SOON CHUN HYANG UNIVERSITY

SAIL Seminar 23.12.14

A Novel Method on Real-time Operation of Multiple Vision-AI Models for In-Aircraft Security of Unmanned Aerial Vehicle

순천향대학교 미래융합기술학과
석사과정 김병훈

Index
Contents

1 Introduction

2 Methods

3 Experiment

4 Conclusion

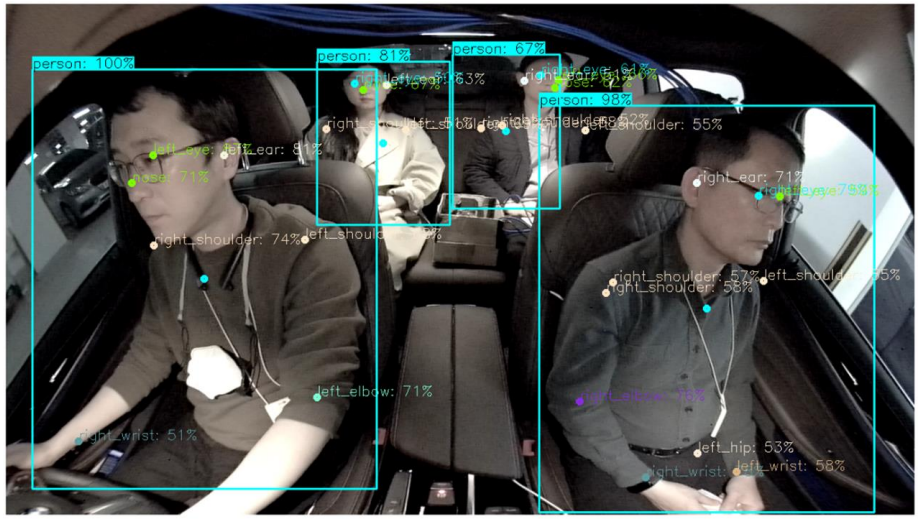
1. Introduction

Background

- ✓ 무인 항공기(UAV) 기술의 발전은 상업용 배송부터 감시 등 다양한 분야에서 새로운 가능성의 시대를 열었음
- ✓ 무인 항공기가 보편화되고 정교해 짐에 따라 자율적인 작동과 의사 결정을 가능하게 하는 On-board system이 장착됨

  * On-board System : 특정 장치나 운송 수단의 내부에 내장되어 작동되는 시스템
- ✓ 그러나 무인 항공기 사용이 급격히 증가하면서 항공기 내 보안에 대한 우려가 커지고 있음(주로 금지 품목의 운송이나, 항공기내 위험 행동)
- ✓ 자율 주행의 특성상 위험 행동에 대한 직접적인 제지가 어려움이 따라 이를 감지하고 완화할 수 있는 시스템을 갖춰야 함
- ✓ 이를 해결하기 위해 비전 기반 AI모델을 활용한 무인 항공기 내부 모니터링 시스템 제안

Anomaly behavior example



LG mobility in-vehicle safer system

1. Introduction

Contribution

Limitations of previous research

- ✓ 제한된 환경(UAV) 내에서 여러 AI모델을 동시에 작동하기 어려움
- ✓ 위험 상황을 감지하기 위해 필요한 모델(위험 물품 감지, 위험 인물 식별 등)들은 복잡하고 리소스가 많이 필요함
- ✓ 위 모델을 실시간 및 병렬로 실행하는 것은 UAV에 탑재된 컴퓨팅 성능, 메모리 등의 한계로 기술적 문제가 있음

Contribution

1. 기존 연구의 한계점을 해결하기 위해 규칙 기반 System을 제안
2. 무인 항공기의 기내 보안에 맞게 여러 AI모델의 실시간 작동을 최적화 하는 데 중점을 둠
3. 계산 부하의 균형을 맞추고 리소스 할당을 최적화 하여, 데이터의 효율적인 처리를 보장하는 것을 목표

2. Methods

Overall concept of the proposed method

- ✓ 본 연구에서는 효율적인 기내 안전 시스템 구축을 위해 위험도에 기반한 비전 AI 모델을 구성하는 방법을 제안
- ✓ 우리는 위험 단계를 4가지로 정의하고, 위험 단계에 따라 실행할 모델을 결정하는 로직을 사용함

Situation	Description
Level 1	Normal person detected (not criminal) or non-dangerous object detected
Level 2	dangerous object detected
Level 3	Detect dangerous object and normal person at once, and there is no interaction.
Level 4	Specific interaction(hold) between dangerous object and person, or dangerous person detected

Table 1. Defining risk situations

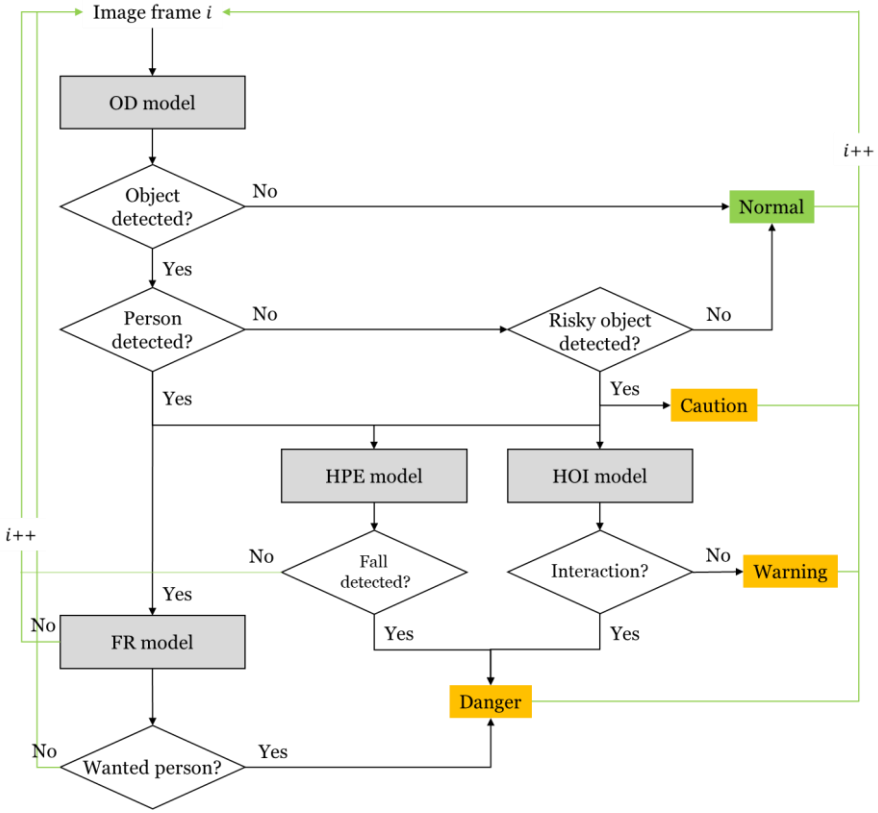


Figure1. rule-based workflow

2. Methods

Human and risky object detection

Object detection

- 본 논문은 이미지 프레임 내의 모든 사람과 위험 물품을 감지하기 위해 YOLOv8 모델을 선택
- 객체 감지 작업에서 높은 성능을 발휘하는 YOLO(You Only Look Once)모델 시리즈는 연구에 널리 사용
- 2023년 Ultralytics는 YOLO 객체 감지 모델의 최신 버전인 YOLOv8을 출시

성능 향상을 위해 크게 2가지의 수정사항을 반영

1. Anchor-free design
   (기존)미리 정해진 앵커 박스에 의존하여 객체의 위치와 치수 추정
   (수정)앵커 박스의 제약없이 객체의 위치와 크기 직접 예측(정확도 향상)
2. Replace the first 6x6 Conv with a 3x3 Conv
   백본 모델의 커널 크기가 절반으로 줄어듬(파라미터 감소)

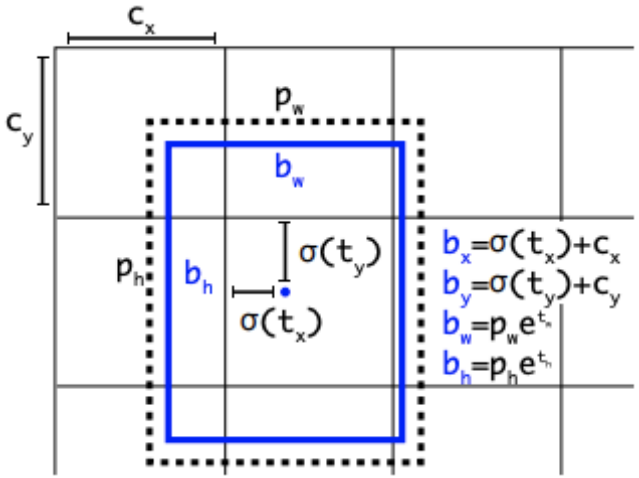


Figure 2. Anchor-based design

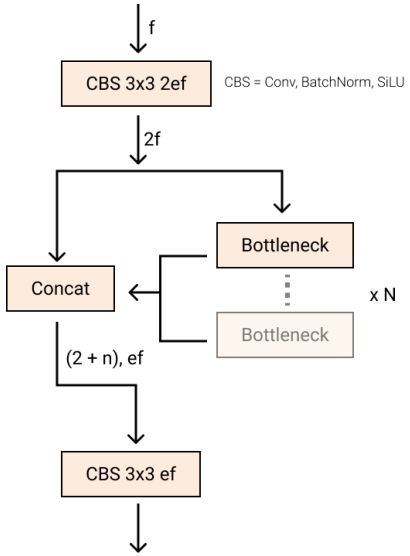


Figure 3. 3x3 Conv in the Backbone

2. Methods

Human and risky object detection

Object detection

- 이러한 변경사항을 가진 YOLOv8은 기존의 분류기 기반 시스템보다 여러가지 장점이 있음
  1. 이미지의 전체적인 컨텍스트를 고려
  2. R-CNN기반 모델과 달리 하나의 네트워크 평가로 예측을 수행(빠른 속도)

- 우리의 목표를 달성하기 위해서는 실시간 추론이 필요하기 때문에, 속도 측면에서 경쟁 모델보다 우위에 있는 YOLOv8 선택
- 저자가 제공한 사전 훈련된 모델은 추가적인 훈련 없이도 성능이 우수하나, 특정 위험 물품에 대한 추가학습 진행

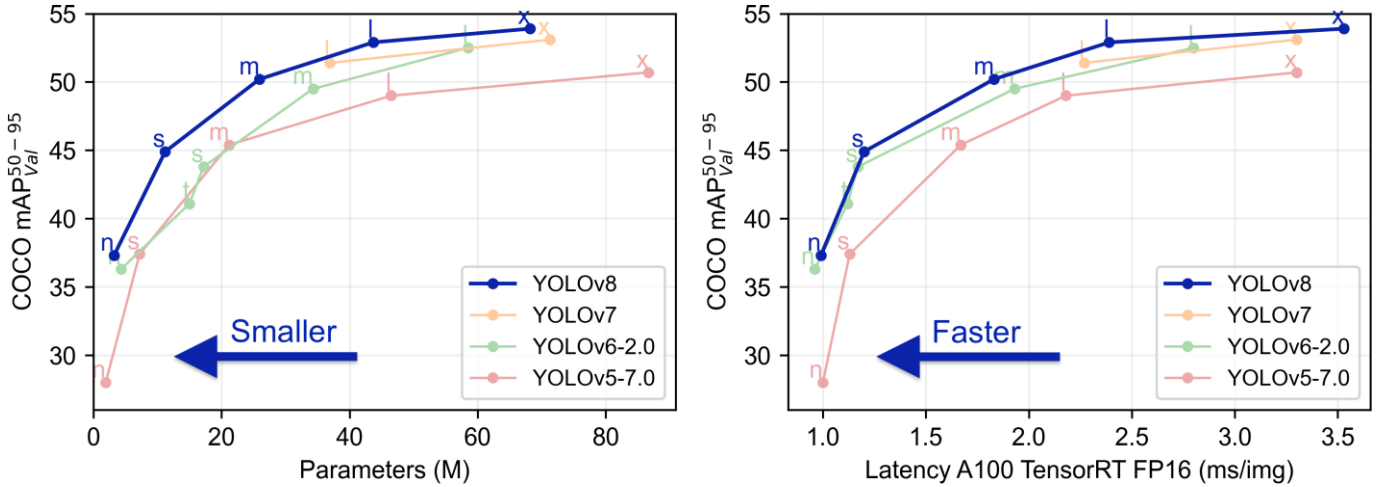
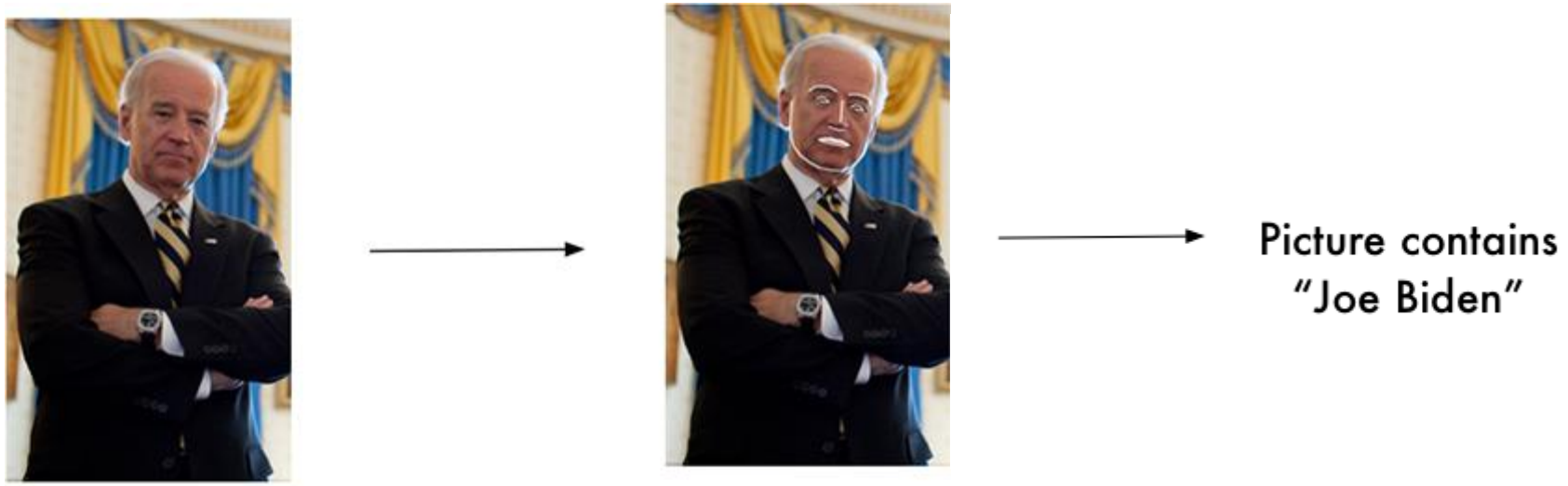


Figure 4. Performance of different YOLOv8 models taken from YOLOv8 GitHub repository

2. Methods

Face recognition

- 본 논문은 이미지 내에 사람이 인식되면, 그 사람의 신원을 확인하여 범죄자를 식별하기 위해 Face recognition을 실행함
- Face detection을 수행한 뒤, 특징 점(landmark)을 추출하고 KNN을 사용하여 학습된 이미지에서 유사한 인물 선정
  - 특징 점 종류 : 턱, 좌측 눈썹, 우측 눈썹, 콧대, 코 끝, 좌측 눈, 우측 눈, 윗입술, 아랫입술
- Face recognition 모델 중 가장 경량화 된 모델로, 기존 연구에 따르면 cpu에서 가장 빠른 속도를 보이므로 위 모델을 선정



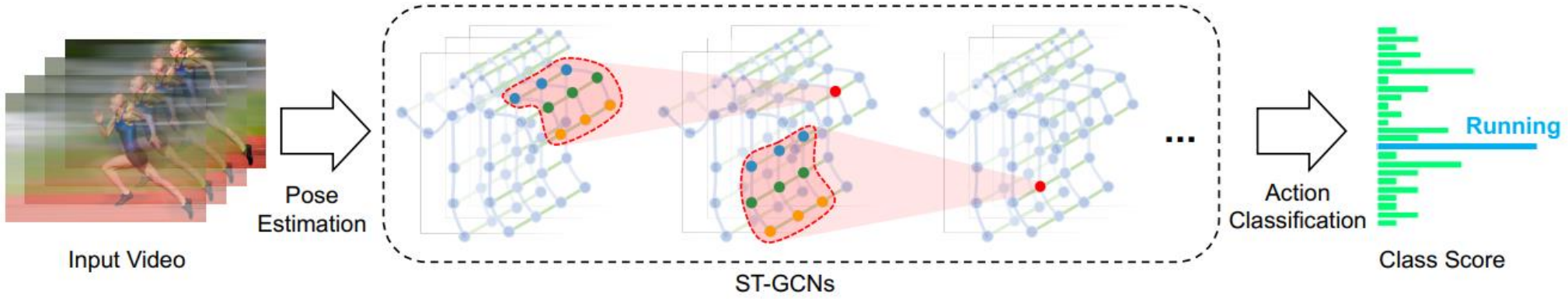
Dlib Face recognition: https://github.com/ageitgey/face_recognition

2. Methods

Fall detection

1. YOLOv3-Tiny 모델을 Person만 학습시켜서 person에 대한 detection을 실시
2. AlphaPose 모델을 사용하여 골격 추출
3. ST-GCN 모델을 사용하여 동작 예측
   - Action Label : Standing, Walking, Sitting, Lying Down, Stand up, Sit down, Fall Down

Fall Down-Lying down 상태가 n초 이상 지속되면 위험 상태로 판단

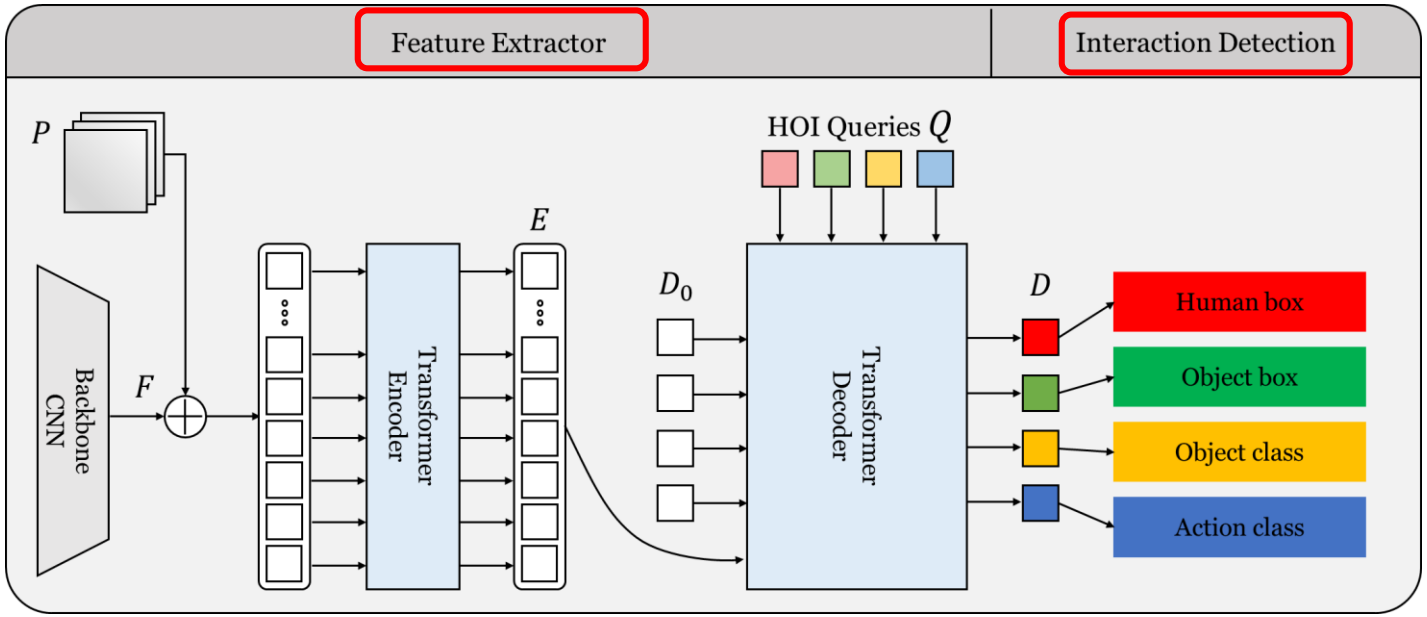


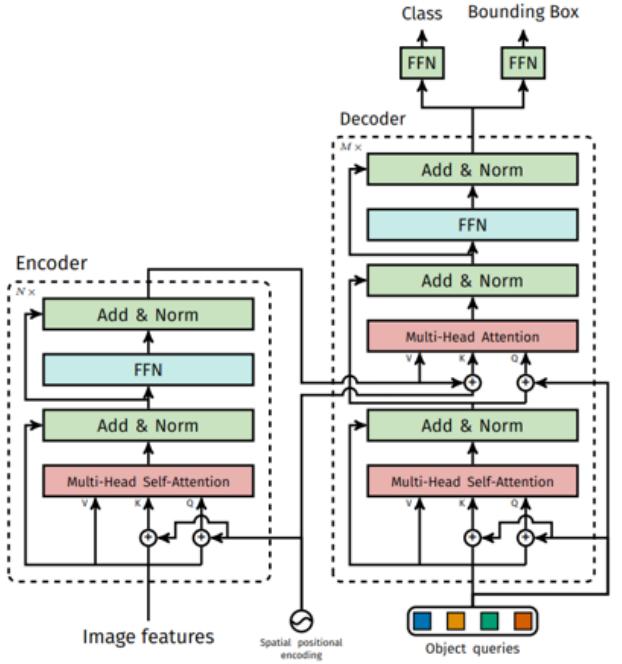
ST-GCN : Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition : https://arxiv.org/pdf/1801.07455.pdf
AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time : https://arxiv.org/abs/2211.03375
YOLOv3: https://github.com/ultralytics/yolov3

2. Methods

Human-Object interaction detection

- ✓ QPIC은 DETR(End-to-End Object Detection with Transformers)를 기반으로 인간-사물-상호작용 쌍을 예측하는 모델
- ✓ Transformer encoder-decoder 구조를 사용하여 image-wide한 feature를 추출
- ✓ FFN Layers 를 통해 Human bbox, Object bbox, object class, action class를 출력



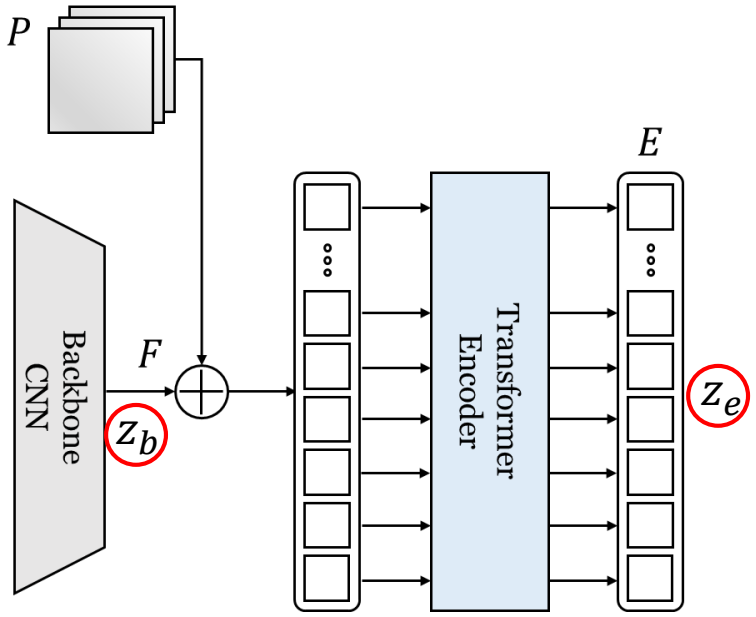


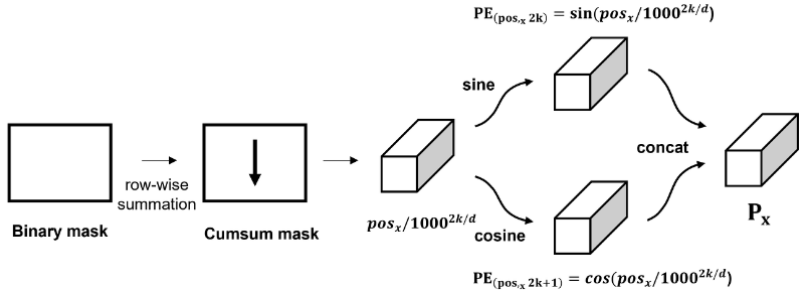
2. Methods

Human-Object interaction detection

Feature Extraction (Transformer Encoder)

1. 인코더는 $z_b \in \mathbb{R}^{D_c \times H' \times W'}$의 특징 맵 z_b와, positional information를 포함하는 $p \in \mathbb{R}^{D_c \times H' \times W'}$을 입력으로 함
2. Self-Attention 메커니즘 사용 → 특징 맵 $z_e \in \mathbb{R}^{D_c \times H' \times W'}$을 추출(Image-wide contextual information 포함)





Positional encoding 얻는 방법

1. Feature map과 동일한 해상도의 binary mask에 대하여 y축 방향으로 누적합을 구함
   누적합 : y축에 대한 절대적인 위치
2. 각 위치에 대한 sin, cos 변환을 수행
3. X축에 대해서도 동일하게 수행
4. 각 결과를 합쳐서 feature map과 동일한 차원의 positional encoding 을 구함

이렇게 도출된 P는 Encoder의 모든 multi-head self-attention layer에서 query와 key에 더해짐

2. Methods

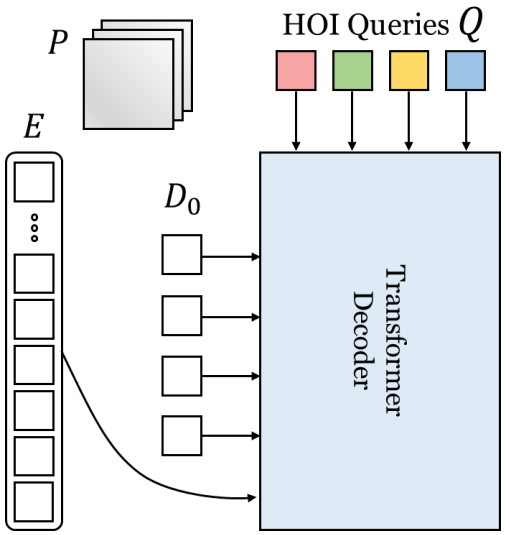
Human-Object interaction detection

Feature Extraction (Transformer Decoder)

1. 디코더는 인코더에서 추출 된 $z_e \in \mathbb{R}^{D_c \times H' \times W'}$ 특징맵, positional encoding p, HOI 쿼리 $Q = \{q_i | q_i \in \mathbb{R}^{D_c}\}_{i=1}^{N_q}$ 를 입력으로 함

   Q 벡터란 : 나타날 수 있는 모든 Human-object-interaction

2. 인간-물체 상호작용을 감지하기 위해, 이미지 전체의 정보를 포함한 embedding vector $\mathrm{D} = \{d_i | d_i \in \mathbb{R}^{D_c}\}_{i=1}^{N_q}$ 를 출력



HOI 쿼리 Q 예시

탐지할 수 있는 Object가 person, cup, chair만 있다고 가정

person-cup-잡다(o)
person-chair-앉다(o)
person-cup-먹다(x)

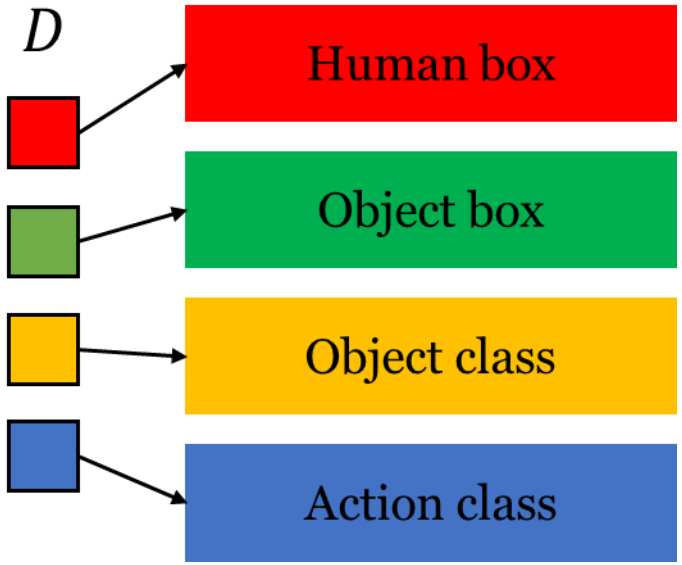
이러한 벡터들이 사전에 입력되어 들어감으로써, 쿼리에 대한 제한을 둠

2. Methods

Human-Object interaction detection

Interaction detection

디코더의 출력에 FFN을 적용하여 4개의 결과를 얻고, 임계값을 넘는 모든 {Human, Object, Action} 을 출력



3. Experiment

Data description

Object detection

- Roboflow weapon dataset 활용
- Train 7729 / Val 509 / Test 187
- Class : 도끼, 칼, 권총, 커터칼 등 30개
- Epoch 100 / batch size 16

Face recognition

- VGGFace2 dataset 중 95명 이미지 활용
- Train: 인물 별 50~300 장 이미지
- Test: 81장 이미지

HOI detection

- V-COCO & Aihub의 "행동 분류 및 상호작용 인식용 한국형 비전 데이터"
- Train 3000 / Val 2867 / Test 163
- Action class : 29개
- Epoch 150 / batch size 2

3. Experiment

Evaluation Metric

Object detection

- 정확도
- 정밀도
- 민감도
- mAP(mean average precision)
- Inference time

Face recognition

- 정확도
- Inference time

HOI detection

- 정확도
- 정밀도
- 민감도
- mAP
- 클래스 별 AP(average precision)
- Inference time

3. Experiment

Experiment

Object detection

Model	precision	Recall	mAP(50)	mAP(50-95)	Inference Time(1 image)
YOLOv8	0.487	0.627	0.569	0.408	0.005(s)

Face recognition

Model	Accuracy	Inference Time(1 image)
YOLOv8	0.927	0.089(s)

3. Experiment

Experiment

HOI detection

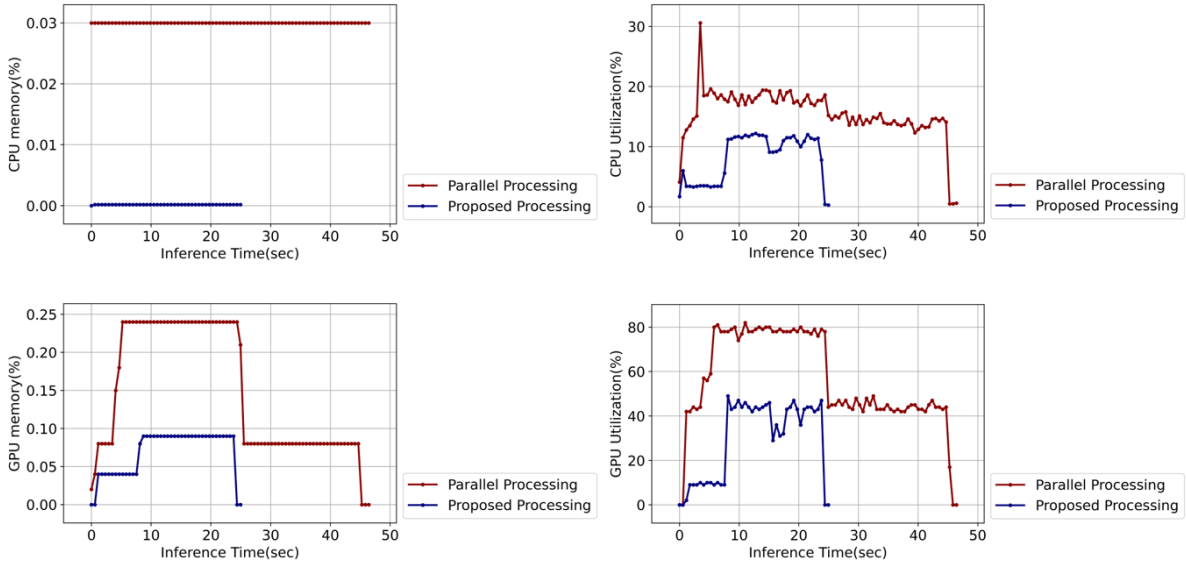
Model	Backbone	FPS	mAP	Accuracy	precision	Recall	AP(잡다)	Inference Time
Qpic-R50	resnet50	120	65.6	67.5	64.1	53.2	48.5	0.008(s)
Qpic-R101	resnet101	110	70.3	73.6	66.7	55.6	48.2	0.009(s)

3. Experiment

Experiment

Measuring resources (최신화 X)

	Conventional sequence	Proposed method
Cpu util(%)	14.70	7.13
Cpu memory(%)	0.03	< 0.01
Gpu util(%)	52.27	25.29
Gpu memory(%)	0.14	0.06
Inference time(s)	46.6	29.1



4. Conclusion

생각해봐야 할 부분

- ✓ 리소스 측정의 다양화
- ✓ 시나리오를 여러 개 짜야하지 않을까
- ✓ 위험 단계를 잘 탐지했다 이런 결과가 나와야 하지 않을까

Thanks