(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년08월29일
(11) 등록번호 10-2437755
(24) 등록일자 2022년08월24일

(51) 국제특허분류(Int. Cl.)
H04N 7/01 (2006.01) *H04N 21/231* (2011.01)
H04N 21/2662 (2011.01) *H04N 21/44* (2011.01)
H04N 21/6437 (2011.01)
(52) CPC특허분류
H04N 7/0127 (2013.01)
H04N 21/23113 (2013.01)
(21) 출원번호 10-2021-0025338
(22) 출원일자 2021년02월25일
심사청구일자 2021년02월25일
(56) 선행기술조사문헌
KR102207370 B1
비특허문헌 1(ICTC2020)

기술이전 희망 : **기술양도**

(73) 특허권자
인천대학교 산학협력단
인천광역시 연수구 아카데미로 119 (송도동)
(72) 발명자
황광일
인천광역시 연수구 송도문화로28번길 81, 105동 3701호(송도동, 송도더샵그린스퀘어)
이정훈
인천광역시 남동구 담방로48번길 7-6, 201호(만수동)
이승수
인천광역시 남동구 청능대로718번길 7, 101동 602호(논현동, 소래마을풍림아파트)
(74) 대리인
특허법인충정

전체 청구항 수 : 총 9 항 심사관 : 박재학

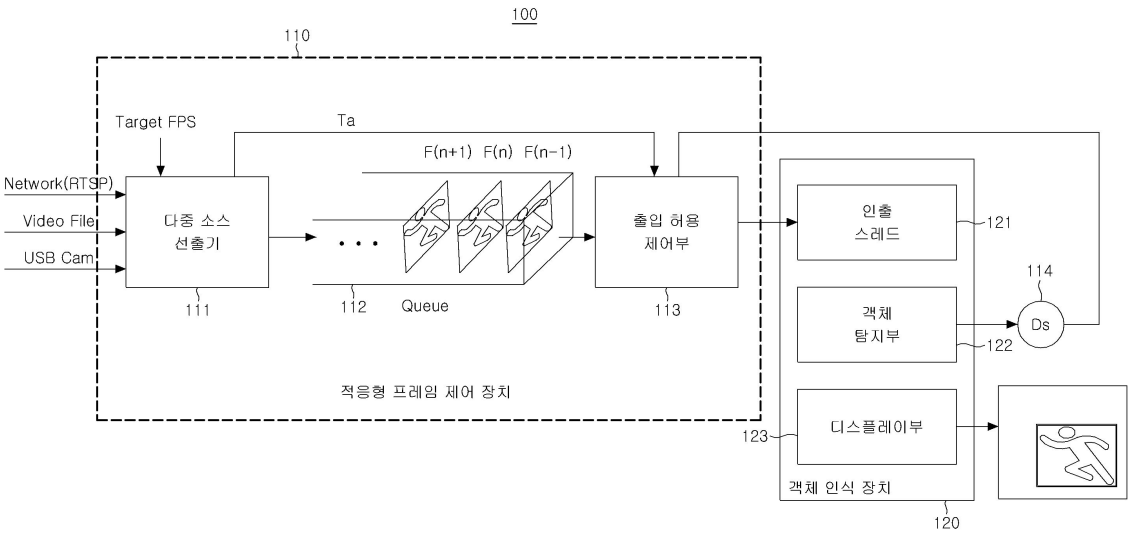
(54) 발명의 명칭 **실시간 객체 인식을 위한 적응형 프레임 관리 시스템 및 이를 이용한 적응형 프레임 제어 방법**

(57) 요 약

실시간 객체 인식을 위한 적응형 프레임 관리 시스템 및 이를 이용한 적응형 프레임 제어 방법은 입력소스로부터 들어오는 입력 프레임 레이트와 객체 인식 장치의 객체 인식 서비스 레이트를 비교하여 적응형으로 프레임을 제어함으로써 실시간 객체 인식 서비스를 구현한다.

본 발명은 기존의 YOLO에 비해 실시간 객체 탐지 서비스가 가능하여 YOLO의 높은 인식률과 편의성을 제공할 수 있는 효과가 있다.

대 표 도



(52) CPC특허분류
H04N 21/2662 (2013.01)
H04N 21/44004 (2013.01)
H04N 21/6437 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1425141169
과제번호	S2920482
부처명	중소벤처기업부
과제관리(전문)기관명	중소기업기술정보진흥원
연구사업명	맞춤형 기술파트너 지원사업
연구과제명	NVR과 연동가능한 인공지능 기반 다중 객체추적 솔루션 개발
기 여 율	1/1
과제수행기관명	인천대학교산학협력단
연구기간	2019.06.22 ~ 2022.03.21

명 세 서

청구범위

청구항 1

복수의 입력소스로부터 영상 프레임을 입력받아 큐(Queue)로 전달하고, 입력 영상으로부터 들어오는 영상 프레임의 도착 시간 간격(T_a)을 측정하는 다중 소스 선출기;

상기 영상 프레임에서 객체의 존재를 추정하여 객체 인식 결과를 출력하는 객체 인식 장치;

상기 객체 인식 장치에 연결되어 상기 영상 프레임에 대해 객체 인식에 소요되는 객체 인식 서비스 시간(D_s)을 측정하는 객체 인식 서비스 시간 측정부; 및

상기 큐의 출력단에 연결되어 상기 큐로부터 상기 영상 프레임을 각각 수신하고, 상기 다중 소스 선출기로부터 수신한 도착 시간 간격(T_a)과 상기 객체 인식 서비스 시간 측정부로부터 수신한 객체 인식 서비스 시간(D_s)을 비교한 결과에 따라 상기 큐에서 인출된 영상 프레임을 상기 객체 인식 장치로 전달할 지 또는 폐기할 지 판단하는 출입 허용 제어부를 포함하고,

상기 출입 허용 제어부는 상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)보다 작은 경우, 상기 큐에서 인출된 일부 영상 프레임을 폐기(Drop)하고, 나머지 영상 프레임을 상기 객체 인식 장치로 전달하는 실시간 객체 인식을 위한 적응형 프레임 관리 시스템.

청구항 2

삭제

청구항 3

복수의 입력소스로부터 영상 프레임을 입력받아 큐(Queue)로 전달하고, 입력 영상으로부터 들어오는 영상 프레임의 도착 시간 간격(T_a)을 측정하는 다중 소스 선출기;

상기 영상 프레임에서 객체의 존재를 추정하여 객체 인식 결과를 출력하는 객체 인식 장치;

상기 객체 인식 장치에 연결되어 상기 영상 프레임에 대해 객체 인식에 소요되는 객체 인식 서비스 시간(D_s)을 측정하는 객체 인식 서비스 시간 측정부; 및

상기 큐의 출력단에 연결되어 상기 큐로부터 상기 영상 프레임을 각각 수신하고, 상기 다중 소스 선출기로부터 수신한 도착 시간 간격(T_a)과 상기 객체 인식 서비스 시간 측정부로부터 수신한 객체 인식 서비스 시간(D_s)을 비교한 결과에 따라 상기 큐에서 인출된 영상 프레임을 상기 객체 인식 장치로 전달할 지 또는 폐기할 지 판단하는 출입 허용 제어부를 포함하고,

상기 출입 허용 제어부는 상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)보다 작은 경우, 최신의 영상 프레임만을 전달하기 위하여 상기 큐의 테일(tail)에 있는 영상 프레임을 제외하고, 상기 큐의 나머지 패킷들을 폐기(Drop)하는 실시간 객체 인식을 위한 적응형 프레임 관리 시스템.

청구항 4

청구항 1 또는 청구항 3에 있어서,

상기 출입 허용 제어부는 상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)보다 크거나 같은 경우, 상기 큐에서 인출된 영상 프레임을 그대로 상기 객체 인식 장치로 전달하는 실시간 객체 인식을 위한 적

응형 프레임 관리 시스템.

청구항 5

청구항 3에 있어서,

상기 폐기되는 영상 프레임의 수는 상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)의 차이에 의해 달라지는 실시간 객체 인식을 위한 적응형 프레임 관리 시스템.

청구항 6

복수의 입력소스로부터 영상 프레임을 입력받아 큐(Queue)로 전달하고, 입력 영상으로부터 들어오는 영상 프레임의 도착 시간 간격(T_a)을 측정하는 다중 소스 선출기;

상기 영상 프레임에서 객체의 존재를 추정하여 객체 인식 결과를 출력하는 객체 인식 장치;

상기 객체 인식 장치에 연결되어 상기 영상 프레임에 대해 객체 인식에 소요되는 객체 인식 서비스 시간(D_s)을 측정하는 객체 인식 서비스 시간 측정부; 및

상기 큐의 출력단에 연결되어 상기 큐로부터 상기 영상 프레임을 각각 수신하고, 상기 다중 소스 선출기로부터 수신한 도착 시간 간격(T_a)과 상기 객체 인식 서비스 시간 측정부로부터 수신한 객체 인식 서비스 시간(D_s)을 비교한 결과에 따라 상기 큐에서 인출된 영상 프레임을 상기 객체 인식 장치로 전달할 지 또는 폐기할 지 판단하는 출입 허용 제어부를 포함하고,

상기 다중 소스 선출기는 복수의 입력소스로부터 영상 프레임을 입력받는 다중 소스 먹스;

상기 다중 소스 먹스로부터 입력되는 영상 프레임을 읽어 들이는 시간과, target FPS(Frame Per Second)와 비교하여 target FPS보다 입력 프레임 레이트(Input Frame Rate)가 높은 경우, target FPS에 동기를 맞춰주는 기능을 수행하는 싱크로나이저(Synchronizer); 및

상기 싱크로나이저에 연결되고, 상기 큐의 입력단에 연결되며, 각각의 입력 영상 프레임 간의 도착 시간 간격(T_a)을 측정하고, 상기 출입 허용 제어부로 전송하는 프레임 처리부를 포함하는 실시간 객체 인식을 위한 적응형 프레임 관리 시스템.

청구항 7

복수의 입력소스로부터 영상 프레임을 입력받아 큐(Queue)로 전달하고, 입력 영상으로부터 들어오는 영상 프레임의 도착 시간 간격(T_a)을 측정하는 다중 소스 선출기;

상기 영상 프레임에서 객체의 존재를 추정하여 객체 인식 결과를 출력하는 객체 인식 장치;

상기 객체 인식 장치에 연결되어 상기 영상 프레임에 대해 객체 인식에 소요되는 객체 인식 서비스 시간(D_s)을 측정하는 객체 인식 서비스 시간 측정부; 및

상기 큐의 출력단에 연결되어 상기 큐로부터 상기 영상 프레임을 각각 수신하고, 상기 다중 소스 선출기로부터 수신한 도착 시간 간격(T_a)과 상기 객체 인식 서비스 시간 측정부로부터 수신한 객체 인식 서비스 시간(D_s)을 비교한 결과에 따라 상기 큐에서 인출된 영상 프레임을 상기 객체 인식 장치로 전달할 지 또는 폐기할 지 판단하는 출입 허용 제어부를 포함하고,

상기 출입 허용 제어부는,

상기 다중 소스 선출기로부터 상기 도착 시간 간격(T_a)을 수신하고, 상기 객체 인식 서비스 시간 측정부로부터 상기 객체 인식 서비스 시간(D_s)을 수신하는 허용 컨트롤러; 및

상기 큐의 출력단에 연결되어 상기 큐로부터 상기 영상 프레임을 수신하고, 상기 허용 컨트롤러의 결과값이 상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)보다 작은 경우, 최신의 영상 프레임만을 전달하기 위하여 상기 큐의 테일(tail)에 있는 영상 프레임을 제외하고, 상기 큐의 나머지 패킷들을 폐기(Drop)하고, 상기 허용 컨트롤러의 결과값이 상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)보다 크거나 같은 경우, 상기 큐에서 인출된 영상 프레임을 그대로 상기 객체 인식 장치로 전달하는 디멀티플렉서를 포함하는 실시간 객체 인식을 위한 적응형 프레임 관리 시스템.

청구항 8

복수의 입력소스로부터 영상 프레임을 입력받아 큐(Queue)로 전달하고, 입력 영상으로부터 들어오는 영상 프레임의 도착 시간 간격(T_a)을 측정하는 다중 소스 선출기;

상기 영상 프레임에서 객체의 존재를 추정하여 객체 인식 결과를 출력하는 객체 인식 장치;

상기 객체 인식 장치에 연결되어 상기 영상 프레임에 대해 객체 인식에 소요되는 객체 인식 서비스 시간(D_s)을 측정하는 객체 인식 서비스 시간 측정부; 및

상기 큐의 출력단에 연결되어 상기 큐로부터 상기 영상 프레임을 각각 수신하고, 상기 다중 소스 선출기로부터 수신한 도착 시간 간격(T_a)과 상기 객체 인식 서비스 시간 측정부로부터 수신한 객체 인식 서비스 시간(D_s)을 비교한 결과에 따라 상기 큐에서 인출된 영상 프레임을 상기 객체 인식 장치로 전달할 지 또는 폐기할 지 판단하는 출입 허용 제어부를 포함하고,

상기 출입 허용 제어부는 상기 객체 인식 장치에서 한 프레임의 처리가 완료되어 다음 새로운 프레임을 인출(Fetch)하는 순간에 동작(Trigger)되는 실시간 객체 인식을 위한 적응형 프레임 관리 시스템.

청구항 9

복수의 입력소스로부터 영상 프레임을 입력받아 큐(Queue)로 전달하고, 입력 영상으로부터 들어오는 영상 프레임의 도착 시간 간격(T_a)을 다중 소스 선출기에서 측정하는 단계;

객체 인식 장치에서 상기 영상 프레임에서 객체의 존재를 인식하는 단계;

상기 객체 인식 장치에 연결된 객체 인식 서비스 시간 측정부는 영상 프레임에 대해 객체 인식에 소요되는 객체 인식 서비스 시간(D_s)을 측정하는 단계; 및

상기 큐로부터 상기 영상 프레임을 수신하고, 상기 다중 소스 선출기로부터 수신한 도착 시간 간격(T_a)과 상기 객체 인식 서비스 시간 측정부로부터 수신한 객체 인식 서비스 시간(D_s)을 비교한 결과에 따라 상기 큐에서 인출된 영상 프레임을 상기 객체 인식 장치로 전달할 지 또는 폐기할 지 판단하는 단계를 포함하고,

상기 객체 인식 장치로 전달할 지 또는 폐기할 지 판단하는 단계는,

상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)보다 작은 경우, 최신의 영상 프레임만을 전달하기 위하여 상기 큐의 테일(tail)에 있는 영상 프레임을 제외하고, 상기 큐의 나머지 패킷들을 폐기(Drop)하는 실시간 객체 인식을 위한 적응형 프레임 제어 방법.

청구항 10

삭제

청구항 11

복수의 입력소스로부터 영상 프레임을 입력받아 큐(Queue)로 전달하고, 입력 영상으로부터 들어오는 영상 프레

임의 도착 시간 간격(T_a)을 다중 소스 선출기에서 측정하는 단계;

객체 인식 장치에서 상기 영상 프레임에서 객체의 존재를 인식하는 단계;

상기 객체 인식 장치에 연결된 객체 인식 서비스 시간 측정부는 영상 프레임에 대해 객체 인식에 소요되는 객체 인식 서비스 시간(D_s)을 측정하는 단계; 및

상기 큐로부터 상기 영상 프레임을 수신하고, 상기 다중 소스 선출기로부터 수신한 도착 시간 간격(T_a)과 상기 객체 인식 서비스 시간 측정부로부터 수신한 객체 인식 서비스 시간(D_s)을 비교한 결과에 따라 상기 큐에서 인출된 영상 프레임을 상기 객체 인식 장치로 전달할 지 또는 폐기할 지 판단하는 단계를 포함하고,

상기 객체 인식 장치로 전달할 지 또는 폐기할 지 판단하는 단계는,

상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)보다 크거나 같은 경우, 상기 큐에서 인출된 영상 프레임을 그대로 상기 객체 인식 장치로 전달하는 단계를 포함하는 실시간 객체 인식을 위한 적응형 프레임 제어 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 실시간 객체 인식을 위한 적응형 프레임 관리 시스템에 관한 것으로서, 더욱 상세하게는 입력소스로부터 들어오는 입력 프레임 레이트와 객체 인식 장치의 객체 인식 서비스 레이트를 비교하여 적응형으로 프레임을 제어함으로써 실시간 객체 인식 서비스를 구현하는 실시간 객체 인식을 위한 적응형 프레임 관리 시스템 및 이를 이용한 적응형 프레임 제어 방법에 관한 것이다.

배 경 기 술

[0002] YOLO(You Only Look Once)는 지속적인 버전 업데이트를 통한 사용의 편리성과 높은 객체 인식률 보장으로 인해 최근 많은 지능형 영상 application의 객체인식 소프트웨어로써 사용되고 있다.

[0003] 또한, 최근에는 고성능의 임베디드 보드를 이용한 다양한 인공지능 기반 솔루션들이 개발되고 있다. YOLO는 성공적인 실시간 객체 인식을 위해서 고사양의 하드웨어(GPU 등)를 요구한다.

[0004] 최근에는 이러한 YOLO를 기반으로 하는 객체 인식을 이용한 다양한 애플리케이션들이 개발되고 있다.

[0005] 도 1은 종래 기술에 따른 YOLO 객체 인식 서비스의 구조를 나타낸 도면이고, 도 2는 종래 기술에 따른 YOLO 리얼 타임 프로세싱을 나타낸 도면이다.

[0006] 대다수의 네트워크 카메라는 네트워크로 전송되는 실시간 동영상의 프레임 레이트를 보장할 수 있는 RTSP(Real Time Streaming Protocol) 프로토콜을 이용한다. RTSP는 네트워크로 전송되는 프레임들을 버퍼링하기 위한 RTSP 큐를 가지고 있다.

[0007] 실시간 객체 인식을 위한 프레임 처리 속도는 실제로 수백 ms 정도이며, 애플리케이션에 따라 3 내지 5 fps 또는 10 fps 정도의 실시간 객체 인식 속도를 요구한다.

[0008] 하지만, 기존 YOLO의 객체 인식 서비스 속도가 RTSP로부터 들어오는 프레임 도착 간격보다 느릴 경우, YOLO는 실시간 처리에 대한 심각한 문제를 야기할 수 있다.

[0009] YOLO는 객체 인식 동안 들어오는 영상 프레임들을 큐에 저장하고, 객체 인식이 끝날 때마다 순차적으로 큐로부터 꺼내어 다음 프레임을 처리한다.

[0010] 객체 인식 처리 시간(Ts)이 입력 영상 프레임의 도착 시간 간격(T_a)보다 길 경우, 이어서 들어오는 영상 프레임들은 큐에서 대기하는 시간이 점점 길어지게 된다. 이때, 도 2에 도시된 바와 같이, 각 입력 영상 프레임들의 총 서비스 지연 시간(D(n))은 다음의 수학식 1과 같다.

수학식 1

[0011] $$D(n) = \begin{cases} D_q^{(n-1)} + D_s^{(n-1)} + T_a + D_s^n, & (D_q^{(n-1)} + D_s^{(n-1)} - T_a > 0) \\ D_s^n, & (D_q^{(n-1)} + D_s^{(n-1)} - T_a \leq 0) \end{cases}$$

[0012] 여기서, D_q^n 은 n번째 영상 프레임의 큐에서의 대기 시간, T_a는 입력 영상 프레임의 도착 시간 간격, D_s^n은 YOLO의 객체 인식에 소요되는 객체 인식 서비스 시간이다.

[0013] 각 영상 프레임에 대한 객체 인식 완료 시간은 시간이 갈수록 점점 누적되어 길어지고, 애플리케이션이 요구하는 데드라인(Deadline)을 초과하게 된다.

[0014] 객체 인식 애플리케이션은 누적 지연이 시간이 갈수록 점점 더 커지고, 실시간이 아닌 훨씬 이전 시간의 영상 프레임을 처리하는 결과를 초래하는 문제점이 발생한다. 도 2와 같이, 객체 인식 처리 속도가 지연될 경우, 이후의 영상 프레임들에 지연이 누적되어 실시간 처리가 불가능해지는 것을 알 수 있다.

선행기술문헌

특허문헌

[0015] (특허문헌 0001) 한국 등록특허번호 제10-1921709호

발명의 내용

해결하려는 과제

[0016] 이와 같은 문제점을 해결하기 위하여, 본 발명은 입력소스로부터 들어오는 입력 프레임 레이트와 객체 인식 장치의 객체 인식 서비스 레이트를 비교하여 적응형으로 프레임을 제어함으로써 실시간 객체 인식 서비스를 구현하는 실시간 객체 인식을 위한 적응형 프레임 관리 시스템 및 이를 이용한 적응형 프레임 제어 방법을 제공하는데 그 목적이 있다.

과제의 해결 수단

[0017] 상기 목적을 달성하기 위한 본 발명의 특징에 따른 실시간 객체 인식을 위한 적응형 프레임 관리 시스템은,

[0018] 복수의 입력소스로부터 영상 프레임을 입력받아 큐(Queue)로 전달하고, 입력 영상으로부터 들어오는 영상 프레임의 도착 시간 간격(T_a)을 측정하는 다중 소스 선출기;

[0019] 상기 영상 프레임에서 객체의 존재를 추정하여 객체 인식 결과를 출력하는 객체 인식 장치;

[0020] 상기 객체 인식 장치에 연결되어 상기 영상 프레임에 대해 객체 인식에 소요되는 객체 인식 서비스 시간(D_s)을 측정하는 객체 인식 서비스 시간 측정부; 및

[0021] 상기 큐의 출력단에 연결되어 상기 큐로부터 상기 영상 프레임을 각각 수신하고, 상기 다중 소스 선출기로부터 수신한 도착 시간 간격(T_a)과 상기 객체 인식 서비스 시간 측정부로부터 수신한 객체 인식 서비스 시간(D_s)을 비교한 결과에 따라 상기 큐에서 인출된 영상 프레임을 상기 객체 인식 장치로 전달할 지 또는 폐기할 지 판단하는 출입 허용 제어부를 포함한다.

[0023] 출입 허용 제어부는 상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)보다 작은 경우, 상기 큐에서 인출된 일부 영상 프레임을 폐기(Drop)하고, 나머지 영상 프레임을 상기 객체 인식 장치로 전달한다.

[0025] 출입 허용 제어부는 상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)보다 작은 경우, 최신의 영상 프레임만을 전달하기 위하여 상기 큐의 테일(tail)에 있는 영상 프레임을 제외하고, 상기 큐의 나머지 패킷들을 폐기(Drop)한다.

[0027] 출입 허용 제어부는 상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)보다 크거나 같은 경우, 상기 큐에서 인출된 영상 프레임을 그대로 상기 객체 인식 장치로 전달한다.

[0029] 출입 허용 제어부는 다중 소스 선출기로부터 도착 시간 간격(T_a)을 수신하고, 객체 인식 서비스 시간 측정부로부터 객체 인식 서비스 시간(D_s)을 수신하는 허용 컨트롤러; 및

[0030] 큐의 출력단에 연결되어 상기 큐로부터 상기 영상 프레임을 수신하고, 허용 컨트롤러의 결과값이 상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)보다 작은 경우, 최신의 영상 프레임만을 전달하기 위하여 상기 큐의 테일(tail)에 있는 영상 프레임을 제외하고, 상기 큐의 나머지 패킷들을 폐기(Drop)하고, 상기 허용 컨트롤러의 결과값이 상기 도착 시간 간격(T_a)이 상기 객체 인식 서비스 시간(D_s)보다 크거나 같은 경우, 상기 큐에서 인출된 영상 프레임을 그대로 상기 객체 인식 장치로 전달하는 디멀티플렉서를 포함한다.

[0032] 본 발명의 특징에 따른 실시간 객체 인식을 위한 적응형 프레임 제어 방법은,

[0033] 복수의 입력소스로부터 영상 프레임을 입력받아 큐(Queue)로 전달하고, 입력 영상으로부터 들어오는 영상 프레임의 도착 시간 간격(T_a)을 다중 소스 선출기에서 측정하는 단계;

[0034] 객체 인식 장치에서 상기 영상 프레임에서 객체의 존재를 인식하는 단계;

[0035] 상기 객체 인식 장치에 연결된 객체 인식 서비스 시간 측정부는 영상 프레임에 대해 객체 인식에 소요되는 객체 인식 서비스 시간(D_s)을 측정하는 단계; 및

[0036] 상기 큐로부터 상기 영상 프레임을 수신하고, 상기 다중 소스 선출기로부터 수신한 도착 시간 간격(T_a)과 상기 객체 인식 서비스 시간 측정부로부터 수신한 객체 인식 서비스 시간(D_s)을 비교한 결과에 따라 상기 큐에서 인출된 영상 프레임을 상기 객체 인식 장치로 전달할 지 또는 폐기할 지 판단하는 단계를 포함한다.

발명의 효과

[0037] 전술한 구성에 의하여, 본 발명은 기존의 YOLO에 비해 실시간 객체 탐지 서비스가 가능하여 YOLO의 높은 인식률과 편의성을 제공할 수 있는 효과가 있다.

[0038] 본 발명은 YOLO 내부의 하드웨어를 변경하지 않고, 기존의 YOLO의 객체 인식 성능을 그대로 사용하면서 네트워크로 전송되는 실시간 동영상을 처리할 수 있는 효과가 있다.

도면의 간단한 설명

[0039] 도 1은 종래 기술에 따른 YOLO 객체 인식 서비스의 구조를 나타낸 도면이다.

도 2는 종래 기술에 따른 YOLO 리얼 타임 프로세싱을 나타낸 도면이다.

도 3은 본 발명의 실시예에 따른 실시간 객체 인식을 위한 적응형 프레임 관리 시스템의 구성을 나타낸 도면이다.

도 4는 본 발명의 실시예에 따른 다중 소스 선출기의 내부 로직을 나타낸 도면이다.

도 5는 본 발명의 실시예에 따른 출입 허용 제어부의 내부 로직을 나타낸 도면이다.

도 6은 본 발명의 실시예에 따른 적응형 프레임 관리 시스템을 이용한 리얼 타임 프로세싱을 나타낸 도면이다.

도 7은 적응형 프레임 제어 장치를 가진 객체 인식 장치와 기존의 객체 인식 장치의 객체 인식 서비스 시간을

비교한 결과를 나타낸 도면이다.

도 8은 객체 인식 장치의 하드웨어별 객체 탐지 시간을 나타낸 도면이다.

도 9 내지 도 13은 객체 인식 장치의 각각 다른 하드웨어에서 실행된 RTSP 전송에 대한 각 영상 프레임의 총 처리 지연시간을 나타낸 도면이다.

도 14는 실제 본 실험에서 각 시간대별 객체 인식을 처리하고 있는 영상 프레임을 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

[0040] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0041] 종래의 YOLO는 실시간 객체 인식 처리를 향상하기 위해서 내부 모델을 모두 수정해야 하기 때문에 YOLO의 사용자 편의성 측면에서 문제점이 있으며, 애플리케이션에서 새로운 기법의 적용을 위해서 해당 객체 인식의 성능 자체에 대한 추가적인 검증이 필요한 문제점도 발생한다.

[0042] 그러나 본 발명은 네트워크로 전송되는 실시간 동영상을 처리하기 위하여 YOLO 내부의 하드웨어를 변경하지 않고, YOLO가 실행되고 있는 시스템 하드웨어의 능력에 의존하지 않으며, 기존의 YOLO의 객체 인식 성능을 그대로 사용할 수 있다.

[0043] 본 발명은 기존의 YOLO의 객체인식 성능을 그대로 유지하면서도 실시간 객체인식 처리를 가능하게 하기 위한 적응형 프레임 제어(Adaptive Frame Control, AFC) 방법을 제공한다.

[0044] 본 발명은 YOLO의 다양한 입력 소스에 대해 일관된 객체 인식 서비스를 가능하게 하고, 특히, IP 카메라 연결에 많이 사용되는 RTSP(Real Time Streaming Protocol) 기반의 YOLO 객체 인식 Application에서 발생할 수 있는 실시간 처리에 관한 문제점을 해결할 수 있다. AFC는 기존의 YOLO와 연동하여 YOLO의 인식 성능을 그대로 활용하면서도 다양한 입력소스로부터의 프레임 입력(Frame Input)을 최적화함으로써 다양한 애플리케이션에서 요구하는 실시간 객체 인식을 가능하게 한다.

[0046] 도 3은 본 발명의 실시예에 따른 실시간 객체 인식을 위한 적응형 프레임 관리 시스템의 구성을 나타낸 도면이고, 도 4는 본 발명의 실시예에 따른 다중 소스 선출기의 내부 로직을 나타낸 도면이고, 도 5는 본 발명의 실시예에 따른 출입 허용 제어부의 내부 로직을 나타낸 도면이다.

[0047] 본 발명의 실시예에 따른 실시간 객체 인식을 위한 적응형 프레임 관리 시스템(100)은 적응형 프레임 제어 장치(110) 및 객체 인식 장치(120)를 포함한다.

[0048] 적응형 프레임 제어 장치(110)는 다중 소스 선출기(Multi Source Prefetcher)(111), 큐(Queue)(112) 및 출입 허용 제어부(113)를 포함한다.

[0049] 다중 소스 선출기(111)는 다중 소스 먹스(Multi-Source Mux)(111a), 싱크로나이저(Synchronizer)(111b) 및 프레임 처리부(111c)를 포함한다.

[0050] 객체 인식 장치(120)는 인출 스레드(121), 객체 탐지부(122) 및 디스플레이부(123)를 포함한다.

[0051] 큐(112)는 가장 먼저 들어온 영상의 프레임이 먼저 출력되는 선입선출(First In, First Out) 구조이고, 입력소스로부터 영상의 프레임을 수신하여 저장한다.

[0052] 큐(112)는 입력단에 다중 소스 선출기(111)에 연결되고, 출력단에 출입 허용 제어부(113)에 연결된다.

[0053] 다중 소스 먹스(111a)는 비디오 채널 멀티플렉서(Video Channel Multiplexer)로서 복수의 CCTV(Closed Circuit Television) 카메라의 출력 신호선들과 입력 포트가 각각 접속되고, CCTV의 출력 신호선들에 디지털 비디오 레코더(Digital Video Recorder, DVR) 또는 네트워크 비디오 레코더(Network Video Recorder, NVR)의 입력 신호선들이 각각 접속될 수 있다.

[0054] 이외에 다중 소스 먹스(111a)는 다양한 입력소스에 접속될 수 있다.

[0055] 다중 소스 먹스(111a)는 객체 인식 장치(120)의 실행 시 복수의 입력소스로부터 영상의 프레임을 받아들일 수 있다. 여기서, 입력소스는 RTSP 기반의 네트워크 카메라, Video 파일, USB 카메라, NVR/DVR, CCTV 등을 포함할 수 있다.

[0056] 각 입력소스로부터 들어오는 영상 프레임들은 객체 인식 장치(120)에서 처리할 수 있는 속도로 처리되기 때문에 일관된 처리 속도를 만들 수 없다. 예를 들어, 저장된 CCTV 동영상 파일에 대한 객체 인식을 수행할 경우, 객체 인식 장치(120)가 실행되는 하드웨어가 고사양인 상황(객체 탐지 FPS가 30fps 이상)에서는 객체 인식 서비스 시간이 30 fps보다 더 빠르기 때문에 해당 영상으로부터 객체가 인식된 정확한 시간을 파악하기가 어렵다.

[0057] 적응형 프레임 관리 시스템(100)은 객체 인식 장치(120)의 내부 하드웨어를 변경하지 않고, 객체 인식 장치(120)의 프레임 인출 전에 실시간 입력과 처리에 대한 적응형 프레임 제어를 수행한다.

[0058] 적응형 프레임 제어 장치(110)는 다중 소스 선출기(111)와 출입 허용 제어부(113)의 사이에 큐(112)를 사용하여 다양한 입력으로부터 일관된 실시간 처리를 가능하게 한다.

[0059] 다중 소스 먹스(111a)는 다양한 입력소스로부터 애플리케이션에서 요구하는 target FPS(Frame Per Second)에 일관된 프레임 Input을 가능하게 하는 기능을 수행한다.

[0060] 다중 소스 먹스(111a)는 별도의 스레드(Thread)로 동작되고, 사용자가 원하는 다양한 입력에 대한 프레임들을 빠른 속도로 읽는다. 여기서, 스레드는 다중 소스 선출기(111)에서 입력소스를 받아들어 큐(112)로 전달하는 일련의 수행 프로세스를 나타낸다.

[0061] 싱크로나이저(Synchronizer)(111b)는 스레드에서 다중 소스 먹스(111a)로부터 입력되는 영상 프레임들을 읽어 들이는 시간과, target FPS와 비교하여 target FPS보다 입력 프레임 레이트(Input Frame Rate)가 높은 경우, target FPS에 동기를 맞춰주는 기능을 수행한다.

[0062] 싱크로나이저(111b)는 target FPS보다 빠른 속도로 영상 프레임이 인출되는 경우, target FPS에 맞게 일관된 도착 시간 간격(Interarrival Time)(T_a)을 유지할 수 있다.

[0063] 이와 같이 다중 소스 선출기(111)는 싱크로나이저(111b)를 이용하여 객체 인식 장치(120)가 실행하고 있는 플랫폼에 상관없이 모든 입력소스로부터 target FPS(30 fps)에 맞는 일관된 T_a를 유지할 수 있다.

[0064] 싱크로나이저(111b)를 지난 각 입력 영상 프레임은 프레임 처리부(111c)를 거쳐 큐(112)의 입력으로 들어간다.

[0065] 프레임 처리부(111c)는 입력단에 싱크로나이저(111b)를 연결하고, 출력단을 큐(112)의 입력단에 연결하며, 일측이 출입 허용 제어부(113)에 연결된다.

[0066] 프레임 처리부(111c)는 각각의 입력 영상으로부터 들어오는 영상 프레임의 도착 시간 간격(Interarrival Time)(T_a)을 지속적으로 측정하고, 측정한 도착 시간 간격을 출입 허용 제어부(113)로 전송한다.

[0067] 각 입력 영상 프레임의 도착 시간 간격(T_a)은 출입 허용 제어부(113)에서 프레임 허가 제어를 위해 사용되는 파라미터이다.

[0068] 출입 허용 제어부(113)는 큐(112)의 출력단과 객체 인식 장치(120)의 인출 스레드(121)의 사이에 연결되고, 간단하면서도 강력한 프레임 제어를 통해 객체 인식 장치(120)로의 실시간 객체 인식 서비스를 가능하게 한다.

[0069] 출입 허용 제어부(113)는 큐(112)로부터 영상 프레임을 추출하여 객체 인식 장치(120)의 인출 스레드(121)로 전달한다.

[0070] 출입 허용 제어부(113)는 허용 컨트롤러(113a) 및 디멀티플렉서(113b)를 포함한다.

[0071] 객체 인식 서비스 시간 측정부(114)는 객체 인식 장치(120)의 객체 탐지부(122)에 연결되어 영상 프레임에 대해 객체 인식에 소요되는 객체 인식 서비스 시간(D_s)을 측정한다.

[0072] 객체 인식 서비스 시간은 객체 인식 장치(120)에서 한 프레임에 대한 객체 인식 서비스를 수행하는데 걸리는 시간이다.

[0073] 객체 인식 서비스 시간 측정부(114)는 출입 허용 제어부(113)에 연결되어 객체 탐지부(122)에서 측정된 객체 인식 서비스 시간(D_s)을 전달한다.

[0074] 허용 컨트롤러(113a)는 프레임 처리부(111c)에서 측정되는 입력 영상 프레임 간의 도착 시간 간격(T_a)과, 객체 탐지부(122)에서 측정된 객체 인식 서비스 시간(D_s)을 수신한다.

[0075] 출입 허용 제어부(113)는 객체 인식 장치(120)에서 한 프레임의 처리가 완료되어 다음 새로운 프레임을 인출(Fetch)하는 순간에 동작(Trigger)된다.

[0076] 허용 컨트롤러(113a)는 프레임 처리부(111c)에서 측정되는 입력 영상 프레임의 도착 시간 간격(T_a)과, 객체 탐지부(122)에서 측정된 객체 인식 서비스 시간(D_s)을 비교한다.

[0077] 디멀티플렉서(113b)는 허용 컨트롤러(113a)의 결과값에 따라 큐(112)에서 인출된 영상 프레임을 객체 인식 장치(120)로 전달할 지 또는 폐기할 지 판단한다.

[0078] 도착 시간 간격(T_a)이 객체 인식 서비스 시간(D_s)보다 크거나 같은 경우, 고성능의 하드웨어 시스템에서 동작되는 상황으로 객체 인식 서비스 시간이 입력 프레임 레이트보다 빠른 상황이거나 T_a = D_s = 0인 초기 상황이기 때문에 큐에서 객체 인식 장치(120)에서 실행 중에 입력된 영상 프레임만이 존재한다.

[0079] 디멀티플렉서(113b)는 도착 시간 간격(T_a)이 객체 인식 서비스 시간(D_s)보다 크거나 같은 경우, 별다른 조치 없이 큐(112)에서 인출된 영상 프레임을 그대로 객체 인식 장치(120)의 인출 스레드(121)로 전달한다.

[0080] 도착 시간 간격(T_a)이 객체 인식 서비스 시간(D_s)보다 작은 경우, 상대적으로 낮은 성능의 하드웨어에서 객체 인식 장치(120)가 실행되고 있거나 순간적으로 또는 특정 시간대에 다른 프로세스에 의해 GPU 또는 메모리가 부족하여 객체 인식 장치(120)의 객체 인식 서비스 시간이 증가하는 경우이다. 이러한 경우는 빠른 입력 프레임 레이트로 인해 객체 인식 장치(120)가 한 프레임의 객체 인식 서비스하는 동안 큐(112)에 이미 복수의 다음 프레임들이 저장된다.

[0081] 이렇게 큐(112)에 저장된 프레임에 의해 실시간 프레임 처리에 대한 문제가 발생된다.

[0082] 디멀티플렉서(113b)는 도착 시간 간격(T_a)이 객체 인식 서비스 시간(D_s)보다 작은 경우, 객체 인식 장치(120)의 인출 스레드(121)로 가장 최신의 영상 프레임만을 전달하기 위하여 큐(112)의 테일(tail)에 있는 영상 프레임을 제외하고, 현재 큐(112)의 나머지 패킷들을 모두 폐기(Drop)한다. 여기서, 테일(tail)은 큐(112)에 마지막 패킷에 대한 포인터로 큐(112)에 입력된 가장 최신의 프레임을 나타내고, 헤드(head)는 큐(112)에서 첫 번째 패킷에 대한 포인터를 나타낸다.

[0083] 폐기되는 영상 프레임의 수는 현재 T_a와 D_s의 차이에 의해 달라질 수 있다. 예를 들어, 입력 프레임 레이트가 30fps이고, 객체 인식 장치(120)의 서비스 프레임 레이트가 15fps인 경우, 실제 초당 가장 최신의 15 프레임만이 객체 인식 장치(120)의 객체 인식을 수행하고, 나머지 15 프레임이 폐기된다.

[0084] 애플리케이션 측면에서는 객체 인식 장치(120)의 객체 인식 서비스 시간에 수렴되는 속도로 실시간 프레임 처리가 가능하다.

[0085] 애플리케이션 측면에서는 모든 프레임에 대하여 객체 인식을 하지 않아도 전체적인 객체 인식률에 영향을 크게 미치지 않는다.

[0086] 따라서, 애플리케이션 측면에서는 폐기된 영상 프레임들로 인하여 객체 인식 장치(120)의 실시간성을 보장해주는 결과를 초래하며, 애플리케이션의 특성에 알맞은 하드웨어 선택이 가능하여 비용 효율적인 시스템 설계를 가능하게 한다.

[0088] 객체 인식 장치(120)는 심층 신경망(Deep Neural Networks, DNN), 컨볼루션 신경망 (Convolutional deep Neural Networks, CNN), 순환 신경망(Reccurent Neural Network, RNN) 및 심층 신뢰 신경 망(Deep Belief Networks, DBN) 중 어느 하나의 신경망을 이용하여 입력 영상으로부터 특징맵을 추출한다.

[0089] 객체 인식 장치(120)는 딥러닝(Deep learning)을 기반으로 학습부에 의하여 이미 학습이 완료된 모델을 이용하

여서 특징맵을 생성할 수 있다. 딥러닝은 여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화(Abstractions, 다량의 데이터나 복잡한 자료들 속에서 핵심적인 내용 또는 기능을 요약하는 작업)를 시도하는 기계학습(Machine Learning) 알고리즘의 집합으로 정의된다.

[0090] 객체 인식 장치(120)의 객체 탐지부(122)는 영상 프레임에서 객체가 존재할 것으로 추정되는 영역을 추출하고, 추출된 영역으로부터 특징을 나타내는 특징맵을 추출한다.

[0091] 객체 탐지부(122)는 추출한 특징맵을 기초로 영상에서 객체의 존재가 추정되는 적어도 하나의 영역을 추출한다. 영역을 추출하는 방법은 예를 들어 faster RCNN, SSD(Single Shot MultiBox Detector), YOLO(You Only Look Once) 등이 있을 수 있으며, 본 발명의 객체 인식 장치(120)는 YOLO 객체 인식모듈을 일례로 하고 있다.

[0092] 객체 탐지부(122)는 특징맵 중에서 영상의 영역별 클래스의 좌표를 포함하는 특징맵을 선정하고, 선정된 특징맵으로부터 영역을 구별하는 좌표를 식별한 뒤, 식별된 좌표를 개체의 존재가 추정되는 영역으로 추출할 수 있다.

[0093] 객체 탐지부(122)는 물건, 사람, 동물 등 다양한 개체를 하나 또는 2개 이상으로 설정할 수 있다.

[0094] 또한, 객체 탐지부(122)는 추출된 적어도 하나의 영역 각각에 대해서, 해당 객체의 최외곽을 둘러싸는 바운딩 박스(Bounding Box)로서 표시할 수 있다.

[0095] 각각의 바운딩 박스는 영상에서 해당 바운딩 박스의 위치에 개체의 존재 가능성이 있음을 나타낸다.

[0096] 디스플레이부(123)는 영상 정보를 나타내는 프레임을 입력으로 받아 해당 프레임 내에서 개체의 위치 좌표((X1, Y1), (X2, Y2))를 바운딩 박스로 한 결과 정보를 출력한다.

[0097] 객체 인식 장치(120)는 영상 프레임을 입력받아 플레이 했을 때, 초당 15 내지 30 프레임을 처리해서 바운딩 박스가 표시된 영상 정보를 출력한다.

[0099] 도 6은 본 발명의 실시예에 따른 적응형 프레임 관리 시스템을 이용한 리얼 타임 프로세싱을 나타낸 도면이고, 도 7은 적응형 프레임 제어 장치를 가진 객체 인식 장치와 기존의 객체 인식 장치의 객체 인식 서비스 시간을 비교한 결과를 나타낸 도면이다.

[0100] 적응형 프레임 제어 장치(110)에서의 실시간 프레임 처리의 일례를 나타낸 것이다. 도 6과 같이, 적응형 프레임 제어 장치(110)는 모든 입력 영상 프레임에 대해 객체 인식 서비스를 수행하지 않고, 가장 최신의 영상 프레임만을 처리한다.

[0101] 따라서, 실제 처리되는 영상 프레임들에 대한 총 서비스 시간은 다음의 수학식 2와 같다.

수학식 2

[0102] $$D(n) = D_q^n + D_s^n$$

[0103] 적응형 프레임 제어 장치(110)는 큐(112)에 유입된 가장 최신의 영상 프레임을 처리하기 때문에 $0 \le D_q^n \le \frac{T_a}{D_s} \ (T_a < D_s)$ 와 같으며, 상대적으로 작은 값이므로 총 서비스 시간이 거의 D_s^n에 수렴된다.

[0104] 도 7은 각각의 입력 영상 프레임 간의 도착 시간 간격(Interarrival Time)(T_a)이 30 fps이고, 객체 인식 장치(120)의 객체 인식에 소요되는 객체 인식 서비스 시간이 15 fps일 때, 애플리케이션 데드라인이 0.3초인 경우이다.

[0105] 기존의 YOLO의 경우, 큐에 저장된 영상 프레임에 대한 처리 시간은 시간이 경과됨에 따라 딜레이가 누적됨을 볼 수 있다.

[0106] 실제 큐의 사이즈는 유한하기 때문에 유한한 큐를 고려할 경우, 큐 최대 사이즈에 수렴된 지연이 계속해서 발생하게 된다.

[0107] 이와 달리 본 발명의 적응형 프레임 제어 장치(110)는 가장 최신의 프레임만을 객체 인식 서비스에 사용하기 때

문에 누적 지연이 발생되지 않고, 애플리케이션의 데드라인 이내에 각 영상 프레임을 처리할 수 있음을 볼 수 있다.

[0108] 본 발명의 적응형 프레임 제어 장치(110)의 개발 환경 및 YOLO와 비교 실험을 위한 기본적인 환경 구성은 다음의 표 1과 같다.

표 1

OS	Linux 18.04
Programming Environment	C++, OpenCV (4.1.1)
CUDA	Version 10.2
CuDNN	Version 8.0.0
Hardware Platform	Jetson Nano, Jetson TX2, Jetson Xavier NX, Jetson AGX Xavier, Laptop with CPU(i7-8750H), Memory (16GB) and GPU(GTX1060),
Object Detector	YOLOv3, YOLO tiny
Input source	IP Camera (RTSP) (640 x 480, 30fps) USB cam (H.264, 640 x 480, 30fps) Video files (H.264, 640 x 480, 30fps)

[0109]

[0110] 도 8은 객체 인식 장치의 하드웨어별 객체 탐지 시간을 나타낸 도면이고, 도 9 내지 도 13은 객체 인식 장치의 각각 다른 하드웨어에서 실행된 RTSP 전송에 대한 각 영상 프레임의 총 처리 지연시간을 나타낸 도면이다.

[0111] 도 8에서 GTX 1060은 하드웨어 중 가장 좋은 성능을 보이는 시스템이다.

[0112] YOLO416과 YOLO608은 1.4초와 2.8초의 지연이 발생함을 확인할 수 있다. 이러한 결과는 현재 실제 카메라로 보여지는 이벤트 또는 객체가 객체 인식 장치에서 1.5초에서 2.8초 이후에 확인이 되는 것을 보여준다. 즉, 30 FPS의 객체 탐지 속도보다 낮은 FPS의 상황에서는 RTSP 입력소스로부터 이벤트에 대한 실시간 처리를 보장할 수 없다는 것을 증명한다.

[0113] AFC 416, AFC 128은 0.3초 이내의 지연으로 객체 인식 서비스가 수행됨을 볼 수 있으며, YOLO와 같은 시스템, 같은 입력 조건, 같은 FPS 처리율에도 불구하고, 실시간 처리를 보장하는 것을 볼 수 있다. 여기서, AFC는 본 발명의 적응형 프레임 제어(Adaptive Frame Control) 방법이 적용된 시스템을 나타낸다.

[0114] 이러한 객체 인식 서비스 지연의 문제는 도 10 내지 도 13의 저사양의 하드웨어로 내려갈수록 더 심각해짐을 볼 수 있다.

[0115] 만약, 0.5초의 객체 인식 데드라인을 가지는 애플리케이션에서 415×415 영상 사이즈의 객체 인식을 수행하기 원한다면, YOLO의 경우, GTX 1060을 탑재한 컴퓨터로도 요구 사항을 만족할 수 없고, 더 고사양의 시스템을 고려해야 한다.

[0116] 그러나 본 발명의 적응형 프레임 제어(AFC) 방법을 사용하는 경우, 도 10과 같은 조건의 Jetson NX를 사용하여 애플리케이션의 요구 사항을 만족할 수 있다.

[0117] 이때, 실제로 Jetson NX와 GTX 1060 이상의 시스템과의 가격을 비교하면, NX가 적어도 1/6 이상의 낮은 가격으로 애플리케이션을 구성할 수 있다.

[0118] 만약, 영상 사이즈에 대한 요구 사항이 더욱 낮아지는 경우(128×128, 256×256), 도 13에 도시된 바와 같이, Jetson Nano를 하드웨어로 고려할 수 있다.

[0119] 반면에 기존 YOLO의 경우, 객체 인식 서비스 지연이 상당히 심각하게 발생하는 것을 확인할 수 있다.

[0120] 이러한 성능의 차이는 기존의 YOLO의 경우, 입력 프레임 레이트와 객체 인식에 소요되는 서비스 레이트 간의 차이에 대한 문제를 간과하였으나, 본 발명의 적응형 프레임 제어(AFC) 방법에서는 다중 소스 선출기를 통해 입력

단의 프레임 레이트를 조절하고, 별도의 큐(112)를 사용하여 RTSP의 버퍼링에 의존하지 않고, 출입 허용 제어부(113)를 통해 입력 프레임 레이트와 객체 인식 장치(120)의 객체 인식 서비스 레이트에 따라 탄력적으로 프레임을 제어한 결과이다.

[0121] 이와 같이 본 발명의 실험 결과는 하드웨어 제약이 있는 시스템에서 RTSP 입력소스에 대한 AFC의 실시간 객체 탐지 서비스 능력이 기존 YOLO에 비해 상당히 우수함을 입증한다.

[0122] 도 14는 실제 본 실험에서 각 시간대별 객체 인식을 처리하고 있는 영상 프레임을 보여주고 있다.

[0123] 이상에서 본 발명의 실시예는 장치 및/또는 방법을 통해서만 구현이 되는 것은 아니며, 본 발명의 실시예의 구성에 대응하는 기능을 실현하기 위한 프로그램, 그 프로그램이 기록된 기록 매체 등을 통해 구현될 수도 있으며, 이러한 구현은 앞서 설명한 실시예의 기재로부터 본 발명이 속하는 기술분야의 전문가라면 쉽게 구현할 수 있는 것이다.

[0124] 이상에서 본 발명의 실시예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

부호의 설명

[0125] 100: 적응형 프레임 관리 시스템

110: 적응형 프레임 제어 장치

111: 다중 소스 선출기

111a: 다중 소스 먹스

111b: 싱크로나이저

111c: 프레임 처리부

112: 큐

113: 출입 허용 제어부

113a: 허용 컨트롤러

113b: 디멀티플렉서

114: 객체 인식 서비스 시간 측정부

120: 객체 인식 장치

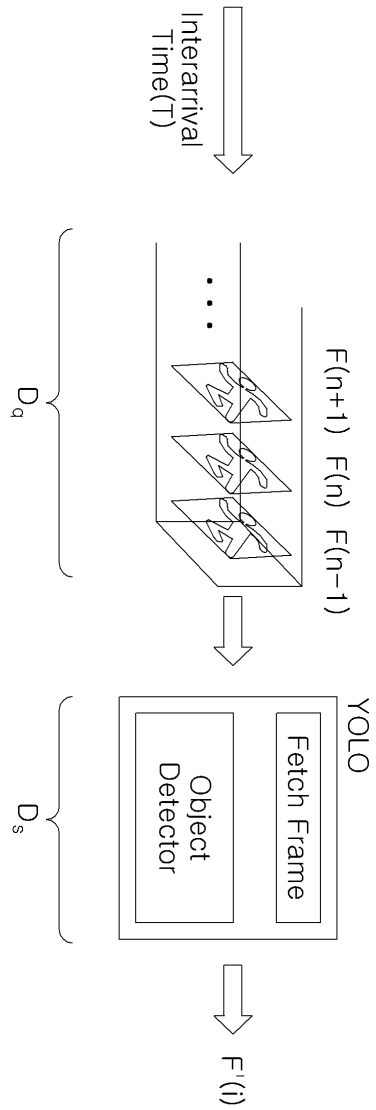
121: 인출 스레드

122: 객체 탐지부

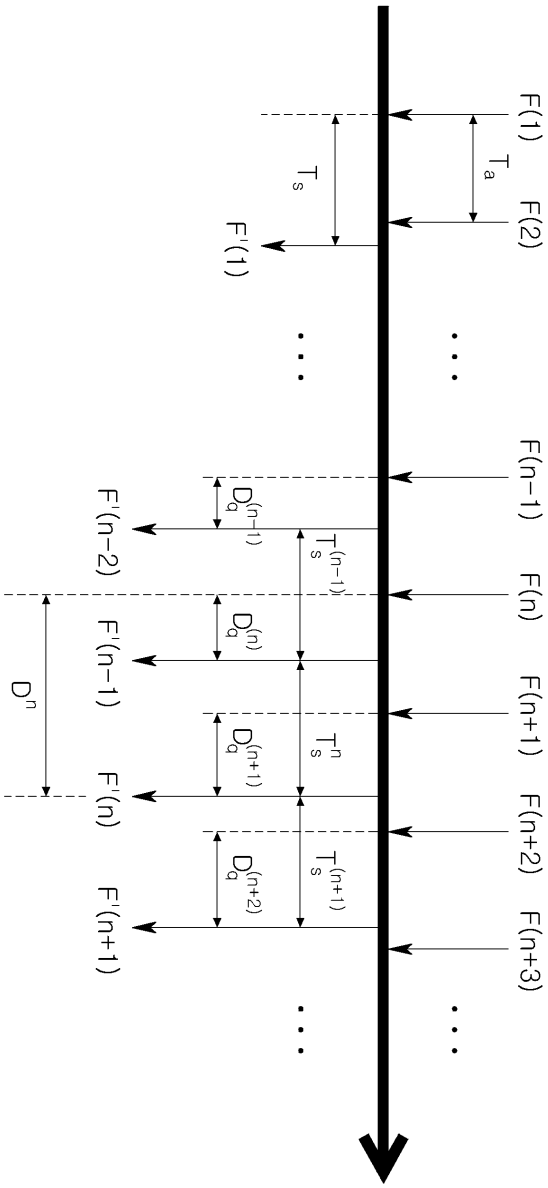
123: 디스플레이부

도면

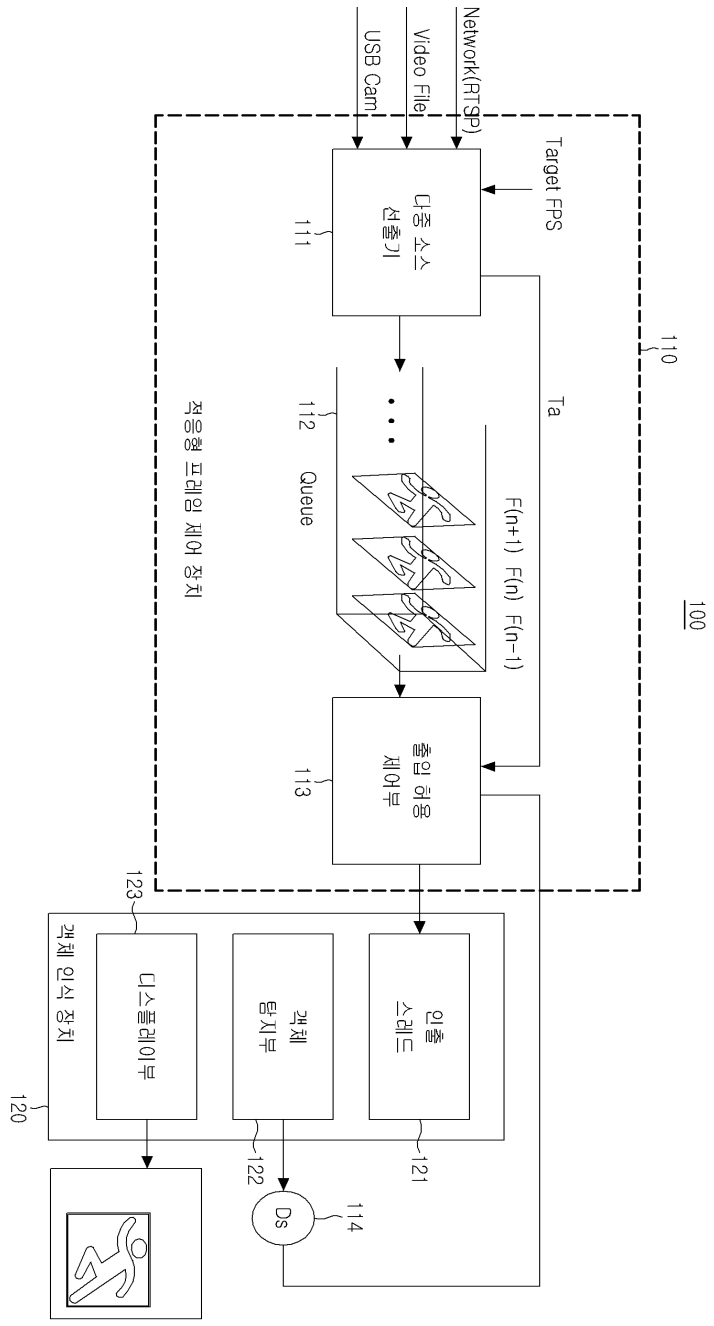
도면1


Interarrival Time(T)
F(n+1) F(n) F(n-1)
D_q
YOLO
Fetch Frame
Object Detector
D_s
F'(i)

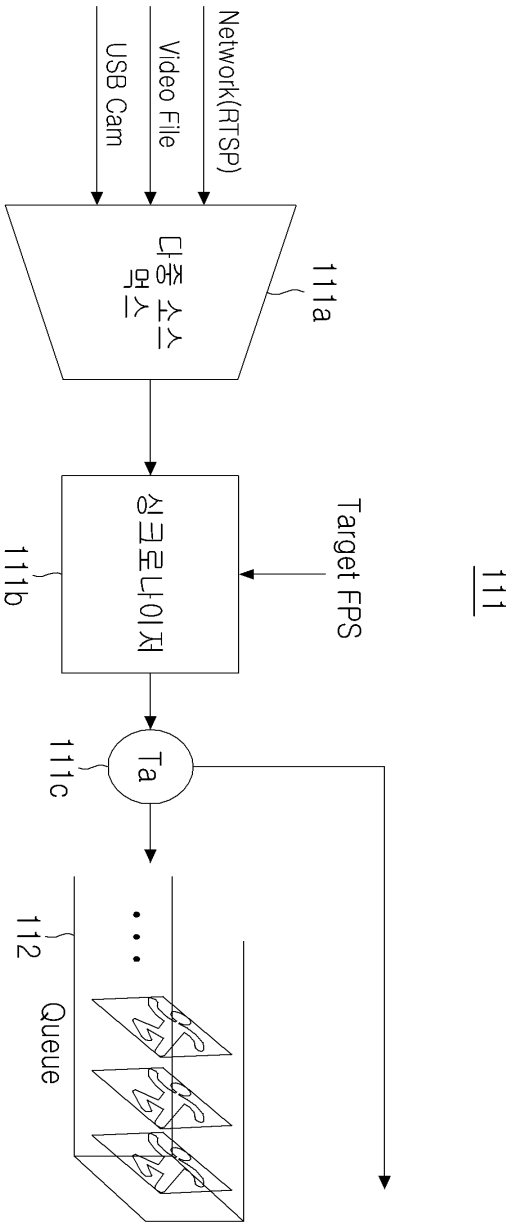
도면2


F(1)
F(2)
F(n-1)
F(n)
F(n+1)
F(n+2)
F(n+3)
T_a
T_s
F'(1)
$D_q^{(n-1)}$
F'(n-2)
$T_s^{(n-1)}$
$D_q^{(n)}$
F'(n-1)
D^n
T_s^n
$D_q^{(n+1)}$
F'(n)
$T_s^{(n+1)}$
$D_q^{(n+2)}$
F'(n+1)

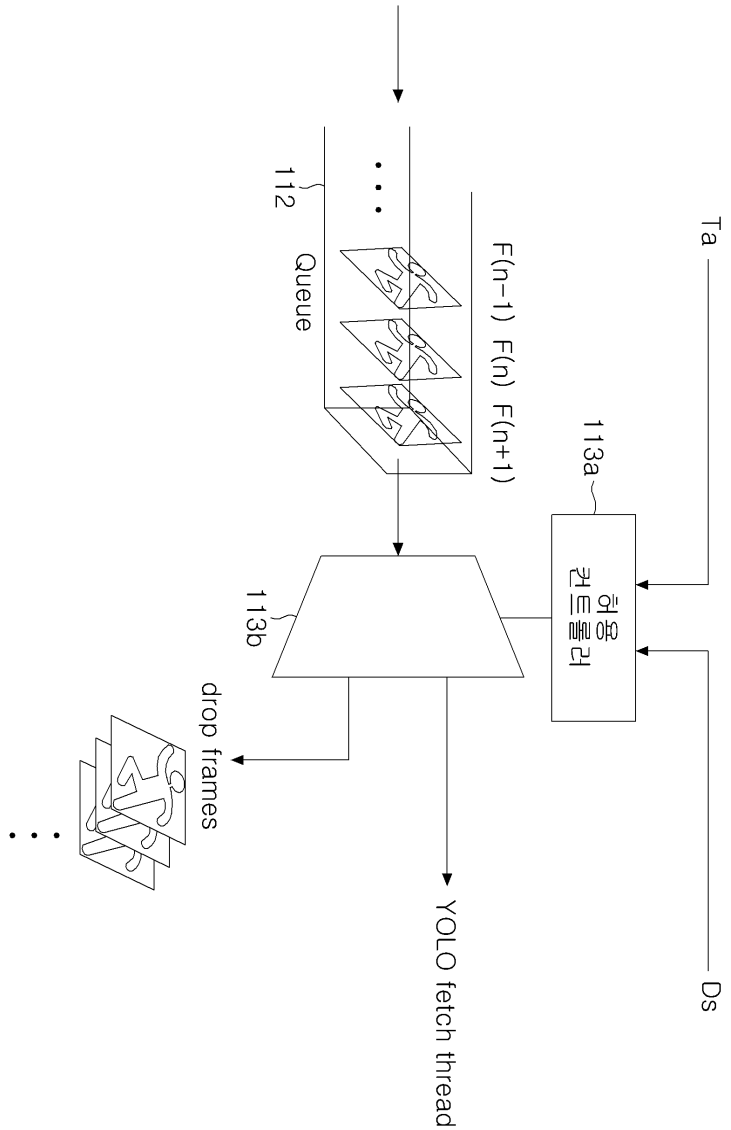
도면3


Network(RTSP)
Video File
USB Cam
Target FPS
다중 소스 선출기
111
110
적응형 프레임 제어 장치
112 Queue
F(n+1) F(n) F(n-1)
Ta
100
출입 허용 제어부
113
123
객체 인식 장치
디스플레이부
객체 탐지부
인출 스레드
120
122
121
Ds
114

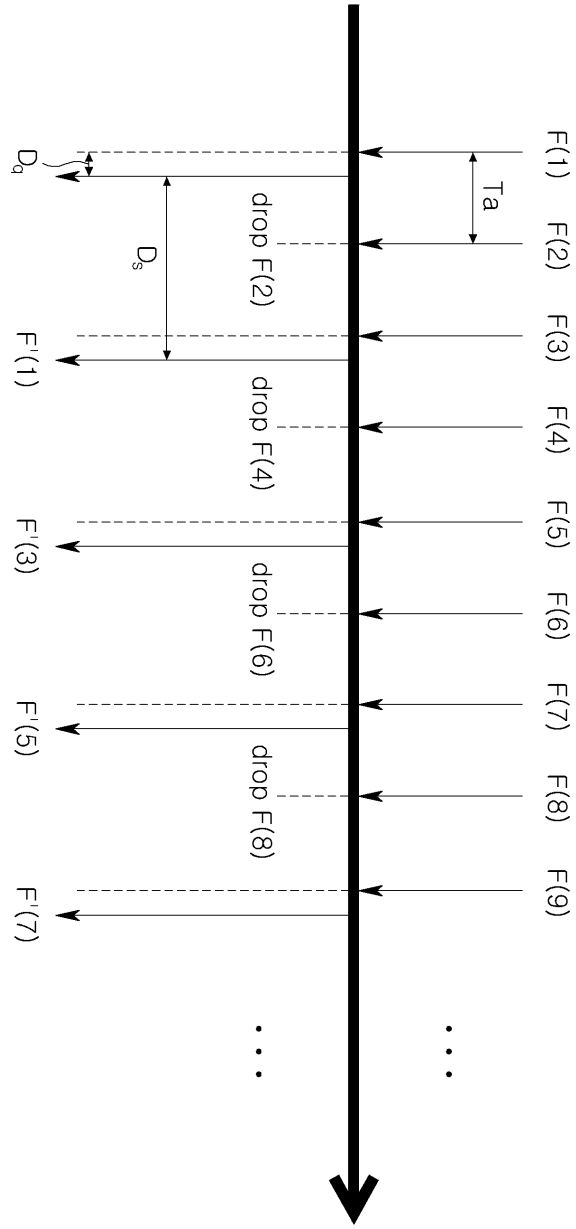
도면4


Network(RTSP)
Video File
USB Cam
다중 소스 멕스
111a
싱크로나이저
111b
Target FPS
Ta
111c
111
112
Queue

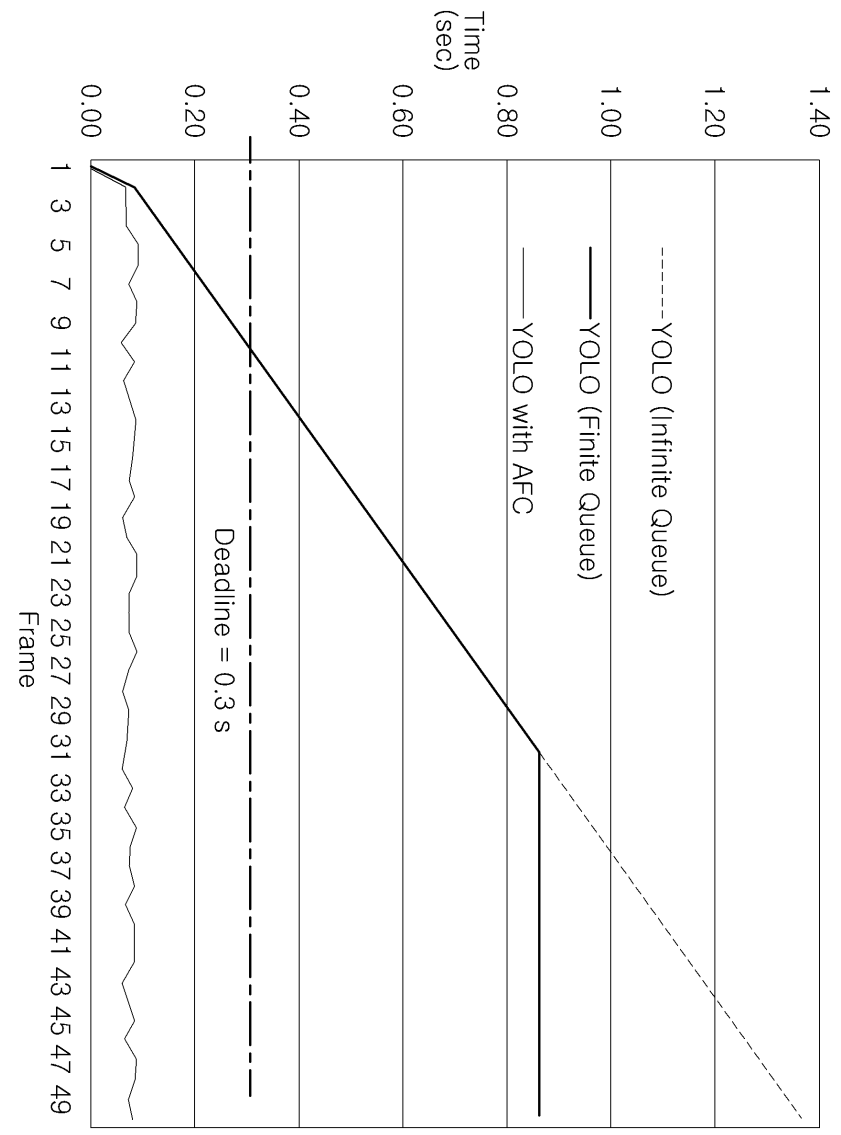
도면5


112
Queue
F(n-1) F(n) F(n+1)
113a
허용
컨트롤러
Ta
Ds
113b
drop frames
YOLO fetch thread

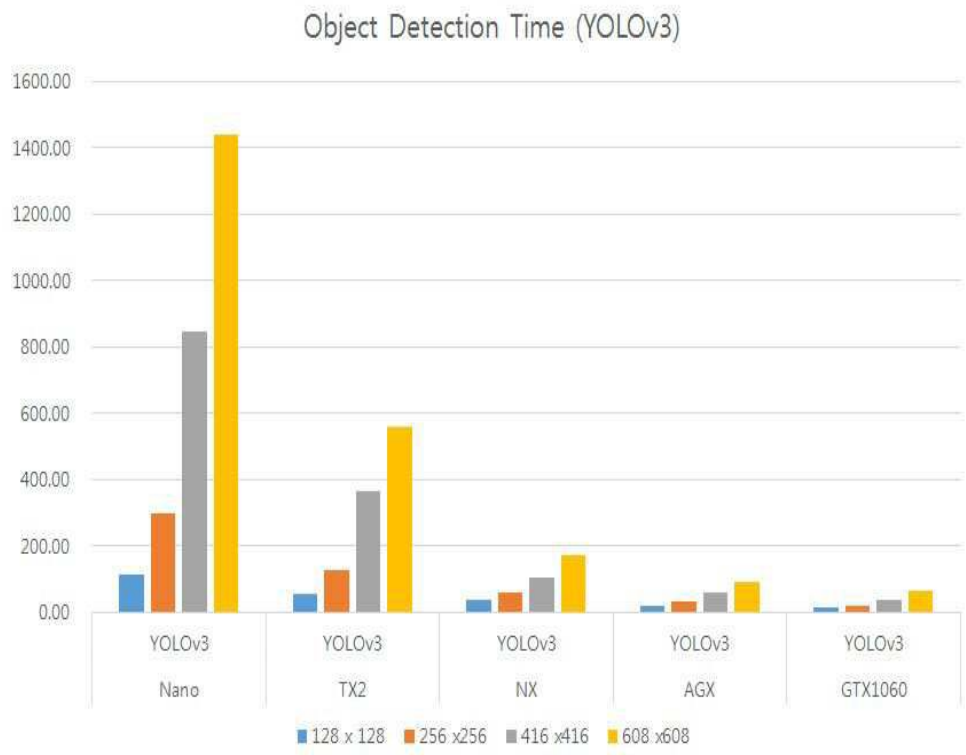
도면6


F(1)
F(2)
F(3)
F(4)
F(5)
F(6)
F(7)
F(8)
F(9)
Ta
D_q
D_s
drop F(2)
drop F(4)
drop F(6)
drop F(8)
F'(1)
F'(3)
F'(5)
F'(7)

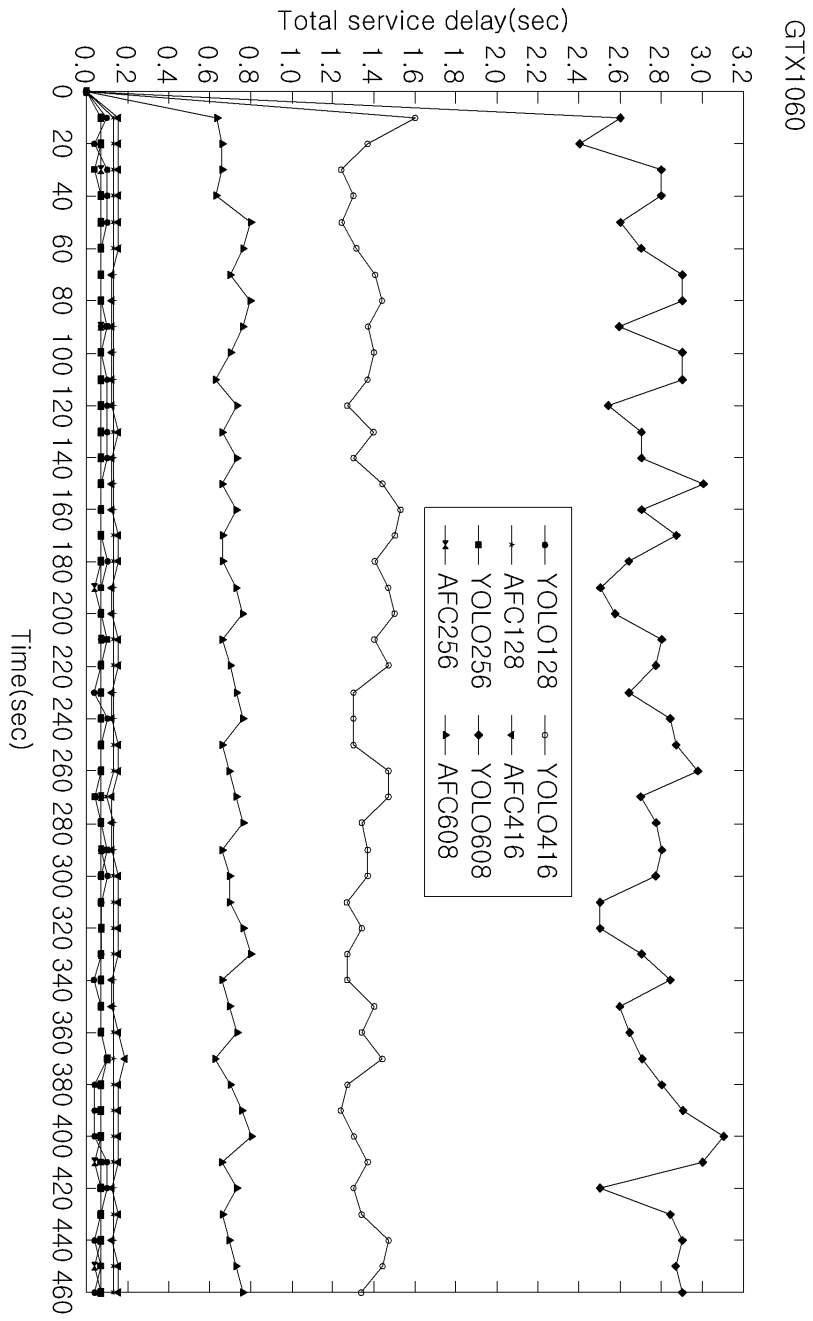
도면7


1.40
1.20
1.00
0.80
0.60
0.40
0.20
0.00
Time (sec)
YOLO (Infinite Queue)
YOLO (Finite Queue)
YOLO with AFC
Deadline = 0.3 s
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
Frame

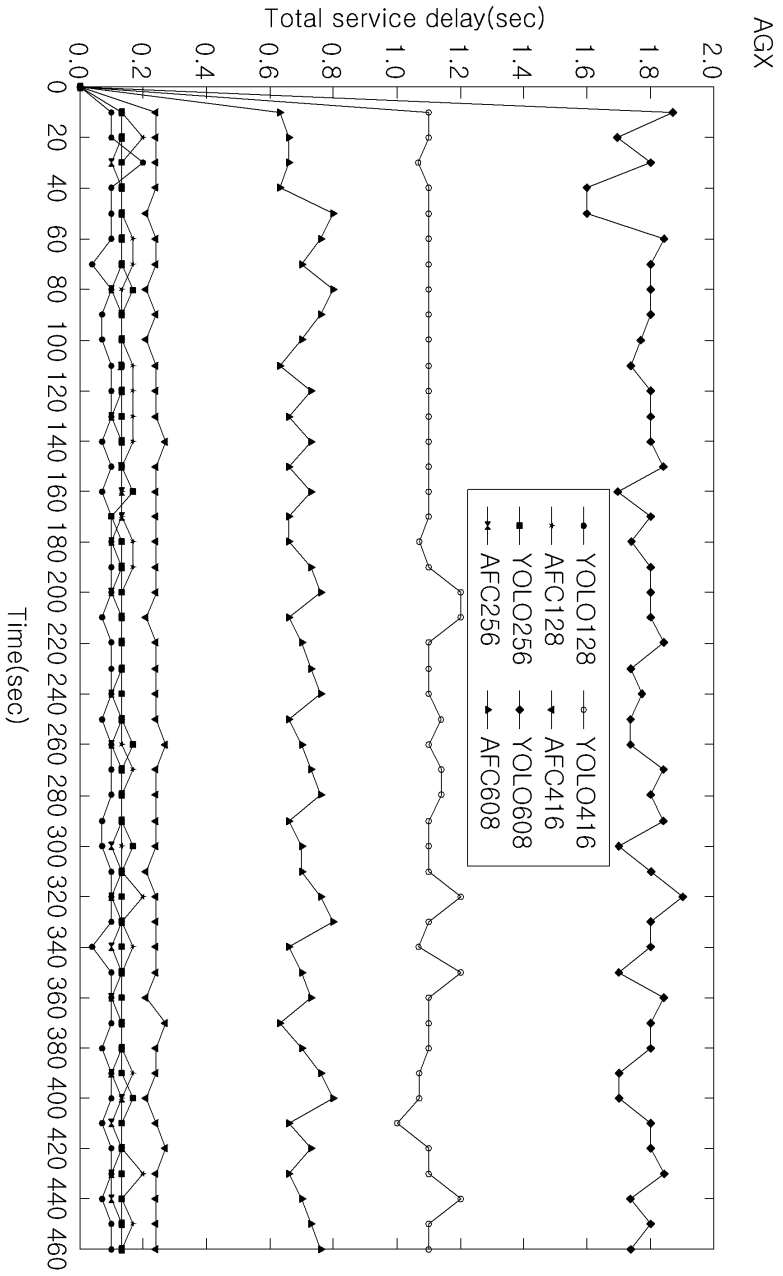
도면8


Object Detection Time (YOLOv3)
1600.00
1400.00
1200.00
1000.00
800.00
600.00
400.00
200.00
0.00
YOLOv3 Nano
YOLOv3 TX2
YOLOv3 NX
YOLOv3 AGX
YOLOv3 GTX1060
128 x 128
256 x256
416 x416
608 x608

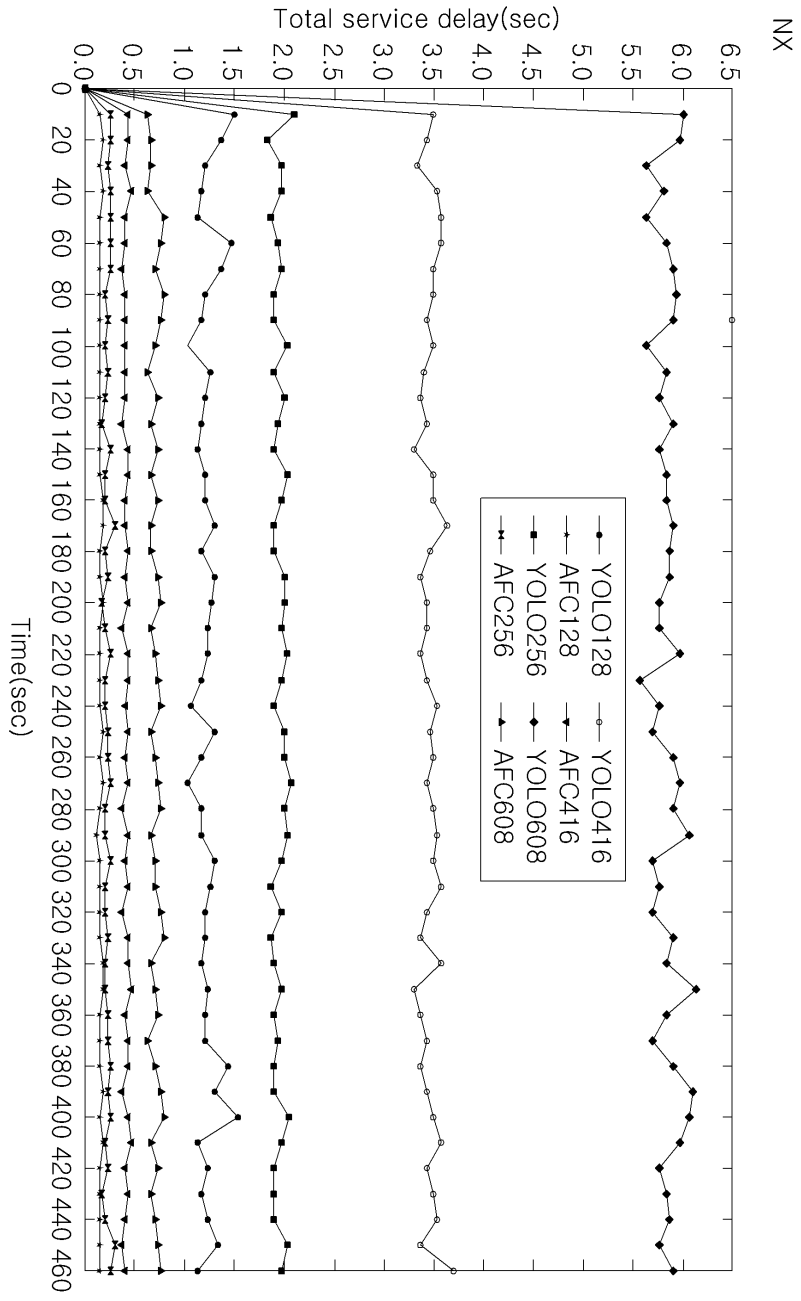
도면9


GTX1060
Total service delay(sec)
Time(sec)
YOLO128
YOLO416
AFC128
AFC416
YOLO256
YOLO608
AFC256
AFC608

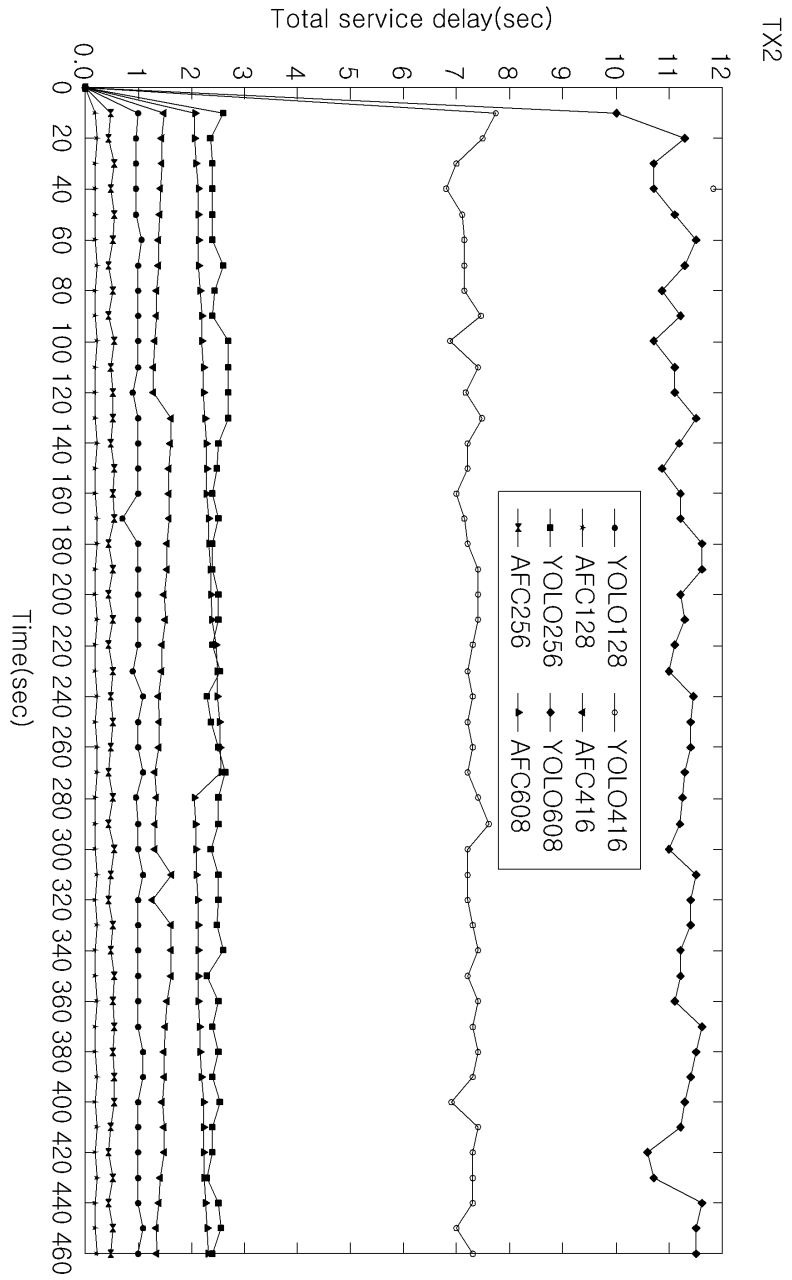
도면10


AGX
Total service delay(sec)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
360
380
400
420
440
460
Time(sec)
YOLO128
YOLO416
AFC128
AFC416
YOLO256
YOLO608
AFC256
AFC608

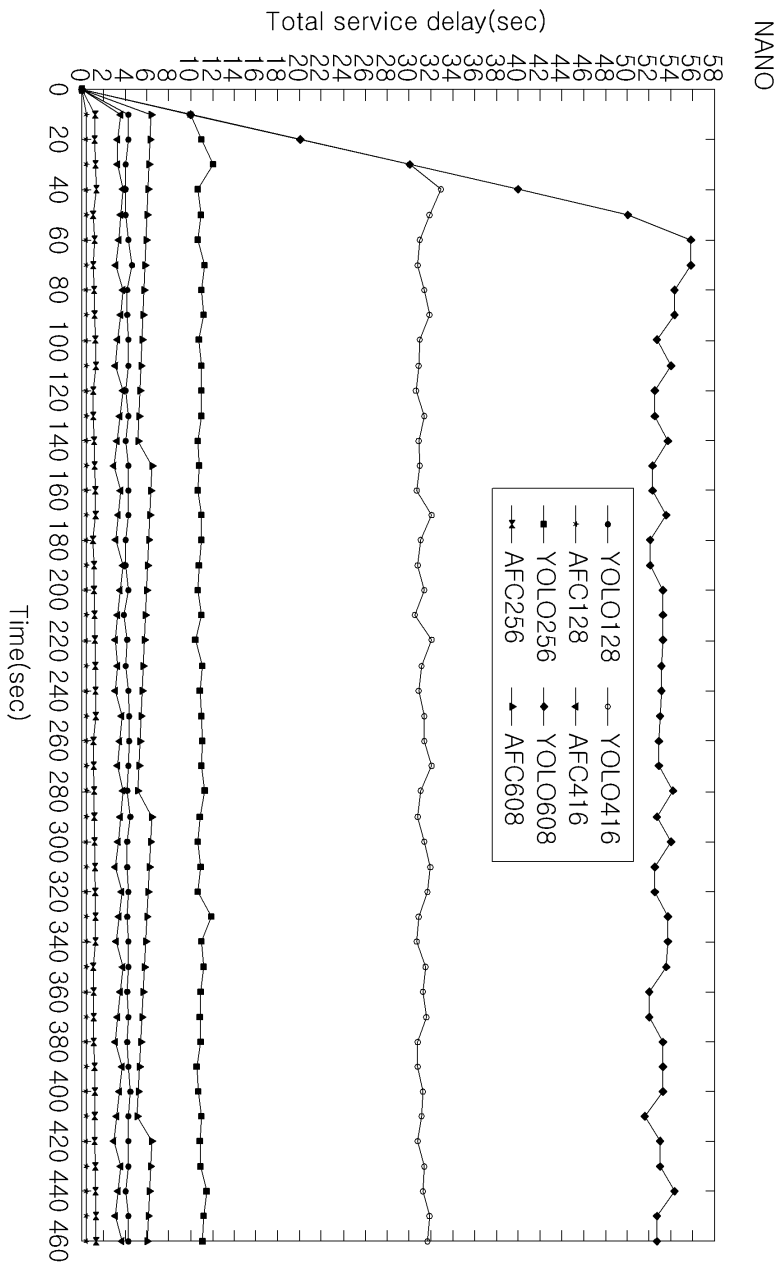
도면11


NX
Total service delay(sec)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
360
380
400
420
440
460
Time(sec)
YOLO128
YOLO416
AFC128
AFC416
YOLO256
YOLO608
AFC256
AFC608

도면12


TX2
Total service delay(sec)
0.0
1
2
3
4
5
6
7
8
9
10
11
12
Time(sec)
0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360 380 400 420 440 460
YOLO128
YOLO416
AFC128
AFC416
YOLO256
YOLO608
AFC256
AFC608

도면13


NANO
Total service delay(sec)
Time(sec)
YOLO128
YOLO416
AFC128
AFC416
YOLO256
YOLO608
AFC256
AFC608

도면14


Ground Truth
GTX 1060 (416x416)
AFC
YOLO
NX (416x416)
AFC
YOLO
Nano (256x256)
AFC
YOLO