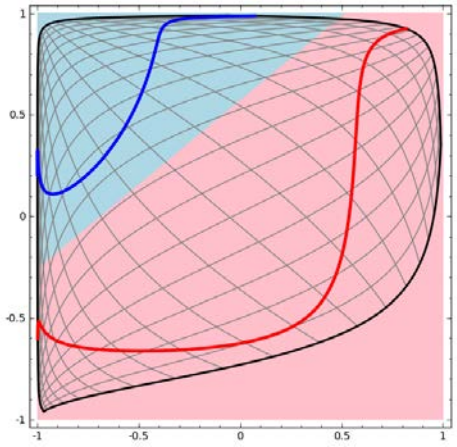
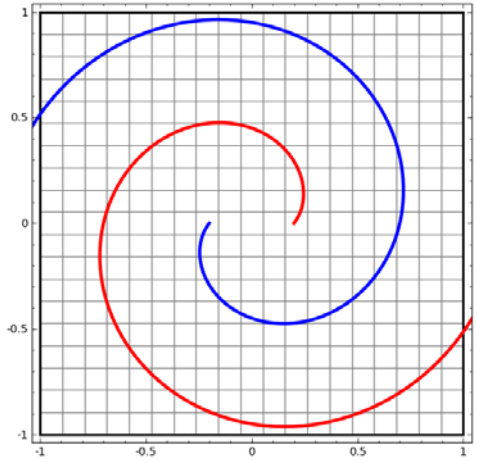
$h_\Theta(\vec{x}) \in \mathbb{R}^4$

$$h_\Theta(\vec{x}) = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \quad h_\Theta(\vec{x}) = \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \quad h_\Theta(\vec{x}) = \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \quad h_\Theta(\vec{x}) = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

Training set: $(\vec{x}^{(i)}, \vec{y}^{(i)}) \quad i = 1, \cdots, m$

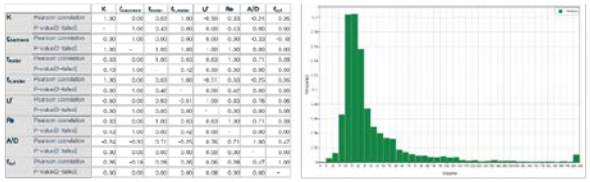
$\vec{y}^{(i)} \in \mathbb{R}^4$

Continuous Visualisation of Layers

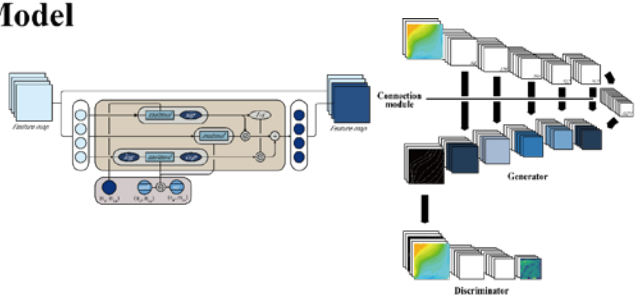







❖ Ship Design & Production Laboratory (SDPL)

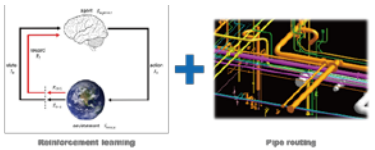
- Studies on fast and accurate initial design and productivity and economical ship design technology through ship design and production computerization.
- Research is being conducted to apply various kinds of design techniques to ship design, such as optimal design, rover design, artificial intelligence design, etc. that add convenience and effectiveness of design.



Development of Design Optimization Model through Component Analysis



Automate Production through Image Processing



Reinforcement Learning

Project

❖ Government support projects

No.	연구 제목	연구 내용	연구 기간	역할	연구개발비 지급 기관
1	3D 모델기반의 해양플랜트 의장설계 시스템 개발	• 3D 모델기반 강화학습을 이용한 배관 자동 설계 알고리즘 개발	2017.04.~ 2019.12.	연구책임자	산업통상 자원부
2	인공지능 기반 조선해양 용접품질 자동검사 및 통합관리 플랫폼 개발	• 인공지능 기법인 CNN을 이용하여 용접검사 이미지의 결함 검출 알고리즘 개발	2018.04.~ 2020.12.		
3	친환경·스마트선박 R&D전문인력양성사업	• 친환경, 스마트선박 개발을 위한 R&D 전문인력양성	2018.03.~ 2023.02.		
4	자율운항선박 기술개발	• 자율운항시스템 원격관리를 위한 인공지능 기반 선박손상 및 선원 관리 시스템 개발	2020.04.~ 2024.12.		해양수산부
5		• 자율운항선박 화재, 침수 피해 최소화를 위한 시스템 신뢰성 평가 및 사고대응 기술 개발	2021.04.~ 2025.12.		

❖ Joint industry projects

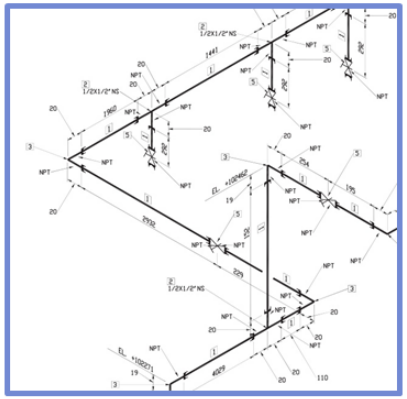
No.	연구 제목	연구 내용	연구 기간	역할	연구개발비 지급 기관
1	Deep learning 기반 선박후류 유동장 분석을 통한 최적 선미형상 도출에 관한 연구(I)	• 선미 유동장 해석결과로부터 선박 선미형상을 도출하는 인공지능 신경망 개발	2019.02.~ 2019.10.	연구책임자	㈜삼성중공업
2	CSA프로시져를 고려한 운동해석 시스템 개발	• 운동 및 구조해석을 위한 전선모델 mass tuning (Shear force, Bending moment)	2020.06.~ 2020.11.		
3	선박용 HVAC capacity 견적이 가능한 인공지능 알고리즘 개발	• DSEC에서 축적된 HVAC Capacity 견적 자료를 활용하여 초기에 HVAC Capacity 추정이 가능한 인공지능 모델 개발	2020.06.~ 2021.05.		㈜DSEC

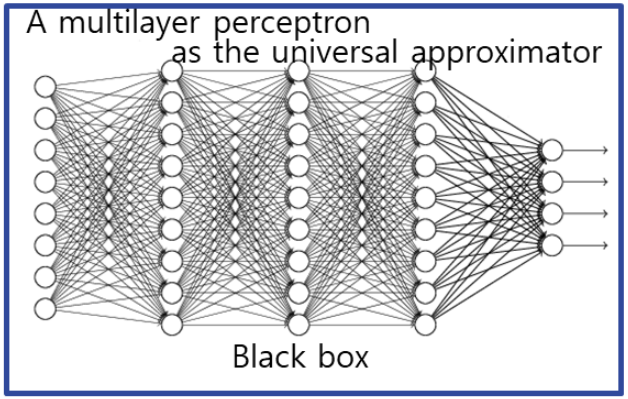
Development of Design Optimization Model through Component Analysis

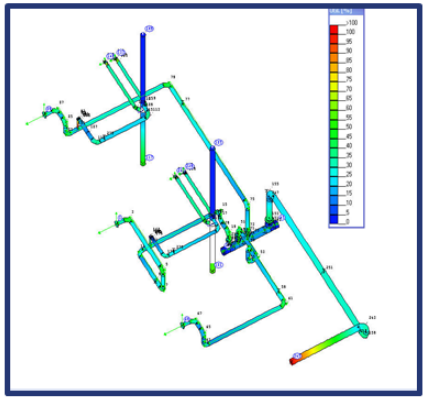


Deep Neural Networks for Stress Prediction in Pipe Design

- To achieve efficiency, the required maximum stress value should be considered in the initial step itself
- Comparing the maximum stress estimated through the DNN model and those measured by actual analysis







01. Feature extraction

- ✓ pipe drawing setting
- ✓ Analysis of stress analysis factor

02. DNN learning

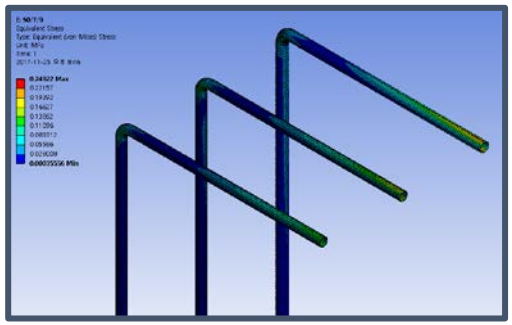
- ✓ Deep Neural network
- ✓ ReLU function, normalisation, Xavier initialization

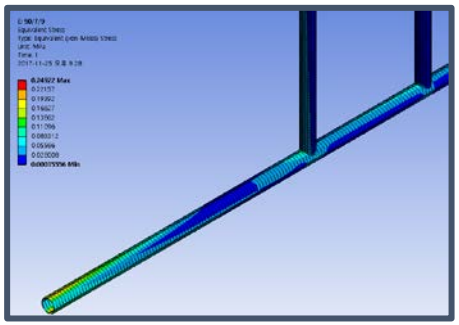
03. Stress estimation

- ✓ Prediction of maximum stress for a new pipe design
- ✓ Validation

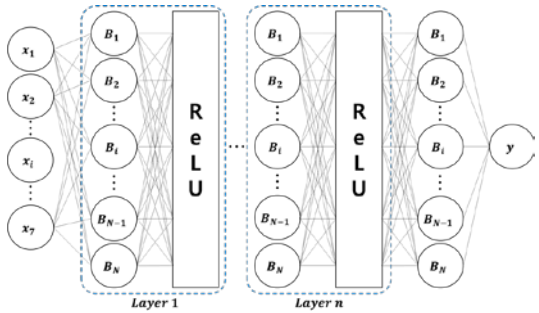
Deep neural network stress prediction process

Deep Neural Networks for Stress Prediction in Pipe Design

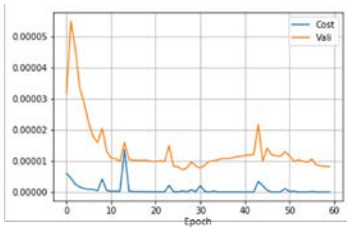




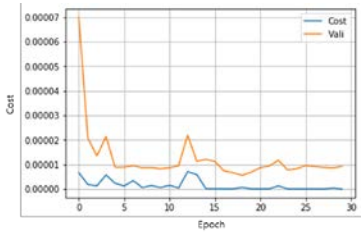
Data generation and extraction



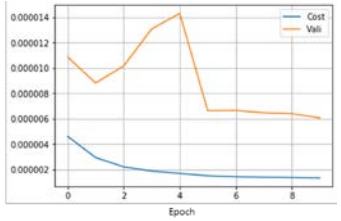
Learning algorithm



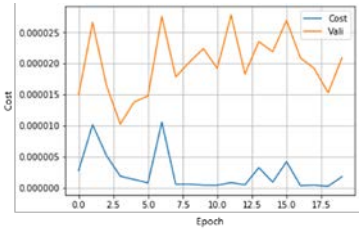
Case 1 loss



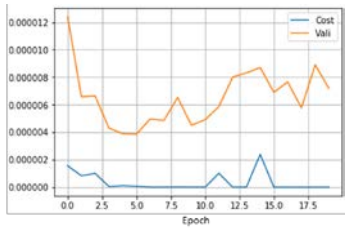
Case 2 loss



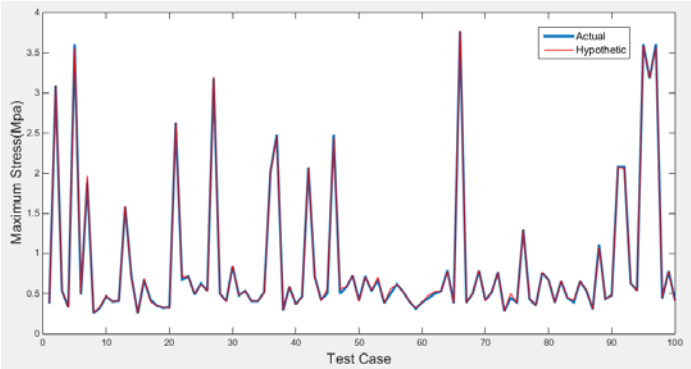
Case 3 loss



Case 4 loss



Case 5 loss



Neural network learning result comparison graph confirmation case 5

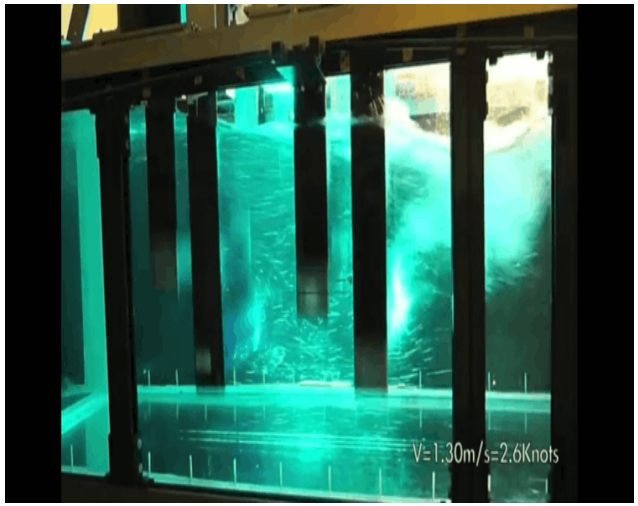
Average error values of training, validation, and test sets

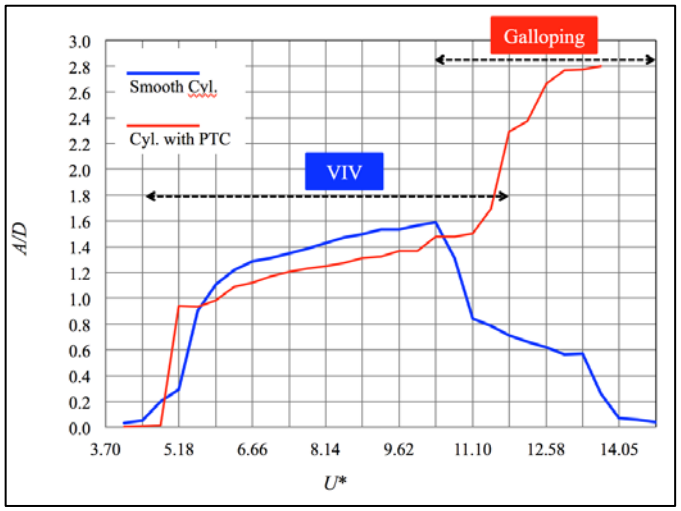
Set	MLR	SVM	Case 1	Case 2	Case 3	Case 4	Case 5
Training	58.51%	13.43%	27.26%	16.46%	1.14%	0.77%	0.67%
Validation			23.75%	20.35%	2.35%	1.08%	1.29%
Test	68.81%	16.08%	20.53%	18.94%	2.77%	1.32%	1.12%

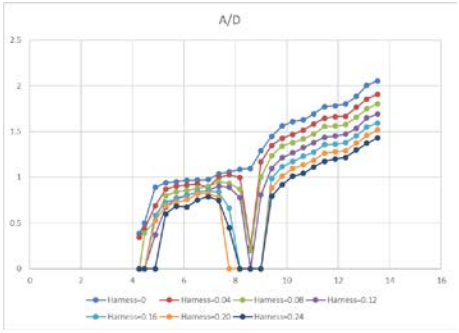
Prediction of dynamic responses of Flow Induced Vibration Using deep learning

❖ *Passive Turbulence Control, PTC*

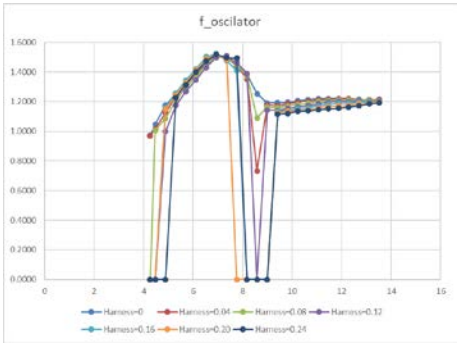
- Estimate the power curve in the initial design phase of the system to select the optimized spring constant and harness damping ratio according to flow rate







Amplitude/Diameter



Oscillator frequency

Correlation Analysis

❖ *Selection of input variables using correlation analysis*

- Generating datasets according to Harness damping ratio and Spring constants
- Selection of input variables through Pearson correlation analysis

		K	$\zeta_{harness}$	f_{motor}	$f_{n,water}$	U*	Re	A/D	f_{cyl}
K	Pearson correlation	1.00	0.00	0.03	1.00	−0.50	0.03	−0.24	0.26
	P-value(2-tailed)	–	1.00	0.43	0.00	0.00	0.43	0.00	0.00
$\zeta_{harness}$	Pearson correlation	0.00	1.00	0.00	0.00	0.00	0.00	−0.33	−0.18
	P-value(2-tailed)	1.00	–	1.00	1.00	1.00	1.00	0.00	0.00
f_{motor}	Pearson correlation	0.03	0.00	1.00	0.03	0.83	1.00	0.71	0.28
	P-value(2-tailed)	0.43	1.00	–	0.42	0.00	0.00	0.00	0.00
$f_{n,water}$	Pearson correlation	1.00	0.00	0.03	1.00	−0.51	0.03	−0.25	0.26
	P-value(2-tailed)	0.00	1.00	0.42	–	0.00	0.42	0.00	0.00
U*	Pearson correlation	−0.50	0.00	0.83	−0.51	1.00	0.83	0.76	0.06
	P-value(2-tailed)	0.00	1.00	0.00	0.00	–	0.00	0.00	0.08
Re	Pearson correlation	0.03	0.00	1.00	0.03	0.83	1.00	0.71	0.28
	P-value(2-tailed)	0.43	1.00	0.00	0.42	0.00	–	0.00	0.00
A/D	Pearson correlation	−0.24	−0.33	0.71	−0.25	0.76	0.71	1.00	0.47
	P-value(2-tailed)	0.00	0.00	0.00	0.00	0.00	0.00	–	0.00
f_{cyl}	Pearson correlation	0.26	−0.18	0.28	0.26	0.06	0.28	0.47	1.00
	P-value(2-tailed)	0.00	0.00	0.00	0.00	0.08	0.00	0.00	–

Input

- Motor frequency (Hz)
- Spring Constants (K)
- Reynolds number
- Natural frequency (Hz)
- Harness damping ratio ($\zeta_{harness}$)

Output

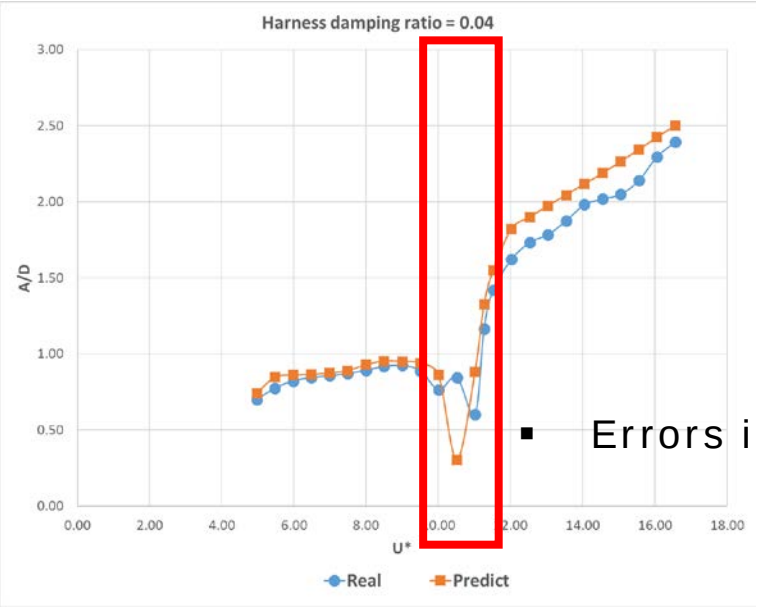
- A/D
- $f_{oscillator}$

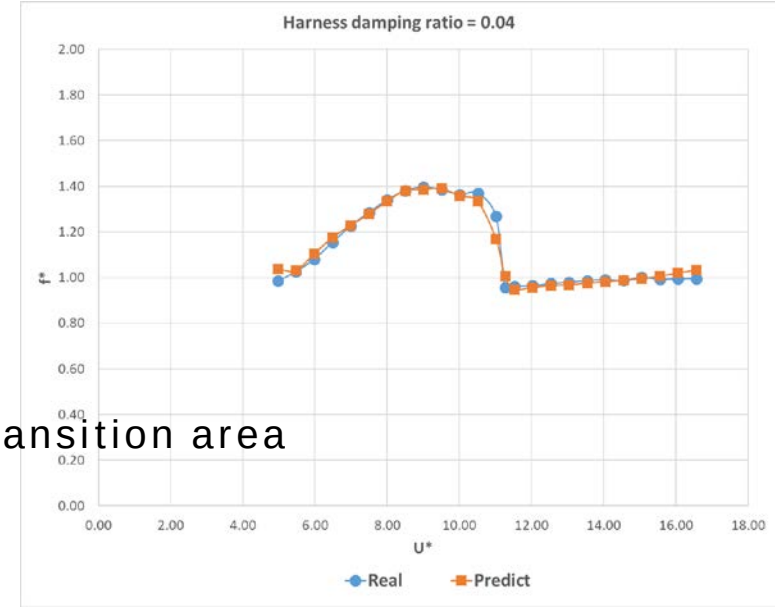
Results(1/2)

- **K= 400 N/m**

$$U^* = \frac{U}{f_{n,water} * D} \qquad f_{n,water} = \frac{1}{2\pi}\sqrt{\frac{k}{m_{osc} + m_a}}$$

- Error in transition area changing from VIV to Galloping





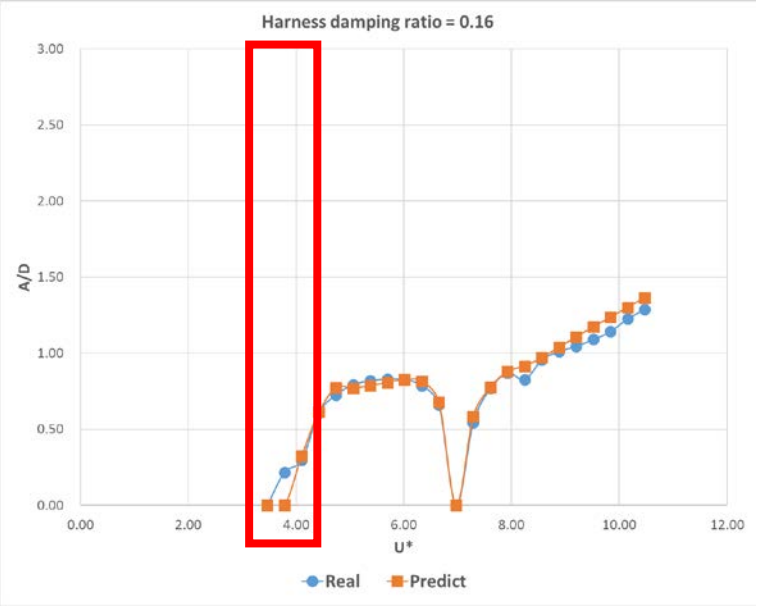
- Errors in transition area

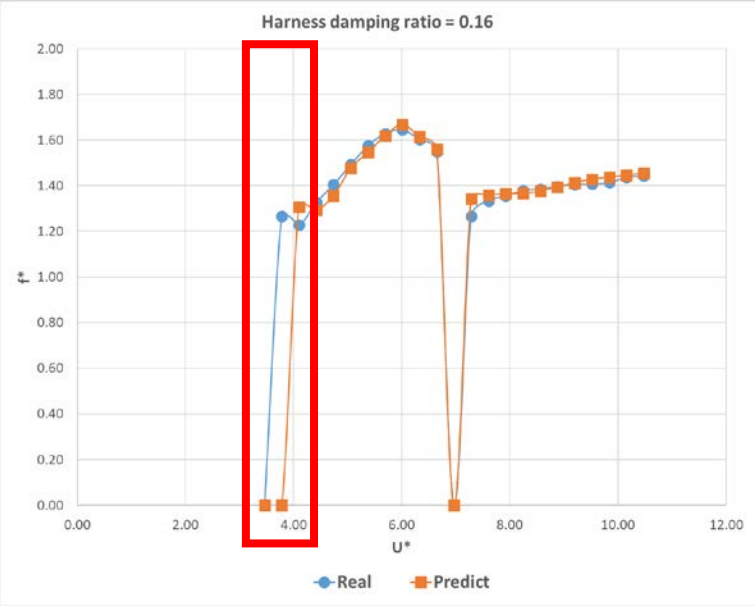
U*		4.97	5.48	5.98	6.48	6.99	7.49	8.00	8.50	9.00	9.51	10.01	10.52	11.02	11.27	11.52	12.03	12.53	13.03	13.54	14.04	14.55	15.05	15.55	16.06	16.56	Average
A/D error	(%)	5.66	9.28	4.56	2.33	2.04	2.04	4.09	3.81	2.77	5.50	12.66	64.23	46.07	13.70	8.99	12.35	9.58	10.58	9.08	6.79	8.33	10.48	9.51	5.49	4.58	10.98
f* error	(%)	5.31	0.45	2.12	1.75	0.22	0.59	0.41	0.08	0.81	0.32	0.49	2.65	7.79	5.23	1.47	0.73	1.00	1.17	1.21	0.94	0.02	0.56	1.33	2.43	3.74	1.71

Results(2/2)

❖ K= 1,000 N/m

- Error in initial nonlinear interval for crossing thresholds for VIV to occur



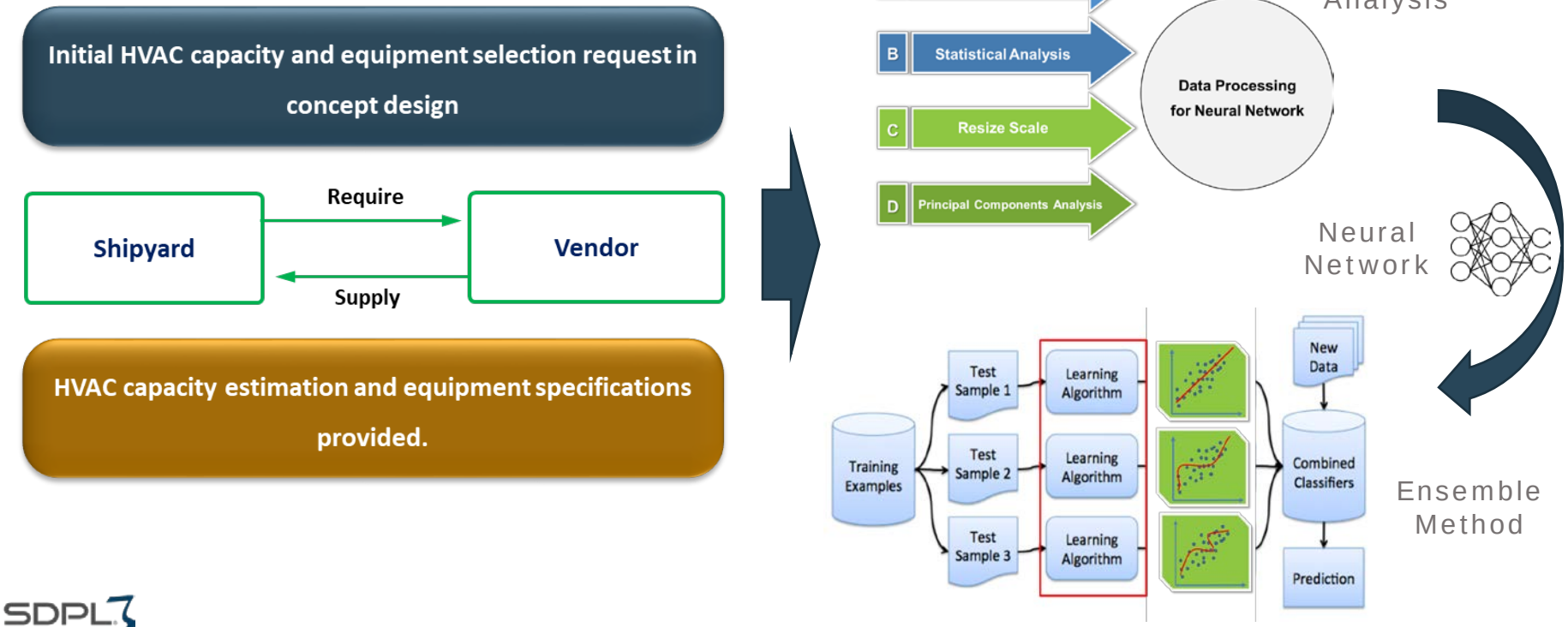


U*		3.46	3.78	4.1	4.42	4.74	5.06	5.38	5.69	6.01	6.33	6.65	6.97	7.29	7.61	7.93	8.24	8.56	8.88	9.2	9.52	9.84	10.16	10.48	Average
A/D error	(%)	0	100	9.61	0.83	6.39	2.89	4.04	2.73	0.25	3.64	2.68	0	7.17	0.89	1.09	10.56	1.47	2.66	5.83	7.48	8.28	5.97	6.08	9.07
f* error	(%)	0	100	6.46	2.51	3.47	1.07	1.86	0.61	1.4	0.69	0.72	0	5.92	1.74	0.64	0.88	0.71	0.18	0.49	1.51	1.64	0.63	0.83	6.38

Development of AI Algorithm Capacity of Estimating HVAC Capacity for ships

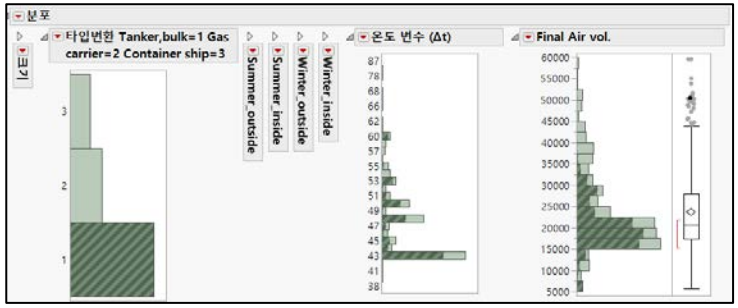
❖ Work process

- Development of AI Algorithms to Estimate HVAC Capacity in the Concept design
- Compare results of multiple deep learning models

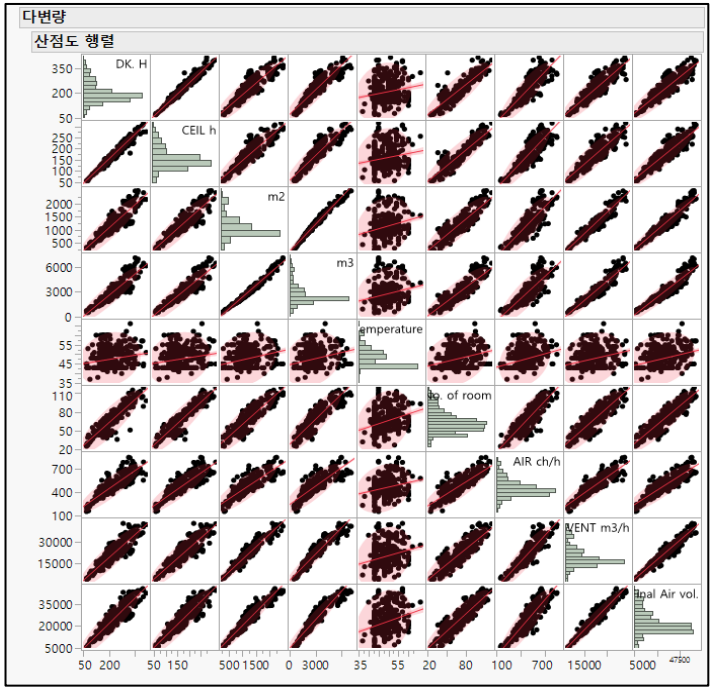


Data analysis & DNN training

❖ Data Analysis Using JMP

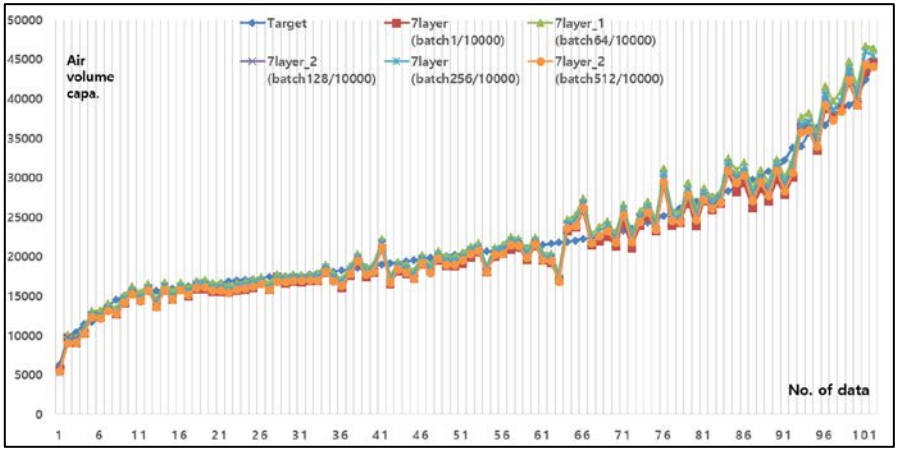


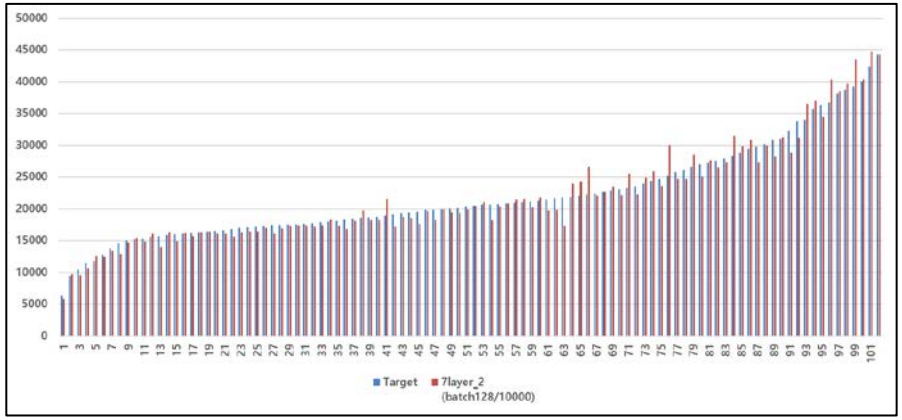
Distribution of data by ship



Correlation analysis

❖ Results





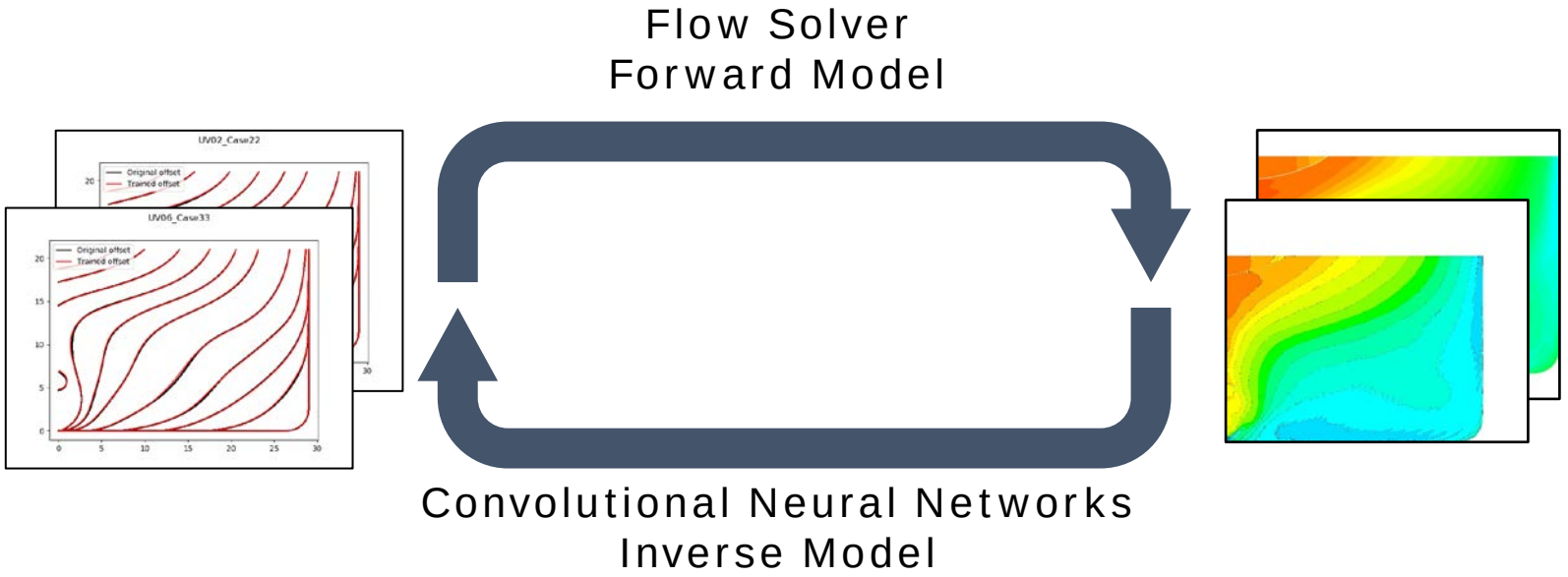
- Selection of neural network features through statistical and correlation analysis
- Comparison of three neural network models

Automate Production through Image Processing



Inverse Design

❖ *Image to shape, inverse design using CNN*

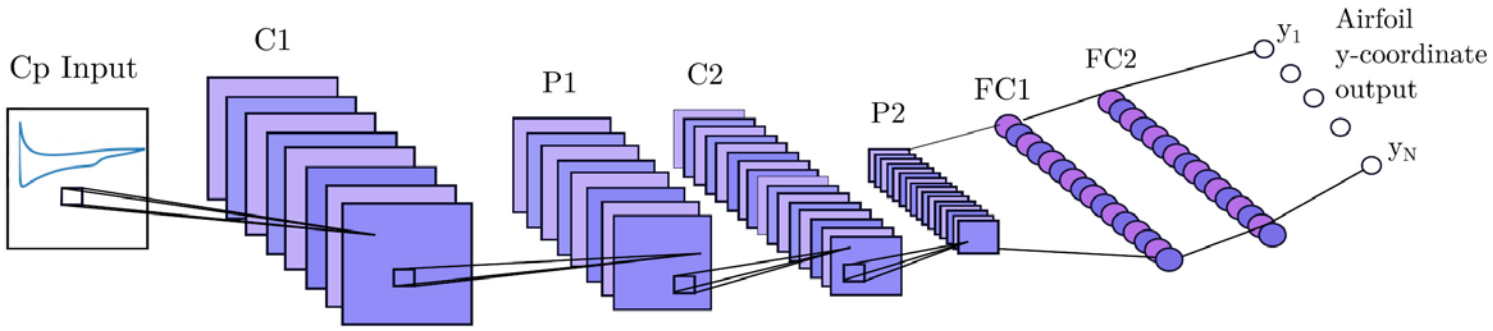


- Inverse design methodology has been studied to reduce the iteration number because both CFD and experiment for a hull form are expensive
- A novel approach is proposed for the inverse design of hull form using convolutional neural networks

Verify the Shape of Lines

❖ *Inverse Design of Airfoil Using a Deep Convolutional Neural Network*

- Obtain the airfoil shape from the coefficient of pressure distribution
- Airfoil y-coordinate output: 70 → Airfoil x, y-coordinate output: 140
- Total number of data: 1,200 → Total number of data: 160

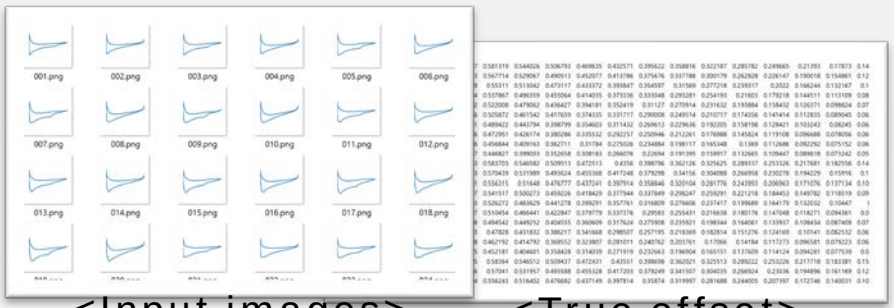


Convolutional Neural Network for Airfoil

Sekar, V., Zhang, M., Shu, C., & Khoo, B. C. (2019). Inverse Design of Airfoil Using a Deep Convolutional Neural Network. *AIAA Journal*, *57*(3), 993-1003.

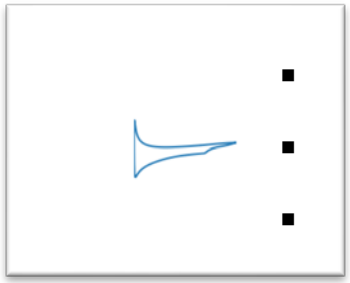
Data Preparation

❖ Training / Test data

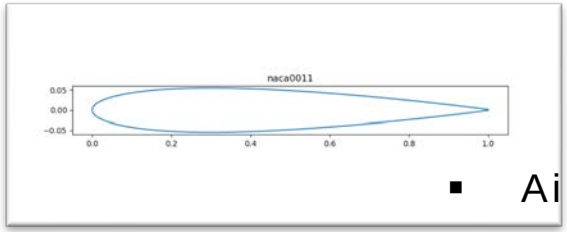


<Input images> <True offset>

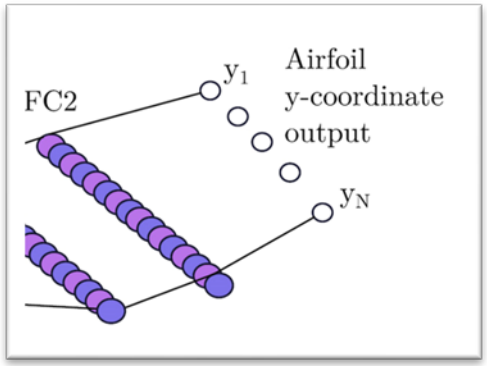
- Dataset: 160
- Training set: 128(80%)
- Test set: 32(20%)



- Cp curve-NACA
- Image size: 144×144
- Image channel: 1



- Airfoil Offset
- X offset: 70
- Y offset: 70

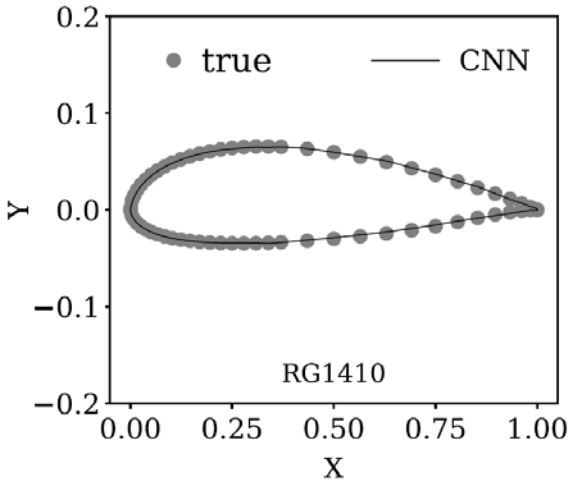


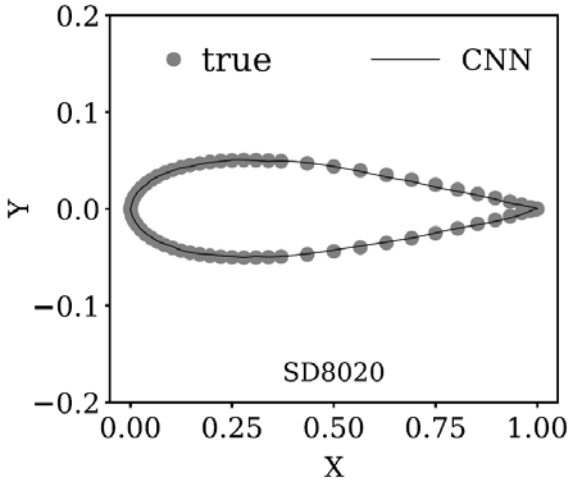
Paper-Airfoil y coordinate output: 70

SDPL-Airfoil x, y coordinate output: 140

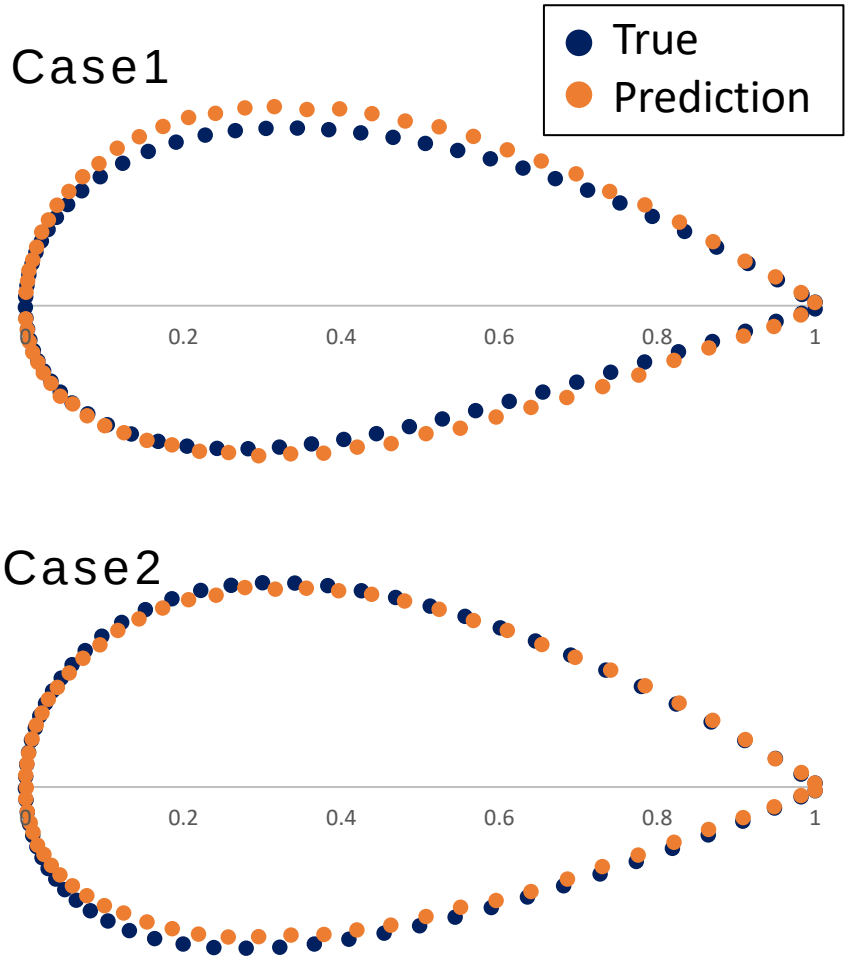
Results

❖ *Prediction of typical testing airfoils*





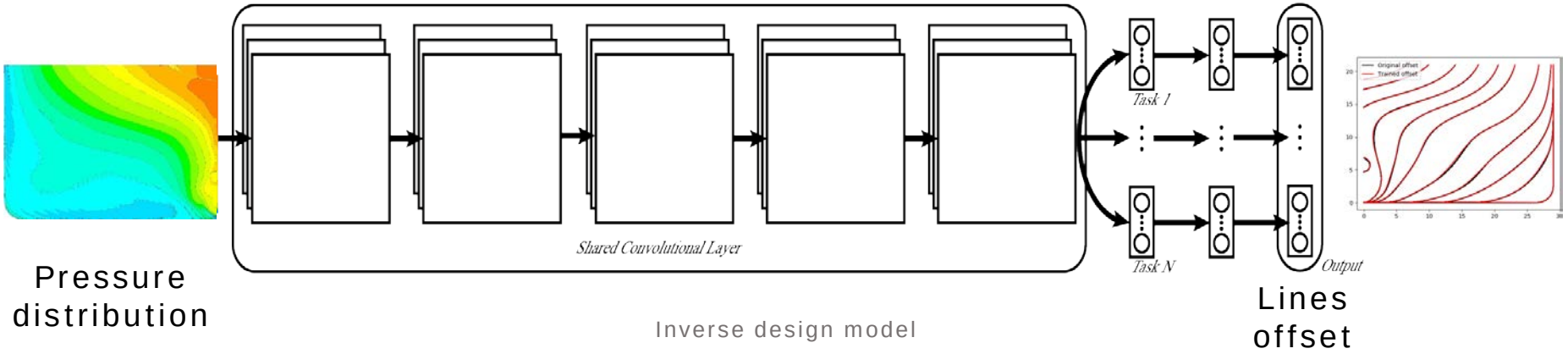
Airfoil y-coordinate output: 70



Airfoil x, y-coordinate output: 140

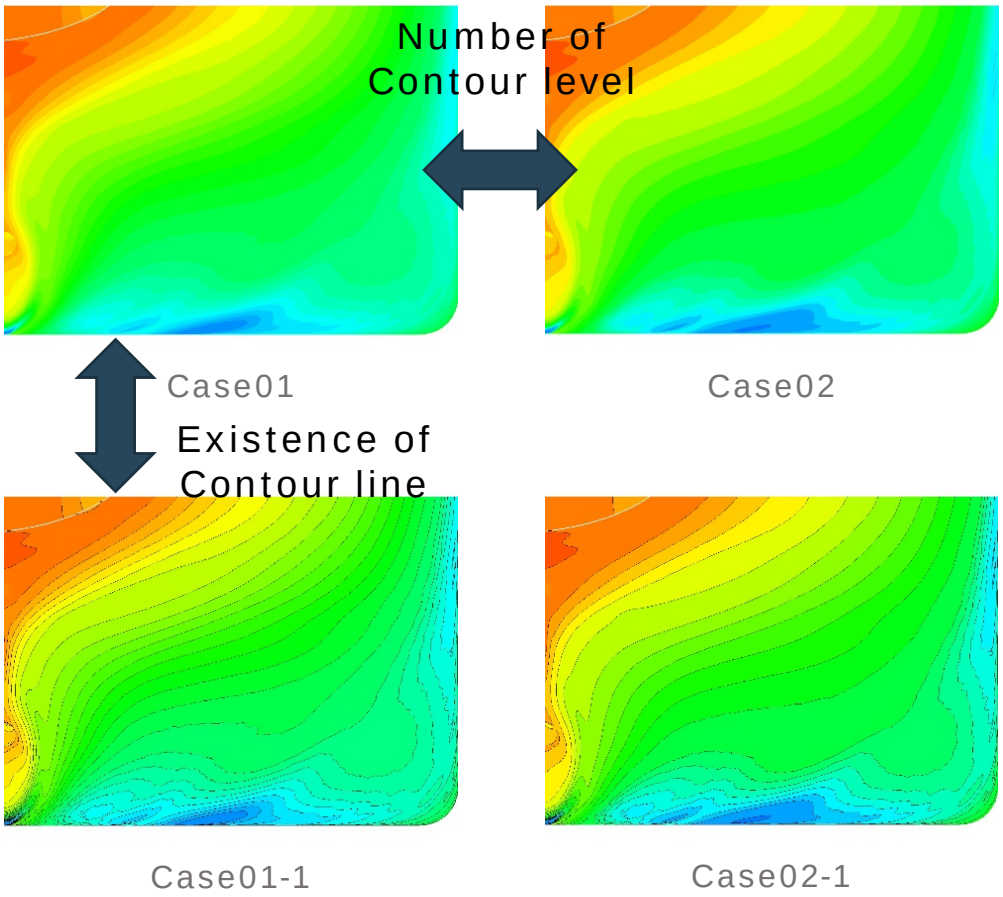
Inverse Design of Stern Hull Form

❖ Concept



- To obtain the stern shape from the pressure distribution of stern
- Generate 1,002 hull forms from a KRISO VLCC (KVLCC)
- Input is the image of pressure distribution on the stern hull surface
- Output is the 14 section curves as stern hull form

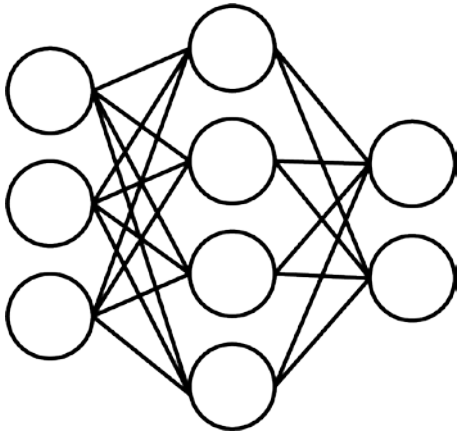
Input Data



- Training data: 80% (800 images)
- Test data: 20% (202 images)

- Contour level: Case01 – 35
  Case02 – 25
- Analysis using STAR-CCM+

Preprocessing

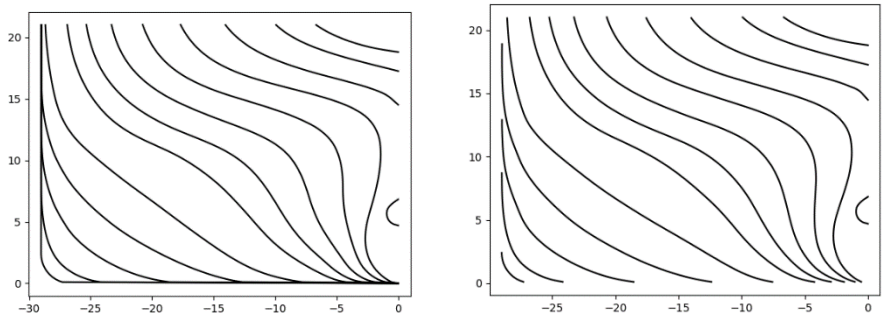


Neural network

- Undetermined number of offsets
- The weight and nodes of the neural network are fixed
- Requires a lot of training data as nodes increase

1. Selecting a reasonable number of outputs from limited data
2. Selecting the number of outputs of a constant size

❖ *Remove the straight part*

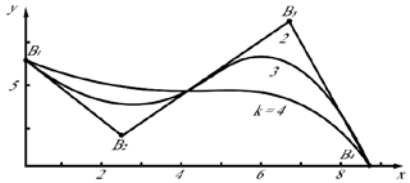


Before & after of stern shape

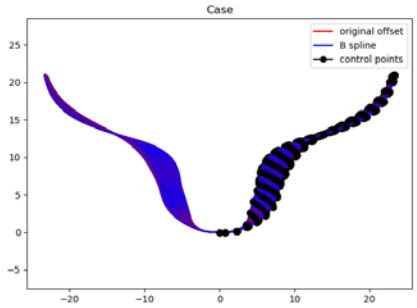
- Remove unnecessary labels
- Effective for B-spline process
- Postprocessing improves accuracy

Dataset

- Obtain the stern shape from the pressure distribution of stern
- Generate 780 data using KRISO VLCC(KVLCC)



Open, Uniform B-spline



Control point

B-spline basis curve

$$S(t) = \sum_{i=0}^{m-n-2} P_i b_{i,n}(t), \mathrm{t} \in [t_n, t_{m-n-1}] \qquad t_0 \le t_1 \le \cdots \le t_{m-1}$$

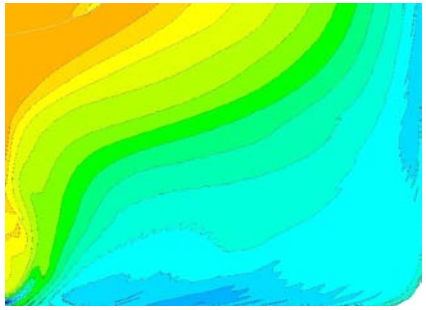
B-spline basis function

$$b_{j,0}(t) := \begin{cases} 1 & if\ t_j \le t < t_{j+1} \\ 0 & otherwise \end{cases}, j = 0, \cdots, m-2$$

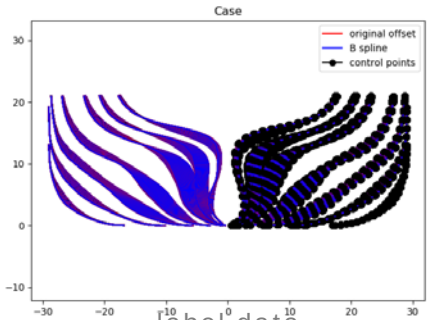
$$b_{j,n}(t) := \frac{t - t_j}{t_{j+n} - t_j} b_{j,n-1}(t) + \frac{t_{j+n+1} - t}{t_{j+n+1} - t_{j+1}} b_{j+1,n-1}(t), \qquad j = 0, \cdots, m-n-2$$

Control point

$$P = S(t)b^T$$



Input data

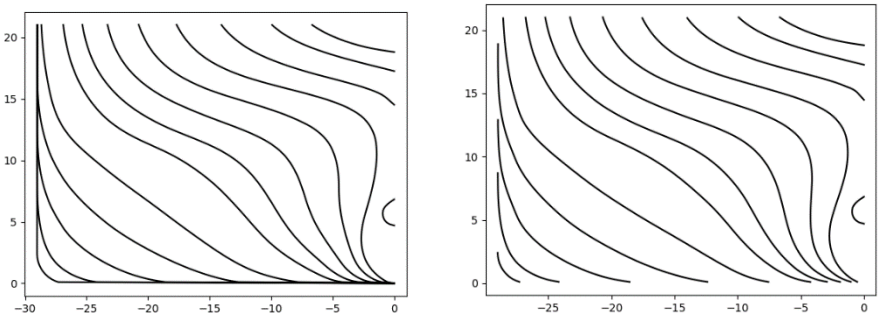


label data

- The number of control point: 24
- The number of section: 7
- Hull form transform: 11x10x7
- Hull form baseline: 10

Output Data

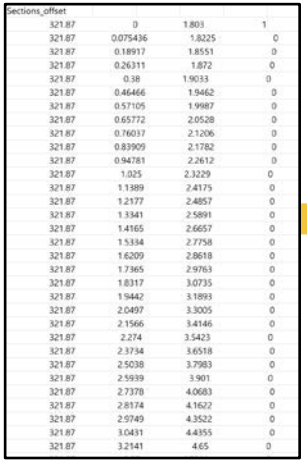
❖ Remove the straight part



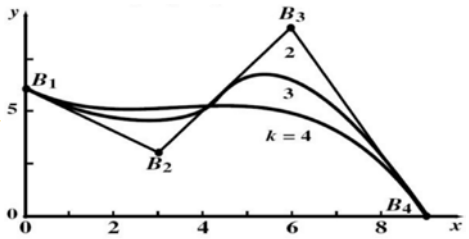
Before & after of stern shape

- Remove unnecessary part of line
- B-spline control points
- Postprocessing improves accuracy

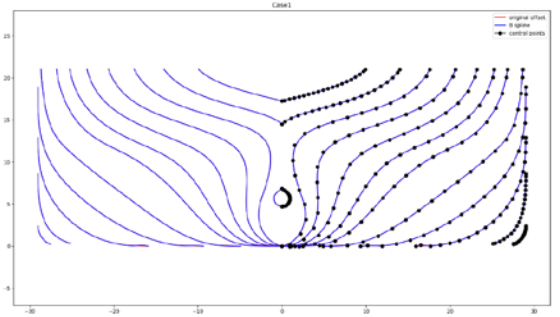
❖ Fixed to size using B-spline



Stern offset



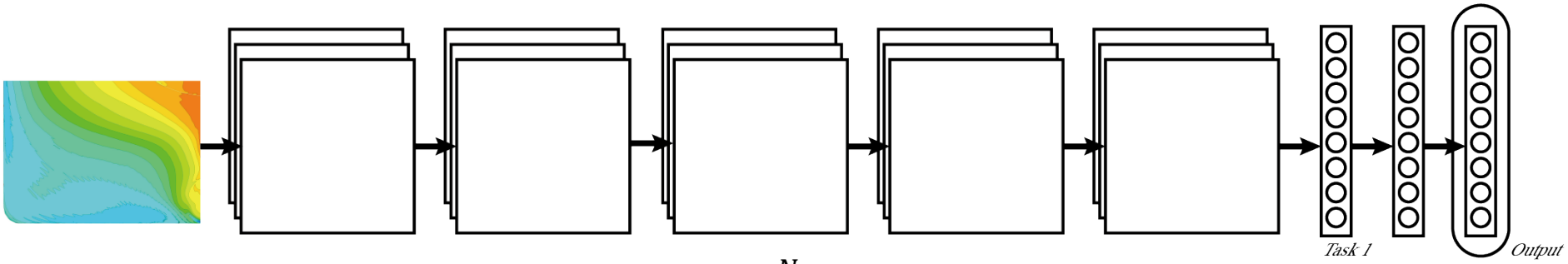
Open, Uniform B-spline



Control point: 23 point in each section

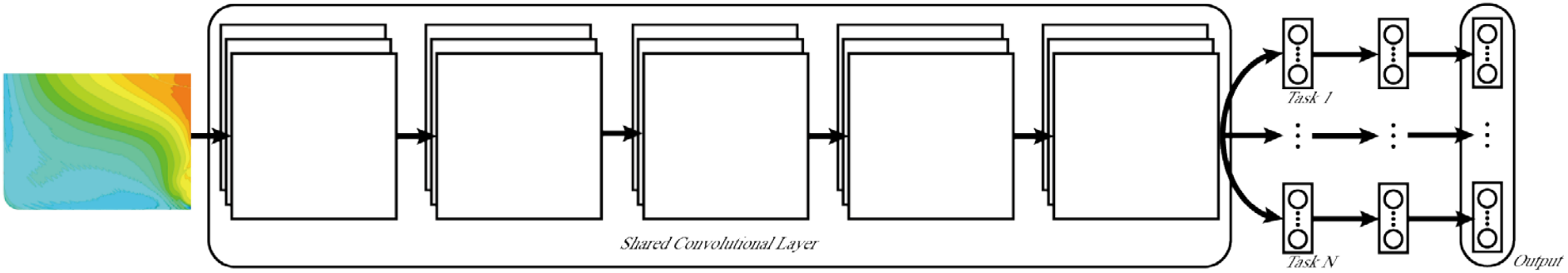
Single-task / Multi-task Learning Model

❖ Single-task learning model



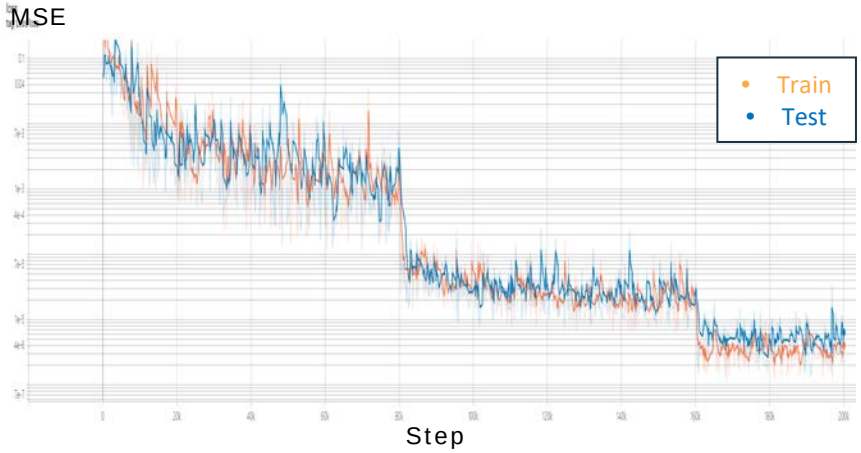
$$l_{single} = \frac{1}{N_{t_cp}} \sum_{i=1}^{N_{t_cp}} (y_i - \hat{y}_i)^2$$

❖ Multi-task learning model



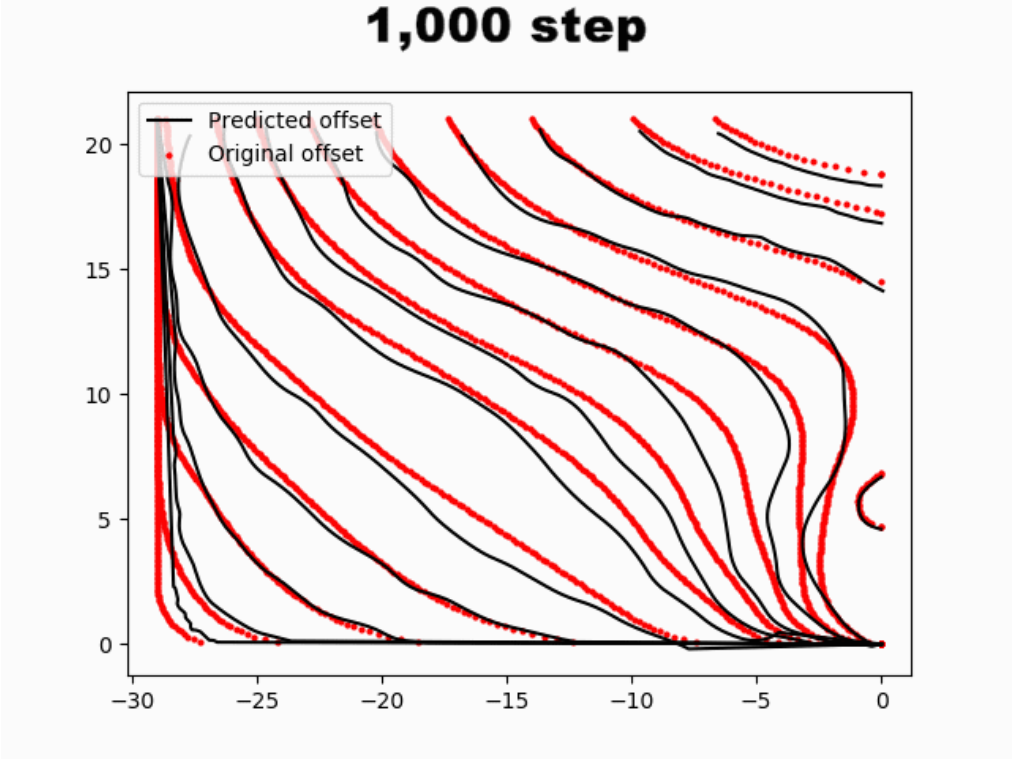
$$l_{total} = l_{multi} = \frac{1}{N_t N_{s_cp}} \sum_{j=1}^{N_s} \sum_{i=1}^{N_{s_cp}} (y_{ji} - \hat{y}_{ji})^2$$

Training Process



Training step	Training loss	Test loss
1,0000	0.0038480	0.0047421
100,000	0.0000239	0.0000245
170,000	0.0000298	0.0000502
200,000	0.0000054	0.0000074

Loss with training step



Original offset vs. Predicted offset

Results

❖ Control point / Offset

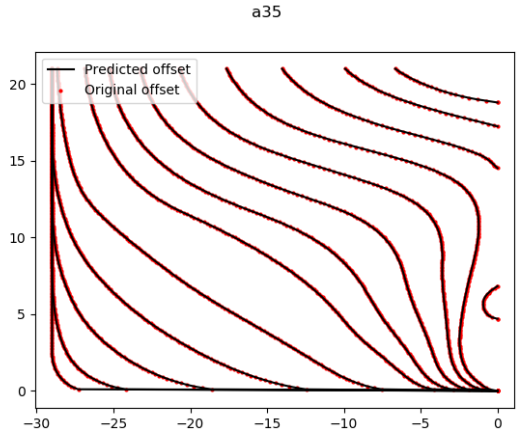
- Evaluation by dividing a certain interval along the z axis
- Root mean square error (RMSE) as evaluation criteria

	Case01	Case01-1	Case02				Case02-1
			Task (single)	Task (conv0 fc3)	Task (conv4 fc3)	Task (conv8 fc3)	
RMSE (mm)	2.227	2.324	3.840	2.000	2.057	1.684	6.560

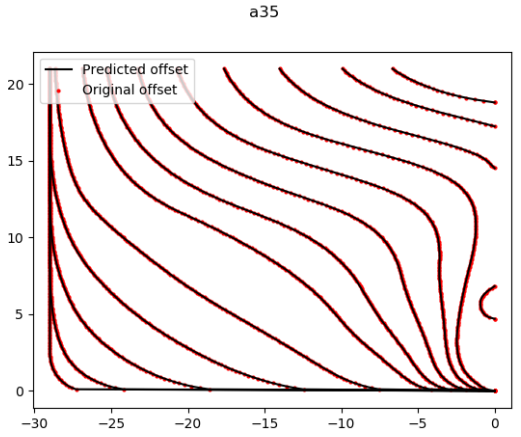
	Case01	Case01-1	Case02				Case02-1
			Task (single)	Task (conv0 fc3)	Task (conv4 fc3)	Task (conv8 fc3)	
RMSE (mm)	4.417	4.694	6.107	4.183	4.379	3.889	12.874
Test time (sec/img.)	0.019	0.019	0.019	0.019	0.047	0.085	0.019

Results

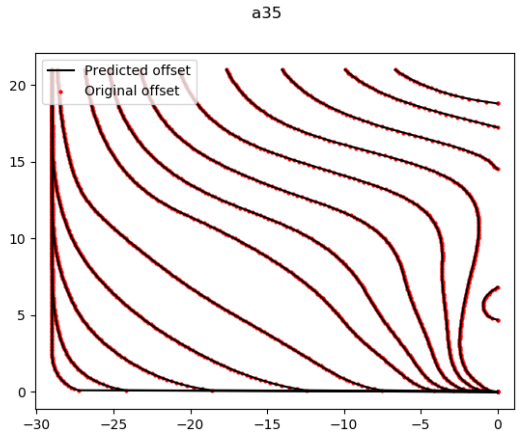
❖ *Single / Multi-task learning model*



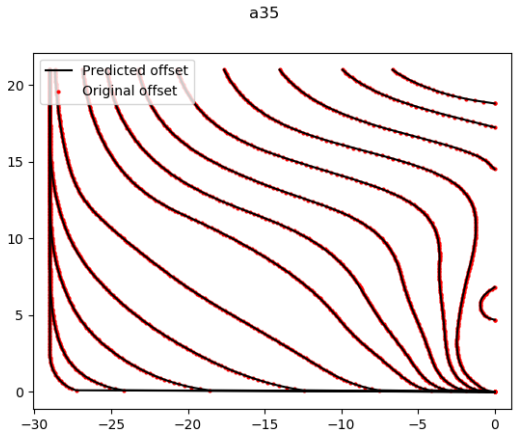
Single-task learning model



Multi-task learning model(Task3)



Multi-task learning model(Task7)



Multi-task learning model(Task11)

Limits of regression model

- Fixed control point of B-spline: 24 points per section
- Error in the process of converting to offset

	Case01	Case01-1	Case02				Case02-1
			Task (single)	Task (conv0 fc3)	Task (conv4 fc3)	Task (conv8 fc3)	
RMSE(Cp)	0.0022	0.0023	0.0038	0.0020	0.0021	0.0017	0.0066
RMSE(Os)	0.0044	0.0047	0.0061	0.0042	0.0044	0.0039	0.0129
Error	+0.0022	+0.0024	+0.0023	+0.0022	+0.0023	+0.0022	+0.0063

Root mean square error according to image case

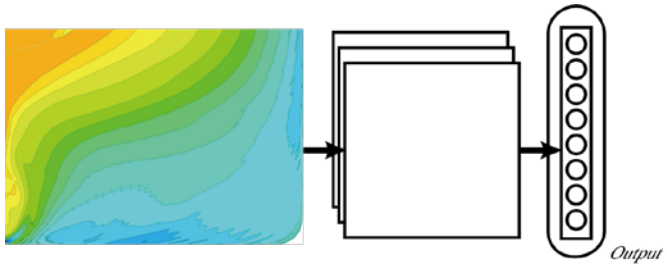


Image to vector

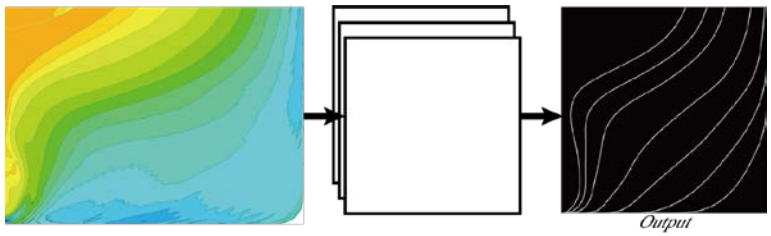


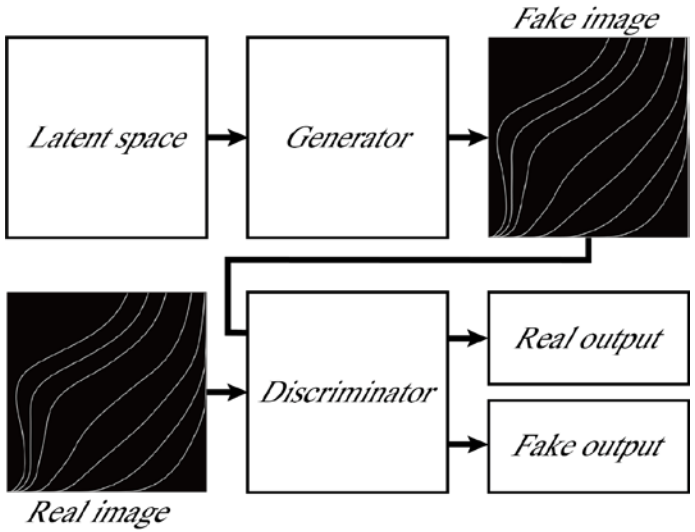
Image to image

❖ Purpose of Image to image

- Minimize restrictions on output form
- Minimize transform error
- Flexibility

  - the pressure distribution of stern
- Improves accuracy for extrapolation

Generative Adversarial Networks



GAN basic concept

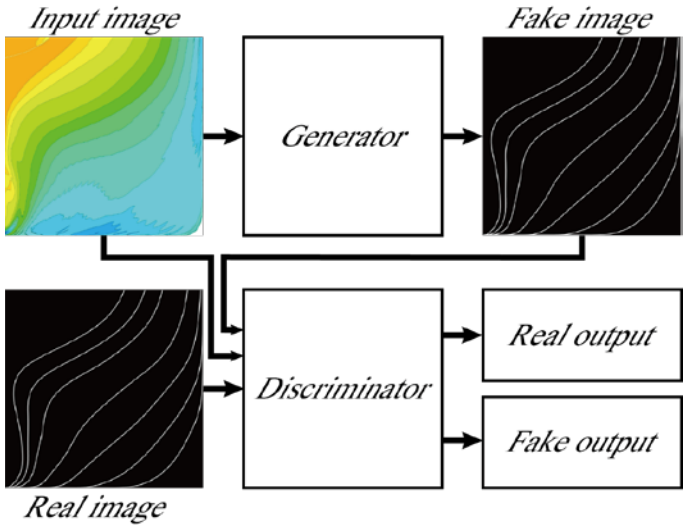
❖ Vector to Image

Discriminator loss

$$Loss_{Dis} = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

Generator loss

$$Loss_{gen} = E_z[\log(1 - D(G(z)))]$$



pix2pix

❖ Image to Image

Discriminator loss

$$Loss_{Dis} = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

Generator loss

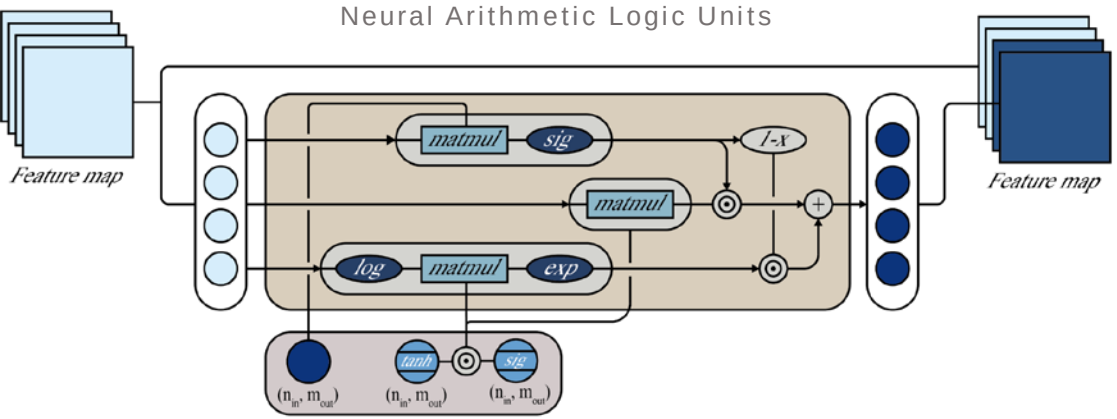
$$Loss_{gen} = E_z\left[\log\left(1 - D(G(z))\right)\right] + \lambda L1loss$$

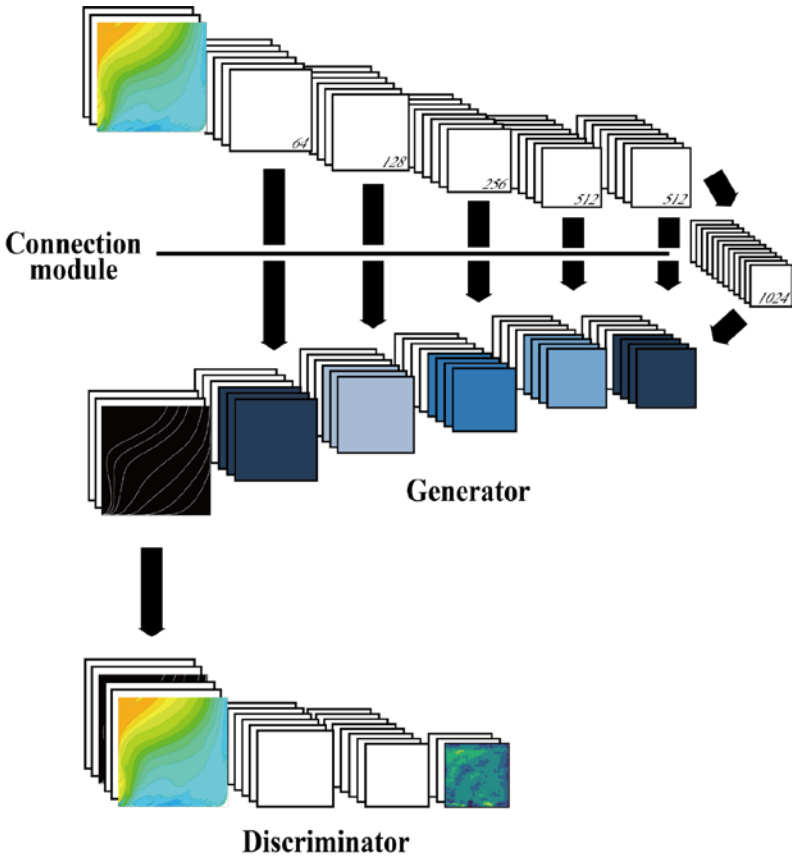
L1 loss – Euclidean distance

$$L1loss = abs(ground\ truth\ - generator)$$

Arithmetic GAN

model





Discriminator loss

$$Loss_{Dis} = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

Generator loss

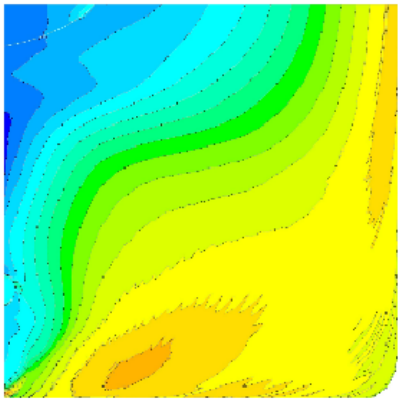
$$Loss_{gen} = E_z\left[\log\left(1 - D(G(z))\right)\right] + \lambda loss_{wce}$$

Weighted cross entropy – Euclidean distance

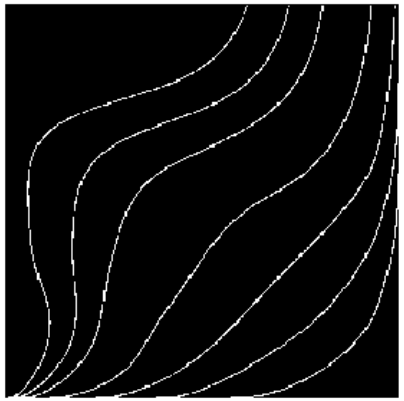
$$Loss_{wce} = ylog(sigmid(x)) * pos + (1 - y)\log(1 - sigmoid(x))$$

Experiments

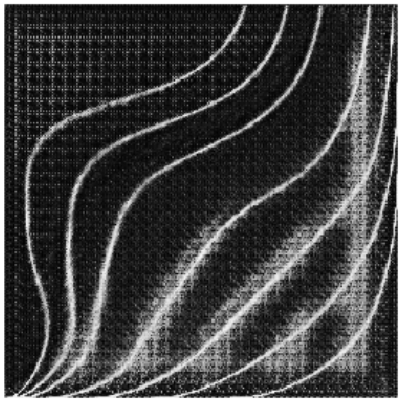
Input Image

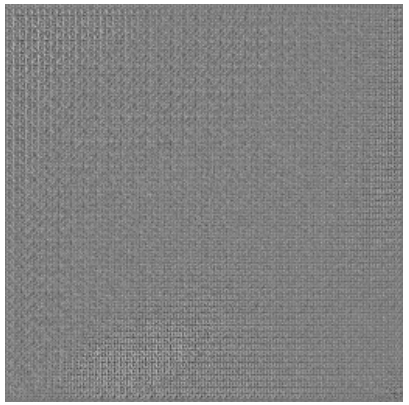


True Mask



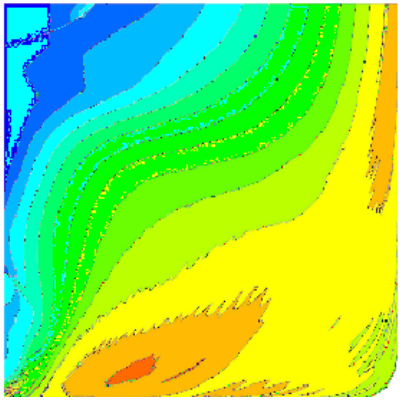
Predicted Mask



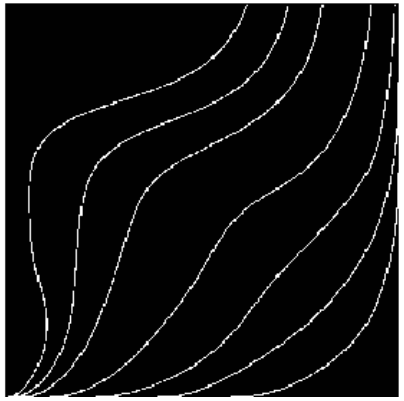


Cross entropy

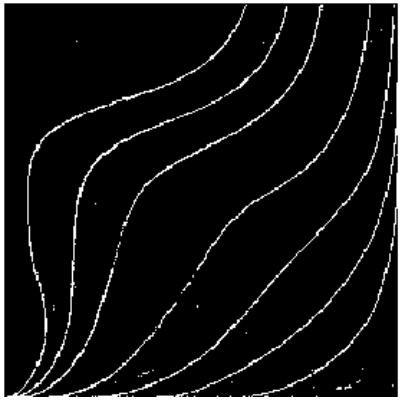
Input Image

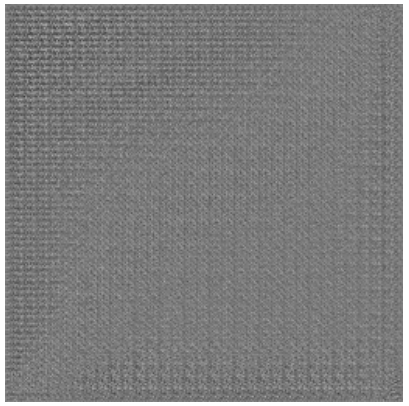


True Mask



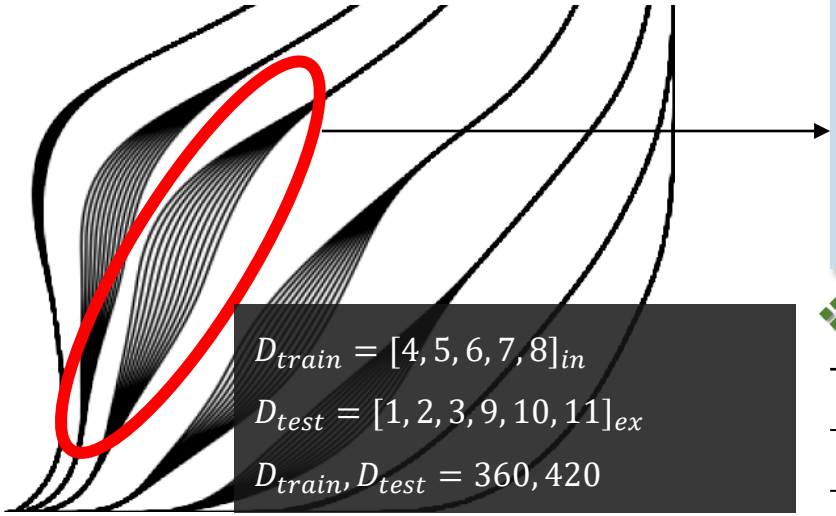
Predicted Mask





Weighted cross entropy

Interpolation, Extrapolation



outer interval: 3

- Line interval: 11
- Hull form baseline: 10
- Transform per baseline: 77
- Total dataset: 11x77 + 10 = 780

❖ Data configuration

	Interval 1	Interval 2	Interval 3	Interval 4
Training set	640	500	360	220
Test set	140	280	420	560

Training set consists of interpolation,
Test set consists of extrapolation

❖ Performance Evaluation

		Ground truth	
		1	0
Prediction	1	True positive	False positive
	0	False negative	True negative

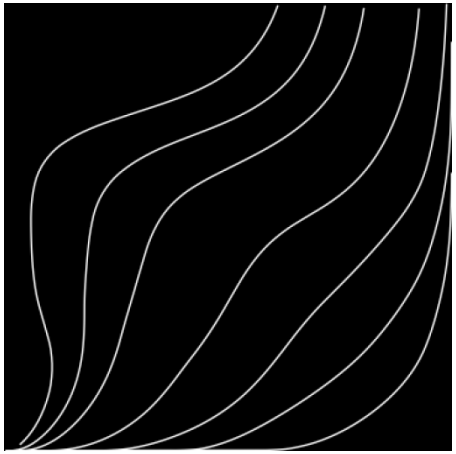
Confusion matrix

$$\text{IoU} = \frac{True\ positive}{True\ positive + False\ positive + Fasle\ negative}$$

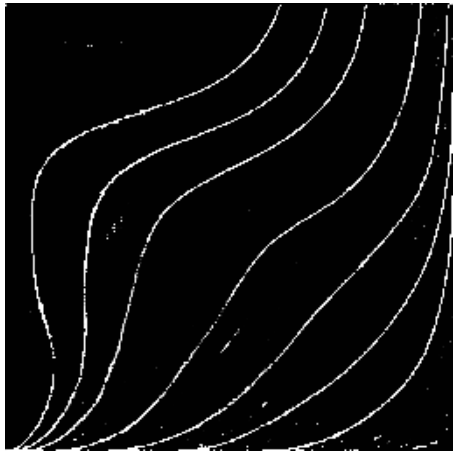
$$\text{mIoU} = \frac{1}{2}\left(\frac{True\ positive}{True\ positive + False\ positive + Fasle\ negative}\right) + \frac{1}{2}\left(\frac{True\ negative}{True\ negative + False\ positive + Fasle\ negative}\right)$$

PASCAL VOC segmentation accuracy

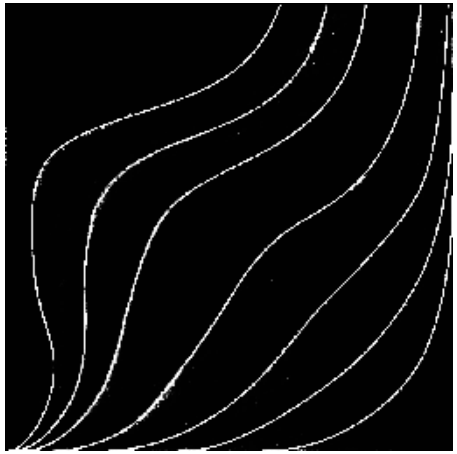
Experiments



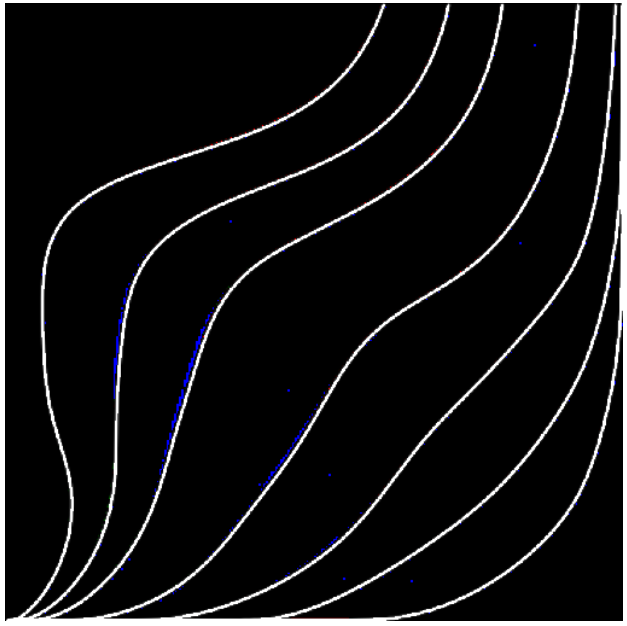
Multi-task learning



Pix2Pix



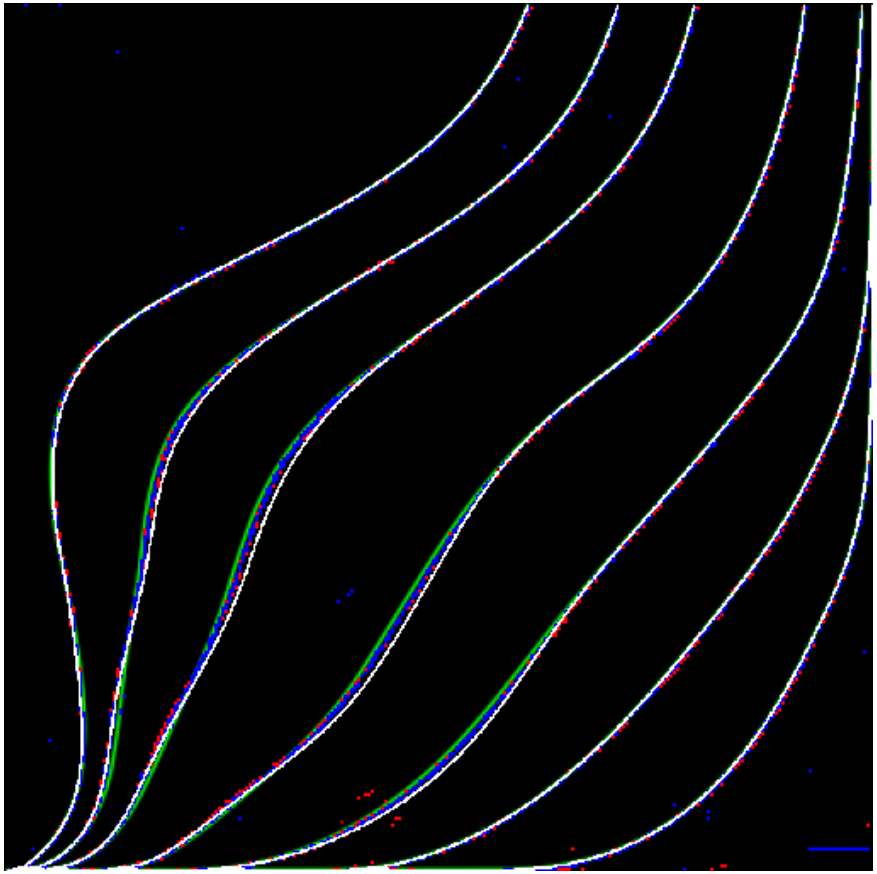
Arithmetic GAN

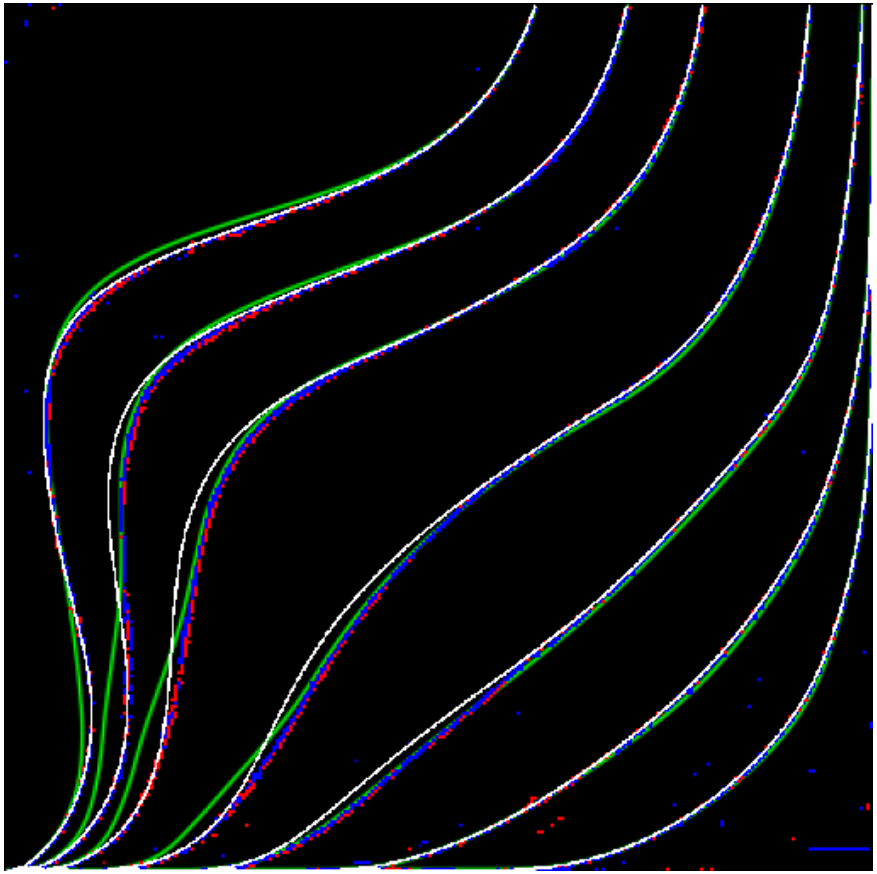


	model	Inter_1	Inter_2	Inter_3	Inter_4
Acc.	Multi	**99.42%**	**99.43%**	**98.35%**	**99.33%**
	Pix2pix	97.84%	97.65%	96.97%	97.10%
	AGAN	98.28%	97.70%	97.59%	97.34%
mIoU	Multi	79.16%	**76.59%**	69.02%	60.29%
	Pix2pix	76.24%	75.38%	70.25%	68.80%
	AGAN	**79.41%**	76.14%	**73.14%**	**71.11%**
IoU	Multi(RMSE)	60.09%(31mm)	**55.26%**(66mm)	40.98%(142mm)	24.81%(196mm)
	Pix2pix	54.29%	53.14%	43.60%	40.56%
	AGAN	**60.58%**	54.64%	**48.75%**	**44.92%**

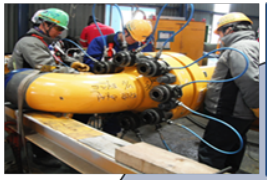
Results

-white: label
-green: Multi-task learning
-red: pix2pix
-blue: Arithmetic GAN

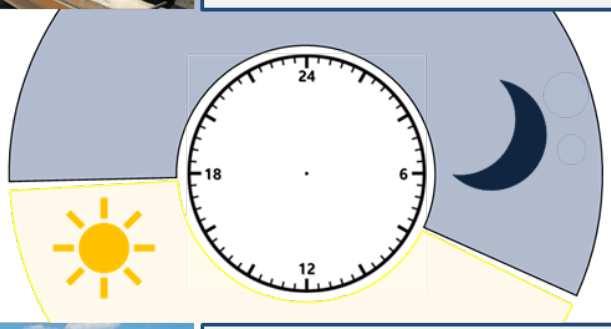




Automatic Detection of Welding Defects Using Faster R-CNN



- RT 특성상 야간 작업 수행
- RT와 용접 결함 검사는 병행이 어려움
- 업무시간의 차이로 인한 시간적 손실 발생





- 주 업무시간에 용접 결함 검사
- 수동적인 작업으로 인한 시수 증가
- 경험적인 작업으로 인한 주관적 결과

- 용접품질에 대한 객관적인 분석 가능
- 실시간 RT 이미지 판독 결과 확보
- 대량의 RT 이미지 자동 판독 가능

용접부 결함 자동 검출

- 결함에 대한 가이드라인 제시
- 주요 결함에 대한 정밀 검사 가능
- 결함 판독 자동화에 따른 객관적 자료 확보

개발 기술의 기대 효과



시수 절감



비용 절감



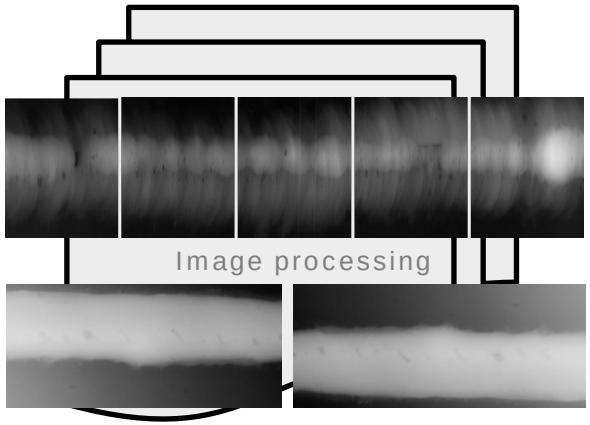
체계화

- Research is needed to secure objectivity in radiographic testing and to enhance the safety of visual inspection at the location of the complaint
- Subjective and experience-based in the identification of welds and information, resulting in inconsistent and unreasonable results

Automatic Detection of Welding Defects Using Faster R-CNN

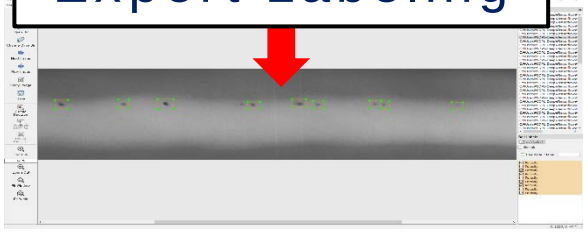
❖ Work process

Dataset



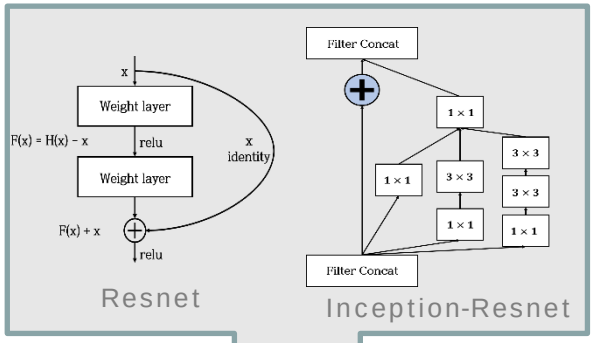
Data augmentation

Expert Labeling

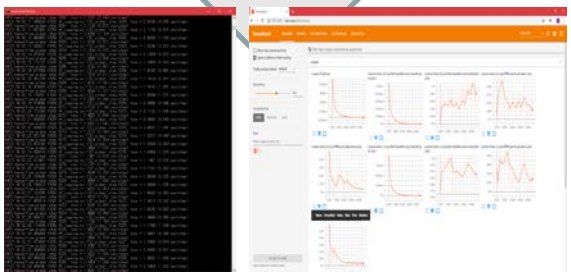


Data labeling

Algorithm Training

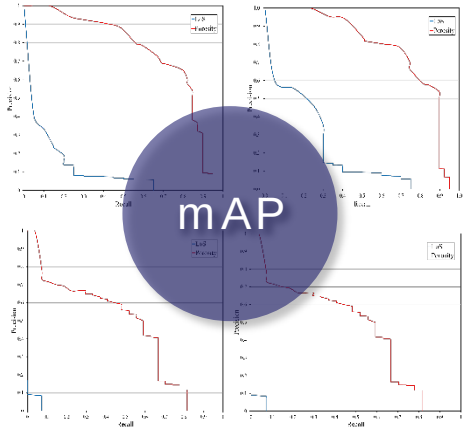


Learning

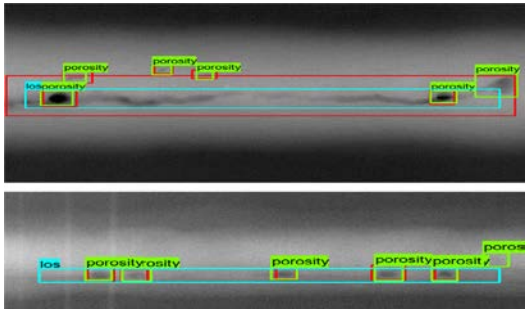


Learning algorithm using TensorFlow

Performance test



PR curve



Prediction

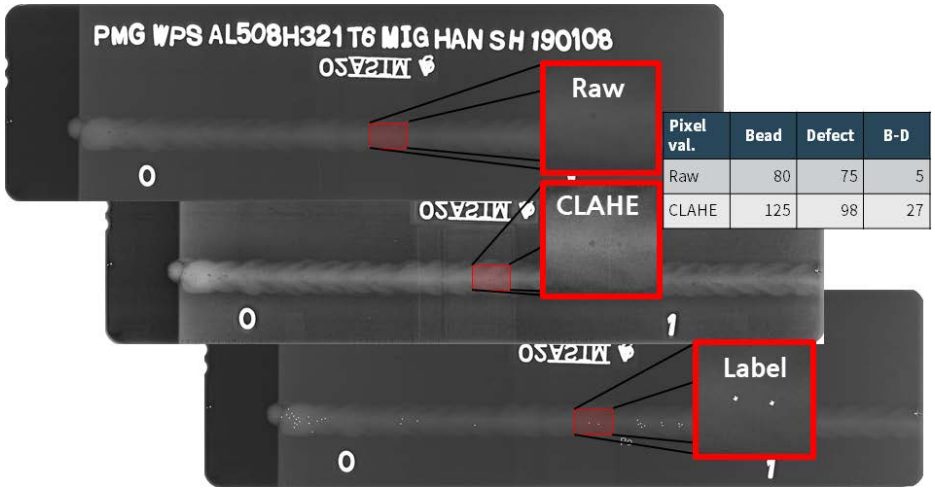
Image Enhancement(1/2)

❖ Image Enhancement

Image Enhancement

- **Histogram Equalization**
- **Denoising**
- **Threshold**

- Using image enhancement techniques, ‘Black’ defects with lower pixel values than the background, such as porosity and lack of penetration, and ‘White’ defects with higher pixel values compared to the background such as Excessive Penetration and Slag Incidence are classified into stages.



Pixel val.	Bead	Defect	B-D
Raw	80	75	5
CLAHE	125	98	27

Defect	Black			White		
Level	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
Pixel	30~	15~30	0~15	30~	15~30	0~15
Defect	83	131	883	20	34	66

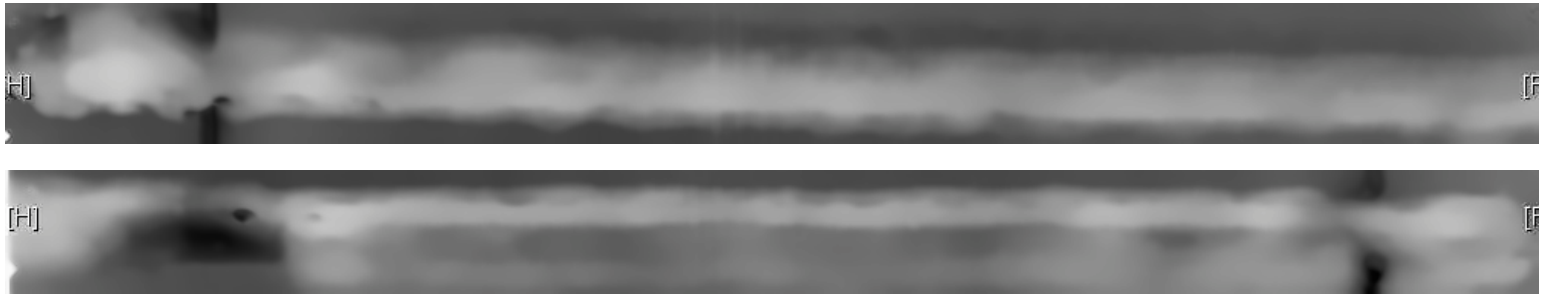
Weld defect before applying Image enhancement

Defect	Black			White		
Level	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
Pixel	30~	15~30	0~15	30~	15~30	0~15
Defect	265	422	420	76	24	10

Weld defect after image enhancement applied

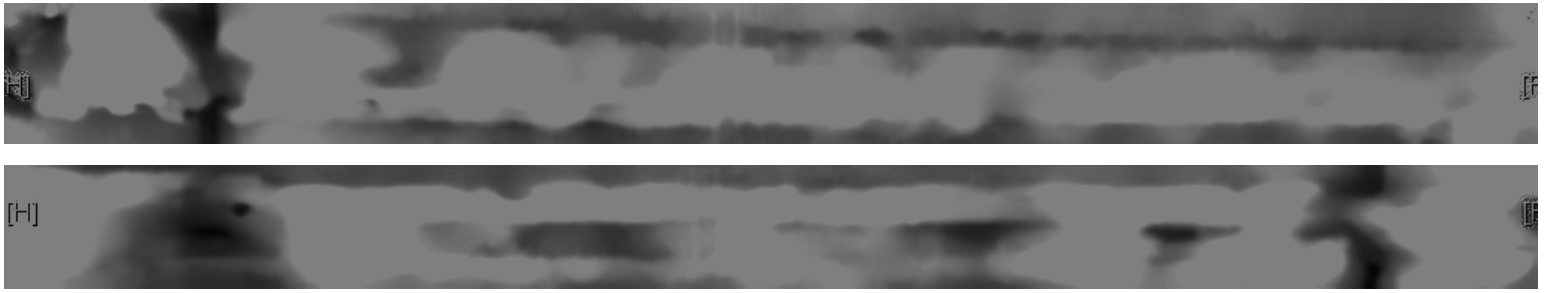
Image Enhancement(2/2)

❖ Denoising



- Stabilize the pixel value by reducing noise for the defect image.
- Eliminate unnecessary noise, eliminate loss of minor defects.

❖ Threshold



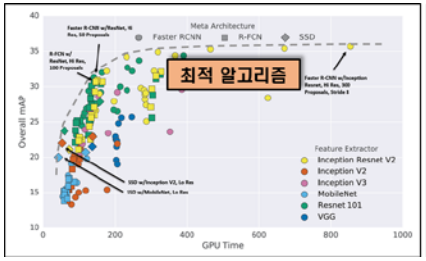
- For defects with positive values during weld defects[0, 1], adjust the threshold after reversing the image

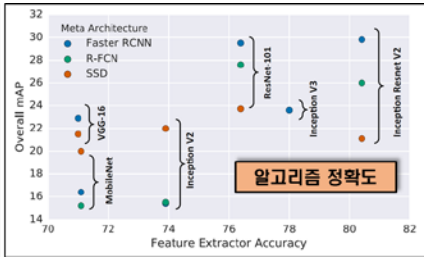
Development Processes and Results

- In international image recognition competitions, high-performance ResNet and Inception-ResNet V2 are selected as comparison targets, with 76% and 80% more performance based on COCO data
- Model performance after applying Data Augmentation are higher, confirming that artificial data growth is effective

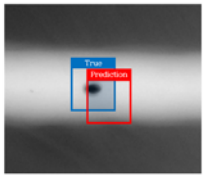
Hyper-parameter Tuning

Algorithm Comparison





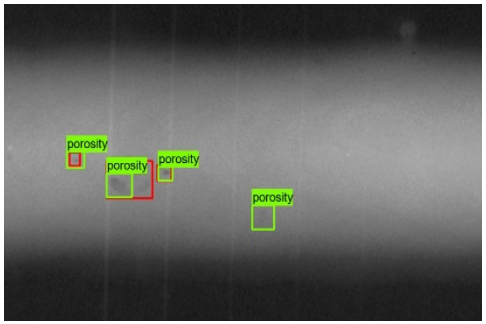
Hyper-parameter



Anchor box size	10	40	80
Aspect ratio	1:1	1:5	5:1

후보 영역 추천 수	500	1,000	2,000
mAP	30.78%	36.68%	28.25%

Development Object Detection Using Faster R-CNN



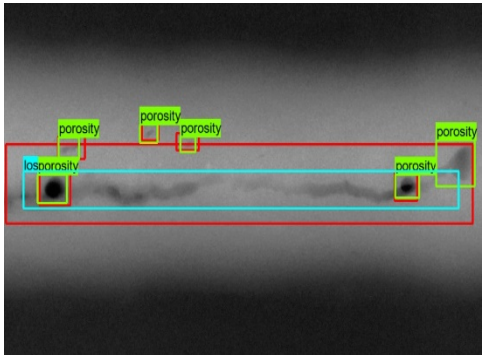


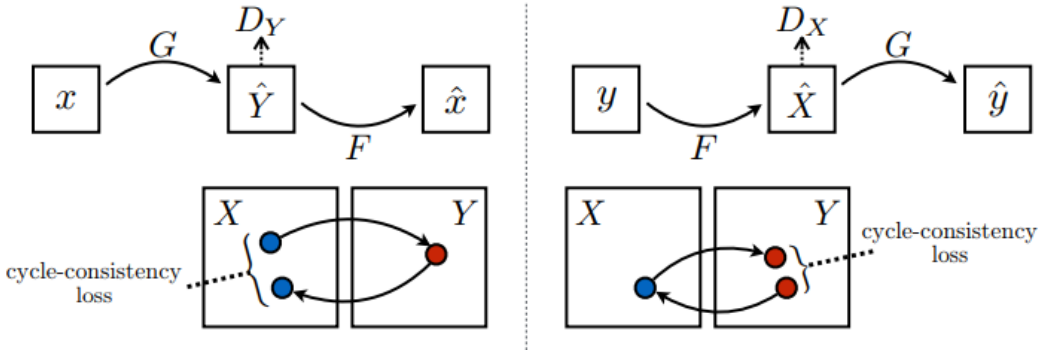
표 1-1 시험결과

횟수	mAP
1	0.804
2	0.812
3	0.818
4	0.811
5	0.784
평균	0.806

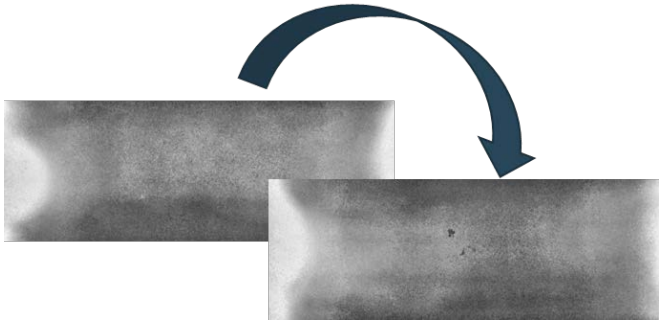
Randomly select 100 images from 320 images

Data Augmentation using GAN

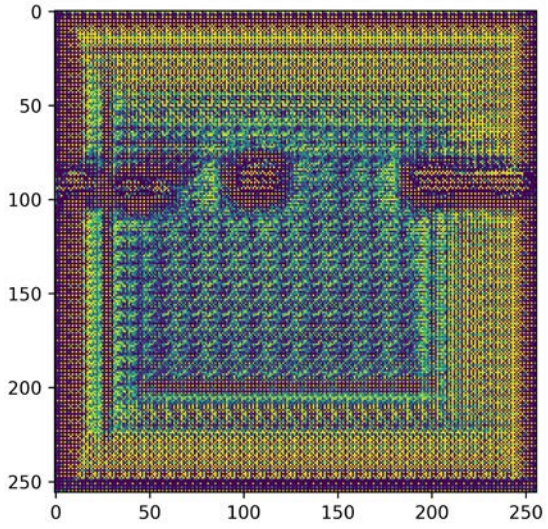
❖ Generative Adversarial Networks



CycleGAN



Convert to Target Domain

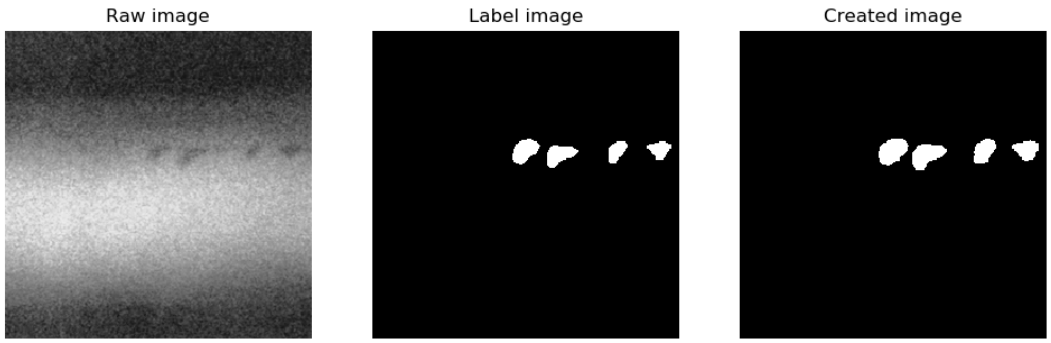


- As a solution to data imbalance, data growth was used, and data growth was required for certain defects in addition to adjustments to the base image, so we studied how to generate data using artificial intelligence
- Generate new images and balanced data with GAN
- Set domain with or without weld defects (no defect **image → defective image)**

Semantic Segmentation

❖ Development Object Detection Algorithm Using Segmentation

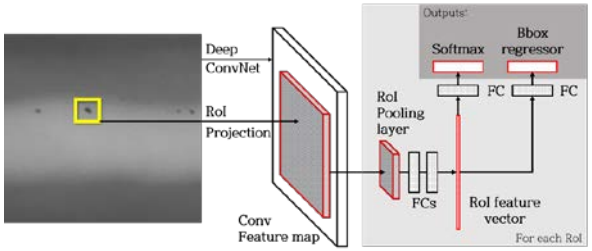
- Develop and apply faster and more accurate algorithms than existing object detection algorithms, taking into account the diversity of image sizes and the characteristics of various patterns of defects.
- Achieving the goal of 'Weld Defect Object Detection Accuracy for Image' by focusing on detection rather than classification of weld defects.



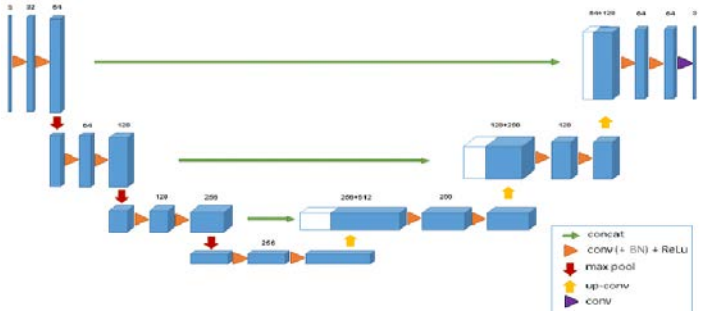
Segmentation Raw / Label / Predicted

Object detection / Segmentation

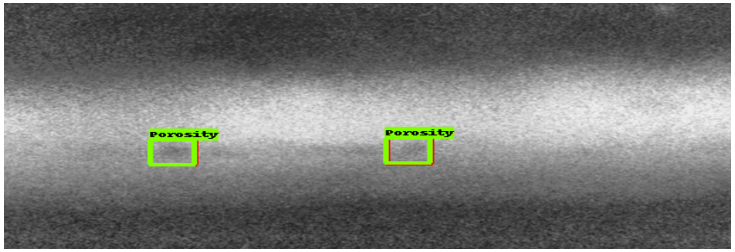
❖ Object detection



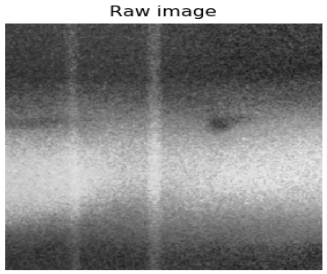
❖ Segmentation

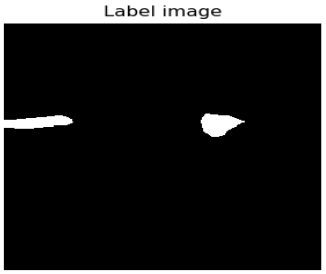


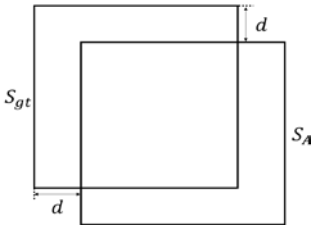
Model



Output







mean Average Precision
- IoU
- Precision & Recall
- Localization & regression

Performance



Accuracy
- Pixel to Pixel
- Confusion matrix
- Acc = TP / (TP +FP+FN)

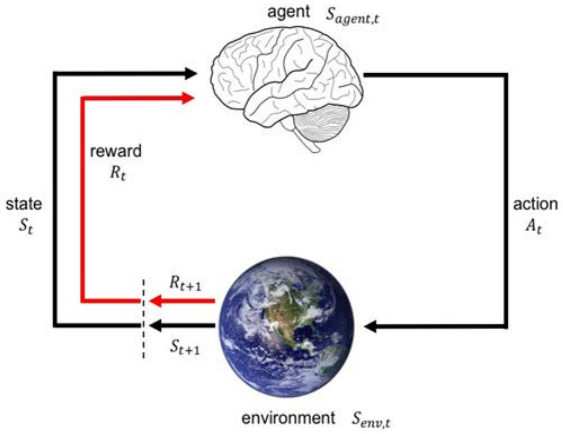
Reinforcement learning



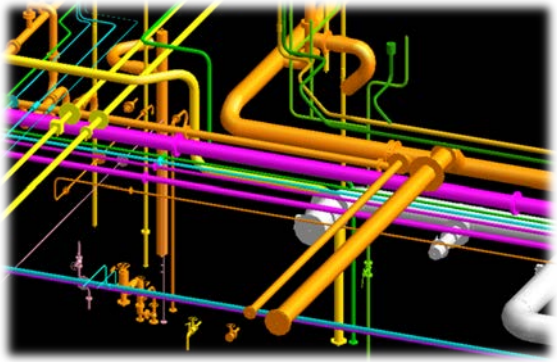
Pipe Routing Using Reinforcement Learning

❖ *Reinforcement learning*

- Agent-environment interaction break down into
- Learning about action(A_t) that maximize reward(R_t) in environment($S_{env,t}$)
- Reinforcement learning provides guidelines at the initial design stage for the pipe routing



Reinforcement learning

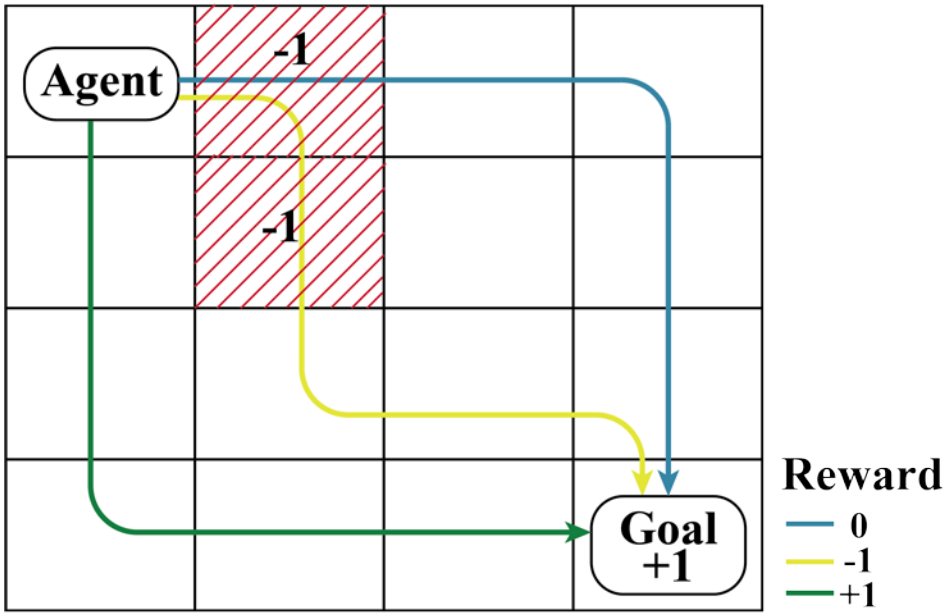


Pipe routing

Pipe Routing Using Reinforcement Learning

❖ Optimal rout selection process

- Reinforcement learning enables agents to explore and exploit the environment to find the best policy
- Quantify the path selected by the agent with the rewards given
- Select the route that will receive the maximum reward as the optimum route

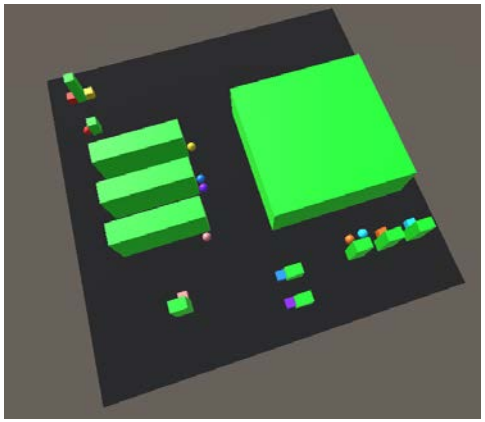


Reward policy

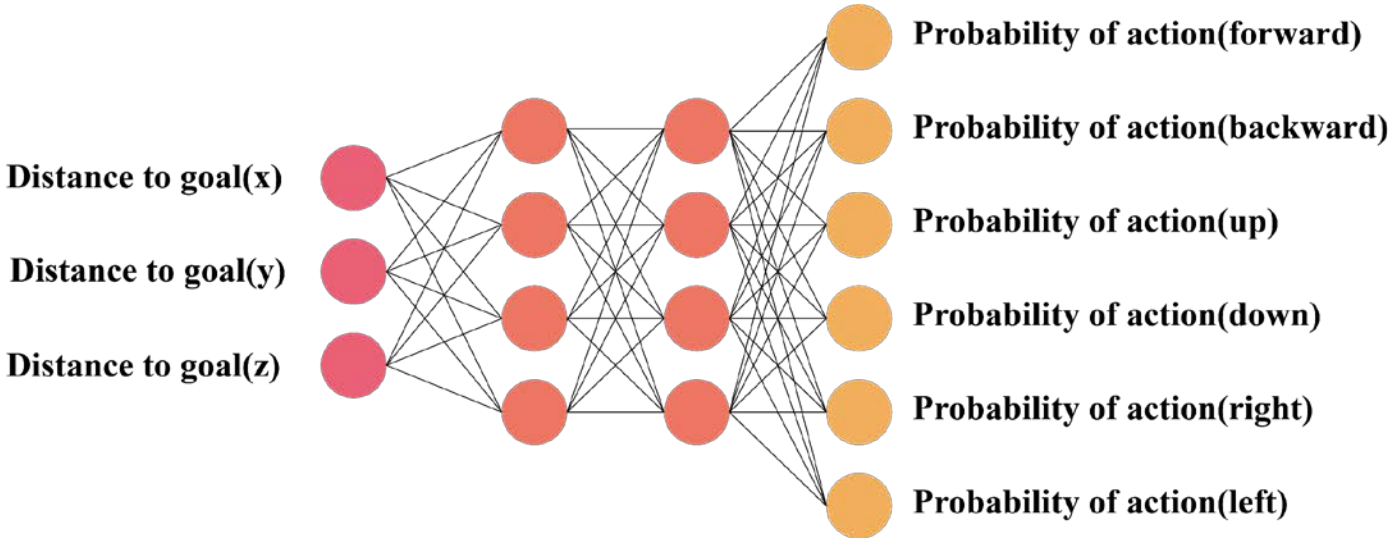
Pipe Routing Using Reinforcement Learning

❖ Proximal Policy Optimization(PPO)

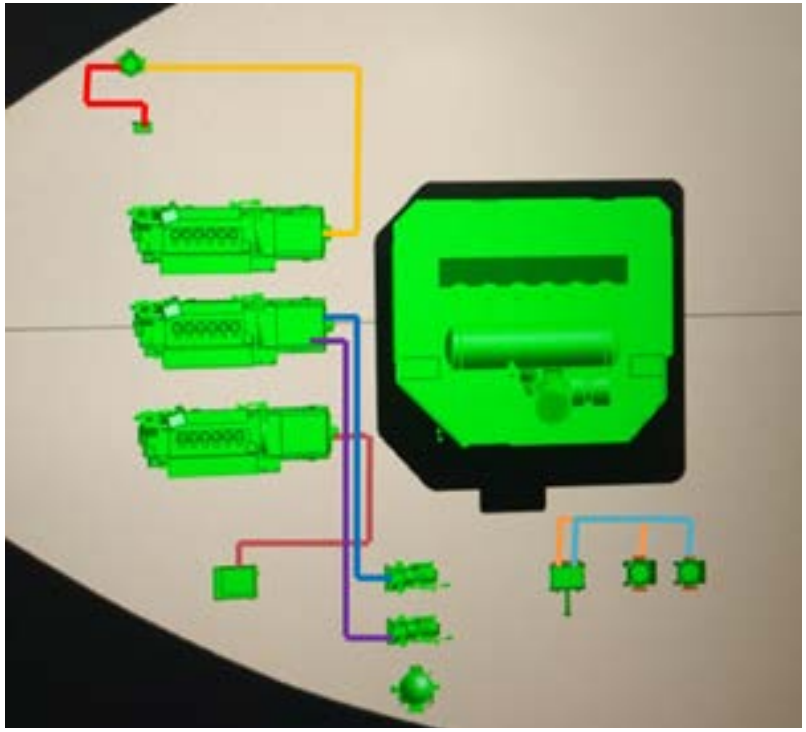
- More effective than other algorithms when the number of states is high and the agent has a variety of behaviors, stable in policy convergence
- Neural network input: distance from the starting point to the arrival point of the pipe (x, y, z)
- Neural network output : Probability of agents to choose each action
- (forward, backward, up, down, left, right)



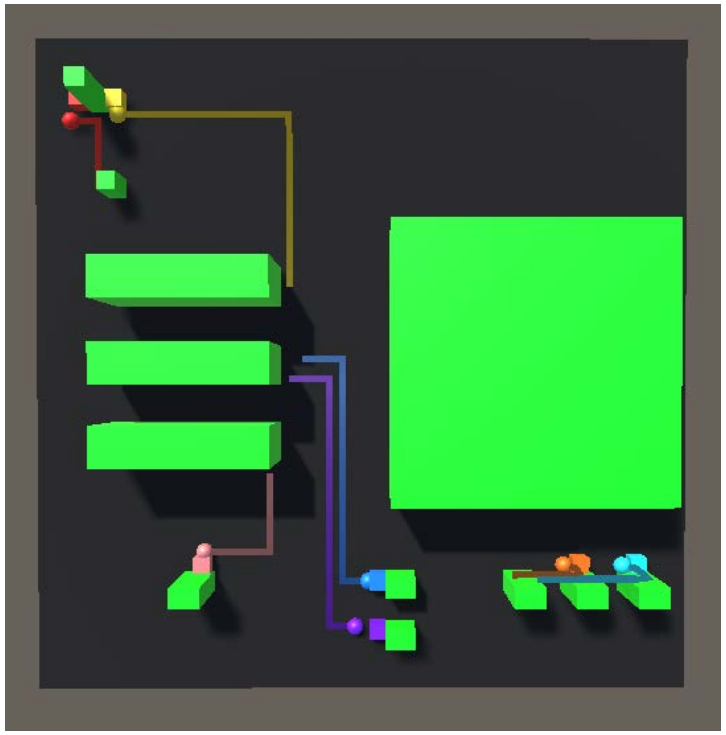
Engine room in Unity



Pipe Routing Using Reinforcement Learning



Ground truth



Reinforcement learning

- Consider physical and economic constraints among piping design constraints
- It takes up to 20 minutes to select a total of 7 piping routes

Q&A