딥러닝 모델의 SoyNet 포팅

SoyNet 팀

발표자: 2022313382 백승렬

목차

과제 개요

- 팀 소개
- 회사 소개(과제 배경)
- 적용 기술

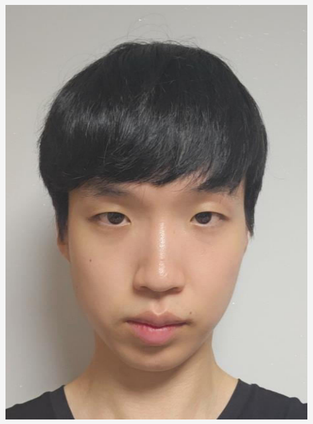
01. 팀 소개

김호재

백승렬

이미향 교수님

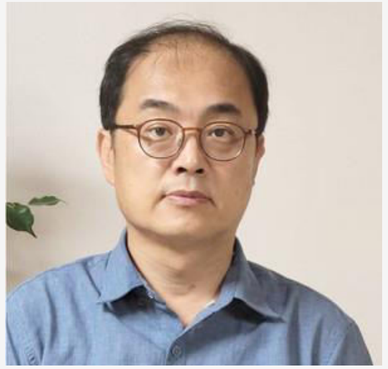
유경석 소이넷 팀장님



박세훈

박제현

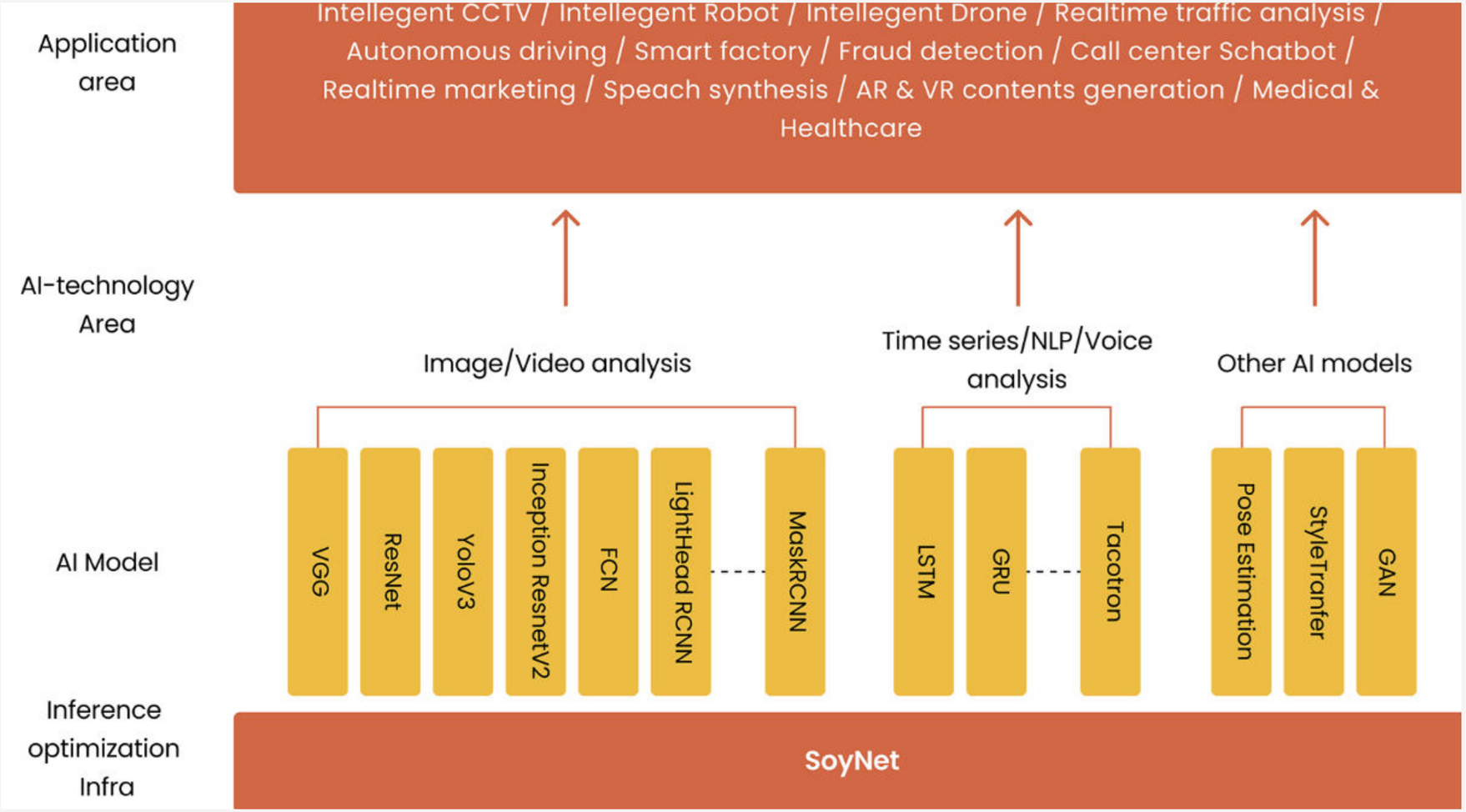
이해성



김용호 소이넷 대표님

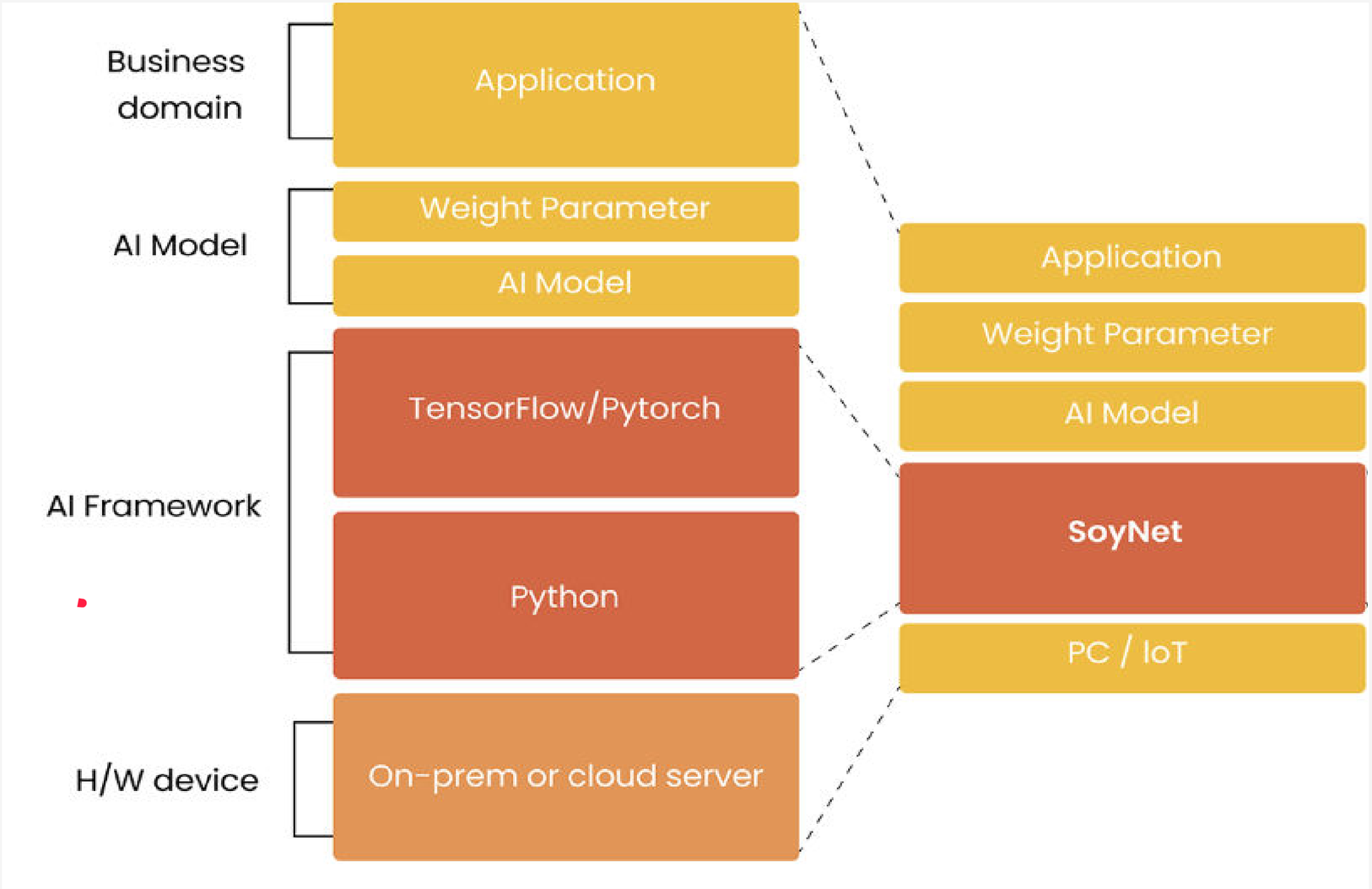
02. 회사 소개: 소이넷

- 추론 최적화 인프라 역할을 하는 솔루션 보유
- 해당 솔루션을 통해 구축해낸 딥러닝 모델의 추론 엔진을 제공



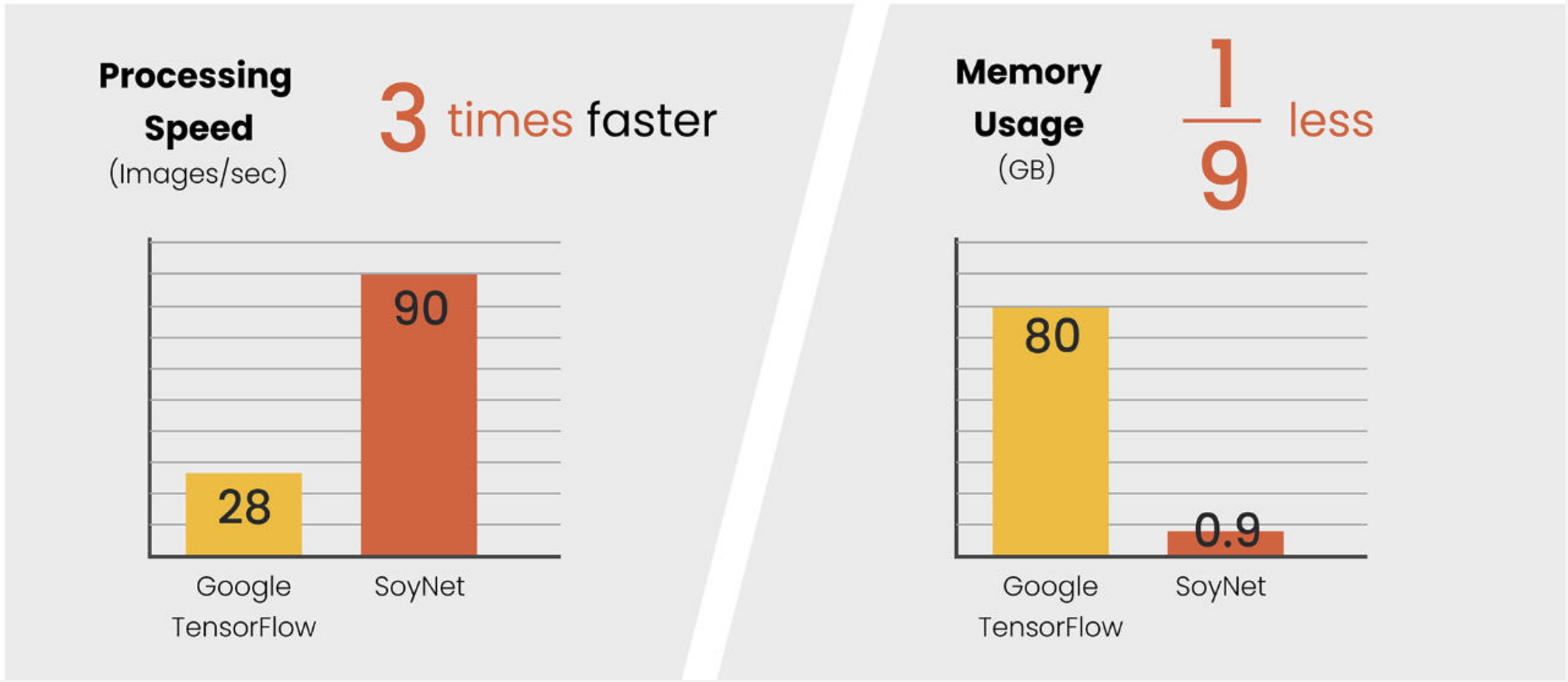
03. 적용 기술: SoyNet.sln

- 소이넷이 보유한 솔루션은 추론 프로그램을 만들 수 있는 인프라임
  → **Python/Tensorflow/PyTorch 와 같이 “프레임워크” 기능을 함.**



03. 적용 기술: SoyNet.sln (cont'd)

- 해당 솔루션이 딥러닝 레이어 단계의 연산에서 일반적으로 다른 프레임워크에 비해 빠르고 메모리를 적게 사용함



과제 내용

- 과제 목표
- 과제 내용
- 수행 내역

01. 과제 목표

- SoyNet 솔루션을 기반으로 추론 엔진 제작
- Python/PyTorch/Tensorflow 기반 기존 모델을 SoyNet 솔루션으로 이식하는 “포팅” 작업

PyTorch

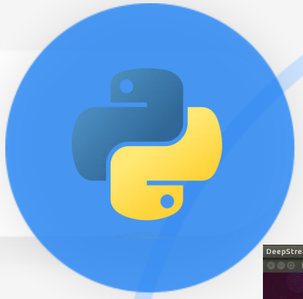
SOYNET

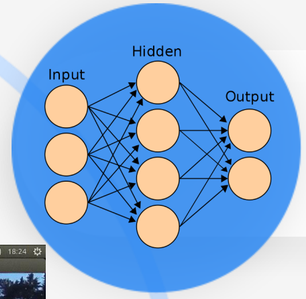
PyTorch/Tensorflow

SoyNet.sln

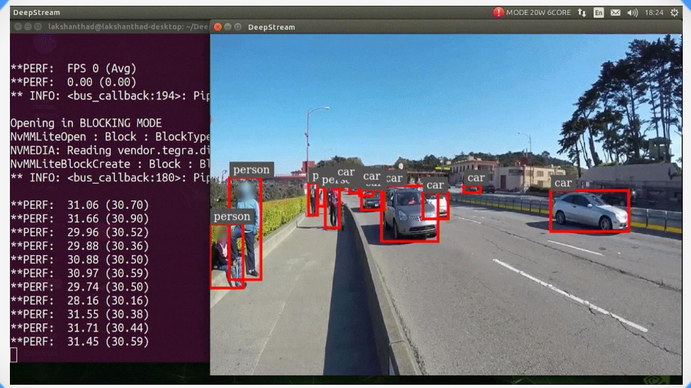
02. 과제 내용

Original Program

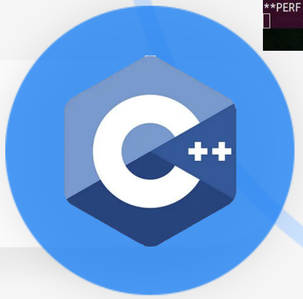




Architecture



SoyNet Solution

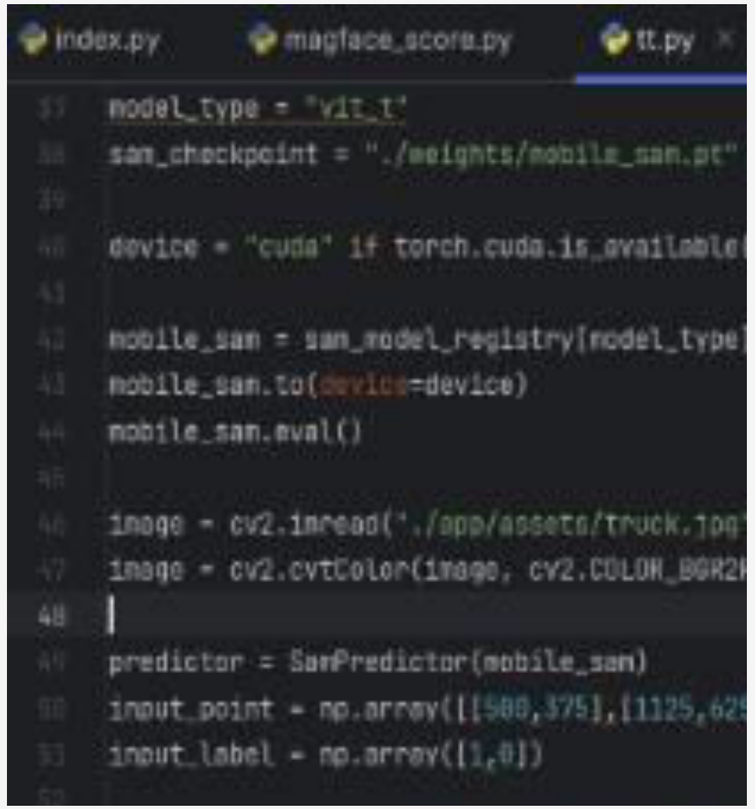


Weights

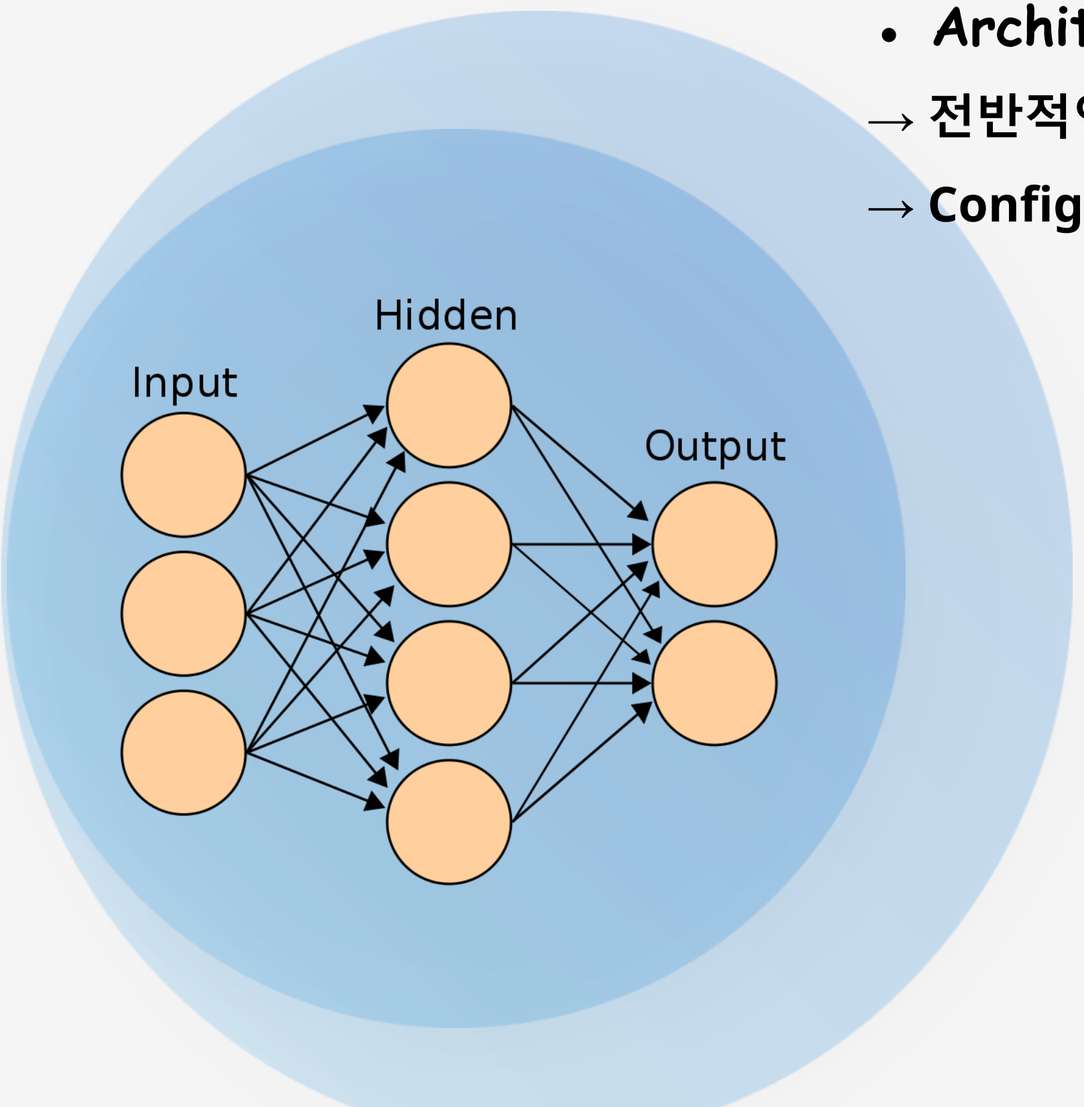
02. 과제 내용

- *Original Program*

→ 이미 Python으로 만들어진 추론 프로그램의 Demo Code



02. 과제 내용



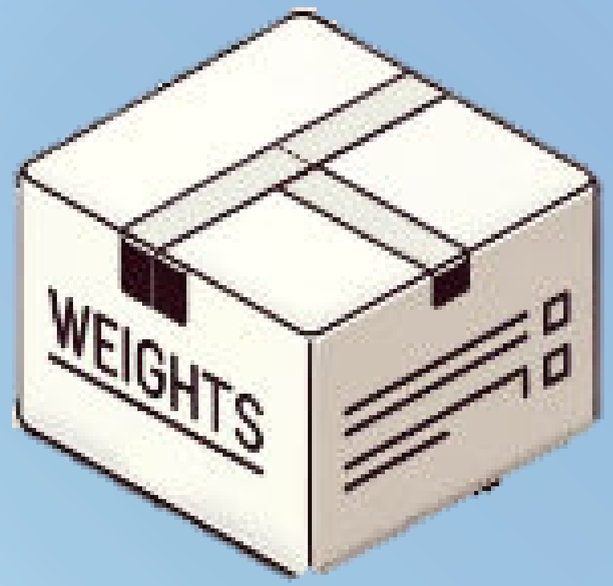
- **Architecture**

→ 전반적인 모델 구성 정보(레이어 등)

→ Config File로 작성

```
[reshape] shape=$BATCH_SIZE,19,7,19,7,128
[trans] order=0,1,3,2,4,5
[reshape] shape=$BATCH_SIZE*361,49,128
[norm] mode=layer axis=2 weight_order=rb eps=1e-5

[dense] hidden=384 weight_order=wa refname=QKV_0
[reshape] shape=$BATCH_SIZE*361,49,4,96
[chunk] axis=3 count=3 refname=Q_IN,K_IN,V_IN
[trans] input=Q_IN order=0,2,1,3 refname=Q
[trans] input=K_IN order=0,2,1,3 refname=K
[trans] input=V_IN order=0,2,1,3 refname=V

#[trans] input=K order=0,1,3,2 refname=K_T
[matmul] input=Q,K trans_b=1 refname=QK_IN
[scale] input=QK_IN scale=0.176777 refname=QK
[eltwise] mode=mobile-sam input=QK
[softmax] refname=ATTN_0
[matmul] input=ATTN_0,V refname=ATTN_0V
[trans] input=ATTN_0V order=0,2,1,3 refname=ATTN_0T

[reshape] input=ATTN_0T shape=$BATCH_SIZE*361,49,128
[dense] hidden=128 weight_order=wa
[reshape] shape=$BATCH_SIZE,19,19,7,7,128
[trans] order=0,1,3,2,4,5
[reshape] shape=$BATCH_SIZE,133,133,128
[slice] start=0,0,0,0 shape=$BATCH_SIZE,128,128,128
[reshape] shape=$BATCH_SIZE,-1,128 refname=X_0_1
[eltwise] input=BLK2_IN,X_0_1 mode=add
```

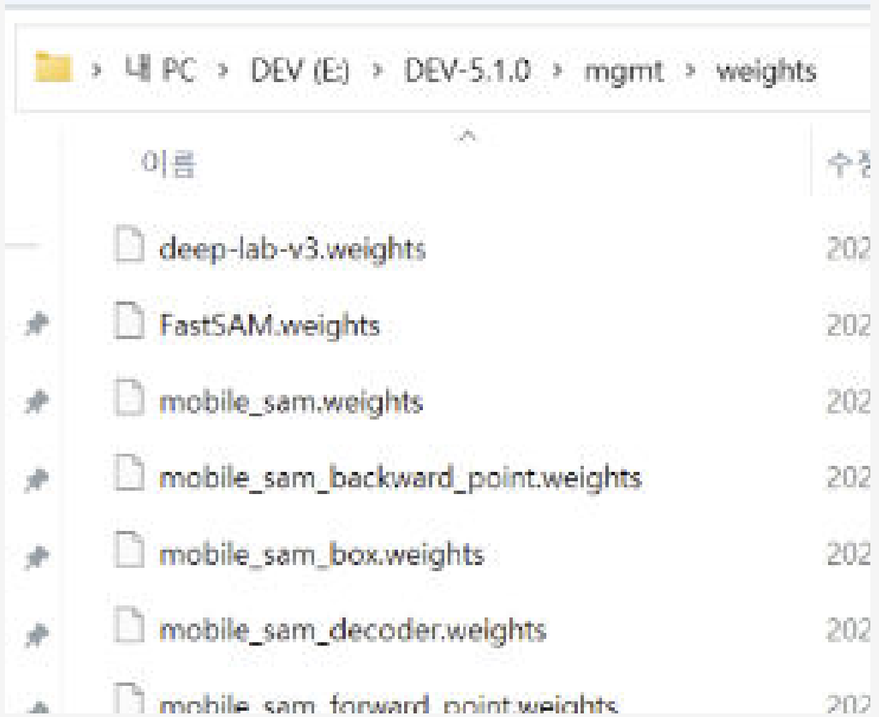
02. 과제 내용



- **Weights**

→ 해당 모델의 학습된 가중치를 엔진이 이해할 수 있는 형식으로 변환

→ [모델 이름].weights



02. 과제 내용

- **SoyNet Solution API**

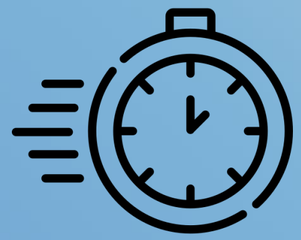
→ 추론 엔진을 생성, 실행하여 결과를 보여주는 C++ 코드

```
feedData(img_encoder_handle, 0, source.data());
inference(img_encoder_handle);
getOutput(img_encoder_handle, 0, image_embed.data());

if (forward_point_encode_max != 0) {
    feedData(forward_point_prompt_handle, 0, point_
    inference(forward_point_prompt_handle);
    getOutput(forward_point_prompt_handle, 0, forwa
}

if (backward_point_encode_max != 0) {
    feedData(backward_point_prompt_handle, 0, point
    inference(backward_point_prompt_handle);
    getOutput(backward_point_prompt_handle, 0, back
}

feedData(box_prompt_handle, 0, box_pt.data());
inference(box_prompt_handle);
getOutput(box_prompt_handle, 0, box_embed.data());
```

02. 과제 내용

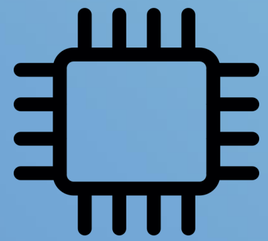
- 성능 측정

pretrained	yes		
porting 시작	########		
porting 종료	########		
porting 작업자	백승렬		
train-code name			
inference-code name	tt.py(batch_size=1), tt1.py(batch_size=2,4)		
weight-down-code name	ww.py		
work 코드			
Benchmark 기기	RTX 3080 12GB		
data	truck.jpg		
batch	1	2	4
precision	f32	f32	f32
data size or length	batch_size x input_height x input_length x 3		
SoyNet fps(입력 전부 사용)	34.49	27.46	17.08
SoyNet GPU memory(입력 전	865	1109	1357
pytorch fps(입력 전부 사용)	34.55	35.08	35.13
pytorch memory(입력 전부 사	1697	1761	1765
Benchmark brief	입력 이미지는 전부 resize되어 들어감. 각 배		
Model Consistency	pytorch	SoyNet	Diff
(정합성 비교)	각 좌표마다	각 좌표마	0



속도

- 원본 Demo & Main 함수에 fps 측정 구현
- 실행 후 확인



메모리 사용량

- cmd에서 nvidia-smi 명령어로 최대 메모리 사용량 측정

정확도

- 각 레이어 별 데이터 값 비교
- 오차값 측정

03. 수행 내역

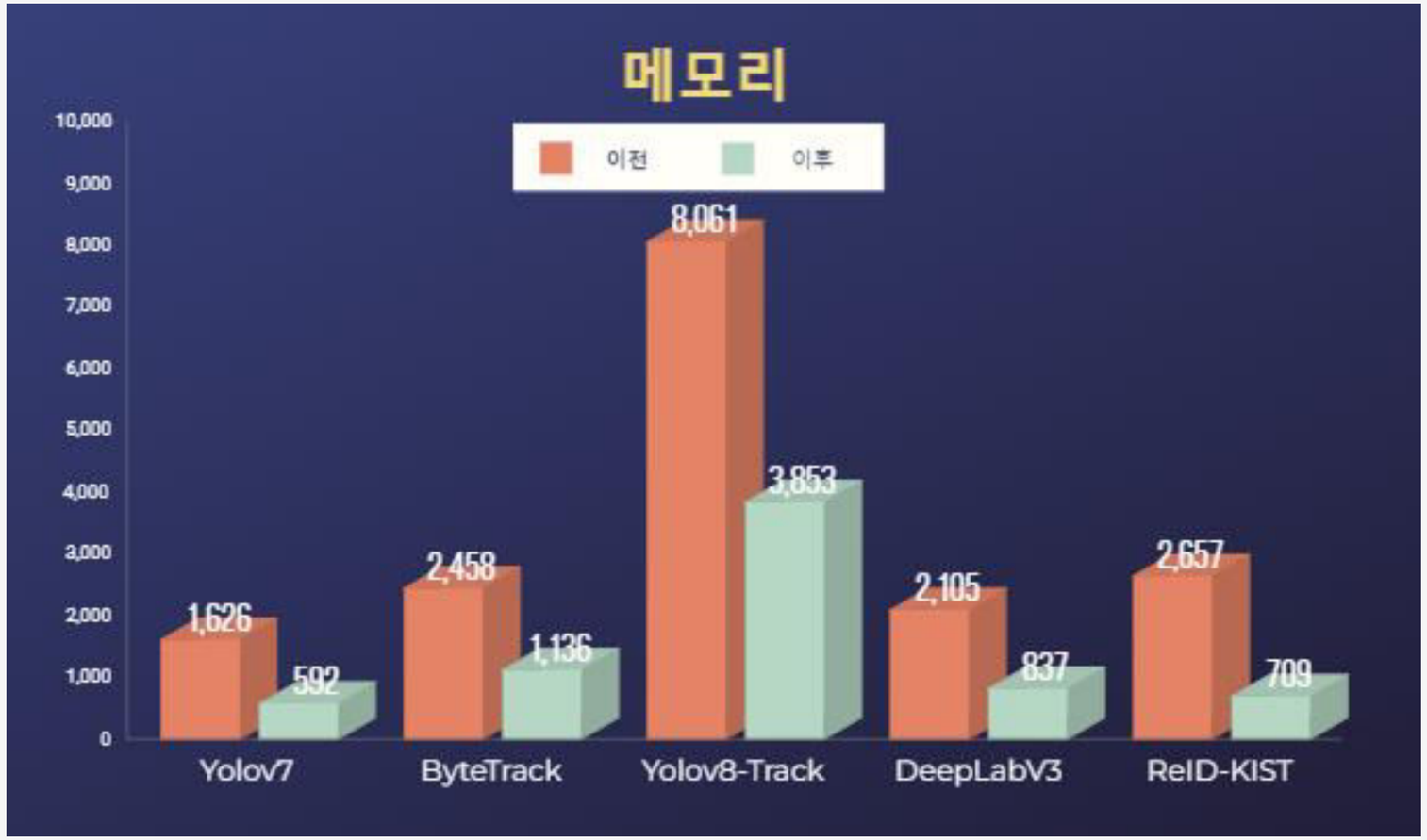
- 주로 이미지를 처리하는 Vision 모델들을 최적화

03. 수행 내역

- Yolo, ByteTrack, DeepLabV3, ReID-Kist 등 10개 이상의 모델을 포팅 완료
- 속도는 모델마다 약 2배~5배 정도 증가

03. 수행 내역

- Yolo, ByteTrack, DeepLabV3, ReID-Kist 등 10개 이상의 모델을 포팅 완료
- 메모리 사용량은 모델마다 약 2~4배 감소



03. 수행 내역

- 결과적으로 최적화된 엔진들을 제공하여 실제 AI 활용에 있어서 드는 비용을 줄일 수 있음
- 저사양의 기기에서도 잘 돌아갈 수 있고(용량 부담 감소), 빠른 속도로 실시간 시스템에 응용 가능

REID-KIST-extraction, REID-KIST-identification, DeepLabv3, Yolov7은 현재 활용 중(국민안전과제 SQI소프트)

CCTV에 적용

CCTV에 찍히는 영상을 프레임 별 입력으로 사용
(최적화 작업을 통해 이러한 real-time 에서의 작업이 가능하게 됨.)

딥러닝 모델들을 활용해 지나가는 사람을 확인

이를 바탕으로 실종자를 찾을 수 있음

2학기 활동

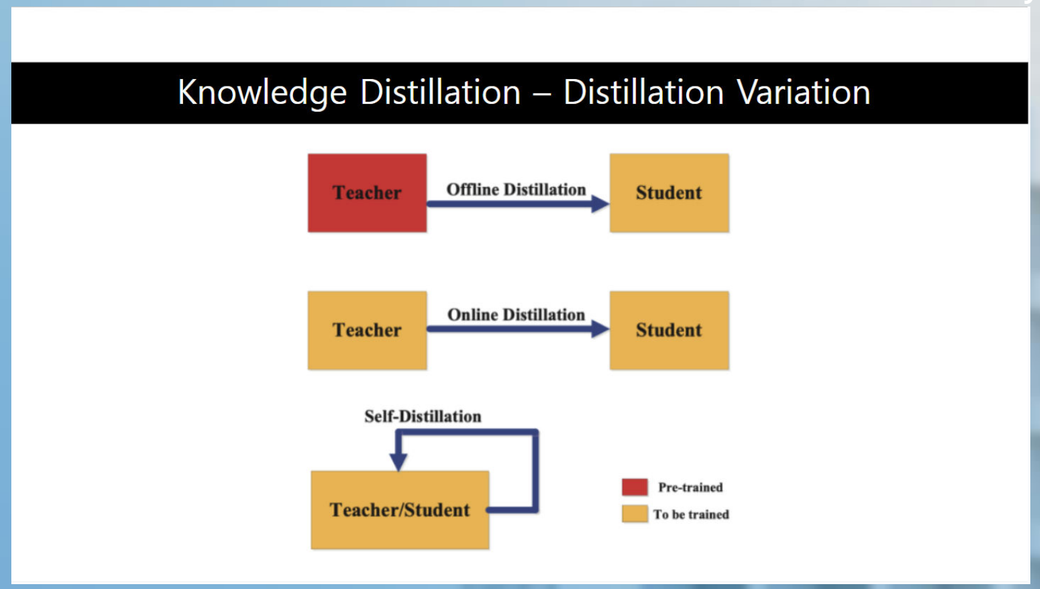
- 논문 스터디
- 공모전 출전

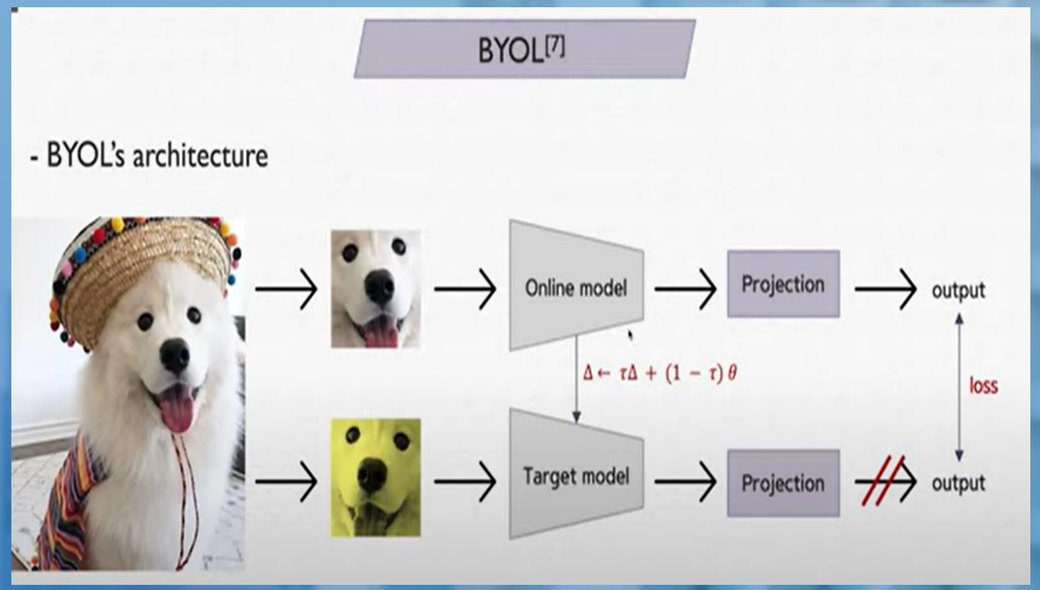
01. 논문 및 기술 스터디

스터디 목표 : 소이넷에서 얻은 경험과 기술을 발전시키고 내재화하기 위함.

개인별 발표 내역

김호재	Domain Adaptive Semantic Segmentation
박세훈	게임개발과 Unity 기초
박제현	3-Tier 시스템과 AWS에 대한 이해
백승렬	Knowledge Distillation
이해성	“Attention is All You Need”, Transformer

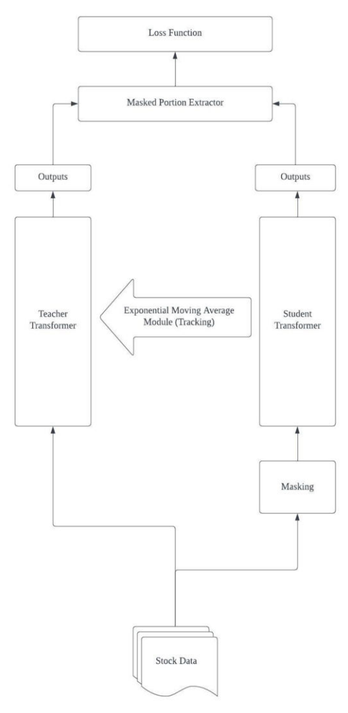


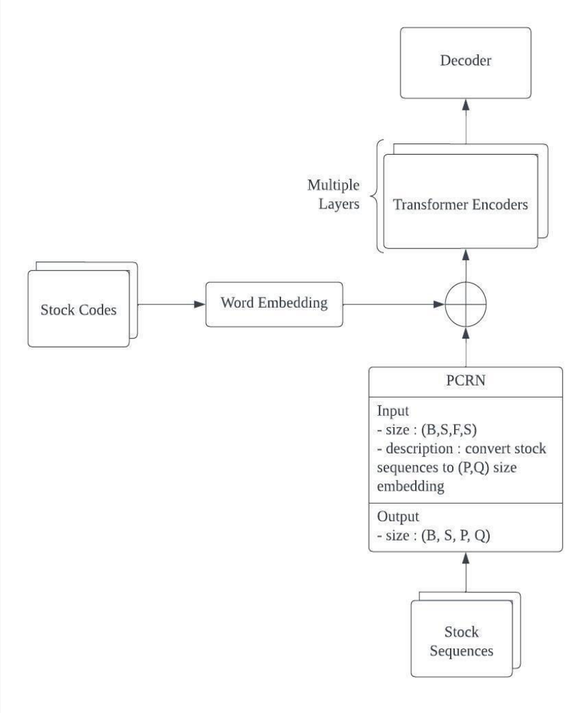


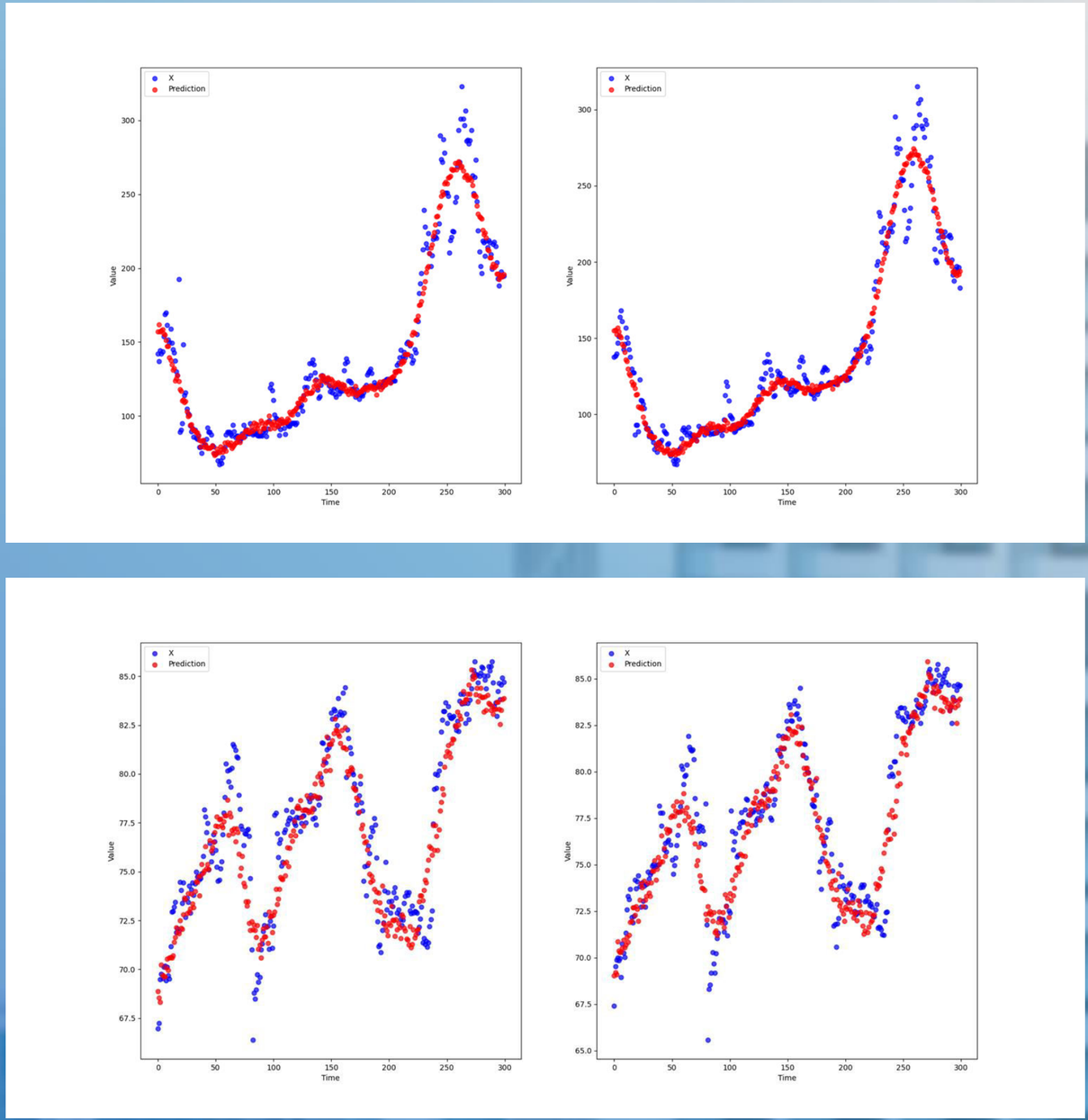
02. 공모전 출전

2023 NH 증권 빅데이터 분석 대회
참가 인원: 김호재, 박제현, 이해성

<Fin2Vec : Transformer 기반 종목 간 상관관계 분석 모델>







THANK YOU

감사합니다.