JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 31 No. 3, pp. 49-59, March 2026
https://doi.org/10.9708/jksci.2026.31.03.049

Training-Free Feature-Level Interpolation with Sparse-Backbone Scheduling for YOLO Detection in Low-FPS Video

Min-Ho Kim*, Kyu-Cheol Cho**

*Student, Dept. of Computer Science, Inha Technical College, Incheon, Korea
**Professor, Dept. of Computer Science, Inha Technical College, Incheon, Korea

[Abstract]

In this paper, we propose a training-free method that mitigates degradation in accuracy and temporal stability while improving system efficiency for YOLO-based detectors in low-frame-rate (Low-FPS) environments. The proposed approach preserves the YOLO backbone and neck, caches feature maps (P3–P5) right before the Detection Head at Anchor frames, and operates in a Sparse-Backbone manner by performing head-only inference on linearly interpolated features from neighboring Anchors for non-anchor frames. Under a system-level end-to-end(E2E) protocol (5 independent runs, mean ± std), it reduces backbone invocations by about 50% while retaining 98.48% of the baseline mean Average Precision at IoU 0.5 (mAP@50), decreasing processing latency (Lat_proc) by 26.60% and improving processing throughput (FPS_proc) by 36.11%. These results confirm that the proposed method can serve as a practical training-free inference module that preserves detection quality and temporal stability while improving system efficiency in Low-FPS and resource-constrained deployments.

▸ **Key words**: Feature-Level Interpolation, Low-FPS Video, YOLO, Training-Free Module, Object Detection

[요 약]

본 연구는 저프레임(Low-FPS) 환경에서 YOLO 기반 탐지기의 정확도 및 시간적 안정성 저하를 완화하고 시스템 효율을 향상시키는 비학습형 방법을 제안한다. 제안 기법은 YOLO의 Backbone–Neck을 유지한 채 Anchor 프레임에서 Detection Head 입력 직전 특징 맵(P3–P5)을 저장하고, 비-Anchor 프레임은 인접 Anchor 특징을 선형 보간한 뒤 head-only 추론을 수행하는 Sparse-Backbone 구조로 동작한다. System-level end-to-end(E2E) 기준(5회 반복, mean ± std)에서 Backbone 호출을 약 50% 줄이면서 mAP@50(mean Average Precision at IoU 0.5)의 98.48%를 유지하였고, 처리 지연 시간(Lat_proc)은 26.60% 감소, 프레임 처리량(FPS_proc)은 36.11% 향상되었다. 이를 통해 제안 방법이 저프레임·자원 제약 환경에서 탐지 품질과 시간적 안정성을 유지하면서 시스템 효율까지 개선할 수 있는 실용적인 비학습형 추론 모듈임을 확인하였다.

▸ **주제어**: 특징 레벨 보간, 저프레임 영상, 욜로, 비학습형 모듈, 객체 탐지

• First Author: Min-Ho Kim, Corresponding Author: Kyu-Cheol Cho
*Min-Ho Kim (btlime3@gmail.com), Dept. of Computer Science, Inha Technical College
**Kyu-Cheol Cho (kccho@inhatc.ac.kr), Dept. of Computer Science, Inha Technical College
• Received: 2026. 01. 09, Revised: 2026. 03. 08, Accepted: 2026. 03. 12.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

최근 드론 감시, 지능형 CCTV, 자율주행 등 영상 기반 지능형 시스템의 확산에 따라 비디오 스트림에서 다수 객체를 실시간으로 안정적으로 탐지하는 기능의 중요성이 커지고 있다[1]. 현장 적용 관점에서는 1-stage 탐지기인 You Only Look Once(이하, YOLO) 계열이 단순한 구조와 빠른 추론 속도를 바탕으로 널리 활용되어 왔다[2][3]. 실제 운용 환경에서는 네트워크 대역폭, 저장 용량 및 엣지 디바이스의 제한된 연산 자원으로 인해 고비용 탐지기를 매 프레임 실행하기 어렵다[4]. 저프레임(Low-FPS) 조건에서는 프레임 간 시간 간격이 증가하면서 객체 이동이 불연속적으로 관측되고, 그 결과 검출 성능 저하 및 시간적 일관성 저하가 발생할 수 있음이 비디오 객체 탐지 연구들에서 논의되어 왔다[5][6].

이를 완화하기 위한 접근으로 비디오 프레임 보간(Video Frame Interpolation, 이하 VFI)을 통해 중간 프레임을 복원하여 시간 해상도를 높이는 연구가 제안되어 왔다[7][8][9]. 그러나 픽셀 공간에서 중간 프레임을 생성하는 방식은 모션 경계나 가려짐(occlusion) 구간에서 시각적 왜곡(블러, 고스트 등)이 발생하기 쉬우며[10], 생성 프레임이 탐지 입력으로 사용될 때 예측 결과의 변동으로 이어질 수 있다. 또한 보간으로 프레임 수가 증가하는 경우, 생성된 모든 프레임에 대해 탐지 네트워크를 반복 수행해야 하므로 연산 부담이 누적될 수 있다[4]. 한편 비디오 전용 탐지기나 추적 기반 접근은 시간 정보를 활용할 수 있으나[6][11][12], 모델 구조 변경이나 별도 학습 절차가 수반되는 경우가 있어 이미 배포된 탐지기에 즉시 적용하기에는 제약이 존재하였다.

본 연구에서는 이러한 제약을 고려하여, 모델 구조 변경이나 재학습 없이 추론 단계에만 삽입 가능한 특징 맵 수준의 프레임 보간 기법(이하, Feature-Level VFI)을 제안하였다. Feature-Level VFI는 픽셀 프레임을 생성하는 대신 Detection Head 입력 직전의 특징 맵(feature map)을 시간 축으로 보간하여 중간 시점의 표현을 구성하는 방식으로 정의하였다. 제안 방법은 YOLO의 구조 및 가중치를 유지한 채[2][3], 짝수 프레임(Anchor)에서만 특징 맵을 추출·저장하고 홀수 프레임에서는 인접 Anchor 특징을 선형 보간하여 생성된 특징에 대해 head-only 추론을 수행하도록 Sparse-Backbone 스케줄링을 적용하였다[5][13]. 이를 통해 계산량이 큰 Backbone 영역은 Anchor 구간에서만 수행하고, 나머지 구간에서는 저장된 특징을 재사용하여 연산 효율 향상을 도모하였다. 또한 출력 프레임 수를 유지하면서 시간 축 정보 손실을 완화하는 추론 모듈로 활용될 수 있다.

- Training-free
- Sparse-backbone scheduling
- Head-input(P3-P5) feature interpolation
- System-level E2E + temporal stability 평가

본 논문의 구성은 다음과 같다. II장에서는 객체 탐지, 비디오 프레임 보간, 저프레임 환경에서의 문제를 정리하고 관련 연구를 고찰한다. III장에서는 제안하는 Feature-Level VFI와 Sparse-Backbone 추론 파이프라인을 설명한다. IV장에서는 비교방법, 평가 지표, 실험 설정 및 결과를 제시하고 분석한다. V장에서는 결론과 한계, 향후 연구 방향을 논의한다.

II. Related Work

2.1 Object Detection

딥러닝 기반 객체 탐지기는 크게 2-stage 방식과 1-stage 방식으로 구분된다[1]. 2-stage 방식은 후보 영역 생성 후 각 영역을 분류·회귀하므로 정확도가 높은 경향이 있으나, 연산량과 지연시간이 증가하기 쉽다. 반면 YOLO, SSD, RetinaNet 등 1-stage 방식은 한 번의 전방향 계산으로 위치와 범주를 동시에 예측하므로 실시간 응용에 적합하다[2][3]. 최근에는 DETR 계열과 비디오의 시간 정보를 활용하는 탐지 기법도 제안되었으나[4][5][6][11][12], 처리 지연과 구현 복잡도가 중요한 환경에서는 단순한 추론 파이프라인과 높은 처리량을 갖는 YOLO 계열이 여전히 실용적 기준선으로 활용된다[2][3]. 이에 따라 본 연구는 YOLO 계열의 Backbone-Neck-Head 분리 구조를 전제로 하며, 추론 단계의 스케줄링과 중간 표현 재사용을 통해 저프레임 문제를 완화하는 방향을 다룬다[5][13].

2.2 YOLO Family and the Scope of YOLO11

YOLO 계열은 실시간 객체 탐지를 목표로 발전해 온 1-stage 탐지기 계열이다[2][3]. 본 연구의 YOLO11은 새롭게 정의한 구조가 아니라, Ultralytics가 제공하는 공식 YOLO11 모델군 중 object detection용 구현체인 YOLO11m detect를 의미한다. Ultralytics는 YOLO11을 detection, segmentation, classification, pose, oriented bounding box (OBB) 등 task-specific

variant로 제공하며, 본 연구는 이 중 detection variant만을 대상으로 한다. 또한 본 연구의 기여는 YOLO11 자체의 구조나 학습 절차를 변경하는 데 있지 않다. 실험에서는 COCO-pretrained 가중치로 초기화한 뒤 데이터셋에서 100 epochs 학습한 동일 checkpoint를 모든 비교 방법에 공통 적용하였으며, Detect head 입력 직전의 P3-P5 feature maps에만 캐싱, 선형 보간, head-only 추론을 적용한다. 즉, 비교는 서로 다른 detector 간이 아니라 동일한 YOLO11m detect와 동일 checkpoint에 대해 추론 단계 실행 전략만 달리한 비교이다.

2.3 Video Frame Interpolation

비디오 프레임 보간(VFI)은 시간적으로 떨어진 두 프레임 사이의 중간 시점 프레임을 추정·생성하여 실질적인 프레임레이트를 높이는 기술이다[4]. 다양한 VFI 방법이 제안되어 왔으며[7][8][9][14], 딥러닝 기반 방법은 중간 프레임 품질을 향상시키는 방향으로 발전해 왔다.

그러나 픽셀 공간에서 중간 프레임을 직접 생성하는 방식은 빠른 이동, 모션 경계, 가려짐 구간에서 시각적 왜곡이 발생하기 쉬우며[10][15], 생성된 프레임이 후속 탐지 입력으로 사용될 때 예측 결과의 변동을 유발할 수 있다. 또한 보간된 프레임까지 포함하여 객체 탐지 네트워크를 반복 수행할 경우 연산 부담이 증가할 수 있다[4]. 이에 따라 본 연구는 비교를 위해 픽셀 기반 프레임 보간 기법(이하, Pixel-Level VFI)과, 대표적인 딥러닝 기반 비디오 프레임 보간(deep-learning-based video frame interpolation, 이하 DL-VFI) 기법인 RIFE(Real-Time Intermediate Flow Estimation)를 테스트 케이스로 두고, III장에서 픽셀 프레임 생성 없이 특징 공간에서 시간 정보를 보완하는 Feature-Level VFI를 제안한다.

2.4 Low-FPS Environments

드론 감시, CCTV, 로봇 비전과 같은 응용 환경에서는 네트워크 대역폭 제한, 저장 용량, 센서 및 연산 자원 제약으로 인해 영상이 충분한 프레임레이트로 획득·처리되지 못할 수 있다[1][4]. 저프레임 환경에서는 프레임 간 시간 간격이 증가하여 객체 이동이 불연속적으로 관측될 수 있으며[5], 그 결과 탐지 결과의 연속성과 안정성이 저하될 수 있다. 본 연구에서는 가장 단순한 Low-FPS 조건으로 stride = 2(즉, 1/2 downsampling)를 기본 설정으로 사용하였고, 추가로 stride = 3/4 및 random, burst, jittered 조건까지 확장하여 평가하였다.

III. Feature-Level VFI

저프레임(Low-FPS) 환경에서는 프레임 간 시간 간격 증가로 객체 이동이 불연속적으로 관측되어 박스 위치·크기 변동과 검출 단절이 확대될 수 있다. 또한 자원 제약 환경에서 매 프레임 Backbone-Neck을 수행하는 방식은 시스템 처리량과 지연시간 측면에서 부담이 크다. 본 장에서는 모델 구조 변경이나 재학습 없이 추론 단계에 삽입 가능한 Feature-Level VFI를 제안하며, Sparse-Backbone 스케줄링과 특징맵 보간으로 시간적 안정성과 system-level E2E 효율을 함께 개선하는 동작 원리를 정리한다.

3.1 Sparse-Backbone Inference Strategy

제안하는 Sparse-Backbone 전략은 프레임을 Anchor 프레임과 Interpolated 프레임으로 구분하고 Anchor에서만 Backbone-Neck을 실행하도록 고정 스케줄링하는 추론 방식이다. 여기서 “Sparse”는 가중치 희소화(pruning/sparsity)가 아니라 Backbone-Neck 실행 빈도를 스케줄링으로 희소화한다는 의미이다. 기본 설정에서는 짝수 인덱스=Anchor, 홀수 인덱스=Interpolated로 정의한다(Anchor stride=2). Anchor 프레임에서는 YOLO11의 Backbone과 Neck을 실행하여 다중 해상도 특징 맵(P3-P5)을 계산하고 Detection Head 입력 직전의 특징을 버퍼에 저장한다.

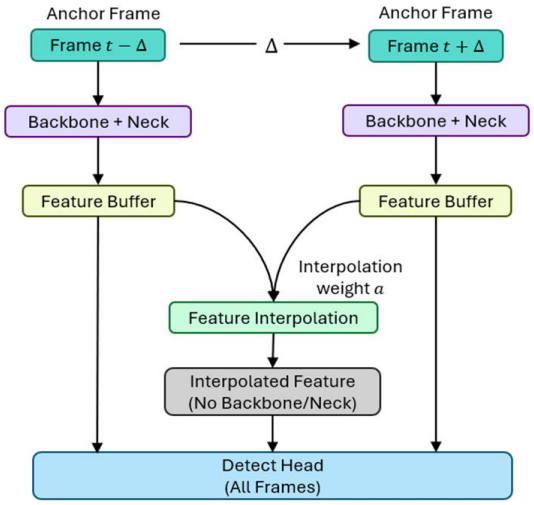

Fig. 1. Proposed Feature-Level VFI Inference Pipeline

Interpolated 프레임에서는 Backbone-Neck 없이 인접한 두 Anchor의 저장 특징을 시간 축에서 보간한 뒤 head-only 추론을 수행한다. 이로써 출력 프레임 수(원본

타임라인)는 유지하면서 Backbone-Neck 호출만 감소한다. stride=3/4 및 불규칙 드롭(random/burst/jittered)에 대한 민감도와 강건성은 IV장에서 추가 검증한다.

3.2 Feature Extraction and Interpolation

본 연구는 YOLO11의 Detection Head 입력 직전 멀티스케일 특징 맵(P3-P5)을 대상으로 Feature-Level 보간을 설계하였다. Anchor 프레임의 특징을 저장하고, Interpolated 프레임에서는 동일 텐서 구조의 중간 시점 특징 맵을 시간 가중치 기반 선형 보간으로 구성한다.

$$F_{t+\alpha}^{(s)} = (1-\alpha)F_t^{(s)} + \alpha F_{t+\triangle}^{(s)}, 0<\alpha<1 \quad \cdots (1)$$

식 (1)에서 사용된 기호의 의미는 다음과 같다.

- t: Anchor 프레임의 시간 인덱스.
- s: 특징 맵의 스케일 인덱스(예: P3, P4, P5).
- $\triangle$: Anchor 간격(stride)
- $\alpha \in [0,1]$: 두 Anchor 프레임 사이에서의 상대적 시간 위치를 나타내는 보간 비율(가중치).
- $F_t^{(s)}$: Anchor 프레임 t에서 추출된 스케일 s의 특징 맵.
- $F_{t+\triangle}^{(s)}$: 다음 Anchor 프레임 t+$\triangle$에서 추출된 스케일 s의 특징 맵
- $F_{t+\alpha}^{(s)}$: 두 Anchor 사이의 상대적 위치에서 보간으로 생성된 스케일 s의 특징 맵.

연구의 기본 설정(Anchor stride = 2)에서는 보간 프레임이 두 Anchor의 중간 시점이므로 α = 0.5로 고정한다. stride = 3/4에서는 여러 중간 프레임의 상대 위치에 따라 α를 달리 적용할 수 있으며, 불규칙 샘플링에서도 인접 Anchor 사이 상대 위치로 α를 정의한다(IV장에서 실험으로 확인). 선형 보간은 학습 없이 적용 가능하며 텐서 연산(덧셈/스칼라 곱)만으로 구현된다. 또한 Pixel-Level VFI와 달리 픽셀 프레임을 생성하지 않고 탐지에 직접 쓰이는 표현(P3-P5)을 시간 축에서 구성하므로, 픽셀 왜곡 전파를 줄이는 방향의 설계이다.

3.3 Inference Pipeline and Complexity

제안 기법은 추론 루프에서 (i) Anchor 프레임: Backbone-Neck-Head 실행 및 P3-P5 저장, (ii) Interpolated 프레임: Feature interpolation + head-only 실행으로 분기한다(Fig. 1). 보간은 동일 시퀀스 내부에서만 수행하며 시퀀스 경계를 넘지 않는다. 모든 프레임에서 Detection Head는 1회 실행되므로 출력 타임라인은 유지되고 Backbone-Neck 실행 빈도만 감소한다. 전체 비용은 다음과 같다

$$C_{prop} \approx N_a(C_{BN}+C_H) + (N-N_a)(C_I+C_H) \quad (2)$$

식 (2)에서 사용된 기호의 의미는 다음과 같다.

- N: 전체 프레임 수.
- N_a: Anchor 프레임 수(Anchor 간격 $\triangle$에 따라 결정됨).
- C_{BN}: 한 프레임에 대한 Backbone-Neck 연산 비용.
- C_H: 한 프레임에 대한 Detection Head 연산 비용.
- C_I: Feature-Level 보간 연산 비용.
- C_{prop}: 제안 기법의 전체 추론 비용.

C_I는 텐서 덧셈과 스칼라 곱 중심의 연산이므로 일반적으로 C_{BN}에 비해 작은 비용으로 취급된다. 메모리 측면에서는 인접 Anchor 특징(P3-P5)을 위한 추가 버퍼가 필요하지만, 저장 대상을 Head 입력 직전 P3-P5와 최근 Anchor 쌍으로 제한하므로 증가 폭은 제한적이다. IV장에서는 모든 비교 방법에 동일한 system-level E2E 측정 프로토콜을 적용하여 효율을 공정하게 비교한다.

IV. Experimental Results

본 연구는 VisDrone2019-VID-test-dev 비디오 데이터셋을 대상으로 수행하였다. 모든 비교 방법에 대해 동일한 입력 해상도 및 전처리 파이프라인을 적용하여 조건을 통제하였다. 본 장의 변화율(%)은 별도 언급이 없는 한 Baseline 대비 상대 변화율이며, FPS 및 Latency는 평균±표준편차(mean±std)로 보고한다.

4.1 Experimental Protocol and Reproducibility

효율 지표는 system-level E2E 범위에서 측정하였다. Low-FPS는 출력 프레임이 감소하므로 처리량/지연을 다음과 같이 분리 정의한다.

- FPS_out: output frames / wall time (원본 타임라인 기준)

- FPS_proc: processed frames / wall time (실제 처리 기준)
- Lat_out(ms), Lat_proc(ms): 각각 1,000/FPS_out, 1,000/FPS_proc

정확도와 시간적 안정성은 원본 타임라인(6,635 frames) 기준으로 정렬해 산출하였다. Low-FPS는 관측 프레임만 처리하므로, 원본 타임라인 정렬 시 누락 시점은 빈 검출(박스 없음)로 처리하여 프레임 드롭이 연속성에 미치는 영향을 보수적으로 반영하였다. Pixel-Level VFI 및 DL-VFI (RIFE)의 프레임 복원 비용은 system-level E2E 시간에 포함하였다. 반복 측정은 5회이며, warmup 50 프레임은 제외하였다. 런타임 지표는 wall-time 특성상 분산이 존재하므로 mean±std로 보고한다.

Table 1. System environment

Parameter	Value
OS	Ubuntu 24.04 LTS
CPU	Intel Xeon Gold 6240 x 2
RAM	DDR4-2933 512GB
GPU	Nvidia Quadro RTX8000 48GB
CUDA	11.8
PyTorch	2.3.0+cu118
Ultralytics	v8.4.13
Precision	FP32

Table 2. Runtime control

Parameter	Value
Dataset	VisDrone2019-VID-test-dev (17 sequences, 6,635 frames)
Input / batch	640x640, batch=1
Thresholds	conf=0.25, IoU=0.5
Runs / warmup	5 runs, warmup 50 frames excluded

Table 3. Metric definitions and reporting policy

Parameter	Value
Evaluation alignment	Original timeline (6,635 frames)
FPS_out	output frames / wall time
FPS_proc	processed frames / wall time
Lat_out / Lat_proc	1,000/FPS_out and 1,000/FPS_proc (ms)
E2E Scope	Preprocess + (VFI if used) + YOLO inference + postprocess
Same pipeline	Same preprocessing/postprocessing and identical thresholds (conf/IoU)

4.2 Comparison Methods

실험은 다음 5가지 설정으로 비교하였다.

- Baseline (Original): 모든 원본 프레임에 대해 YOLO 표준 추론(Backbone-Neck-Head).
- Low-FPS (1/2 Downsampled): 원본 시퀀스의 절반 프레임만 관측(홀수 프레임 제거)하고 관측 프레임만 표준 추론.
- Pixel-Level VFI (Restoration on Low-FPS): Low-FPS에서 누락 시점을 픽셀 선형 보간으로 복원 후 전체 프레임에 표준 추론(E2E에 복원 비용 포함).
- DL-VFI(RIFE) (Restoration on Low-FPS): Low-FPS에서 DL 기반 VFI(RIFE)로 중간 프레임 복원 후 전체 프레임에 표준 추론(E2E 복원 비용 포함).
- Feature-Level VFI (Restoration on Low-FPS): Anchor에서 Head 입력 직전 특징(P3-P5) 저장, 누락 시점은 특징 공간 선형 보간으로 구성 후 head-only 추론(구조/가중치는 Baseline과 동일).

4.3 Evaluation Metrics

정적 탐지 성능은 mean Average Precision at IoU 0.5 (mAP@50), mean Average Precision over IoU 0.5:0.95 (mAP@50:95), 그리고 F1 score로 측정하였다. 시간적 안정성은 Center Jitter, Scale Jitter, Flicker Rate(%), Streak Length로 정량화하였다. 이러한 안정성 지표는 후속 처리 및 이벤트 검출에서의 체감 품질과 직접 연결되며, Jitter 증가는 궤적 불안정과 IDSW(identity switches) 증가로, Flicker 증가는 단속적 검출로 인한 경보 품질 저하로 이어질 수 있다. Streak Length는 동일 객체의 연속 검출 유지 정도를 나타내어 궤적 단절을 간접적으로 반영한다. 또한 tracking-by-detection 기반의 IDF1(identification F1 score), IDSW(identity switches), Fragments 및 정규화 지표인 IDSW/1kTP(IDSW per 1,000 true positives), Frag/1kTP(Fragments per 1,000 true positives)도 함께 평가한다. 작은 성능 차이 해석을 위해 시퀀스 단위 paired bootstrap 95% confidence interval (CI)을 4.8에서 제시한다.

4.4 Detection Performance

검출 성능은 mAP@50, mAP@50:95, F1 Score로 비교하였다(Fig. 2-3).

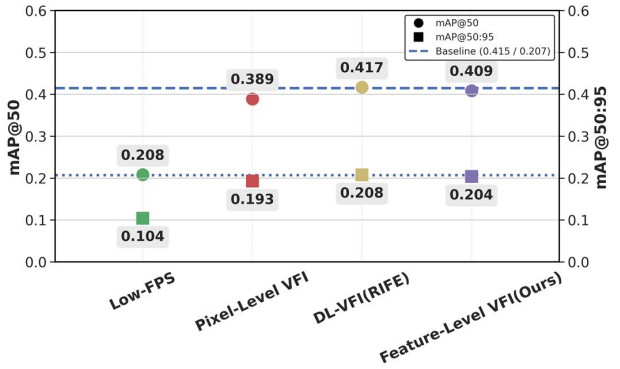

Fig. 2. Detection Performance – mAP@50 and mAP@50:95

원본 타임라인 정렬 기준, Low-FPS는 프레임 누락이 탐지 공백으로 반영되어 mAP@50과 mAP@50:95가 모두 약 49.79% 감소한다. Pixel-Level VFI는 Low-FPS 대비 성능을 회복하지만 mAP@50 -6.23%, mAP@50:95 -7.02%의 하락이 남는다. DL-VFI(RIFE)는 mAP@50 +0.47%, mAP@50:95 +0.26%로 Baseline과 유사한 수준을 보인다. Feature-Level VFI는 mAP@50 -1.52%, mAP@50:95 -1.80%로 감소 폭이 제한되어 정적 정확도를 거의 유지한다.

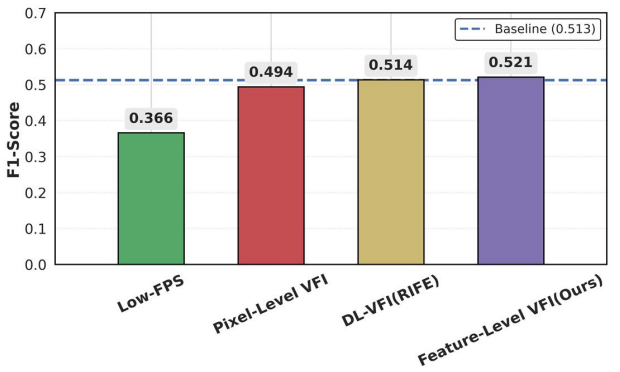

Fig. 3. Detection Performance – F1 Score

F1 Score에서도 Low-FPS는 -28.68%로 큰 폭의 감소가 나타나며, Pixel-Level VFI는 -3.74%로 감소 폭이 제한된다. DL-VFI(RIFE)는 +0.11%로 Baseline과 유사하고, Feature-Level VFI는 +1.54%로 정밀도-재현율 균형이 유지되는 경향을 보인다.

4.5 Temporal Stability and Overall Discussion

비디오 환경에서 검출 결과의 연속성과 안정성을 확인하기 위해 Center Jitter, Scale Jitter, Flicker Rate(%), Streak Length를 비교하였다(Fig. 4-6). Jitter와 Flicker는 낮을수록 안정적이며, Streak Length는 높을수록 연속 검출이 길게 유지됨을 의미한다.

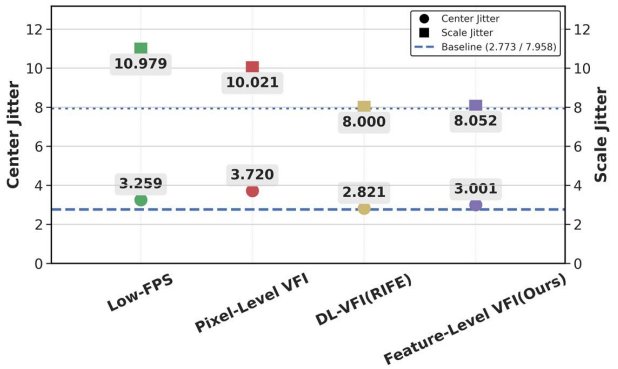

Fig. 4. Temporal Stability – Center/Scale Jitter Comparison

Low-FPS는 Center Jitter +17.54%, Scale Jitter +37.96%로 변동이 크게 증가한다. Pixel-Level VFI는 Center Jitter +34.18%, Scale Jitter +25.93%로 잔여 변동이 크다. DL-VFI(RIFE)는 +1.74% / +0.53%로 Baseline과 유사하며, Feature-Level VFI는 +8.24% / +1.19%로 증가 폭이 제한적이다.

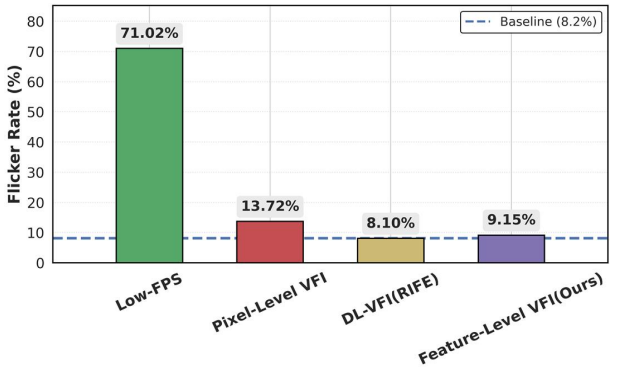

Fig. 5. Temporal Stability – Flicker Rate

Low-FPS의 Flicker는 +769.51%로 급격히 증가한다. Pixel-Level VFI는 +67.96%로 완화하지만 잔여 변동이 남는다. DL-VFI(RIFE)는 -0.82%로 Baseline과 유사하며, Feature-Level VFI는 +12.05%로 증가 폭이 제한된다(절대 변화 약 +0.98%p).

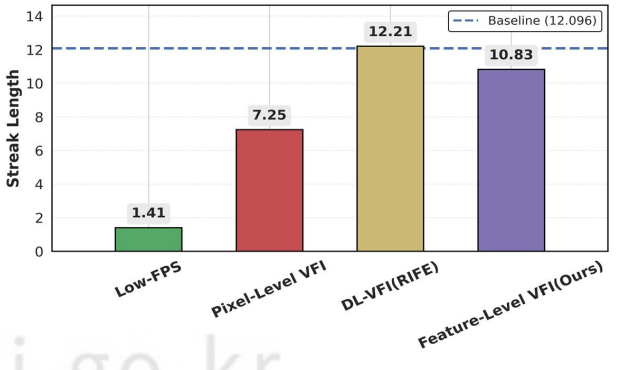

Fig. 6. Temporal Stability – Streak Length

Low-FPS는 Streak Length가 -88.36%로 크게 감소한다. Pixel-Level VFI는 -40.10%로 일부 회복되나 감소 폭이 크다. DL-VFI(RIFE)는 +0.94%로 Baseline과 유사하지만 4.6의 효율 결과와 함께 해석할 필요가 있다. Feature-Level VFI는 -10.51%로 연속 검출 저하를 제한한다.

4.6 Efficiency and Latency Analysis

효율성 평가는 동일한 system-level E2E 기준으로 수행하였다. FPS_out은 Low-FPS에서 과장될 수 있으므로 FPS_proc 및 Lat_proc 중심으로 해석한다(Table 3).

Table 4. Computational Cost - Backbone Count

Parameter	Backbone Count
Baseline	6,635
Low-FPS	3,320
Pixel-Level VFI	6,635
DL-VFI(RIFE)	6,635
Feature-Level VFI	3,320

Backbone Count는 Low-FPS와 Feature-Level VFI에서 3,320으로 감소하며, Pixel-Level VFI와 DL-VFI(RIFE)는 Baseline과 동일하게 6,635가 유지된다.

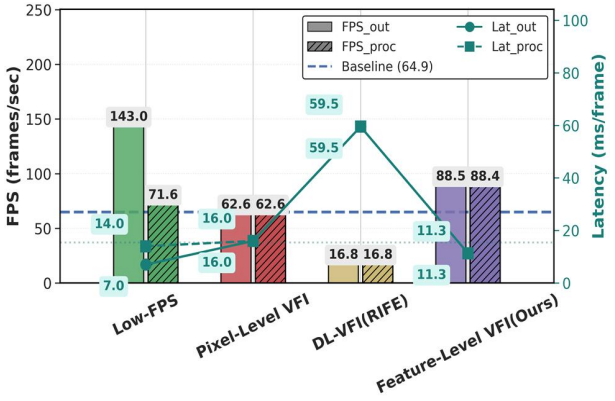

Fig. 7. System E2E Latency and FPS

Baseline은 FPS_proc 64.94±2.34, Lat_proc 15.41±0.56 ms이다. Low-FPS는 FPS_out 143.01±3.42로 증가하지만 FPS_proc은 71.56±1.71 수준이다. Pixel-Level VFI는 복원 오버헤드로 FPS_proc 62.63±3.00, Lat_proc 16.00±0.80 ms로 이득이 제한된다. DL-VFI(RIFE)는 FPS_proc 16.80±0.27, Lat_proc 59.53±0.96 ms로 system-level E2E 비용이 크다. Feature-Level VFI는 출력 타임라인을 유지하면서 FPS_proc 88.39±1.00, Lat_proc 11.31±0.13 ms로 개선된다.

4.7 Robustness Under Larger Strides and Irregular Low-FPS

기본 Low-FPS 설정인 stride=2를 넘어, stride=3/4 및 불규칙 드롭(random/burst/jittered)까지 포함하는 조건에서도 동일한 평가 정책(원본 타임라인 정렬, 시퀀스 경계 보간 금지)을 적용하였다.

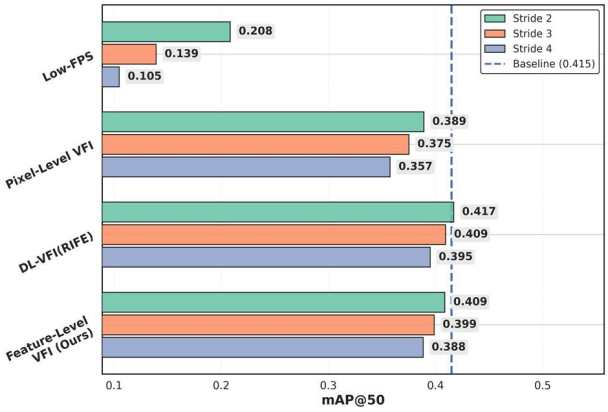

Fig. 8. Stride sensitivity(△=2/3/4) – Detection accuracy

stride가 증가할수록 Low-FPS는 탐지 공백 확대에 따라 정확도가 급격히 감소한다. Pixel-Level VFI는 Low-FPS 대비 회복 효과가 있으나 stride 증가에 따라 하락이 누적된다. DL-VFI(RIFE)는 감소 폭이 비교적 완만하며, Feature-Level VFI도 stride=3/4에서 mAP@50이 각각 -3.91%, -6.37%로 감소 폭이 제한되어 큰 간격에서도 과도한 붕괴가 발생하지 않음을 보인다.

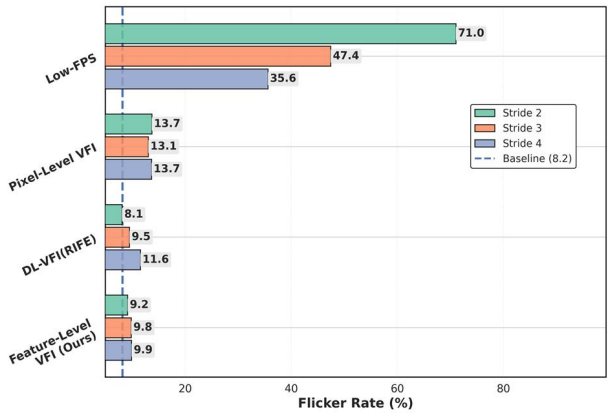

Fig. 9. Stride sensitivity (△=2/3/4) – Temporal stability

Flicker Rate(%)는 Low-FPS에서 높은 수준이 지속되어 프레임 누락이 연속성을 본질적으로 훼손함을 보여준다. Pixel-Level VFI는 Flicker를 완화하지만 Baseline 대비 잔여 변동이 남는다. DL-VFI(RIFE)는 stride 증가 시 Flicker가 증가하는 경향이 관찰되며, Feature-Level VFI는 stride 증가에도 Baseline 인근 범위에서 유지되는 경향을 보인다.

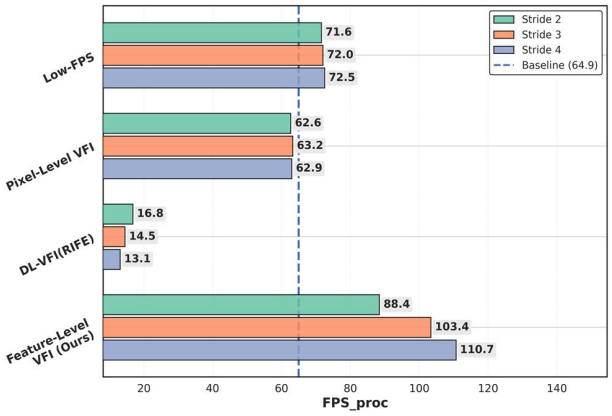

Fig. 10. Stride sensitivity (△=2/3/4) – System E2E efficiency

stride가 커질수록 Feature-Level VFI는 Backbone 호출 감소 효과가 커져 FPS_proc가 증가하는 경향을 보인다. 반면 Pixel-Level VFI는 복원 오버헤드로 처리량 이득이 제한적이며, DL-VFI(RIFE)는 복원 비용으로 FPS_proc가 크게 낮다. Low-FPS는 출력 타임라인이 감소하므로 FPS_out 해석에 주의가 필요하다.

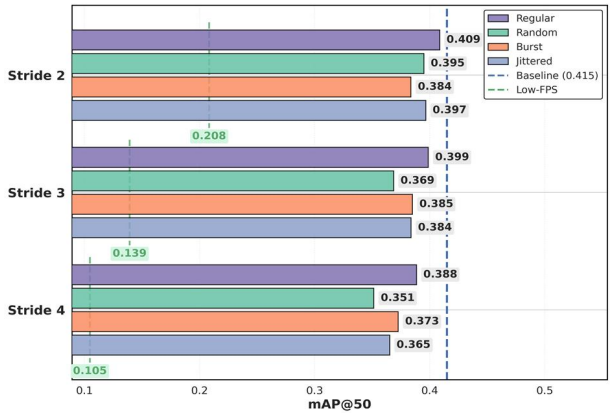

Fig. 11. Irregular Low-FPS(random/burst/jittered) – Robustness in mAP@50

실제 환경에서는 프레임 드롭이 stride 고정으로만 발생하지 않을 수 있으므로, Fig. 11에서는 평균 간격을 유지한 채 앵커 배치의 시간 분포를 random, burst, jittered로 달리하여 Feature-Level VFI의 정확도를 평가하였다. 불규칙 드롭에서도 성능 저하가 제한적이었고, burst 조건에서도 단순 드롭(Low-FPS) 대비 높은 mAP를 유지하는 경향을 보였다.

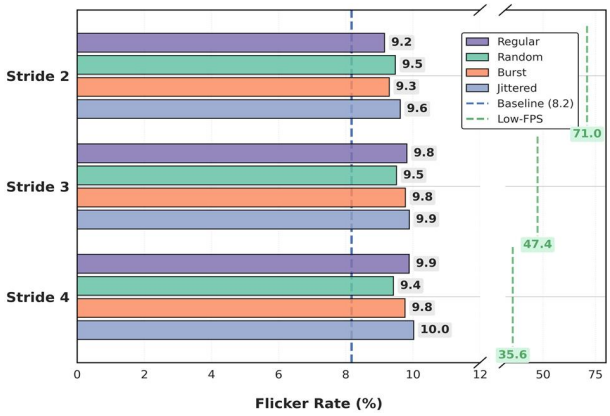

Fig. 12. Irregular Low-FPS(random/burst/jittered) – Robustness in Flicker Rate(%)

Feature-Level VFI는 불규칙 조건에서도 Flicker가 대체로 9-10% 범위로 유지되어 Baseline과 유사한 수준을 보인다. 반면 Low-FPS는 조건과 무관하게 Flicker가 매우 높아 프레임 누락이 시간적 연속성을 크게 저하시킴을 재확인할 수 있다.

4.8 Statistical Reliability

본 연구는 동일 데이터셋과 동일 파이프라인 조건에서 5회 반복 측정을 수행하였고, 주요 효율 지표는 mean±std로 보고하였다. 또한, 작은 성능 차이(mAP 등)에 대한 해석을 보강하기 위해, 시퀀스 단위 paired bootstrap(17 sequences) 기반의 95% 신뢰구간을 추가로 제시한다.

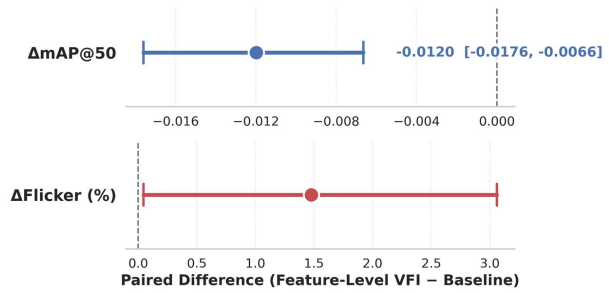

Fig. 13. Paired bootstrap 95% CI for sequence-level differences

paired bootstrap 결과, mAP@50의 평균 차이(Feature-Level VFI – Baseline)는 -1.2%p이며 95% CI는 [-1.76%p, -0.66%p]로 0을 포함하지 않는다.

Flicker(%)의 평균 차이는 +1.48%p이며 95% CI는 [+0.05%p, +3.06%p]로 나타난다. 즉 정확도 저하와 Flicker 증가는 통계적으로 유의할 수 있으나, 변화 규모는 제한적이므로 system-level E2E 효율 개선과의 트레이드오프 범위에서 해석한다.

4.9 Additional Analyses (Ablation & Generalization)

본 절에서는 제안 기법의 설계 선택과 적용 범위를 정량적으로 확인하기 위해 (i) 멀티스케일 보간 범위 어블레이션, (ii) 비학습형 보간 대안 비교, (iii) YOLOv8 일반화를 추가로 평가하였다.

Table 5. Multi-scale ablation (P3/P4/P5) for Feature-Level VFI(△=2)

Scale config	mAP@50	IDF1	IDSW/1kTP	FPS_proc
P3 only	0.4025	0.5598	48.88	84.58
P3 + P4	0.4083	0.5664	40.35	86.97
P3-P5	0.4086	0.5675	39.24	88.38

P3 only 대비 P3-P5는 mAP@50이 약 1.52% 개선되고, IDSW/1kTP는 약 19.72% 감소하여 멀티스케일 보간이 정확도와 추적 일관성 측면에서 유리함을 보인다.

Table 6. Training-free interpolation alternatives for Feature-Level VFI(△=2)

Method	mAP@50	Flicker(%)	FPS_proc
Linear(ours)	0.4086	9.15	88.37
Copy(nearest)	0.3881	12.36	87.4
Motion-gated	0.4049	9.51	85.96

Copy(nearest)는 Linear 대비 mAP@50이 약 5.02% 감소하고 Flicker가 약 35.08% 증가하여 변동성이 크게 악화된다. Motion-gated는 Linear 대비 성능 저하가 작고 Flicker도 유사하며 FPS_proc는 소폭 감소한다. 따라서 선형 보간은 구현이 단순하면서도 성능 저하가 가장 제한적인 기본 선택으로 해석할 수 있다.

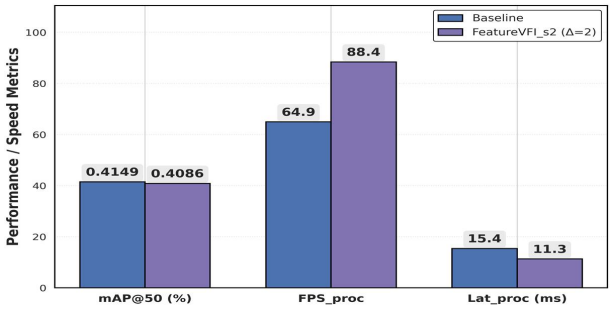

Fig. 14. Generalization to YOLOv8: Baseline vs Feature-Level VFI(△=2)

YOLOv8에서도 Feature-Level VFI는 mAP@50이 -1.43%로 제한적인 감소에 그치며, FPS_proc는 76.11±0.69에서 86.01±1.10으로 증가하고 Lat_proc는 13.14±0.12 ms에서 11.63±0.15 ms로 감소하여, 특정 YOLO 버전에만 국한되지 않는 효율 이득을 확인할 수 있다.

4.10 Failure Modes and Correlation Analysis

Feature-Level VFI는 인접 Anchor 특징을 선형 보간해 중간 시점 특징을 구성하고 head-only로 추론하므로, 프레임 간 변위가 큰 구간(고속 이동, 급격한 경계 변화, 강한 가려짐)에서는 보간 특징이 실제 중간 시점을 충분히 근사하지 못할 수 있다. 이때 박스 위치/크기 불안정이나 검출 누락이 발생하며, stride 증가(3/4) 또는 burst처럼 Anchor 간격이 일시적으로 커지는 조건에서 더 두드러질 수 있다.

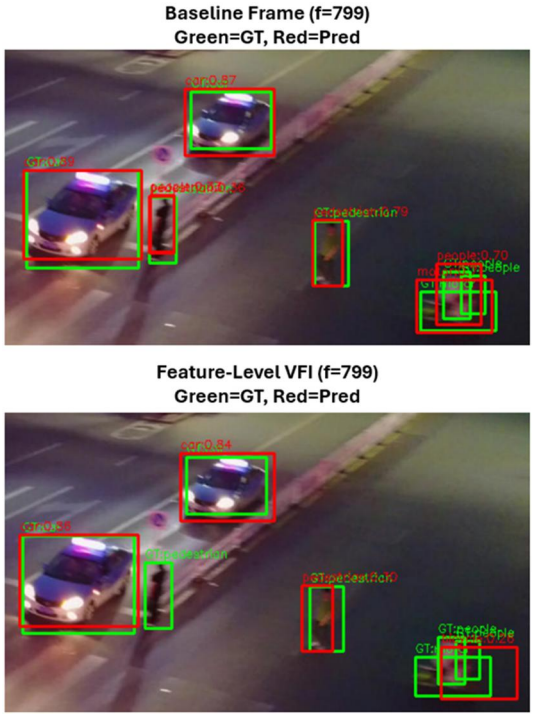

Fig. 15. Qualitative Failure Example: Baseline vs. Feature-Level VFI Predictions

Fig. 15는 동일 장면에서 Baseline과 Feature-Level VFI의 예측 결과를 비교한 예시이다. Baseline에서는 빠르게 이동하는 객체가 비교적 안정적으로 검출되는 반면, Feature-Level VFI에서는 일부 프레임에서 박스가 객체를 벗어나거나 크기가 불안정해지며 검출이 누락되는 사례가 관측된다. 정량적으로도 고속 객체 구간에서의 Recall이 0.49에서 0.25로 감소(-49.04%)하여, 큰 변위

상황에서 선형 특징 보간의 근사 한계가 성능 저하로 연결될 수 있음을 보인다.

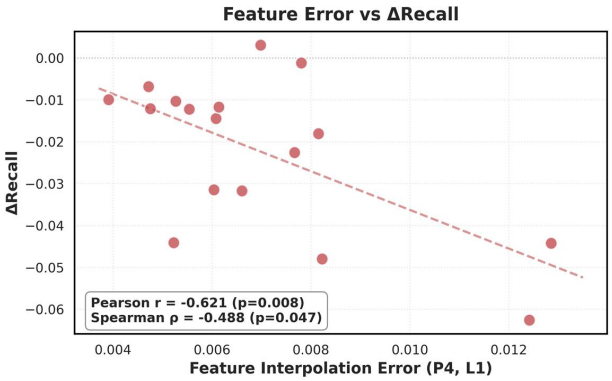

Fig. 16. Error-Recall Correlation (17 sequences)

Fig. 16은 P4 특징맵의 선형 보간 오차(L1)와 Recall 변화(Recall = Feature-Level VFI − Baseline) 간 관계를 시퀀스 단위로 나타낸다. 보간 오차가 커질수록 Recall이 더 감소하는 경향이 관측되며, Pearson r = -0.621 (p = 0.008), Spearman ρ = -0.488 (p = 0.047)로 유의한 음의 상관을 확인하였다. 이는 고속 이동/가려짐 등 특징 변화가 큰 장면에서 보간 오차가 증가하고 탐지 누락이 커질 수 있음을 정량적으로 뒷받침한다.

V. Conclusions

본 연구는 저프레임(Low-FPS) 입력 환경에서 YOLO 기반 객체 탐지기의 성능 저하와 시간적 불안정(박스 흔들림, 검출 단절 등)을, 모델 구조 변경이나 재학습 없이 추론 단계의 스케줄링과 중간 표현 재사용만으로 완화하는 비학습형(training-free) 추론 모듈을 제안하였다. 제안 기법은 YOLO11의 Backbone과 Neck 구조 및 가중치를 유지한 채, Anchor 프레임에서 Detection Head 입력 직전의 멀티스케일 특징 맵(P3-P5)을 저장하고, 비 Anchor 시점에는 인접 Anchor 특징을 선형 보간해 head-only 추론을 수행하는 Sparse-Backbone 기반 Feature-Level VFI로 구현된다. 실험 결과, Backbone-Neck 호출을 약 50%로 제한하면서도 정적 탐지 성능은 Baseline 대비 mAP@50 기준 98.48% 수준으로 유지되었고, Center/Scale Jitter 및 Flicker/Streak 역시 Baseline에 근접한 범위를 보여 저프레임 조건에서도 결과 연속성과 일관성이 비교적 잘 보존됨을 확인하였다. 효율성 측면에서는 전처리-(VFI 포함)-추론-후처리를 포함한 동일한 system-level E2E 프로토콜로 비교한 결과, Feature-Level VFI는 FPS_proc를 36.11% 향상시키고 Lat_proc를 26.60% 감소시켜 품질-효율 트레이드오프 관점에서 실용적인 개선을 제공하였다. Pixel-Level VFI는 복원 오버헤드로 인해 system-level E2E 처리량 이득이 제한적이었고, Low-FPS는 타임라인 누락이 직접 반영되어 안정성 지표가 크게 악화되었다. DL-VFI(RIFE)는 일부 정확도 및 안정성 지표에서 Baseline에 더 근접할 수 있으나, system-level E2E 비용이 크게 증가하여 실시간 제약 환경에서는 적용 부담이 크다. 추가로 stride=3/4 및 불규칙 드롭(random/burst/jittered) 조건에서도 성능 저하가 일정 범위 내로 제한되는 경향을 확인하여, 더 큰 시간 간격과 비정규 저프레임 상황에 대한 적용 가능성을 보였다. 한편 고속 이동이나 강한 가려짐 구간에서는 선형 보간의 근사 한계로 성능 저하가 커질 수 있으며, 이는 stride 증가나 burst 조건에서 더 두드러질 수 있다. 시퀀스 단위 paired bootstrap 분석에서는 mAP 및 Flicker 변화가 통계적으로 유의할 수 있음을 보였으나, 변화 규모가 제한적임을 함께 보고하여 작은 차이에 대한 해석을 보강하였다. 종합하면, 제안 기법은 저프레임 및 자원 제약 환경에서 탐지 품질과 시간적 안정성을 가능한 한 유지하면서, 시스템 관점의 효율 개선까지 달성할 수 있는 training-free 추론 모듈로 활용 가능함을 확인하였다. 향후 연구에서는 빠른 이동 및 가려짐 구간의 오차를 줄이기 위해 경량 특징 정렬(Feature Alignment) 또는 모션 단서 기반의 비학습형 보간을 결합하고, 장면 변화나 객체 속도에 따라 Anchor를 동적으로 선택하는 적응적 스케줄링으로 성능과 효율의 균형을 추가로 안정화하는 방향을 고려할 수 있다.

REFERENCES

[1] L. Jiao, R. Zhang, F. Liu, S. Yang, B. Hou, L. Li, and X. Tang, "New Generation Deep Learning for Video Object Detection: A Survey," IEEE Trans. Neural Netw. Learn. Syst., Vol. 33, No. 8, pp. 3195-3215, Aug. 2022. DOI: 10.1109/TNNLS.2021.3053249

[2] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "YOLOv10: Real-Time End-to-End Object Detection," Proc. Advances in Neural Information Processing Systems (NeurIPS), pp. 107984-108011, Vancouver, Canada, Dec. 2024. DOI: 10.52202/079017-3429

[3] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "DETRs Beat YOLOs on Real-time Object Detection,"

Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 16965-16974, Seattle, USA, Jun. 2024. DOI: 10.1109/CVPR52733.2024.01605

[4] A. Ilioudi, A. Dabiri, B. J. Wolf, and B. De Schutter, "Deep Learning for Object Detection and Segmentation in Videos: Toward an Integration With Domain Knowledge," IEEE Access, Vol. 10, pp. 34562-34576, Mar. 2022. DOI: 10.1109/ACCESS.2022.3162827

[5] G. Sun, Y. Hua, G. Hu, and N. Robertson, "Efficient One-Stage Video Object Detection by Exploiting Temporal Consistency," Proc. European Conf. on Computer Vision (ECCV), Tel Aviv, Israel, Oct. 2022. DOI: 10.1007/978-3-031-19833-5_1

[6] C. Deng, D. Chen, and Q. Wu, "Identity-Consistent Aggregation for Video Object Detection," Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV), pp. 13388-13398, Paris, France, Oct. 2023. DOI: 10.1109/ICCV51070.2023.01236

[7] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, "Real-Time Intermediate Flow Estimation for Video Frame Interpolation," Proc. European Conf. on Computer Vision (ECCV), pp. 624-642, Tel Aviv, Israel, Oct. 2022. DOI: 10.1007/978-3-031-19781-9_36

[8] G. Zhang, Y. Zhu, H. Wang, Y. Chen, G. Wu, and L. Wang, "Extracting Motion and Appearance via Inter-Frame Attention for Efficient Video Frame Interpolation," Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 5682-5692, Vancouver, Canada, Jun. 2023. DOI: 10.1109/CVPR52729.2023.00550

[9] G. Zhang, C. Liu, Y. Cui, X. Zhao, K. Ma and L. Wang, "VFIMamba: Video Frame Interpolation with State Space Models," Proc. Advances in Neural Information Processing Systems (NeurIPS), pp. 107225-107248, Vancouver, Canada, Dec. 2024. DOI: 10.52202/079017-3405

[10] J. Park and N. I. Cho, "Explicit Guidance for Robust Video Frame Interpolation against Discontinuous Motions," Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV), pp. 8464-8473, Tucson, USA, Feb. 2025. DOI: 10.1109/WACV61041.2025.00820

[11] K. A. Hashmi, T. U. Sheikh, D. Stricker, and M. Z. Afzal, "Beyond Boxes: Mask-Guided Spatio-Temporal Feature Aggregation for Video Object Detection," Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV), pp. 8122-8133, Tucson, USA, Feb. 2025. DOI: 10.1109/WACV61041.2025.00788

[12] S. Sarkar, G. Datta, S. Kundu, K. Zheng, C. Bhattacharyya, and P. A. Beerel, "MaskVD: Region Masking for Efficient Video Object Detection," Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV), pp. 1955-1964, Tucson, USA, Feb. 2025. DOI: 10.1109/WACV61041.2025.00197

[13] F. He, Q. Li, X. Zhao, and K. Huang, "Temporal-Adaptive Sparse Feature Aggregation for Video Object Detection," Pattern Recognit., Vol. 127, Art. no. 108587, Jul. 2022. DOI: 10.1016/j.patcog.2022.108587

[14] J. Dong, K. Ota, and M. Dong, "Video Frame Interpolation: A Comprehensive Survey," ACM Trans. Multimedia Comput. Commun. Appl., Vol. 19, No. 2s, Art. no. 78, pp. 1-31, Apr. 2023. DOI: 10.1145/3556544

[15] G. Wu, X. Tao, C. Li, W. Wang, X. Liu, and Q. Zheng, "Perception-Oriented Video Frame Interpolation via Asymmetric Blending," Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2753-2762, Seattle, USA, Jun. 2024. DOI: 10.1109/CVPR52733.2024.00266

Authors

Min-Ho Kim received the Associate of Science (A.S.) degree from the Department of Computer Science at Inha Technical College, Korea, in 2025. He received the Bachelor of Science (B.S.) degree in the same department in 2026. His research interests include artificial intelligence, machine learning systems, and computer vision.

Kyu-Cheol Cho received the B.S., M.S. and Ph.D. degrees in Computer Science and Information Engineering from Inha University, Korea, in 2005, 2007 and 2013, respectively. Dr. Cho joined the faculty of the Department of Computer Science at Inha Technical College, Incheon, Korea, in 2016. He is currently an assistant professor in the Department of Computer Science, Inha Technical College. He is interested in cloud computing, green IT and web programming.