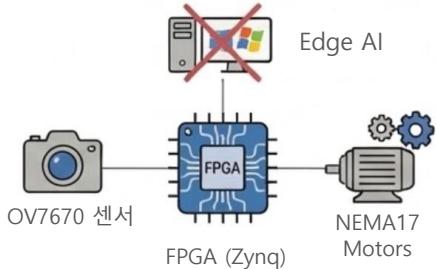


FPGA 기반 객체 정렬 및 추론 엣지 AI 시스템

프로젝트 개요



PC 없이 스스로 판단하는 독립형 엣지 시스템

외부 PC나 OS 없이 FPGA 단독으로 영상을 처리하고 객체를 정렬 및 추론

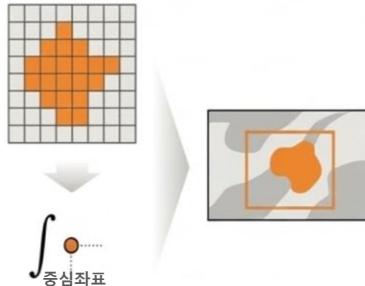
하드웨어 병렬 처리로 지연 시간(Latency) 단축

CPU의 순차 처리 방식 한계를 극복하기 위해 제어 및 추론 과정 하드웨어로직 설계

사용된 핵심 리소스

H/W: Zybo Z7-20, OV7670 모듈, 스텝 모터
S/W: Verilog HDL, C언어

핵심 알고리즘



1단계: 픽셀 중심 좌표법

화면 내 객체의 전체 픽셀의 중심 좌표 산출

2단계: Dynamic ROI (관심영역 동적 재배치)

산출된 중심 좌표를 기준으로 실시간 주변 관심영역 추출

3단계: Line Buffer 영상 데이터 처리

영상 데이터 처리시 요구되는 FPGA 리소스 최적화 및 실시간 CNN 추론 구현

시스템 특장점



Zero-Latency Vision

프레임 버퍼를 제거한 영상 데이터 입력을 통한 실시간 중심 좌표 산출 및 CNN 추론

경량 AI 모델로 98.9% 인식률 달성

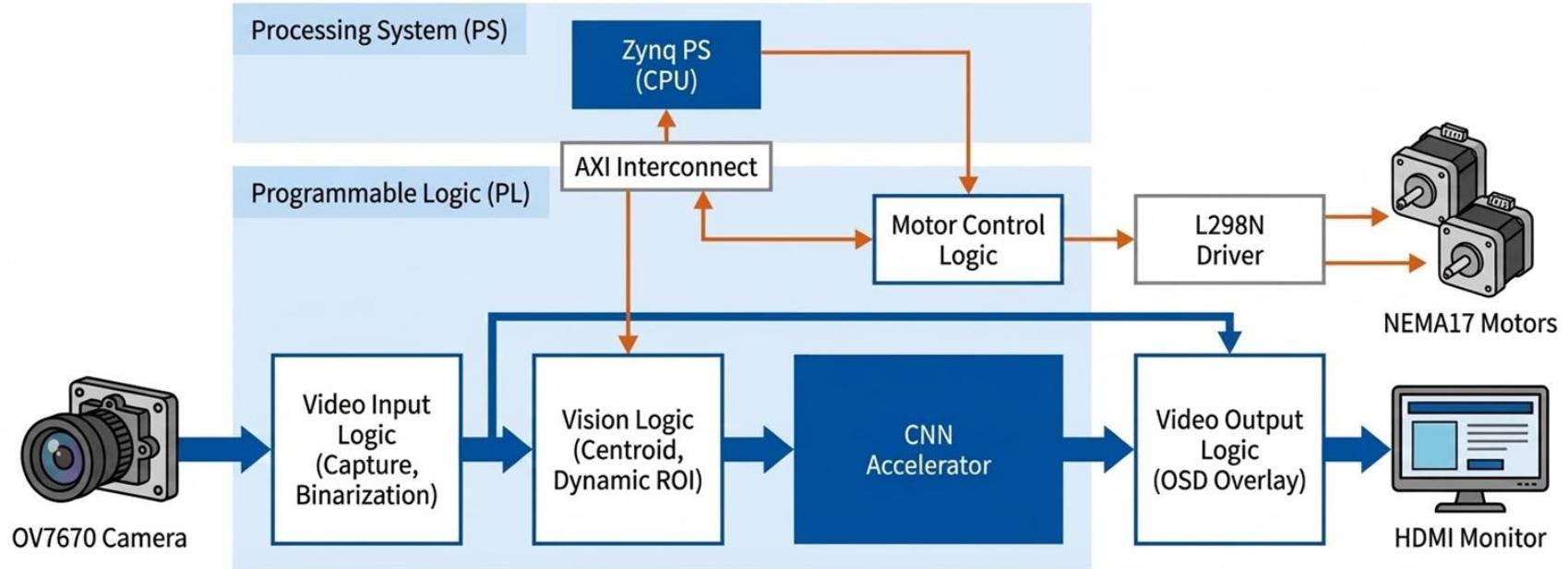
Dynamic ROI를 통한 객체 추적으로 CNN 모델의 추론 정확도 향상



완벽한 실시간 제어

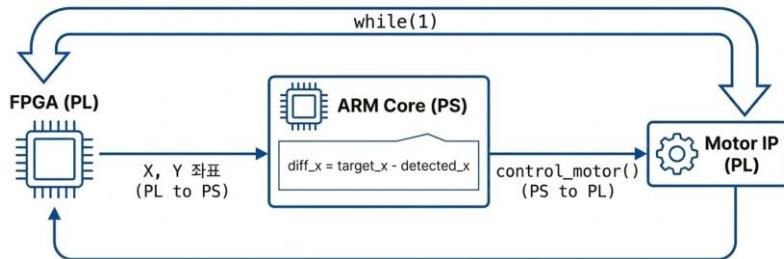
비전 처리 지연 최소화와 FSM 모터 제어로 직을 통한 실시간 객체 위치 정렬

시스템 아키텍처



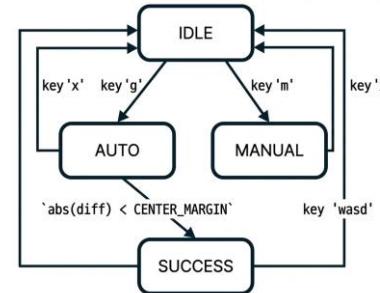
PS(Processing System)

제어 루프 블록도



AXI 버스를 통한 Programmable Logic IP
(모터 제어)에 제어 명령(방향, 속도) 전송

FSM



Processing System에서 FSM 설계에 따라 객체의 현재 중심 좌표와 목표 좌표의 오차(Error) 계산

영상 이진화



다채널 컬러이미지 (YUV Pixel Stream)

하드웨어 비교기 (Comparator)

이진화 이미지

1클릭 내 판권

입력되는 YUV픽셀 값을 하드웨어 비교기 회로가 실시간으로 설정된 임계값(Threshold)과 비교하여 '1' (객체) 또는 '0' (배경)으로 변환

연산량 극소화

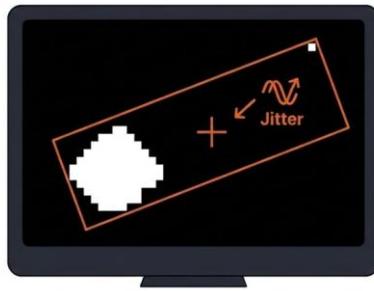
YUV 정보를 의미 있는 바이너리
마스크로 경량화, 후속 연산
(픽셀 중심 좌표 계산 등)의 부하 감소

결과

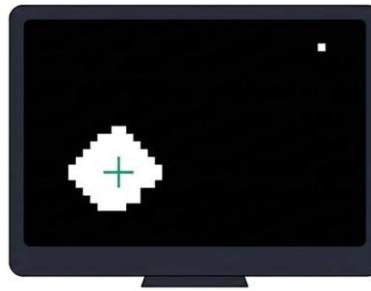
픽셀 단위의 즉각적인 전처리 실현

중심 좌표 문제

MIN / MAX

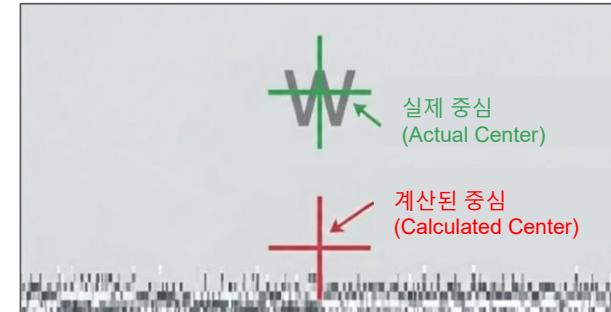


픽셀 중심좌표



단일 노이즈 픽셀이 전체 영역
계산을 왜곡하여 심한 중심 좌
표 변화(지터) 발생

유효 픽셀의 분포 밀도를 분석
하여, 노이즈의 영향을 배제 하
고 안정적인 중심좌표 도출



중심좌표 쓸림 문제 해결

문제

- 픽셀중심좌표 알고리즘 이후에도 Y 좌표가
실제보다 아래로 계산 되는 현상 발견

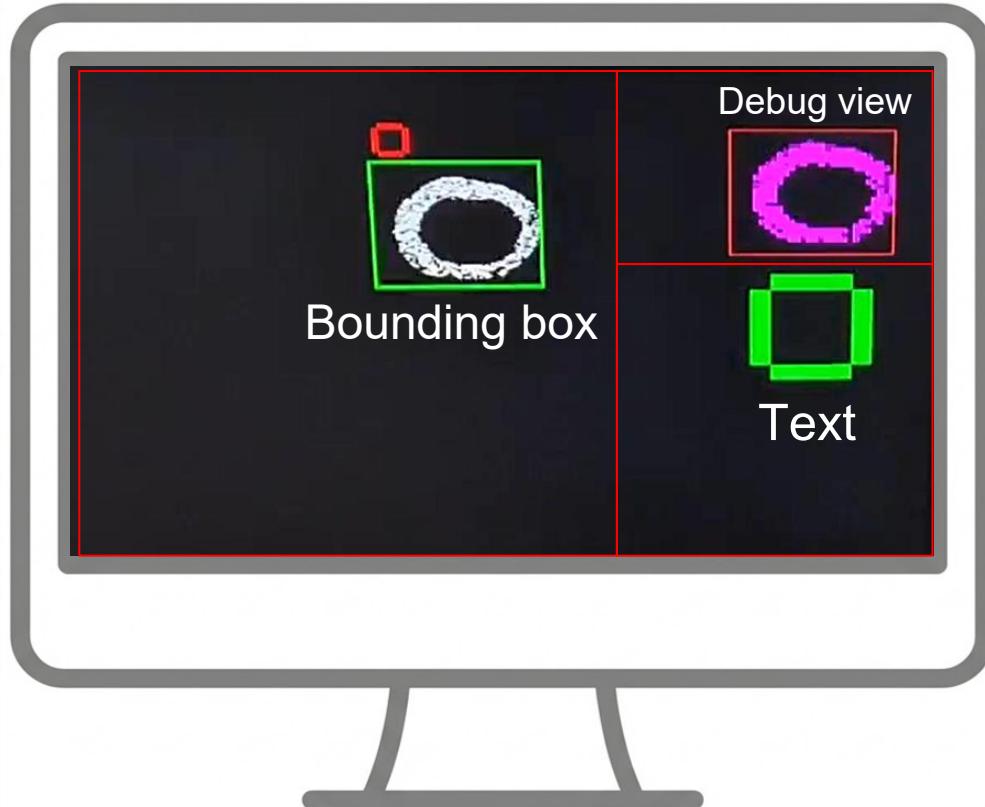
원인 분석

- 이미지 프레임의 하단에 노이즈 발생으로
Y값을 증가시켜서 중심좌표를 아래로 당김

해결

- 노이즈가 생기는 478부터 좌표계산에서
제외

CNN 가속기

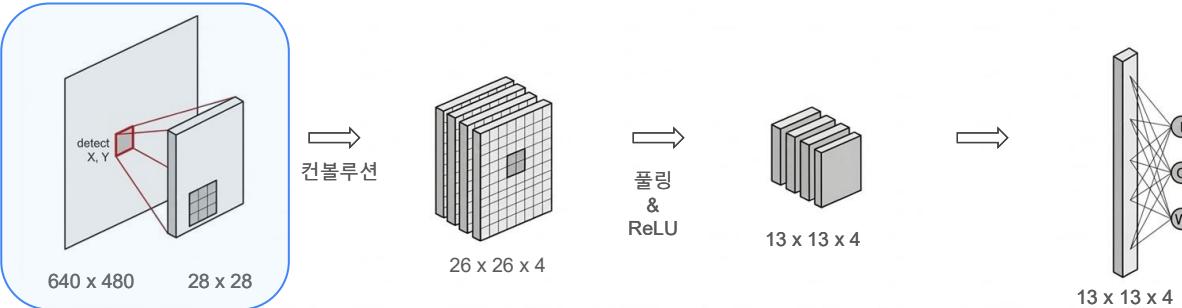


HDMI OSD 출력

기능

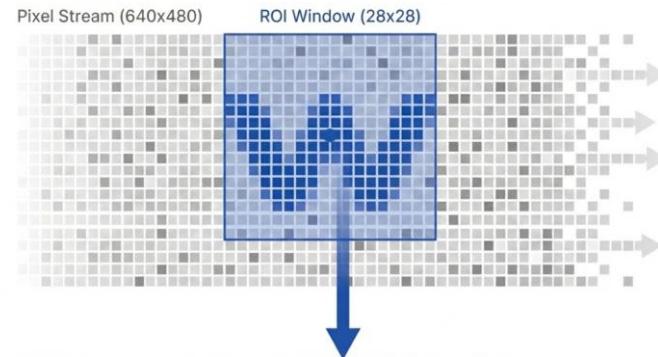
- 텍스트: CNN 추론 결과 폰트
- 바운딩 박스: 중심 좌표를 기준으로 박스 생성
- 디버그 뷰: 바운딩 박스로 추출한 이미지 출력
- 카메라 이미지: 실제 카메라의 이미지 이진화 출력

CNN 가속기 아키텍처 및 데이터 흐름



이미지 전처리

- **실시간 추적:**
ROI 영역을 중심좌표를 이용하여 화면
어디에 있어도 CNN 가능
- **Cropping:**
프레임 저장 대기 없이 데이터 입력
즉시 관심 영역 (28×28) 추출
- **Downscaling:**
 $640 \times 480 > 112 \times 112$ (Debug)
> 28×28 (CNN) 계층적 변환.



Dynamic ROI



화면 구석에 객체 출현

픽셀 좌표중심(X_c, Y_c) 계산

계산된 중심 기준 28X28 ROI 박스 생성

고정된 ROI

문제

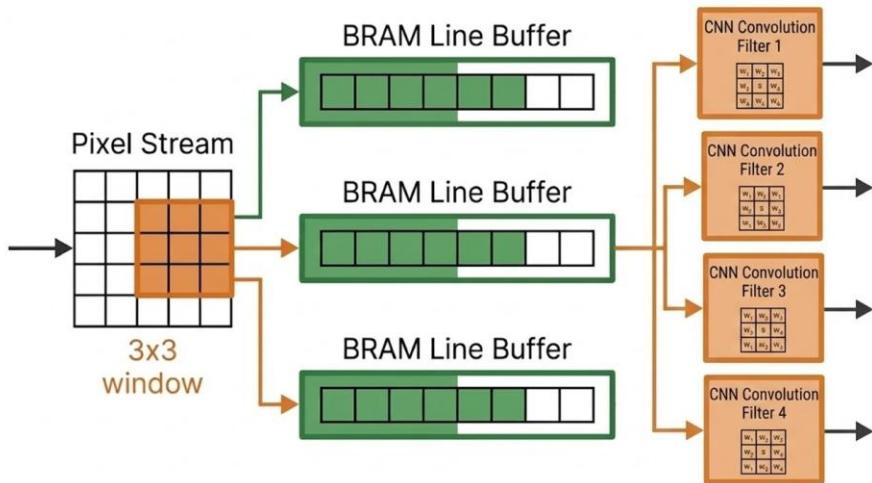
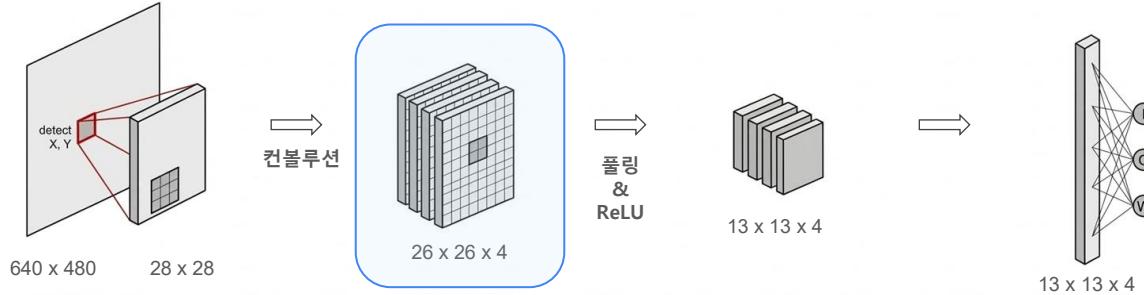
- 고정된 ROI(관심 영역)는
객체가 영역을 벗어나면 추론
불가능

Dynamic ROI

해결책 (Dynamic ROI)

- **중심좌표 추출:**
픽셀 중심 좌표로 객체의 실시간 중심 좌표
(X_c, Y_c)를 먼저 계산
- **ROI 재설정:**
추출된 (X_c, Y_c)를 기준으로, CNN 입력에 맞는 28×28 픽셀
크기의 윈도우(ROI) 실시간 재설정

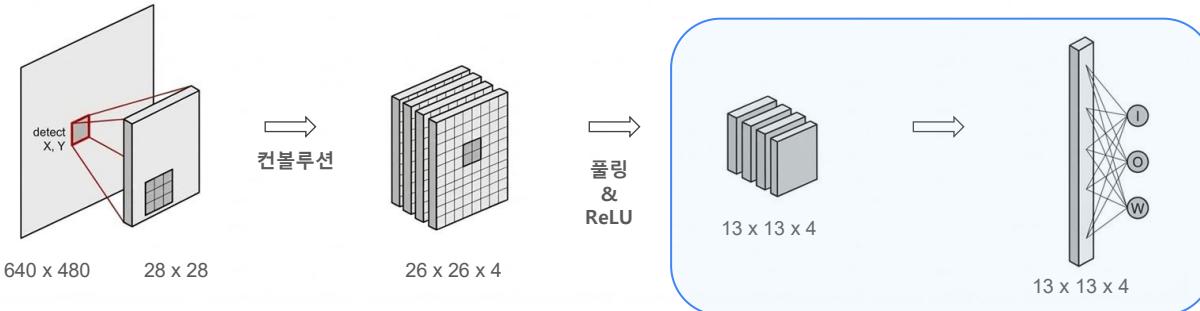
4채널 병렬 컨볼루션 및 특징 추출



4채널 병렬 CNN 가속

- **Feature Diversity:** 4개의 독립적 채널로 다양한 기하학적 특징 동시 추출.
- **Trade-off:** Pooling으로 인한 공간 정보 손실을 채널 확장으로 보완.
- **Line Buffer:** 외부 메모리 없이 내부 BRAM 만으로 3x3 윈도우 실시간 생성.

데이터 압축 및 고속 분류



Pooling layer

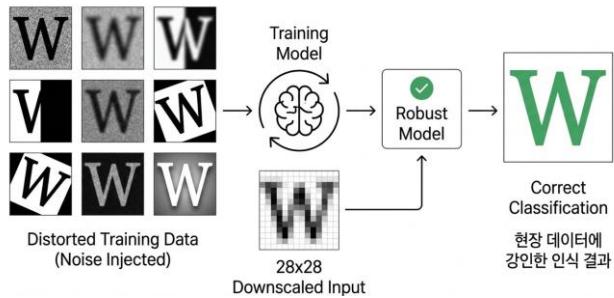
- **ReLU (Rectified Linear Unit):**
데이터에 비선형성(Non-linearity) 주입
모델의 표현력 강화
- **2 x 2 Max Pooling (Stride 2):**
2 x 2 영역에서 가장 큰 값(특징)만 추출
- **FC Layer 연산 부하 감소:**
압축 전: $26 \times 26 \times 4 = 2,704$ 개
압축 후: $13 \times 13 \times 4 = 676$ 개
⇒ 연산량 $\frac{1}{4}$ 감소

Fully connected layer

- **시분할 누적 연산:**
 $13 \times 13 \times 4 = 676$ 개의 데이터 한 번에 계산하지 않음
⇒ 4개의 채널씩 묶어서 $13 \times 13 = 169$ 클럭 동안 누적 연산
- **고속 추론:**
학습된 가중치(Weight)를 외부 메모리 로딩 없이 FPGA
로직 내부에 상수(Constant)로 내장.
⇒ 메모리 대역폭 소모 '0', 데이터 로딩 지연 시간 '0'.

하드웨어 특화 학습 및 양자화

Dataset



CNN 학습 및 인식률 개선

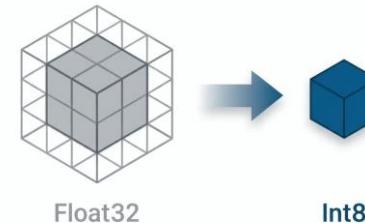
문제

- 카메라 원본(640x480) \Rightarrow CNN(28x28)
축소 시 계단 현상(Aliasing) 및 화질 저하.

해결

- Noise Injection Training (노이즈 주입 학습)
- 깨끗한 이미지가 아닌, 노이즈가 섞인 데이터로 모델을 학습시켜 현장 적응력 강화

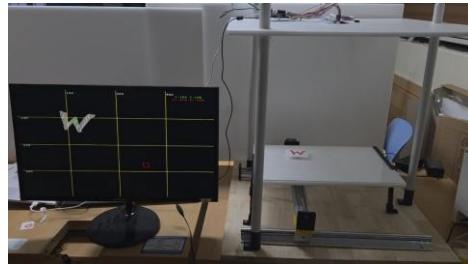
Quantization



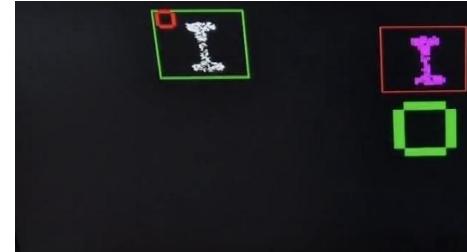
양자화

- **Integer Quantization:**
부동소수점 연산 없이 고정소수점 연산으로 자원 사용 최소화.
- **방식:**
선형 스케일링
- **공식:**
 $\text{Integer} = \text{round}(\text{Float} \times \text{SCALE}(10))$
- **방식:**
실수에 SCALE=10을 곱한 후 반올림

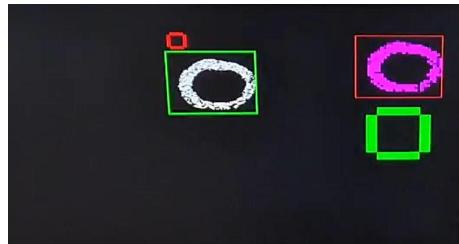
시연 영상



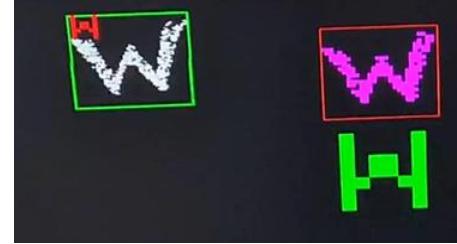
<https://youtu.be/ssYAIElkrek>



<https://youtu.be/pq4ShsVEkw0>



<https://youtu.be/YTFxmHVEm4E>



<https://youtu.be/xhQ2JrhAVRA>