creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

조선대학교
CHOSUN UNIVERSITY

[UCI]I804:24011-200000266641

2018년 2월

석사학위논문

비정상적인 상황 검출을 위한 임베디드 시스템 구현에 관한 연구

조 선 대 학 교 대 학 원

제어계측공학과

김 진 수

조선대학교
CHOSUN UNIVERSITY

비정상적인 상황 검출을 위한 임베디드 시스템 구현에 관한 연구

A Study on Implementation of Embedded System for Abnormal Situation Detection

2018년 2월 23일

조 선 대 학 교 대 학 원

제 어 계 측 공 학 과

김 진 수

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

비정상적인 상황 검출을 위한 임베디드 시스템 구현에 관한 연구

지도교수 반 성 범

이 논문을 공학석사학위신청 논문으로 제출함.

2017년 10월

조 선 대 학 교 대 학 원

제어계측공학과

김 진 수

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

김진수의 공학석사학위논문을 인준함

위원장 조선대학교 교수 곽근창 (인)

위 원 조선대학교 교수 고낙용 (인)

위 원 조선대학교 교수 반성범 (인)

2017년 11월

조선대학교 대학원

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

목 차

제1장 서론 ····· 1
제2장 지능형 영상감시 시스템 관련 연구 ····· 4
제1절 지능형 영상감시 시스템 ····· 4
제2절 비정상적인 상황 검출 알고리즘 ····· 9
제3장 다중 비정상 상황 검출 알고리즘 ····· 13
제1절 학습된 정보 기반 침입자 검출 ····· 14
제2절 색상과 움직임 정보 기반 화재 검출 ····· 16
제3절 인체 움직임 정보 기반 배회 · 낙상 검출 ····· 18
제4장 제안하는 지능형 영상감시 시스템 ····· 23
제1절 임베디드 모듈 구현 ····· 24
제2절 알고리즘 최적화 ····· 27
제3절 임베디드 모듈 최적화 ····· 31
제5장 실험결과 ····· 34
제6장 결론 ····· 42
참고문헌 ····· 43

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

표 목차

표 1-1. 지역별 CCTV 수 및 통제 인원 ····· 2
표 2-1. 컴퓨터 기반 지능형 영상감시 시스템 ····· 6
표 2-2. 소형 디바이스 기반 지능형 영상감시 시스템 ····· 8
표 3-1. 각도 변화량에 따른 가중치 ····· 20
표 4-1. 라즈베리파이 규격 ····· 24
표 5-1. 컴퓨터와 임베디드 모듈 처리 시간 비교 ····· 34
표 5-2. 컴퓨터와 임베디드 모듈 규격 비교 ····· 35
표 5-3. 비정상적인 상황 검출 알고리즘 성능 ····· 40

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

그림 목차

그림 1.1 사건 · 사고 발생 현황 ····· 1
그림 2.1 분산처리를 이용한 지능형 영상감시 시스템 ····· 5
그림 2.2 인체 외형 기반 낙상 검출 시스템 ····· 5
그림 2.3 라즈베리파이와 아두이노를 이용한 침입자 검출 시스템 ····· 7
그림 2.4 라즈베리파이 기반 지능형 영상감시 시스템 블록도 ····· 7
그림 2.5 UAV 제어 시스템 ····· 8
그림 2.6 표정, 조명, 각도 변화 영상을 이용한 얼굴인식 ····· 10
그림 2.7 색상과 옵티컬 플로우를 이용한 화재 검출 ····· 10
그림 2.8 각도와 이동거리를 이용한 배회 검출 ····· 11
그림 2.9 5개 특징을 이용한 낙상 검출 ····· 12
그림 3.1 지능형 영상감시 시스템 흐름도 ····· 13
그림 3.2 Haar-like filter ····· 14
그림 3.3 Cascade 구조 ····· 15
그림 3.4 AlexNet 구조 ····· 16
그림 3.5 객체의 이동거리 측정 ····· 19
그림 3.6 객체의 변화 각도 측정 ····· 19
그림 3.7 객체의 각도 변화량 기준 ····· 20
그림 3.8 Y_{min}, Y_{max}, X_{min}, X_{max} 정의 ····· 21
그림 3.9 낙상 상황에서의 객체 좌표 변화 ····· 22
그림 4.1 제안하는 지능형 영상감시 시스템 구성 ····· 23
그림 4.2 RGB 카메라 객체 검출 결과 ····· 25
그림 4.3 깊이 카메라 객체 검출 결과 ····· 25
그림 4.4 임베디드 모듈 구현 ····· 26
그림 4.5 지능형 영상감시 시스템 최적화 방법 ····· 27
그림 4.6 구간별 처리 시간 ····· 28

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

그림 4.7 python 언어, cython 언어 알고리즘 처리 시간 ·····30
그림 4.8 라즈베리파이 CPU 사용량 ·····31
그림 4.9 라즈베리파이 메모리 사용량 ·····32
그림 4.10 zRam 알고리즘 처리 시간 ·····33
그림 5.1 통합 비정상 상황 검출 알고리즘 처리 시간 비교 ·····36
그림 5.2 전처리 시간 비교 ·····36
그림 5.3 침입자 검출 처리 시간 비교 ·····37
그림 5.4 화재 검출 처리 시간 비교 ·····37
그림 5.5 배회 검출 처리 시간 비교 ·····38
그림 5.6 낙상 검출 처리 시간 비교 ·····38
그림 5.7 비정상적인 상황(침입자)시 알림 ·····39
그림 5.8 비정상적인 상황(화재)시 알림 ·····39
그림 5.9 비정상적인 상황(배회)시 알림 ·····39
그림 5.10 비정상적인 상황(낙상)시 알림 ·····40
그림 5.11 침입자 검출을 위한 학습 영상 ·····41
그림 5.12 비정상적인 상황(화재) 데이터베이스 ·····41
그림 5.13 비정상적인 상황(배회) 데이터베이스 ·····41

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

ABSTRACT

A Study on Implementation of Embedded System for Abnormal Situation Detection

Kim, Jin Su
Advisor : Prof. Pan, Sung Bum
Dept. of Control and Instrumentation Eng.,
Graduate School of Chosun University

The conventional CCTV surveillance system for preventing accidents and incidents is a method for surveillance of multiple monitors by surveillance personnel. However, where one person monitors multiple CCTVs, the person misses 95% of the data after 22 minutes. To address this issue, researchers have studied the computer-based intelligent image surveillance system for notifying people of abnormal situation in the images. However, because the system is involved in the issues of power consumption and costs for building the system, and privacy leaks, the intelligent image surveillance system based on embedded modules has been studied.

This thesis implements the intelligent image surveillance system based on embedded modules for detecting intruders on the basis of learnt information, detecting fires on the basis of colors and motion information, and detecting loitering and fall down objects on the basis of human body motion

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

information. Moreover, the algorithm and the embedded module optimization method are applied to implement real-time processing. The method for optimizing the used algorithm is used for 6 processes of examining subroutines, using positive integers, minimizing remainders and use of division operators, minimizing direct calculation of global variables, minimizing transfer factors and recursive algorithms in function call, and using Cython. The method for optimizing embedded modules is used for 3 processes of removing bottlenecks, using memories, and using zRam.

The intelligent image surveillance system based on embedded modules is implemented in Raspberry Pi. The algorithm processing time is 0.053 seconds on the average by a computer, 0.9499 seconds on the average by Raspberry Pi before optimization, and 0.47 seconds on the average by Raspberry Pi after optimization. The result of comparing before and after optimizing the intelligent image surveillance system shows reduced processing time by 50.52%. Therefore, this suggests real processing of the intelligent image surveillance system based on the embedded modules is possible.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

제1장 서론

최근 살인 사건 및 가정 내 화재, 노인들의 낙상 사고 등 실내·외 환경에서 사건·사고의 수가 증가하고 있어 사람들은 안전에 대한 관심이 증가하고 있다. 이에 따라 사건·사고를 예방하기 위해 경찰의 순찰활동 증가 및 CCTV(Closed Circuit Television) 보급을 증가하고 있으며, 일부 지방자치단체에서는 다양한 목적으로 운영되던 CCTV를 통합해 관리하는 통합관제센터를 설립하여 효율성을 높이고 있다[1][2].

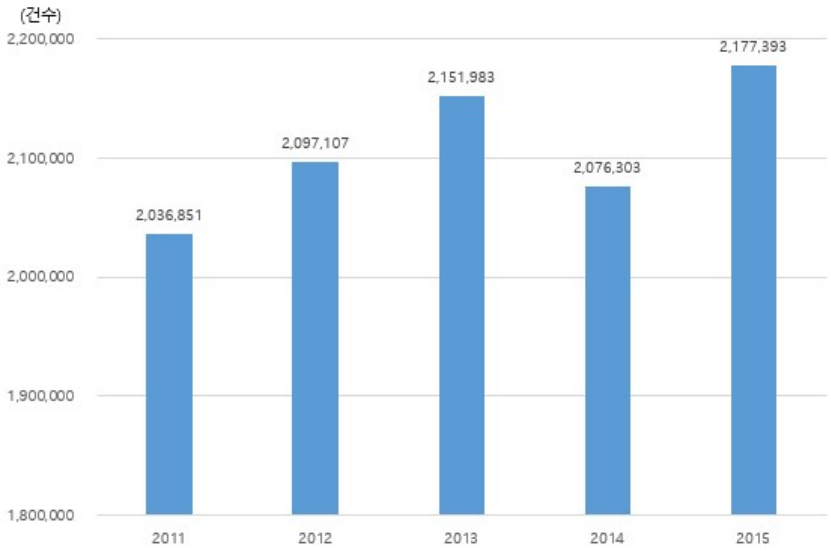


그림 1.1 사건·사고 발생 현황

그러나 대부분 경찰의 순찰활동은 형식적으로 이루어지고 있다는 조사결과가 있다. 차량순찰의 경우 차가 들어가지 못하면 지나가 버리거나 한곳에 머물며 휴식을 취하다 돌아가는 등 여러 문제점이 있으며, 도보 순찰의 경우 주로 의경에 의존하여 효과성이 떨어지며 인력 부족 문제가 있어 경찰의 순찰활동은 범죄 억제력이 부족함을 알 수 있다[3].

CCTV의 경우 서울의 범죄 발생 지역 456개 동 중 113개의 동에 범죄의 빈도가 높았으며, 서울경찰청에서 조사 결과 113개의 동에 상대적으로 CCTV가 적게 설치

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

되어 있었다. 이는 범죄 발생이 CCTV 수와 관련이 있으며, CCTV가 범죄 억제력이 있음을 알 수 있다[4].

기존 CCTV 감시 시스템은 감시 인력이 여러 대의 모니터를 통해 CCTV를 감시하는 방법으로 한사람이 여러 대의 CCTV를 감시할 경우 10분 후 발생하는 사건의 45%를 발견하지 못하며, 22분 후에는 95%를 발견하지 못한다[5]. 표 1-1에서 볼 수 있듯이 평균 1명당 40대의 CCTV를 감시하는 상황으로 사실상 제대로 감시하기 어려운 실정이다[2].

표 1-1 지역별 CCTV 수 및 통제 인원

분류	군포	광명	안성	수원
CCTV (단위: 대)	705	1,527	574	1,131
Control Personnel (단위: 명)	13	33	18	36

이를 보완하기 위한 방법으로 감시 인력이 CCTV를 감시하지 않고 영상 내에 비정상적인 상황이 발생할 경우 알림을 주는 지능형 영상감시 시스템에 관한 연구가 활발히 이루어지고 있다. 지능형 영상감시 시스템의 검출 분야로는 침입자 검출, 화재 검출, 배회 검출, 낙상 검출, 폭행 검출 등이 있으며 이 중 비교적 활발히 연구되고 있는 분야는 침입자 검출, 화재 검출 분야이다. 그러나 대부분의 범죄가 발생하기 전에 이루어지는 행동은 자신이 범죄를 수행할 주변 환경을 살펴보는 배회 활동으로, 배회하는 사람을 예의주시함으로 다양한 범죄를 사전에 예방할 수 있다[6]. 또한, 2015년 통계청 자료에 따르면 국민 100명 중 15명이 낙상을 경험하였으며, 낙상으로 인한 사망률은 10만 명당 4.6명으로 질병이환 및 사망의 외인 항목 중 3번째로 높게 나타나고 있다[7]. 낙상사고는 매년 증가할 것이라는 분석이 나오고 있어 낙상사고 발생 시 상태를 알릴 수 있는 시스템이 갖춰지지 않을 경우 좋지 않은 결과를 가져올 수 있다.

현재 지능형 영상감시 시스템은 컴퓨터를 기반으로 한 연구가 활발히 진행되고 있지만, 소비전력 및 장비구축·유지비용 등의 단점이 있어 실제 환경에서 활용하기에는 어려움이 있다. 이에 대한 대책으로 저비용 및 경량화, 낮은 소비전력으로 구동이 가능한 특징을 지닌 소형 디바이스 기반 지능형 영상감시 시스템에 관한

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

연구가 진행되고 있다.

본 논문에서는 임베디드 모듈 기반 비정상적인 상황 검출 시스템 구현을 제안한다. 학습된 정보를 이용한 침입자 검출, 색상과 움직임 정보를 이용한 화재 검출, 인체 움직임 정보를 이용한 배회·낙상 검출의 비정상적인 상황을 검출하며, 임베디드 모듈을 이용하기 위해 라이브러리 구성을 진행하고 실시간 지능형 영상감시를 위해 제안한 6개의 알고리즘 최적화 및 3개의 임베디드 모듈 최적화를 수행한다. 최적화 후 지능형 영상감시 시스템의 비정상적인 상황 검출 알고리즘은 50.52% 처리 시간이 감소되었다.

본 논문의 구성은 다음과 같다. 2장에서는 지능형 영상감시 시스템 관련 연구로 기존 지능형 영상감시 시스템과 비정상적인 상황 검출 알고리즘에 대해 설명한다. 3장에서는 임베디드 모듈 기반 다중 비정상 상황 검출 알고리즘에 대해 설명하며, 4장에서는 제안하는 지능형 영상감시 시스템 구현에 대해 설명한다. 5장에서는 실험 결과, 6장에서는 결론으로 논문을 마무리한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

제2장 지능형 영상감시 시스템 관련 연구

비정상적인 상황이란 개인적, 통계적, 사회문화적, 전문적 기준에서 일반적인 행동 및 환경에서 벗어난 상황으로 정의되며 종류로는 침입, 방화, 배회, 낙상, 폭력 등이 있다. 기존 비정상적인 상황 검출 알고리즘은 검출하려는 종류에 따라 다양한 검출 방법이 있으며, 기존 지능형 영상감시 시스템은 컴퓨터를 이용한 방법과 소형 디바이스를 이용한 방법이 있다.

제1절 지능형 영상감시 시스템

영상 내에 비정상적인 상황이 발생한 경우 알림을 주는 지능형 영상감시 시스템은 컴퓨터를 이용한 방법과 소형 디바이스를 이용한 방법으로 나눌 수 있으며 컴퓨터 기반 지능형 영상감시 시스템에 비해 소형 디바이스 기반 지능형 영상감시 시스템의 연구가 비교적 활발히 일어나지 않고 있다. 그러나 컴퓨터 기반 지능형 영상감시 시스템은 구축·유지비용, 소비전력, 설치제약, 사생활 유출의 문제점이 있어 실제 환경에서 활용하기에 적합하지 않다. 실제 환경에서 효율적으로 활용하기 위해 소형 디바이스 기반 지능형 영상감시 시스템의 연구가 진행되고 있지만, 아직 초기 단계에 머물러 있다.

기존 컴퓨터를 이용한 지능형 영상감시 시스템은 첫 번째로 다수의 카메라를 이용한 객체 추적 시스템이 있다. DSP(Digital Signal Processor) 기반 IP 카메라에서 입력되는 영상을 AVS(Audio Video Coding Standard)를 이용해 인코딩을 수행하고, RTSP(Real Time Streaming Protocol)를 이용해 IP 네트워크로 전송한다. IP 네트워크에서는 처리 시간 단축을 위해 분산처리를 수행하고 원격 데이터 접근을 지원하며, IP 네트워크에 연결된 각 서버는 실시간 처리를 위해 GPU를 활용하여 네트워크에서 입력된 영상에 대한 디코딩 및 영상처리 알고리즘을 수행한다. 수행하는 알고리즘은 입력되는 영상에서 GMM(Gaussian Mixture Model)을 이용한 객체 검출 및 추적하는 알고리즘으로 31ms의 처리 시간을 나타낸다[8].

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

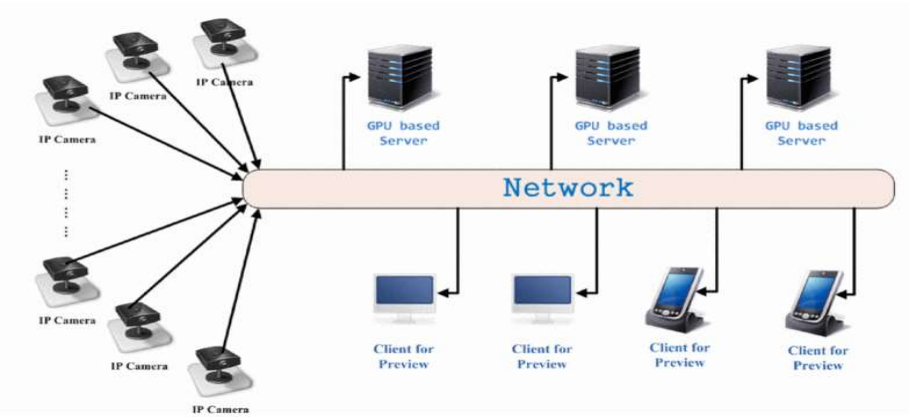


그림 2.1 분산처리를 이용한 지능형 영상감시 시스템

두 번째로 인체 외형 기반 낙상 검출 시스템이 있다. 실시간으로 카메라에서 입력되는 영상을 이용해 컴퓨터에서 낙상 검출을 수행하고 네트워크를 통해 병원 서버에 전송하여 저장 및 모니터링하는 시스템으로, 낙상 검출 방법은 입력되는 영상에서 GMM을 이용해 객체를 검출하고 동적 특징을 사용하는 방법보다 계산량이 적은 정적 특징인 인체 형상의 종횡비 및 경사각 특징을 KNN(K-Nearest Neighbor)을 이용하여 분류한다. 낙상 검출 시 메시지를 통해 낙상이 발생함을 알리며, 86.11%의 성능을 나타낸다[9].

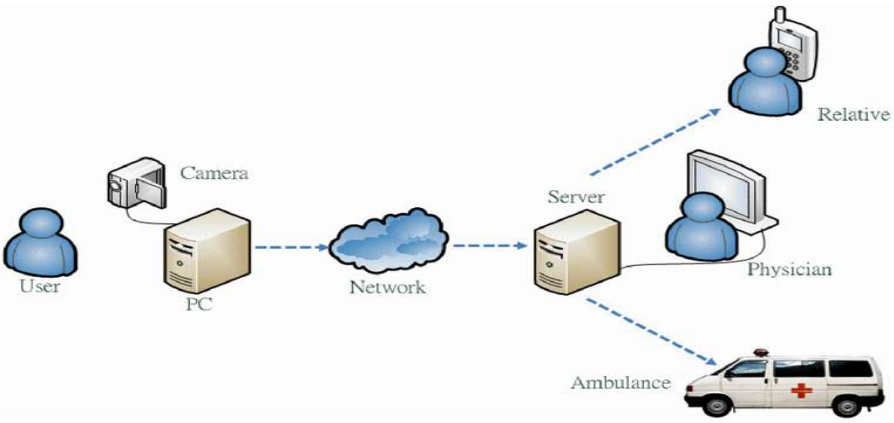


그림 2.2 인체 외형 기반 낙상 검출 시스템

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

표 2-1은 컴퓨터 기반 지능형 영상감시 시스템의 상용화된 제품 정보를 나타낸다[10][11][12][13]. 컴퓨터 기반 지능형 영상감시 시스템은 각각의 카메라에서 입력된 영상을 중앙서버로 전송 후 중앙서버에서 침입자 검출, 화재 검출, 배회 검출, 낙상 검출 등 다양한 검출을 복합적으로 수행하는 특징이 있다.

표 2-1 컴퓨터 기반 지능형 영상감시 시스템

제품명	회사	검출종류
지능형 영상보안 시스템	에스원	얼굴 검출, 보호구 착용여부 검출, 낙상 검출, 도주 검출, 배회 검출, 화재 검출
지능형 영상감시 시스템	DSN	침입자 검출, 화재 검출, 연기 검출, 낙상 검출, 도난 검출
IQ-Series	CRIFAS	침입자 검출, 배회 검출, 낙상 검출, 화재 검출
IVS	NUUO	배회 검출, 사람 및 차량 카운트, 침입자 검출

기존 소형 디바이스를 이용한 지능형 영상감시 시스템은 첫 번째로 2개의 버전으로 구성된 실시간 침입자 감시 시스템이 있다. 첫 번째 버전은 라즈베리파이와 아두이노로 구성된 단일 보드 컴퓨터로 구성되어 있으며, 얼굴인식을 이용한 침입자 검출 시 검출시간의 지연이 있다. 두 번째 버전은 컴퓨터로 구성되어 있으며 얼굴인식을 이용한 침입자 검출 시 지연 없이 실시간으로 동작이 가능하다. 그림 2.3은 첫 번째 버전의 침입자 검출 시스템으로 MOG2 방법을 이용해 움직임을 검출하고 크기를 이용해 인체 여부를 판별, Haar-like feature를 이용한 얼굴 검출, fisherfaces 기반 얼굴인식을 통해 침입자를 검출한다. 침입자가 발생하면 메일을 통해 사용자에게 알림을 주며, 웹 인터페이스를 이용해 원격으로 영상을 확인할 수 있다. 얼굴인식 학습데이터는 92*92 영상을 이용하며, 자연 및 인공조명을 이용한 학습데이터는 최고 90%, 인공조명만 이용한 경우 최고 60%의 성능을 나타낸다[14].

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

그림 2.3 라즈베리파이와 아두이노를 이용한 침입자 검출 시스템

두 번째로 라즈베리파이를 이용한 침입자 감시 시스템이 있다. 사물인터넷 기반의 영상 처리 장치로 클라우드와 같은 원격 서버에 데이터 전송이 가능하여 입력되는 영상 내에 움직임이 감지 될 경우 영상을 클라우드 서버에 저장하여 추후 확인이 가능한 시스템이다. 차영상을 이용해 객체를 검출하고 Haar-like feature를 이용해 인체 여부를 판별하며, 인체로 판별되면 GSM 모듈을 이용해 사용자에게 메시지를 전송함으로 비정상적인 상황이 발생함을 알린다. 그림 2.4는 라즈베리파이를 이용한 지능형 영상감시 시스템의 블록도이며, 4대의 카메라를 활용하여 640*480 영상을 초당 24프레임 처리하고 80%의 성능을 나타낸다[15].

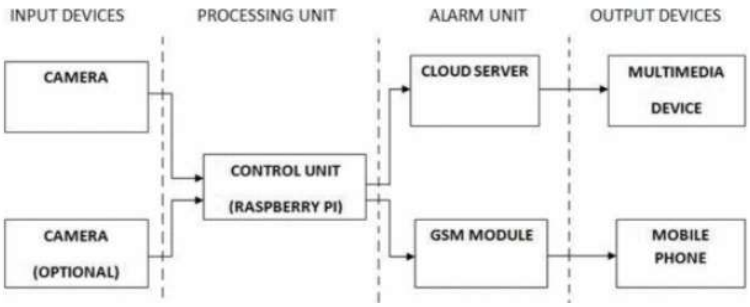


그림 2.4 라즈베리파이 기반 지능형 영상감시 시스템 블록도

마지막으로 QuaRC 기반 단일 Gumsitx를 이용한 UAV(Unmanned Aerial Vehicle) 화재 검출 시스템이 있다. 화재의 색상정보와 움직임 정보를 이용해 화재를 판단하며, 색상 정보를 이용하기 위해 Lab 색상을 활용하고 움직임 정보를 이용하기

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

위해 옵티컬 플로우를 활용한다. 비선형 시스템과 컨트롤러를 사용하면 계산 시간이 늘어날 수 있으며, 채터링 효과를 유발할 수 있어 SMC(Sliding Mode Control)과 LQR(Linear Quadratic Regulator)를 사용한다[16].

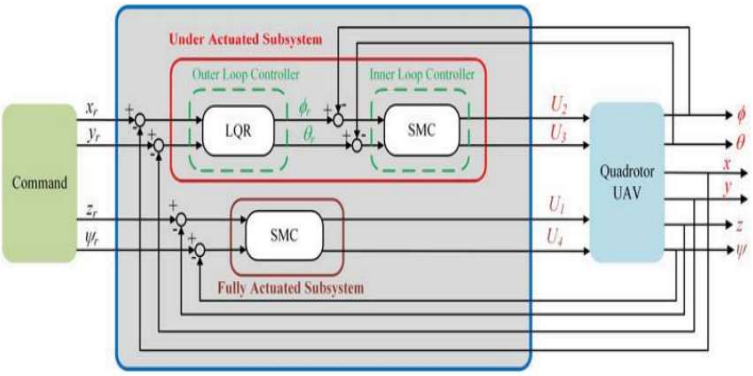


그림 2.5 UAV 제어 시스템

표 2-2는 소형 디바이스 기반 지능형 영상감시 시스템의 상용화된 제품 정보를 나타낸다[17][18][19]. 소형 디바이스 기반 지능형 영상감시 시스템은 카메라에서 입력된 영상을 이용하여 침입자 검출, 화재 검출, 움직임 검출 등의 검출을 단일적으로 수행한다. 컴퓨터 기반 지능형 영상감시 시스템의 사생활 유출, 소비전력, 구축·유지비용의 단점을 보완하고 실제 환경에서 효율적으로 활용하기 위해서는 다양한 검출들을 복합적으로 수행하는 소형 디바이스 기반 지능형 영상감시 시스템의 연구가 필요하다.

표 2-2 소형 디바이스 기반 지능형 영상감시 시스템

제품명	회사	검출종류
Home IP Camera	easyN	사람 검출, 움직임 검출, 원격감시
IoT 홈캠2	KT	움직임 검출, 원격감시
홈 CCTV 미니	LG	적외선 모니터링, 침입자 검출, 원격감시

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

제2절 비정상적인 상황 검출 알고리즘

영상을 이용하여 침입자를 검출하기 위해서는 영역 내에 들어온 사람이 누구인지 판별이 필요하다. 사람을 판별하는 방법으로는 옷 색상, 객체 크기·비율, 얼굴인식 등이 있지만 옷 색상은 고정되어 있는 색상이 아니라 시간이 지남에 따라 달라지기 때문에 사용자로 등록하기에 적합하지 않다. 객체 크기·비율의 경우 영상입력장치와 객체와의 거리에 따라 객체의 크기가 다르게 입력되며 옷, 모자 등에 의해 객체비율이 다르게 입력되는 오검출되어 적합하지 않다.

초기 얼굴인식 방법은 PCA(Principle Component Analysis)를 이용한 방법으로, PCA를 얼굴 영상에서 수행하여 얻어진 eigenface를 이용하여 얼굴을 인식한다[20]. 그 후 PCA 알고리즘의 성능을 향상하기 위해 얼굴 영상을 DCT(Discrete Cosine Transform) 계수를 이용하여 재구성된 이미지와 결합 후 학습영상으로 사용한 방법이 있으며, PCA의 단점을 보완하기 위해 PCA 기반의 DiaPCA(Diagnonal PCA), 클러스터를 기반으로 한 특징 투영 방법을 사용한 얼굴인식 방법이 있다[21][22][23]. 추가적으로 PCA와 LDA(Linear Discriminate Analysis)를 결합하여 인식률을 상향한 방법으로 PCA와 LDA를 기반으로 특징을 추출하고 NNC(Nearest Neighbor Classifier)를 이용하여 분류한 방법이 있으며, 1~5m 거리별 얼굴 이미지를 사용하여 PCA와 LDA를 기반으로 특징을 추출하여 얼굴을 인식한 방법이 있다[24][25]. LBP(Local Binary Pattern)를 이용하여 얼굴 영상에서 특징을 추출하고 CNN(Convolution Neural Networks)을 이용하여 얼굴을 인식한 방법이 있으며, 딥러닝의 RBM(Restricted Boltzmann Machines)을 이용한 얼굴인식 방법으로 원영상, 표정변화, 조명변화 영상을 학습이미지로 사용하고 -25도, -15도, 15도, 25도 각도 변화의 영상을 검증이미지로 사용한 방법으로 각도에 따른 인식률은 -15도 경우 99.9%로 가장 높은 인식률을 나타낸다[26][27].

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

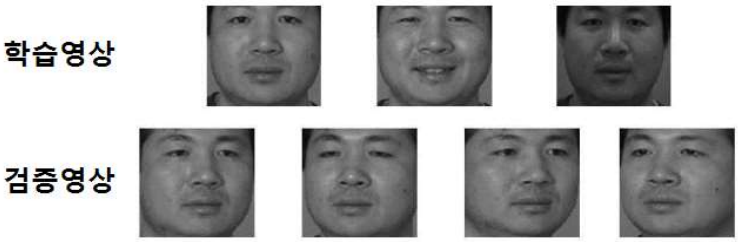


그림 2.6 표정, 조명, 각도 변화 영상을 이용한 얼굴인식

기존 화재를 검출하는 방법은 센서를 이용한 방법과 영상을 이용한 방법으로 나뉜다. 센서를 이용한 방법으로 온도 센서, 연기 센서, CO2 센서를 이용하여 취득한값을 퍼지 규칙의 매개변수로 활용하여 화재를 검출한 방법이 있으며, AMS MOX 센서, PID alphasense 센서, NDIR CO2 alphasense 센서 등 8개의 센서를 결합하여 만든 장치를 이용하여 화재를 검출한 방법이 있다[28][29].

색상을 이용하여 화재를 검출한 방법은 RGB 색 공간을 이용한 방법으로 RGB 색 공간에서 화재의 색상을 검출한 후 화재 확산 등의 추가 특징 정보를 이용하여 화재를 판단한 방법이 있다[30]. 또한, 목재를 이용한 화재, 알코올을 이용한 화재 및 전기합선 등 다양한 환경에서의 RGB 색상의 표준편차를 계산하여 화재를 검출한 방법이 있으며, HSI 색 공간 및 옵티컬 플로우의 특징을 이용하여 화재를 검출한 방법으로 4개의 미국 국립표준연구소 데이터베이스와 2개의 취득한 데이터베이스를 이용하여 실험한 결과 평균 91.6%의 인식률을 나타낸다[31][32]. 마지막으로 다기능 융합을 기반으로 화재 연기를 인식하여 화재를 검출한 방법으로 HSI 색 공간, 2차원 DWT(Discrete Wavelet Transform), 픽셀 비율, 옵티컬 플로우의 특징을 이용하여 SVM(Suppor Vector Machine)으로 화재를 검출한다[33].

(a) 화재 영상 (b) 화재 검출영상

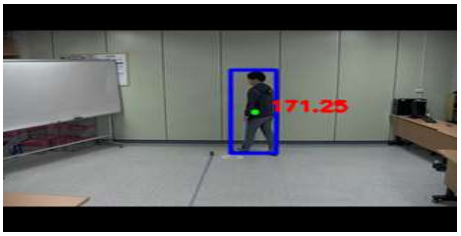
그림 2.7 색상과 옵티컬 플로우를 이용한 화재 검출

Collection @ chosun

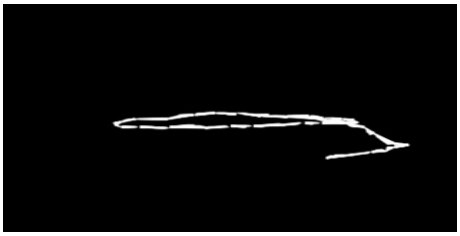
조선대학교
CHOSUN UNIVERSITY

기존 배회를 판단하는 방법은 영상에서 객체의 시간을 측정하는 방법으로 PETS 데이터베이스를 사용하여 실험한 결과 평균 90.77%의 인식률을 나타낸다[2][34][35]. 그러나 단순히 객체의 시간을 측정하는 방법은 약속장소에서 사람을 기다리거나 버스정류장에서 버스를 기다리는 등 사람이 일정 장소 안에 오랫동안 있는 행동에 오검출이 발생하게 된다. 이를 보완하기 위해 n by m 크기로 영상을 분할하고 객체의 블록시간을 측정하여 배회를 판단한 방법이 있지만 영상 입력장치와 객체와의 거리에 따른 오검출이 발생한다[36][37].

위의 단점을 보완한 방법으로 배회 행동을 하는 사람은 일반사람에 비해 방향 변화가 많이 일어나는 특징을 이용하여 객체의 각도를 측정하여 배회를 판단한 방법이 있다[38]. 또한, 배회 판단 조건을 2개 사용하여 검출한 방법으로 객체 시간 측정과 객체 블록시간을 사용하여 배회를 검출한 방법[39], 객체 시간 측정과 각도를 사용하여 배회를 검출한 방법[40], 각도와 객체의 이동거리를 사용하여 배회를 검출한 방법이 있다[41].



(a) 객체 각도 변화량



(b) 객체 밸런스 포인트 이동궤적

그림 2.8 각도와 이동거리를 이용한 배회 검출

기존 낙상을 검출하는 방법은 센서를 이용한 방법과 영상을 이용한 방법으로 나뉜다. 센서를 이용한 방법은 3축 가속도 센서를 사람의 몸에 부착하여 센서값과 가속도값이 임계값을 넘었을 시 낙상으로 판별한 방법이 있다[42]. 또한, 핸드폰의 가속도 센서를 이용한 방법으로 평상시의 가속 패턴을 ADL(Activities of Daily Living)으로 저장하고 실시간 NNR(Nearest Neighbor Rule)을 기반으로 ADL과 비교하여 낙상으로 판단한 방법이 있으며[43], 가속도 센서와 압력 센서를 이용하여 낙상을 판단한 방법이 있다[44].

영상을 이용한 방법은 객체의 움직임에 따라 변화하는 원을 설정하고 원의 변

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

화와 영상의 수직, 수평 히스토그램 특징값을 사용하여 SVM으로 낙상을 판단한 방법이 있다[45]. 또한 검출된 객체에 바운딩 박스를 설정하고 바운딩 박스의 변화하는 가속도를 이용하여 낙상을 판단한 방법이 있으며[46], 객체의 종횡비, 유효면적 비, 검출된 객체의 특징점, 신체 축 각도, 신체 윤곽의 원심 비율의 총 5개의 고유벡터값을 특징으로 사용하여 SVM으로 낙상을 판단한 방법으로 낙상의 인식률은 94.5%를 나타낸다[47].

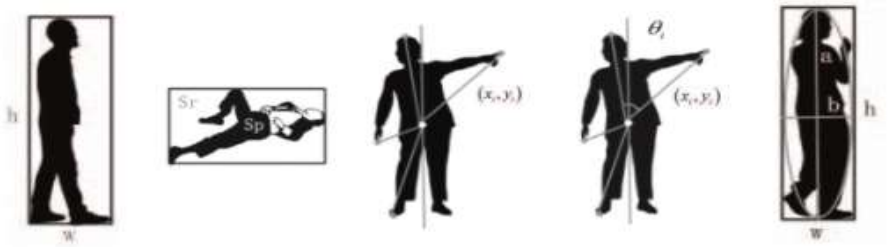


그림 2.9 5개 특징을 이용한 낙상 검출

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

제3장 다중 비정상 상황 검출 알고리즘

그림 3.1은 지능형 영상감시 시스템의 흐름도를 나타낸다. 임베디드 모듈을 이용하여 입력되는 영상에서 적응형 차영상을 이용하여 움직이는 객체를 검출한 후 정확성을 위해 침식연산을 수행한다. 일정시간 동안 객체가 검출되지 않으면 배경영상을 갱신하며, 객체가 검출되면 객체의 크기 정보를 이용하여 인체 여부를 판단한다. 인체가 아닐 경우 색상 정보와 움직임 정보를 이용하여 화재를 검출하며, 인체로 판단될 경우 학습된 정보를 이용한 침입자 검출, 밸런스 포인트의 각도 변화와 이동거리를 이용한 배회 검출, 인체 움직임 정보와 가속도 정보를 이용한 낙상 검출을 수행한다. 비정상적인 상황이 검출되면 사전에 정의된 사용자에게 ssmtp 및 mock 라이브러리를 활용해 영상을 전송함으로 비정상적인 상황이 발생함을 알린다.

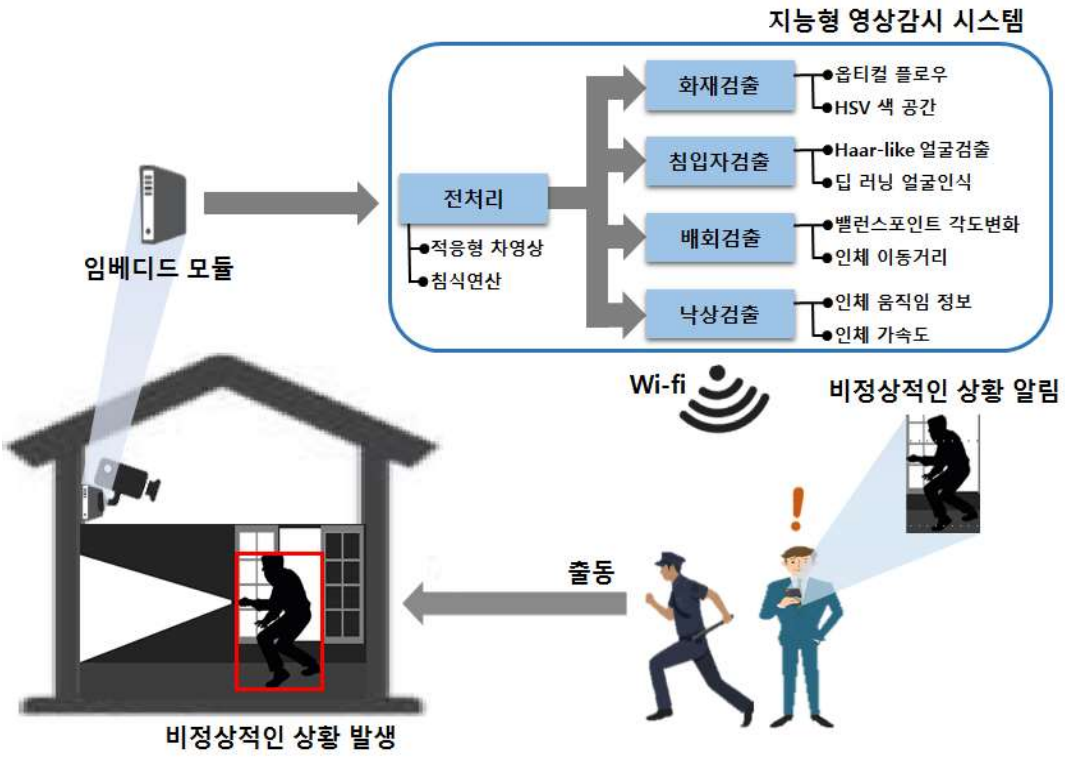


그림 3.1 지능형 영상감시 시스템 흐름도

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

침입자 검출, 화재 검출, 배회 검출, 낙상 검출 알고리즘의 실시간 처리를 위해 제안한 서르부틴 확인, 양의 정수형 사용, 나머지·나누기 연산자 최소화, 전역변수 직접계산 최소화, cython 언어 사용, 함수호출 시 전달인자·재귀함수 최소화, 총 6개의 알고리즘 최적화 방법을 수행하며, 제안한 병목 현상 제거, 메모리 활용, zRam 사용, 총 3개의 임베디드 모듈 최적화 방법을 수행한다. 본 장에서 임베디드 모듈 기반 침입자 검출, 화재 검출, 배회 검출, 낙상 검출 알고리즘에 대해 설명하며, 4장에서 제안한 알고리즘 최적화 및 임베디드 모듈 최적화 방법, 지능형 영상감시 시스템 구현에 대해 설명한다.

제1절 학습된 정보 기반 침입자 검출

본 논문에서는 그림 3.1과 같이 침입자를 검출하기 위해 Haar-like feature 기반의 얼굴 검출을 수행하며, 검출된 얼굴을 딥 러닝(Deep Learning)을 이용하여 학습한 정보를 기반으로 얼굴인식을 수행한다. Haar-like feature는 Haar-like filter에 의해 계산되며 검정영역의 밝기와 흰 영역의 밝기 차이를 이용한다. Haar-like filter는 그림 3.2와 같은 형태로 구성되며, Haar-like feature는 단일 Haar-like filter를 사용하는 것뿐만 아니라 다수의 Haar-like filter를 결합해 Adaboost 알고리즘으로 선형 조합하여 사용한다[48].

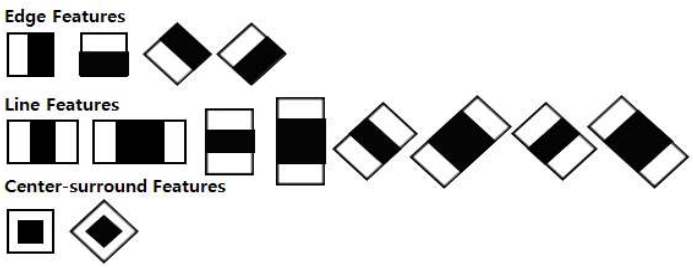


그림 3.2 Haar-like filter

Adaboost의 기본 개념은 약한 분류기를 선형적으로 결합하여 최종적으로 높은 검출 성능을 가진 강한 분류기를 생성하는 것으로, 연산속도 향상을 위해 그림 3.3과 같이 True, False 분류를 Cascade 구조로 만들어서 적용한다. Cascade 구조

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

는 하위단계에서 약한 분류기를 적용하고 상위단계로 올라갈수록 강한 분류기를 적용하는 방법으로, 검출하려는 영상과 유사도가 없는 영상에 대해 초기 분류기에서 False로 분류하여 계산량이 적은 장점이 있다[49].

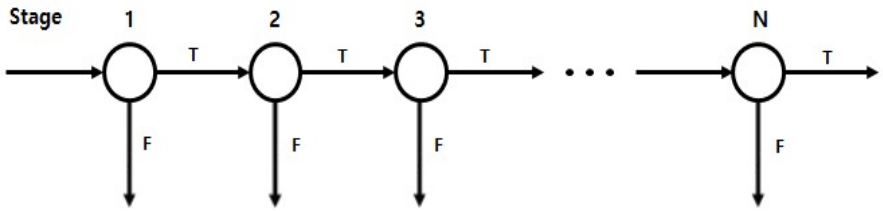


그림 3.3 Cascade 구조

딥 러닝은 다양한 비선형 변환기법의 조합을 통해 영상인식을 수행하는 기계학습의 일종으로, 텐서플로우, 카페, 토치 등이 있다. 그중 텐서플로우는 구글에서 만든 오픈소스 라이브러리로 현재 범용적으로 사용되고 있으며 텐서플로우의 가장 큰 장점인 데이터 플로우는 그래프를 통해 파라미터의 변화 양상이나 구조도를 보여줌으로써 연결관계와 흐름도를 쉽게 파악이 가능하다. 또한, 코드 수정 없이 CPU/GPU 동작이 가능하며 계산 구조와 목표 함수만 정의하면 자동으로 미분 계산을 처리하는 특징이 있다.

본 논문에서는 얼굴인식을 수행하기 위해 텐서플로우의 AlexNet을 사용한다. AlexNet은 GPU를 기반으로 한 구조로 5개의 컨볼루션 레이어와 3개의 맥스 풀링, 3개의 풀리 커넥티트 레이어로 구성되어 있다. 컨볼루션 레이어는 특징을 추출하는 역할을 수행하며 특징을 추출하는 기능인 필터와 필터값을 비선형값으로 바꾸어 주는 활성함수로 이루어진다. 맥스 풀링은 컨볼루션 레이어에서 필터를 적용하여 얻은 데이터의 크기를 줄이는 부분으로, n by m의 크기로 자른 후 그 영역에서 가장 큰값을 사용한다. 풀리 커넥티드 레이어는 추출된 특징값을 이용하여 데이터를 분류하는 부분이다. 본 논문에서는 ReLU(Rectified Linear Unit)함수와 Softmax함수를 사용한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

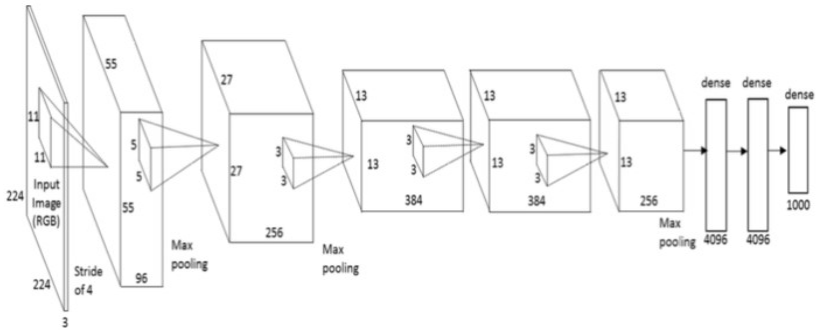


그림 3.4 AlexNet 구조

제2절 색상과 움직임 정보 기반 화재 검출

화재를 검출하는 방법은 크게 2가지로 센서를 이용하여 검출하는 방법과 영상을 이용하여 검출하는 방법이 있다. 센서를 이용하는 방법은 화재 센서, 온도 센서 등을 이용하여 센서에 들어온 정보를 기반으로 화재를 검출하지만, 화재 검출 범위가 영상을 이용하는 방법에 비해 좁을 뿐만 아니라 화재의 유무를 판별할 뿐 화재의 크기 및 확산속도 등 추가적인 정보를 얻을 수 없다.

본 논문에서는 그림 3.1의 흐름도와 같이 HSV 색 공간을 이용한 화재의 색상 정보와 옵티컬 플로우를 이용한 화재의 움직임 정보를 사용하여 화재를 검출한다[32]. 기본 색상 정보인 RGB 색 공간은 0~255 사이값을 가지고 있지만, 물체는 동일한 색상을 띄고 있어도 표면의 불균일한 빛으로 인해 표면 전체에서 다른 밝기 또는 채도를 가질 수 있어, 물체를 검출하기 위해서는 색상, 채도, 명도로 구성되어 있는 HSV 색 공간으로 변환하여 검출해야 한다[50].

HSV 색 공간에서 H는 색상으로 0~360도 범위를 가지며 색 구성 요소를 나타낸다. S는 채도로 색상의 진함 정도를 나타내며, V는 명도로 밝은 정도를 나타낸다. HSV 색 공간 계산을 위한 공식은 식(3.1)~식(3.4)과 같다[51]. R', G', B'는 R, G, B값을 0~1 범위로 정규화를 시킨 값이며, R', G', B'를 이용하여 식(3.1)과 같이 C_{max}, C_{min}, △을 계산하고 식(3.2)~(3.4)를 이용하여 H, S, V를 계산한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

$$\begin{aligned} C_{\max} &= \max(R', G', B') \\ C_{\min} &= \min(R', G', B') \\ \Delta &= C_{\max} - C_{\min} \end{aligned} \tag{3.1}$$

$$H = \begin{cases} 0^o & \Delta = 0 \\ 60 \times (\dfrac{G' - B'}{\Delta}) & C_{\max} = R' \\ 60 \times (\dfrac{B' - R'}{\Delta} + 2) & C_{\max} = G' \\ 60 \times (\dfrac{G' - B'}{\Delta} + 4) & C_{\max} = B' \end{cases} \tag{3.2}$$

$$S = \begin{cases} 0 & C_{\max} = 0 \\ \dfrac{\Delta}{C_{\max}} & C_{\max} \neq 0 \end{cases} \tag{3.3}$$

$$V = C_{\max} \tag{3.4}$$

옵티컬 플로우는 영상 내의 움직이는 객체의 정보를 이용하여 객체를 추정하는 방법으로, 영상에 대한 명암 패턴의 이동을 속도 벡터인 u, v로 정의하여 영상 내의 객체 정보에 부가하고 연속해서 영상 내의 동일한 속도를 가지는 영역을 하나의 객체로 판단하여 검출한다[52]. 화재의 움직임을 검출하기 위해 Shi & Tomas 특징 추출방법을 사용해 영상 내의 특징점을 추출하며 Lucas-Kanade 옵티컬 플로우를 이용해 반복적으로 점들을 추적한다. 명암 패턴의 이동 속도인 u, v를 이용하여 화재를 판단하기 위한 식(3.5), 식(3.6)을 계산한다.

$$\begin{aligned} ofp_u &= u_{n-1} \\ ofp_v &= v_{n-1} \\ ofn_u &= u_n \\ ofn_v &= v_n \end{aligned} \tag{3.5}$$

$$of_u = ofp_u - 3(\sqrt{(ofp_v - ofn_v)^2 + (ofp_u - ofn_u)^2})\cos(\arctan\frac{ofp_v - ofn_v}{ofp_u - ofn_u}) \tag{3.6}$$

$$of_v = ofp_v - 3(\sqrt{(ofp_v - ofn_v)^2 + (ofp_u - ofn_u)^2})\sin(\arctan\frac{ofp_v - ofn_v}{ofp_u - ofn_u})$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

HSV 색 공간을 이용한 화재의 색상 정보와 옵티컬 플로우를 이용한 화재의 움직임 정보를 이용하여 화재를 판단하는 공식은 식(3.7)과 같다.

$$\begin{gathered} H > TH_H,\ S > TH_S,\ V > TH_V \\ \text{and} \\ ofp_v - of_v > TH_{ofv} \\ \text{and} \\ ofp_u - of_u < TH_{ofu} \end{gathered} \tag{3.7}$$

제3절 인체 움직임 정보 기반 배회·낙상 검출

배회란 '정해진 목적지 없이 특정 공간을 이리저리 돌아다닌다.' 로 정의할 수 있으며[4], 대부분의 범죄사건 전에 이루어지는 행동은 자신이 범죄를 수행할 주변 환경을 살펴보는 배회 활동으로 배회 활동하는 사람을 검출하여 예의주시함으로 범죄를 사전에 예방할 수 있다[6]. 일반적으로 정상적인 보행자는 동선이 일정해 특정 공간에서의 움직임과 각도 변화량이 적은 반면, 비정상적인 보행자는 동선이 일정하지 않아 특정 공간에서의 움직임과 각도 변화량이 많다[5]. 본 논문에서는 위의 비정상적인 보행자 특징을 이용하여 배회를 검출하기 위해 아래 조건 2개를 이용하여 판단한다[53].

[조건1] 객체의 밸런스 포인트 이동거리
[조건2] 객체의 밸런스 포인트 각도 변화량

그림 3.5는 객체의 이동거리 측정 방법을 나타낸다. 객체가 t-1에서 t로 이동한 거리는 L(t-1), t에서 t+1로 이동한 거리를 L(t)라 할 경우 객체의 대표 무게중심점의 좌표 변화량을 이용하여 L(t-1), L(t)을 계산한다. 객체의 대표 무게중심점은 일정 구간마다 현재 무게중심점으로 객체의 이동궤적을 정규화시켜 잡음에 의한 오검출을 줄이기 위해 사용한다. 객체가 t에서 t+1로 이동한 거리 L(t)를 구하는 공식은 식(3.8)과 같다. X는 객체의 길이 변화량, Y는 객체의 높이 변화량, H는 객체의 높이로 입력영상과 검출된 객체의 거리에 따른 이동거리 측정이 다르게 계산되는 오계산을 보완한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

$$L(t) = \frac{\sqrt{X^2 + Y^2}}{H} \tag{3.8}$$

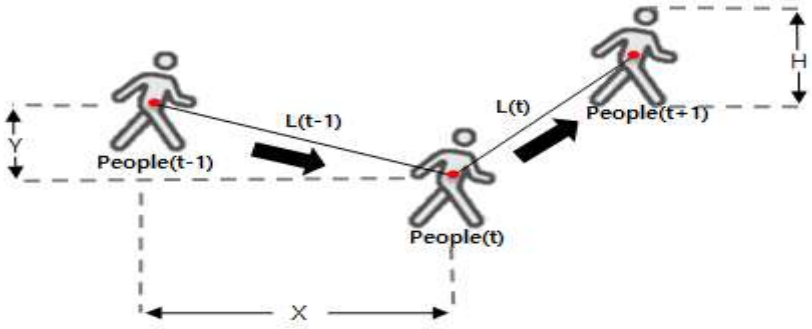


그림 3.5 객체의 이동거리 측정

그림 3.6은 객체의 변화 각도를 측정하는 방법으로 객체가 t-2에서 t-1로 이동할 때 생기는 각도 θ(t-1), t-1에서 t로 이동할 때 생기는 각도는 θ(t)이다. 객체의 대표 무게중심점의 X의 변화량과 Y의 변화량을 이용하여 각도를 계산하며, 구하는 공식은 식(3.9)와 같다. 각도의 변화량은 계산한 θ(t-1), θ(t)의 차를 이용해 구한다.

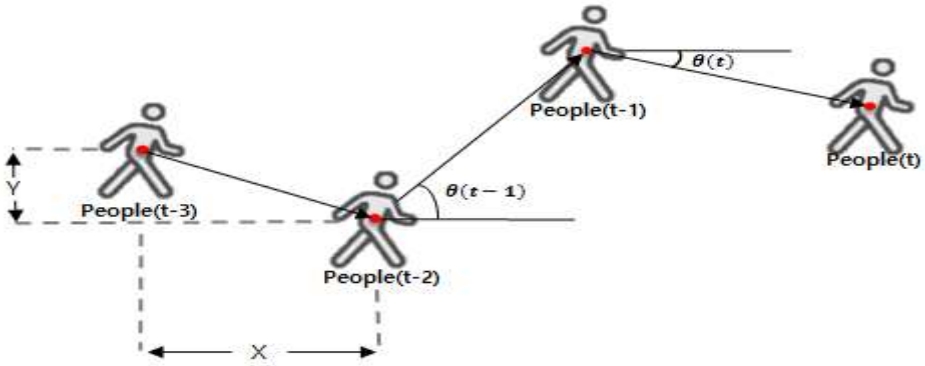


그림 3.6 객체의 변화 각도 측정

$$\theta(t) = \arctan\left(\frac{Y}{X}\right) \tag{3.9}$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

객체의 각도 변화량 기준은 그림 3.7과 같다. 객체의 현재 진행 방향은 화살표 방향인 0° 로, 현재 진행 방향에서 좌·우로 변화하는 각도를 5단계로 그룹화하여 측정한다. 각도의 변화량에 따른 가중치는 표 3.1과 같으며, 각도의 변화량이 적은 경우 낮은 가중치를 부여하고 각도의 변화량이 큰 경우 높은 가중치를 부여함으로 객체의 방향변화에 민감도를 부여하였다.

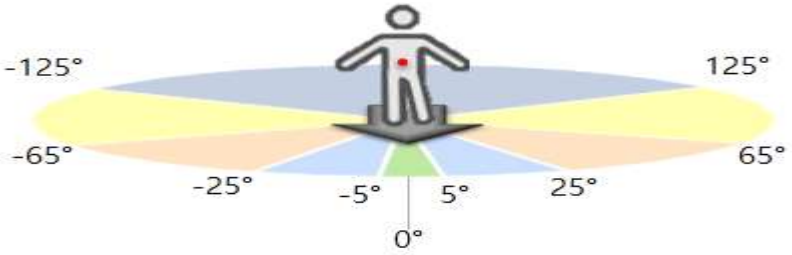


그림 3.7 객체의 각도 변화량 기준

표 3-1 각도 변화량에 따른 가중치

각도 변화량	가중치
- 5 ~ 0 또는 0 ~ 5	0.01
-25 ~ -5 또는 5 ~ 25	0.02
-65 ~ 25 또는 25 ~ 65	0.04
125 ~ 65 또는 65 ~ 125	0.07
-125 미만 또는 125 초과	0.10

객체의 각도 변화량에 따른 가중치 부여 방법과 객체의 높이에 반비례하는 이동거리 측정 방법을 사용하여 배회를 판단하는 공식은 식(3.10)과 같다.

$$Angle_{weight} > TH_a \quad and \quad Move_{len} > TH_l \tag{3.10}$$

낙상을 검출하기 위해 그림 3.1의 흐름도와 같이 객체의 움직임 정보와 가속도 정보를 이용한다[54]. 움직임 정보와 가속도 정보를 계산하기 위한 기준점으로 Y_{min}, Y_{max}, X_{min}, X_{max}를 사용하며 Y_{min}은 Y축 최소좌표, Y_{max}는 Y축 최대좌표, X_{min}은 X축 최소좌표, X_{max}는 X축 최대좌표로 그림 3.8과 같다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

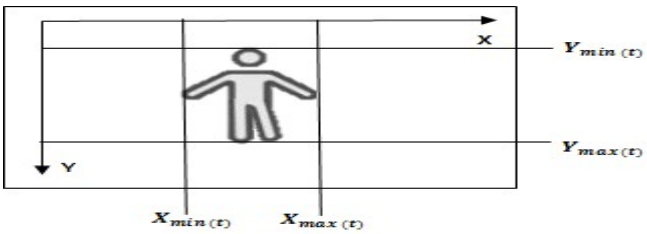


그림 3.8 Y_{min}, Y_{max}, X_{min}, X_{max} 정의

그림 3.9는 정상 상태에서 낙상 상태로 변화 시 각 좌표값의 변화를 나타낸다. 그림 3.9 (a)는 전방낙상의 경우로 Y_{min}과 Y_{max}값이 증가하며, X_{min}과 X_{max}값의 변화가 거의 없다. 그림 3.9 (b)는 후방낙상의 경우로 Y_{min}값이 증가하며, Y_{max}와 X_{min}, X_{max}값의 변화가 거의 없다. 그림 3.9 (c)는 좌측낙상의 경우로 Y_{min}값이 증가하며 X_{min}, X_{max}값이 감소, Y_{max}값의 변화가 거의 없다. 그림 3.9 (d)는 우측낙상의 경우로 Y_{min}, X_{min}, X_{max}값이 모두 증가, Y_{max}값의 변화가 거의 없다. 낙상상태의 경우 Y_{min}값이 증가하고, 좌·우측 낙상의 경우 X_{min}, X_{max}의 증·감이 뚜렷하게 나타나는 특징을 이용해 식(3.11)을 기반으로 움직임 정보를 이용한 낙상을 검출한다.

$$F_a = \frac{X_{\max} + Y_{\min}}{2}, \quad F_b = \frac{X_{\min} + Y_{\max}}{2} \tag{3.11}$$

추가적으로 객체의 가속도 정보를 이용하기 위해 식(3.12)를 기반으로 객체의 가속도 정보를 구한다. 식(3.12)에서 k는 프레임 수, V_k는 k번째 가속도값, t_k는 k번째 프레임의 시간이다.

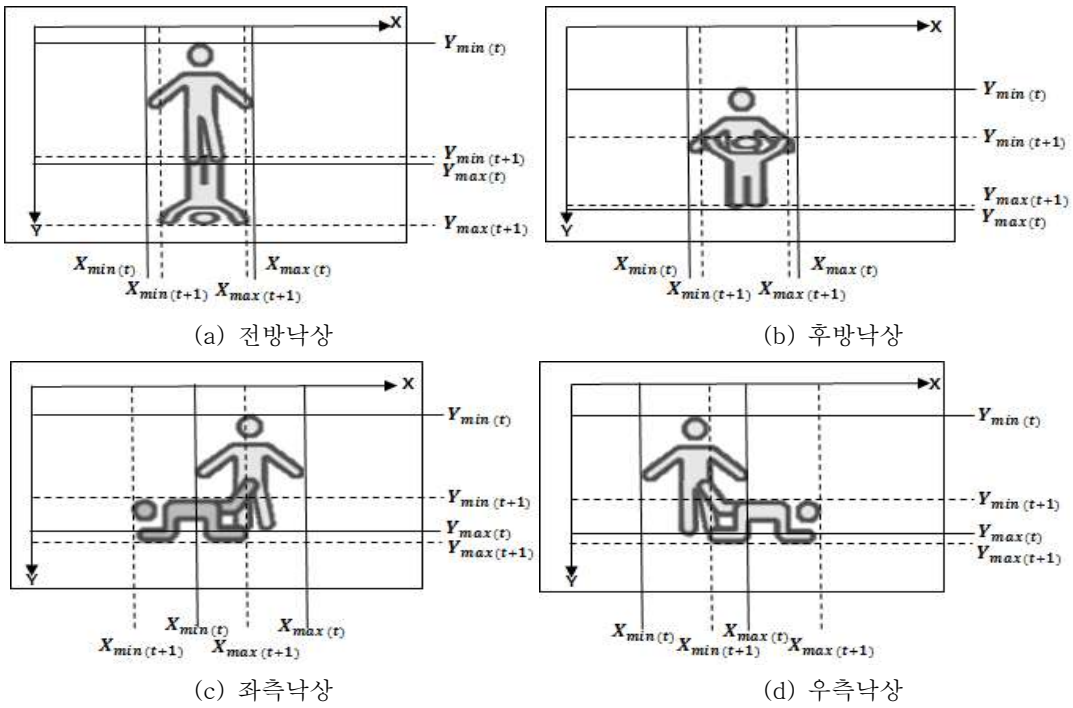
$$V_k = \frac{Y_{\min(k)} - Y_{\min(k-1)}}{t_{(k)} - t_{(k-1)}} \tag{3.12}$$

객체의 움직임 정보 이용 방법과 가속도 정보를 사용하여 낙상을 판단하는 공식은 식(3.13)과 같다.

$$F_a > F_b \text{ and } V_k > T_v \tag{3.13}$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY



(a) 전방낙상 (b) 후방낙상

(c) 좌측낙상 (d) 우측낙상

그림 3.9 낙상 상황에서의 객체 좌표 변화

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

제4장 제안하는 지능형 영상감시 시스템

지능형 영상감시 시스템의 종류는 컴퓨터를 기반으로 한 방법과 소형 디바이스를 기반으로 한 방법으로 나눌 수 있다. 현재 소형 디바이스를 기반으로 한 연구에 비해 컴퓨터를 기반으로 한 연구가 활발히 진행되고 있지만, 소비전력 및 장비 구축 ·유지비용, 사생활 유출 등의 단점이 있어 실제 환경에서 활용하기에는 어려움이 있다. 또한 현재 상용화되어 있는 소형 디바이스 기반 지능형 영상감시 시스템은 움직임 검출, 사람 검출 등 복합적이지 않은 검출을 수행한다. 따라서 본 논문에서는 그림 4.1과 같이 저비용 및 경량화, 낮은 소비전력으로 구동이 가능한 소형 디바이스를 이용하여 지능형 영상감시 시스템을 구성하며 다중 비정상 상황 검출 알고리즘의 실시간 처리를 위해 6개의 알고리즘 최적화 방법 및 3개의 임베디드 모듈 최적화 방법을 제안한다.

그림 4.1 제안하는 지능형 영상감시 시스템 구성

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

제1절 임베디드 모듈 구현

임베디드 모듈이란 특정한 작업을 수행하도록 기계 또는 전자 장치의 두뇌 역할을 하는 마이크로프로세서를 장착해 설계한 시스템으로 정의된다. 임베디드 모듈은 저비용 및 경량화, 낮은 소비전력으로 구동이 가능한 특징이 있으며, 오픈소스 하드웨어의 종류로는 라즈베리파이, 오렌지파이, 비글보드, 아두이노 등이 있지만 범용성 및 성능 면에서 라즈베리파이가 월등해 많은 분야에서 활용하고 있다. 라즈베리파이는 신용카드 크기의 싱글 보드 컴퓨터로 라즈베리파이2에 비해 라즈베리파이3는 프로세서가 32Bit에서 64Bit, 속도가 900MHz에서 1.2GHz로 증가해 실시간 지능형 영상감시 시스템의 소형 디바이스로 적합하다[55].

표 4-1 라즈베리파이 규격

항목	라즈베리파이 B+	라즈베리파이2	라즈베리파이3
프로세서	BCM2835 32Bit Single Core	BCM2836 32Bit QUAD Core	BCM2837 64Bit QUAD Core
그래픽	Videocore Ⅳ	Videocore Ⅳ	Videocore Ⅳ
속도	700 MHz	900 MHz	1.2 GHz
메모리	512MB SDRAM 400MHz	1GB SDRAM 400MHz	1GB SDRAM 400MHz
전원	5V/1.8A	5V/1.8A	5V/2.5A

일반 RGB 카메라에서 출력되는 영상을 이용해 객체 검출을 수행할 경우 그림 4.2와 같이 조명이 부족한 환경 및 야간 상황에서 객체를 검출하지 못하거나 오검출하는 문제점이 있다. 이러한 문제점을 해결하기 위해 임베디드 모듈의 영상입력 장치로 키넥트의 깊이 카메라를 활용한다. 깊이 카메라는 IR 프로젝터에서 투광한 적외선이 IR 카메라로 반사되어 돌아오는 시간을 이용해 거리 정보를 얻는 방식으로, 본 논문에서는 깊이 카메라의 거리 정보를 기반으로 객체를 검출한다. 색상 정보가 아닌 적외선 정보를 이용하므로 조명이 부족한 환경 및 야간 상황에서 그림 4.3과 같이 객체 검출이 가능하다.

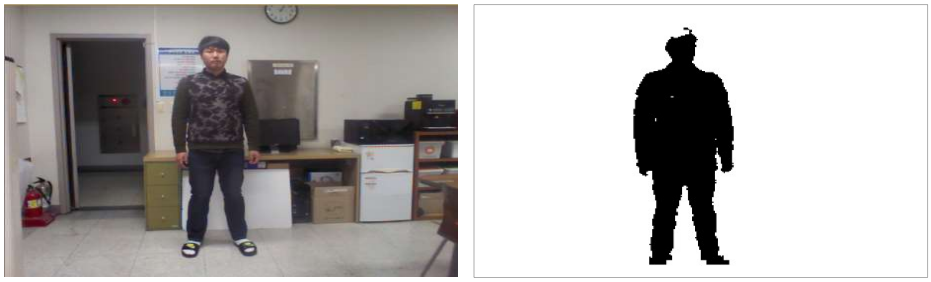
Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

(a) 주간 상황에서 RGB 카메라를 이용한 객체 검출 결과

(b) 야간 상황에서 RGB 카메라를 이용한 객체 검출 결과

그림 4.2 RGB 카메라 객체 검출 결과



(a) 주간 상황에서 깊이 카메라를 이용한 객체 검출 결과

(b) 야간 상황에서 깊이 카메라를 이용한 객체 검출 결과

그림 4.3 깊이 카메라 객체 검출 결과

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

그림 4.4는 임베디드 모듈 구현을 표현하며, 임베디드 모듈 영상입력장치로 키넥트를 사용하기 위해 libfreenect를 활용한다. libfreenect는 OpenKinect에서 키넥트를 리눅스환경에서 사용할 수 있게 개발한 라이브러리로, libfreenect를 활용하기 위해 python과 OpenCV 관련 항목인 python-dev, ipython, python-scipy 등 5개의 필요 라이브러리를 구성하며, build-essential, libxi-dev, git-core, freeglut3, gspca kernel driver 등 11개의 관련 라이브러리를 구성한다. 추가적으로 임베디드 모듈에서 키넥트를 사용하기 위해서는 외부 전원이 있는 USB 허브를 사용해야 하며 이를 위해 libusb를 활용해 RGB 카메라와 깊이 카메라 정보를 전송한다.

그림 4.4 임베디드 모듈 구현

임베디드 모듈을 이용하여 얼굴인식 기반 침입자 검출을 수행하기 위해 pip, linux-armv7l, mock 등 5개의 필요 라이브러리를 구성하여 텐서플로우 환경을 구축한다. 화재 검출, 배회 검출, 낙상 검출을 수행하기 위해 시각화 이미지 패키지 matplotlib, skiamge, 컴파일러 및 링커 플래그를 추가하는 pkg-config, 모듈 활용을 위한 cmake, 이미지 및 비디오 파일을 사용하기 위한 libjpeg-dev, libavocodev-dec, libxine2-dev 등 9개, 비디오 스트리밍을 위한 libgstreamer-dev 등 2개를 구성한다.

임베디드 모듈 및 알고리즘 최적화를 수행하기 위해 로드 밸런싱, 스왑핑, 오

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

버클록, cython 구성한다. 로드 밸런싱을 하기 위해 cgroup-bin을 사용하며, 스왑핑을 하기 위해 swappiness을 활용, 오버클록을 변경하기 위해 config의 arm-freq를 활용, cython을 사용하기 위해 setuptools를 사용한다. 라즈베리파이의 동작 성능을 비교하기 위해 와탭을 사용하며, 와탭은 CPU, 메모리의 성능을 원격으로 감시할 수 있는 소프트웨어로 와탭 리포지토리 및 서버 모니터 패키지를 구성하고 스크립트를 이용한다. 비정상적인 상황 발생 시 알림을 전송하기 위한 ssmtp 및 영상을 전송하기 위한 mock 라이브러리를 구성한다.

제2절 알고리즘 최적화

컴퓨터에 비해 성능이 부족한 임베디드 모듈은 실시간 다중 비정상 상황을 검출하기 위해 알고리즘 최적화 과정 및 임베디드 모듈 최적화 과정이 필수적이다. 본 논문에서는 6개의 알고리즘 최적화 방법, 3개의 임베디드 모듈 최적화 방법을 제안하여 임베디드 모듈의 처리성능을 높이고 알고리즘의 처리 시간을 단축시킨다.

그림 4.5 지능형 영상감시 시스템 최적화 방법

1. 서브루틴 확인

알고리즘의 최적화 과정 중 어설픈 최적화(Premature Optimization)를 하지 않기 위해 알고리즘의 서브루틴 처리 시간을 측정하여 최적화를 수행할 부분을 탐색한다. 서브루틴은 전처리, 침입자 검출, 화재 검출, 배회 검출, 낙상 검출 부분으

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

로 나누었으며 측정된 결과는 그림 4.6과 같다. 침입자 검출의 평균 처리 시간은 0.579초로 가장 크게 나타났으며, 1초당 1.72 프레임을 처리한다.

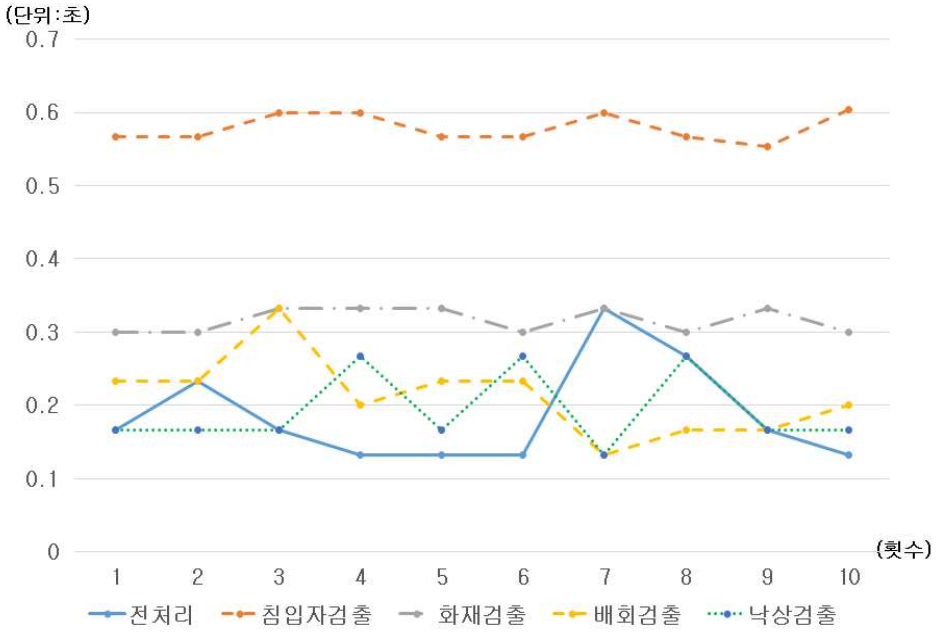


그림 4.6 구간별 처리 시간

2. 양의 정수형 사용

프로그램의 기본 연산자는 int형으로 char, short형으로 선언 후 알고리즘 구동 시 계산되는 과정에서 int형으로 변환하여 계산하고 원래의 char, short형으로 변환하는 불필요한 과정을 거치게 되어 처리 시간을 증가시킨다. 또한, 프로세스는 양의 정수형 연산이 음의 정수형과 양의 정수형을 같이 사용한 경우에 비해 연산이 빠르기 때문에 사용하는 값이 음수가 아닐 경우 양의 정수형으로 선언해 사용한다.

3. 나머지, 나누기 연산자 최소화

표준적인 프로세스의 나눗셈 연산은 분모와 분자 32bit의 경우 20~140의 실행

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

사이클을 가지고 있어 다른 연산자에 비해 처리 시간이 오래 걸린다. 따라서 2의 승수로 연산을 할 경우 빠른 연산자 중 하나인 shift 연산을 이용하며, 그렇지 않은 경우 곱셈으로 대체하여 사용한다.

4. 전역변수 직접계산 최소화

전역변수는 레지스터에 직접 할당해 사용할 수 없어 포인터를 사용해 간접 할당하거나, 함수 호출을 이용하는 방법으로 전역변수를 사용할 수 있다. 포인터와 함수 호출을 할 경우 메모리 낭비가 발생하게 되어 처리 시간이 증가하며, 컴파일러는 전역변수의 값을 직접 사용할 때마다 다시 읽어 들이는 오버로드가 발생한다. 따라서 전역변수의 사용을 자제하고, 사용할 경우 지역변수에 값을 대입 후 지역변수를 계산하는 방법을 사용한다.

5. 함수호출 시 전달인자, 재귀함수 최소화

함수 호출시 전달인자가 4개 이상이 되면 스택을 통해 인자를 전달하여 할당되는 스택의 크기만큼 메모리 접근이 발생한다. 또한 구조체를 전달인자로 전달할 경우 구조체의 모든 값이 스택을 통해 전달되어 메모리 접근이 발생한다. 따라서 전달인자가 4개 이상일 경우 구조체를 선언하고 구조체에 대한 포인터를 넘기는 방식을 사용한다. 추가적으로 알고리즘에서 재귀함수 호출 시 시스템 내부의 스택을 사용해 메모리 접근이 발생하며, 스택을 초기화하지 않기 때문에 처리 시간이 증가하고 오버플로우가 발생할 수 있다. 따라서 반복문 또는 반복 호출, 꼬리 재귀함수를 사용해 재귀함수 사용을 최소화한다.

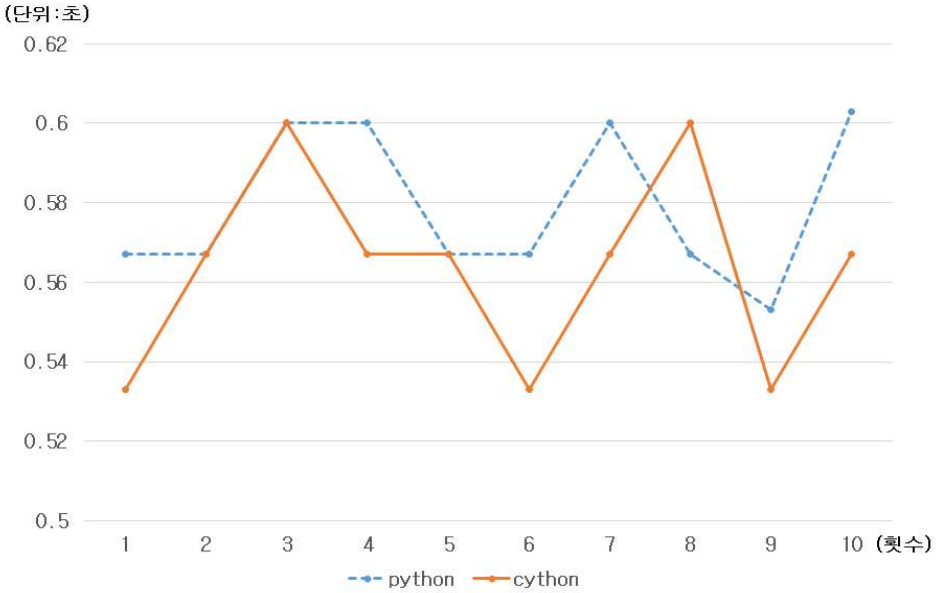
6. cython 언어 사용

cython 언어는 python 언어의 빠른 생산성과 c 언어의 실행속도 장점을 합쳐둔 형태로 단계적 타이핑 언어(Progressively Typed Language or Gradual Typing)다 [56]. 그림 4.7은 python 언어 알고리즘과 cython 언어 알고리즘의 처리 시간 결과로, 서브루틴의 침입자 검출 알고리즘 처리 시간 비교 결과 python 언어는 평균 0.5791초, cython 언어는 평균 0.5634초로 2.71%의 처리 시간 감소가 있으며 서브

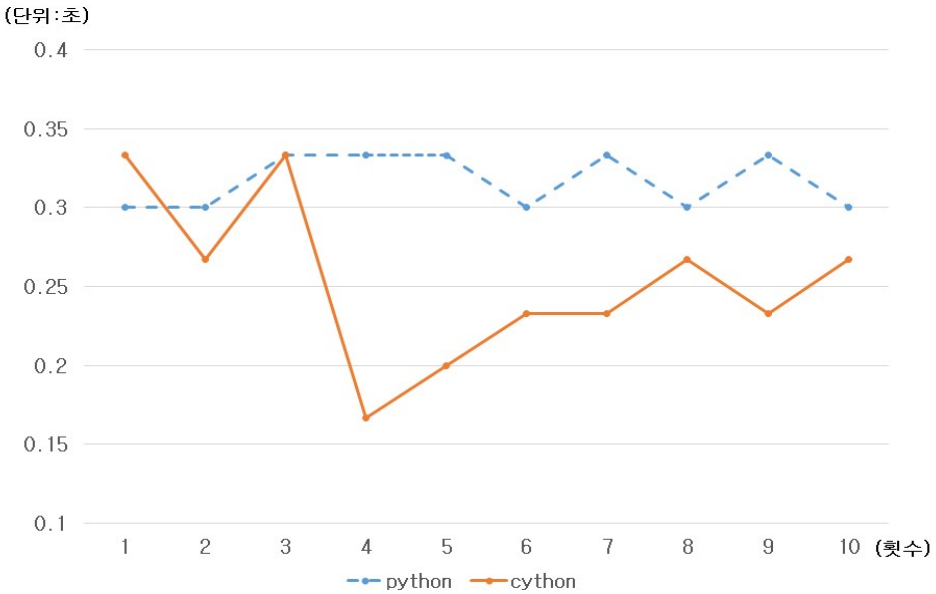
Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

루틴의 화재 검출 알고리즘 처리 시간 비교 결과 python 언어는 평균 0.3165초, cython 언어는 평균 0.2533초로 19.96% 처리 시간이 감소된다.



(a) 서브루틴 - 침입자 검출



(b) 서브루틴 - 화재 검출

그림 4.7 python 언어, cython 언어 알고리즘 처리 시간

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

제3절 임베디드 모듈 최적화

1. 병목 현상 제거

병목 현상이란 전체 시스템의 성능이나 용량이 하나의 구성 요소로 인해 제한받는 현상으로 CPU의 코어 한곳에 집중되어 사용될 경우 속도 저하의 원인이 된다. 또한, CPU의 클록 수를 제한해 사용하고 있어 처리 속도를 제한하고 있다. 따라서 CPU의 코어들 중 한곳에 집중되지 않도록 로드 밸런싱 작업 및 오버클록을 구성해 사용한다. 그림 4.8은 알고리즘 구동 중 CPU 사용량을 표시한 것으로 기본 상태에 비해 오버클록 및 로드 밸런싱을 한 후의 CPU 사용이 증가함을 확인할 수 있다.

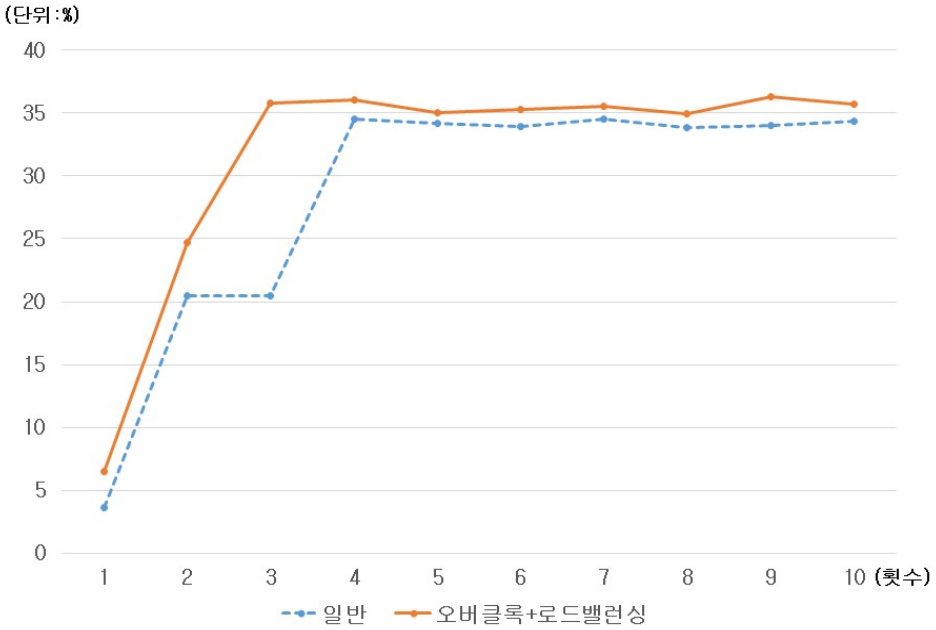


그림 4.8 라즈베리파이 CPU 사용량

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

2. 메모리 활용

알고리즘 구동 시 디스크 기반 계산방법과 메모리 기반 계산방법으로 나뉜다. 그러나 디스크 접근속도에 비해 메모리 접근속도가 월등히 좋아 메모리 기반 계산방법의 처리 속도가 빠르다. 따라서 메모리 기반 계산방법을 사용하기 위해 디스크 캐싱, 스왑핑, 램디스크/램드라이브를 사용한다. 그림 4.9는 알고리즘 구동 중 메모리 사용량을 표시한 것으로 스왑핑 후 메모리 사용이 증가함을 확인할 수 있다.

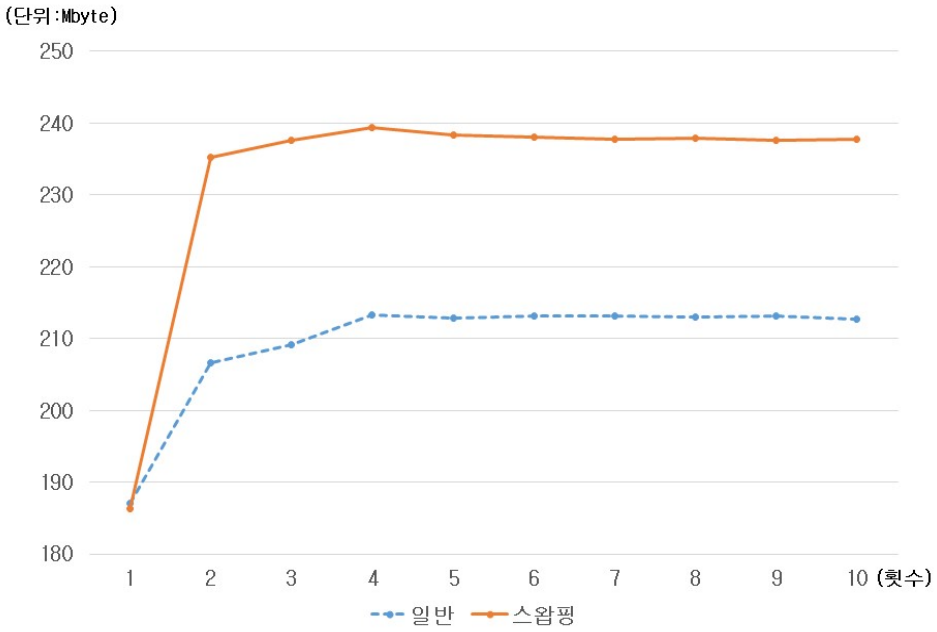


그림 4.9 라즈베리파이 메모리 사용량

3. zRam 사용

zRam은 메모리 사용량이 일정 수준 이상이 되면 메모리 영역의 데이터를 zRam 영역으로 압축 후 옮기는 방법이다. 메모리가 충분한 컴퓨터에서 사용할 경우 데이터 이동과정으로 인해 알고리즘 속도 저하의 원인이 될 수 있으나, 메모리가 충분하지 않은 임베디드 모듈에서 사용할 경우 메모리 부족으로 인한 속도저하 문제

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

를 해결하여 속도를 향상하는 방법이다. 그림 4.10은 zRam의 사용여부에 따른 처리 시간 비교 결과를 나타낸다. zRam 미사용의 경우 평균 처리 시간이 0.2533초이며 zRam 사용의 경우 평균 처리 시간이 0.2467초로 2.6%의 처리 시간이 감소된다 [57].

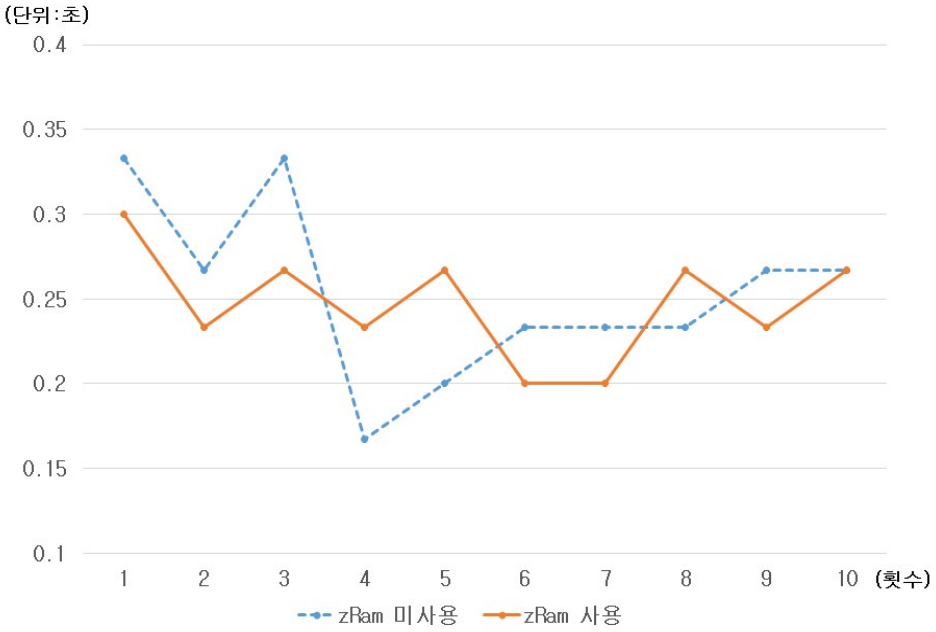


그림 4.10 zRam 알고리즘 처리 시간

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

제5장 실험결과

본 장에서는 지능형 영상감시 시스템의 성능을 확인하기 위해 컴퓨터와 임베디드 모듈의 알고리즘 처리 시간 및 규격, 최적화 전·후의 통합 비정상 상황 검출 알고리즘 및 각 서브루틴 알고리즘의 처리 시간을 비교한다. 표 5-1은 컴퓨터와 임베디드 모듈의 처리 시간을 표시한 내용으로, 컴퓨터의 영상 입·출력 알고리즘의 처리 시간은 평균 0.033초이며 최적화 전 임베디드 모듈의 처리 시간은 평균 0.2133초로 6.46배, 최적화 수행 후 임베디드 모듈의 처리 시간은 평균 0.136초로 4.12배의 처리 시간이 차이난다. 또한 컴퓨터의 비정상적인 상황 검출 알고리즘 처리 시간은 평균 0.053초이며 최적화 전 임베디드 모듈의 처리 시간은 평균 0.9499초로 17.9배, 최적화 수행 후 임베디드 모듈의 처리 시간은 평균 0.47초로 8.86배의 처리 시간이 차이난다.

표 5-1 컴퓨터와 임베디드 모듈 처리 시간 비교

구분	컴퓨터		임베디드 모듈			
			최적화 전		최적화 후	
	영상 입출력	알고리즘 구동	영상 입출력	알고리즘 구동	영상 입출력	알고리즘 구동
1	0.033	0.067	0.167	0.867	0.133	0.467
2	0.033	0.033	0.2	1.033	0.133	0.367
3	0.033	0.067	0.2	0.9	0.1	0.4
4	0.033	0.067	0.233	1	0.167	0.533
5	0.033	0.033	0.167	0.933	0.133	0.6
6	0.033	0.033	0.2	0.9	0.133	0.4
7	0.033	0.067	0.233	0.933	0.133	0.533
8	0.033	0.033	0.267	0.967	0.133	0.5
9	0.033	0.067	0.233	0.933	0.167	0.433
10	0.033	0.067	0.233	1.033	0.133	0.467
평균	0.033	0.053	0.2133	0.9499	0.136	0.47

표 5-2는 컴퓨터와 임베디드 모듈 규격을 비교한 내용으로, 컴퓨터는 x86 기반의 Intel i7-6700K CPU를 사용하며 임베디드 모듈은 ARM 기반의 BCM2837 64Bit QUAD Core를 사용한다. 제조사가 다른 두 CPU는 정량적인 비교가 불가능하지만, 갤럭시 S5 엑시노스 5410의 성능과 3배 이상 차이가 나는 성능을 보여준 i7-4770

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

에서 알 수 있듯이 코어 수, 클록 등의 차이를 통해 대략 12배 이상의 성능차이가 남을 확인할 수 있어 임베디드 모듈에 비해 컴퓨터의 성능이 좋다[58]. 그러나 무게는 230배 이상, 부피는 364배 이상, 소비전력은 44배로 임베디드 모듈이 적으며, 컴퓨터의 장비구축 비용은 156만원 이상인 반면 임베디드 모듈의 장비구축 비용은 4만원대로 33배 이상 차이가 난다.

표 5-2 컴퓨터와 임베디드 모듈 규격 비교

구분	컴퓨터	임베디드 모듈 (라즈베리파이)
CPU	Intel i7-6700K	BCM2837 64Bit QUAD Core
그래픽	GTX980Ti	Videocore Ⅳ
속도	4.0GHz	1.2 GHz
메모리	16GB DDL4L	1GB SDRAM 400MHz
전원	220V	5V
무게	10.52Kg	45g
크기	42 x 16.5 x 40.1 cm	8.5 x 5.6 x 1.6 cm
가격	156만원 이상	4만원대

본 논문에서 제안한 6개의 알고리즘 최적화 방법과 3개의 임베디드 모듈 최적화 방법을 수행한 지능형 영상감시 시스템의 최적화 전·후의 성능을 비교한다. 그림 5.1은 통합 비정상 상황 검출 알고리즘의 처리 시간을 비교한 결과로 최적화 전 알고리즘의 처리 시간은 평균 0.9499초인 반면 최적화 후 알고리즘의 처리 시간은 평균 0.47초로 50.52% 감소되었다.

그림 5.2부터 그림 5.6은 통합 비정상 상황 검출 알고리즘의 각 서브루틴 처리 시간을 나타낸다. 그림 5.2는 전처리 처리 시간을 비교한 결과로 최적화 전 처리 시간은 평균 0.1866초이며 최적화 후 처리 시간은 평균 0.1398초로 25.08% 감소되었다. 그림 5.3은 침입자 검출 알고리즘 처리 시간을 비교한 결과로 최적화 전 처리 시간은 평균 0.5791초이며 최적화 후 처리 시간은 평균 0.5534초로 4.43% 감소되었다. 그림 5.4는 화재 검출 알고리즘 처리 시간을 비교한 결과로 최적화 전 처리 시간은 평균 0.3165초이며 최적화 후 처리 시간은 평균 0.2432초로 23.15% 감소되었다. 그림 5.5는 배회 검출 알고리즘 처리 시간을 비교한 결과로 최적화 전 처리 시간은 평균 0.2132초이며 최적화 후 처리 시간은 평균 0.1533초로 28.09% 감소되었다. 마지막으로 그림 5.6은 낙상 검출 알고리즘 처리 시간을 비교한 결과

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

로 최적화 전 처리 시간은 평균 0.1936초이며 최적화 후 처리 시간은 평균 0.1534초로 20.76% 감소되었다.

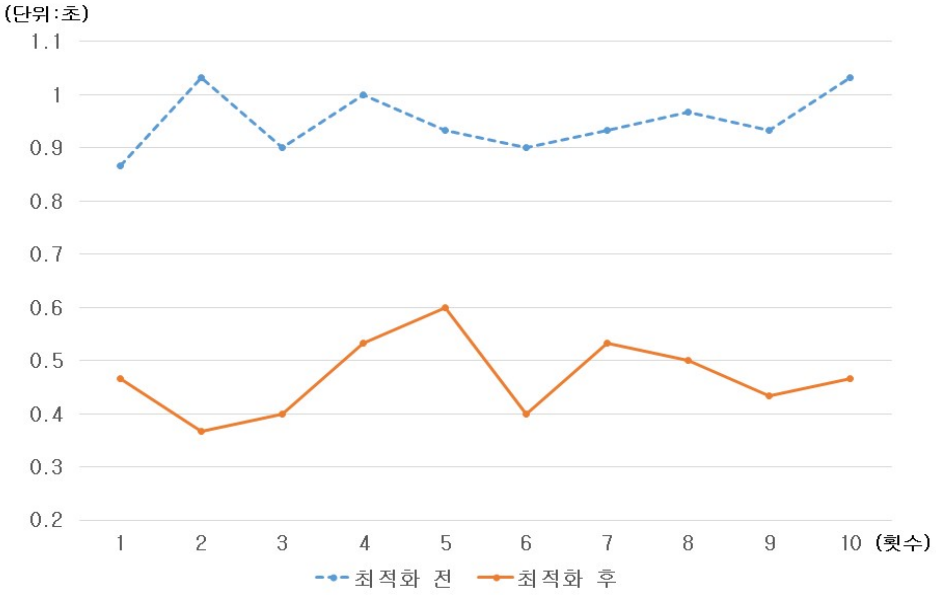


그림 5.1 통합 비정상 상황 검출 알고리즘 처리 시간 비교

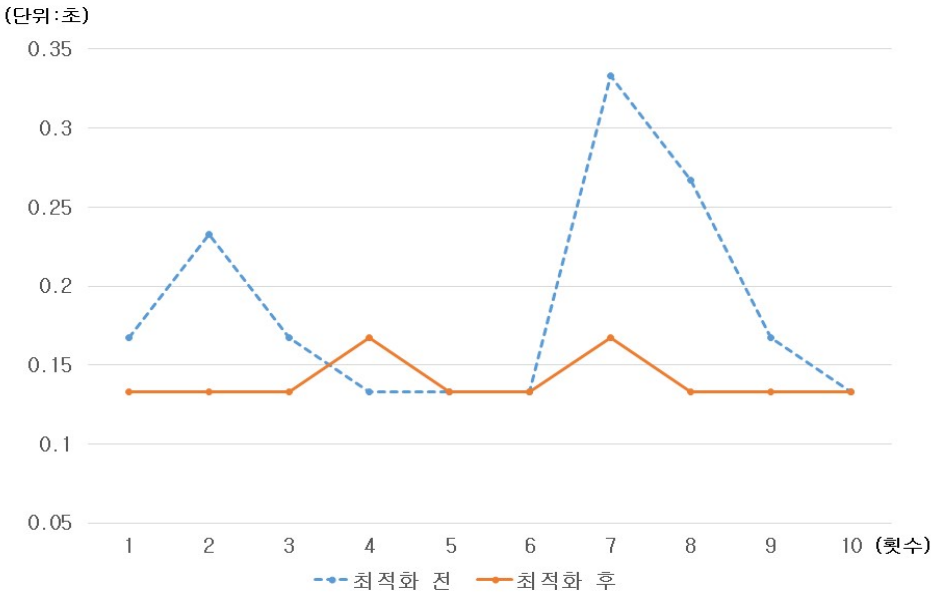


그림 5.2 전처리 시간 비교

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

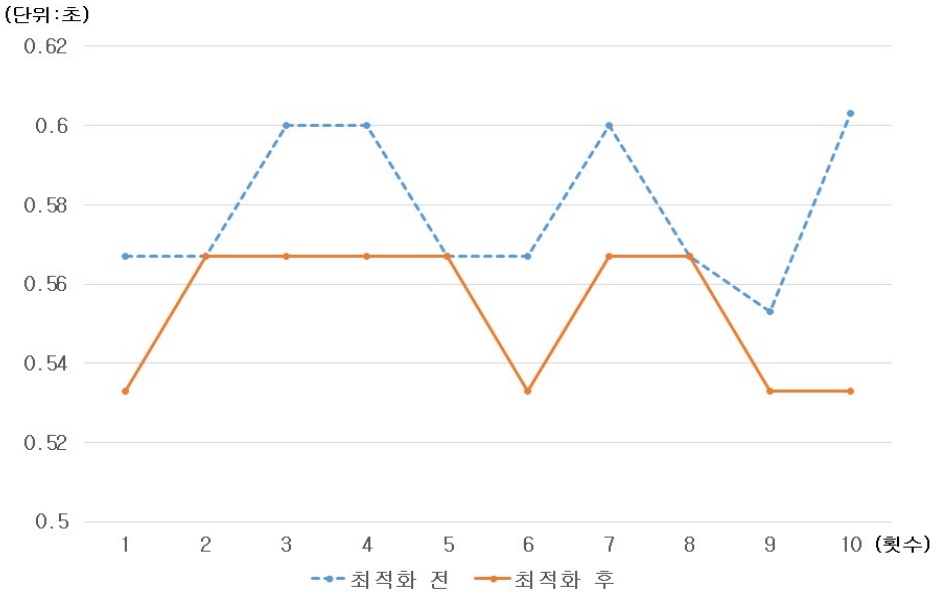


그림 5.3 침입자 검출 처리 시간 비교

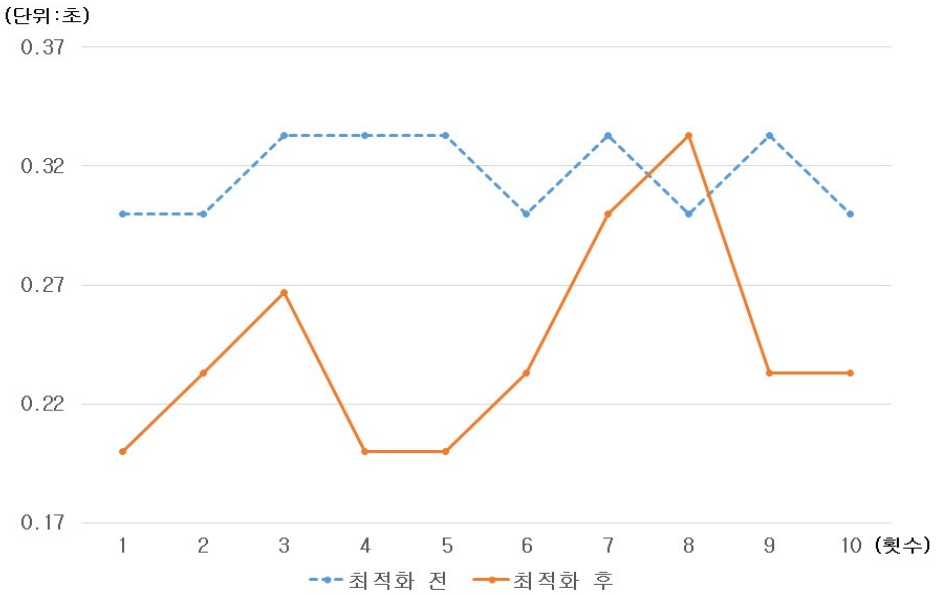


그림 5.4 화재 검출 처리 시간 비교

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

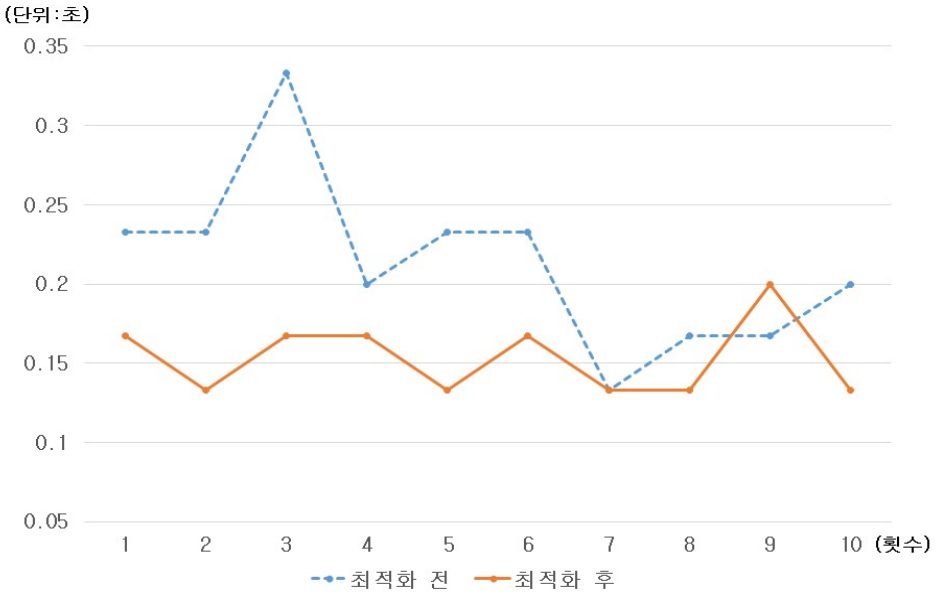


그림 5.5 배회 검출 처리 시간 비교

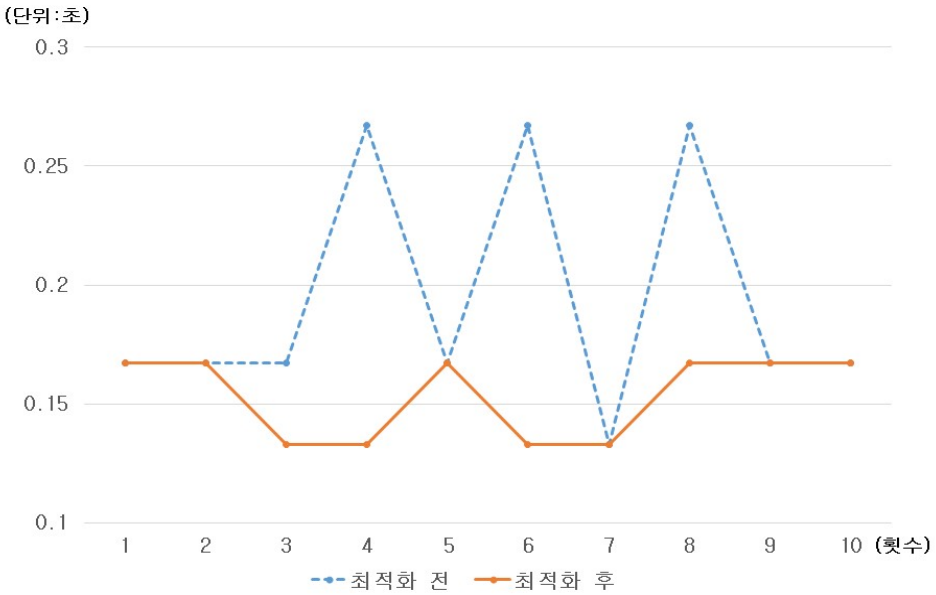
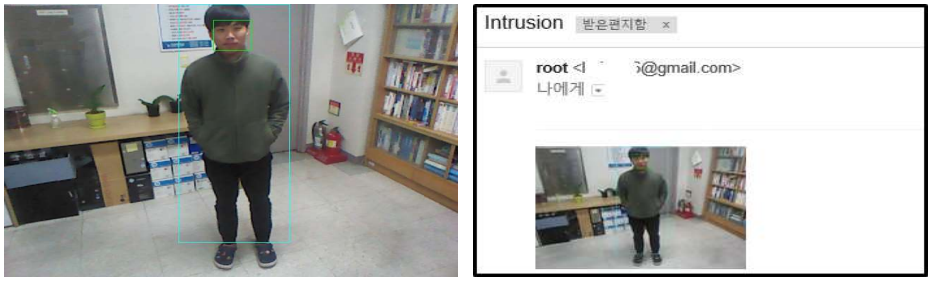


그림 5.6 낙상 검출 처리 시간 비교

Collection @ chosun

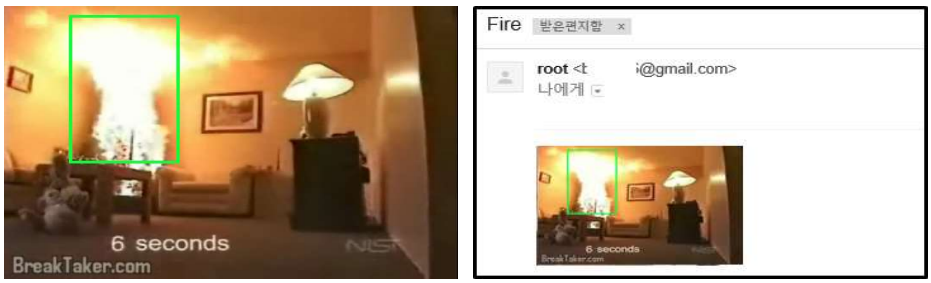
조선대학교
CHOSUN UNIVERSITY

그림 5.7부터 그림 5.10은 침입자, 화재, 배회, 낙상의 비정상적인 상황 발생 시 비정상적인 상황을 검출한 결과와 이를 원격으로 사전에 정의된 사용자에게 전달하여 비정상적인 상황이 발생함을 알린 결과를 나타낸다.



(a) 침입자 검출 (b) 침입자 검출 후 사용자 알림

그림 5.7 비정상적인 상황(침입자)시 알림



(a) 화재 검출 (b) 화재 검출 후 사용자 알림

그림 5.8 비정상적인 상황(화재)시 알림

(a) 배회 검출 (b) 배회 검출 후 사용자 알림

그림 5.9 비정상적인 상황(배회)시 알림

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

(a) 낙상 검출 (b) 낙상 검출 후 사용자 알림

그림 5.10 비정상적인 상황(낙상)시 알림

표 5.3은 비정상적인 상황 검출 알고리즘의 성능을 나타낸다. 침입자 검출을 수행하기 위해 그림 5.11과 같이 거리별 얼굴 회전 영상을 이용하여 원영상 및 조명변화 영상으로 학습을 진행하며, 지능형 영상감시 시스템을 이용해 실시간 얼굴 검출 및 인식 실험으로 성능을 평가하였다. 화재 검출의 성능을 평가하기 위해 미국 국립표준연구소(NIST)의 화재 데이터베이스를 이용하며, 실내 가정환경에서 소파 화재 및 건조한 나무 화재 영상으로 구성된 데이터베이스를 이용해 실험을 진행하였다[59]. 배회 검출의 성능을 평가하기 위해 PETS 데이터베이스 및 자체 데이터베이스를 이용하였다. PETS 데이터베이스는 실내외 환경에서 영상 내에 다수의 인원이 등장하고 그중 정해진 인원이 배회를 수행하며[60], 자체데이터베이스는 실내외 환경에서 한 명의 객체가 배회를 수행한다. 낙상 검출의 성능을 평가하기 위해 지능형 영상감시 시스템을 이용해 실시간 구동으로 성능을 평가하였다.

표 5-3 비정상적인 상황 검출 알고리즘 성능

구분	데이터베이스	인식률
침입자 검출	실시간 실험	88.51%
화재 검출	미국 국립표준연구소 화재 데이터베이스	92.63%
배회 검출	PETS 데이터베이스 + 자체 데이터베이스	80%
낙상 검출	실시간 실험	93.54%

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

그림 5.11 침입자 검출을 위한 학습 영상

(a) NIST 데이터베이스 1

(b) NIST 데이터베이스 2

그림 5.12 비정상적인 상황(화재) 데이터베이스

(a) PETS 데이터베이스

(b) 자체 데이터베이스

그림 5.12 비정상적인 상황(배회) 데이터베이스

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

제6장 결론

본 논문에서는 임베디드 모듈을 이용한 지능형 영상감시 시스템을 구현하였으며, 6개의 알고리즘 최적화 및 3개의 임베디드 모듈 최적화 방법을 제안하였다. 학습된 정보를 기반으로 얼굴을 인식하여 침입자를 검출하며 색상 정보와 움직임 정보를 이용한 화재 검출, 객체의 밸런스 포인트의 변화 각도와 이동거리를 이용한 배회 검출, 객체의 움직임 특징과 가속도를 이용한 낙상 검출의 비정상적인 상황 검출을 수행하였다. 실시간 처리를 위해 제안한 서브루틴 확인, 양의 정수형 사용, 나머지·나누기 연산자 최소화, 전역변수 직접계산 최소화, cython 언어 사용, 함수호출 시 전달인자·재귀함수 최소화, 총 6개의 알고리즘 최적화 방법을 수행하였으며 병목 현상 제거, 메모리 활용, zRam 사용, 총 3개의 임베디드 모듈 최적화 방법을 수행하였다.

제안한 지능형 영상감시 시스템과 컴퓨터의 알고리즘 처리 시간 및 규격을 비교한 결과 알고리즘 처리 시간 8.86배, CPU 성능은 대략 12배로 컴퓨터가 좋지만 무게 230배, 부피 364배, 소비전력 44배, 장비구축 비용 33배 이상 지능형 영상감시 시스템이 월등해 실제 환경에 적합함을 확인하였다. 또한 지능형 영상감시 시스템의 최적화 전·후 처리 시간 비교결과 통합 비정상 상황 검출 알고리즘은 50.52% 처리 시간이 감소되었으며, 각 서브루틴의 비교 결과 전처리 25.08%, 침입자 검출 4.43%, 화재 검출 23.15%, 배회 검출 28.09%, 낙상 검출 20.76%으로 평균 20.3% 처리 시간이 감소되었다. 기존 처리 시간 단축 연구는 HTTP 헤더 최대수명 지정, Gzip 압축송수신, 자바스크립트 자원 재배치 등 5가지 최적화 기법을 이용해 모바일 애플리케이션 처리 시간을 단축한 연구로 평균 시간 30% 감소된 결과가 있지만, 이는 운영체제 및 구동 알고리즘이 다르기 때문에 정량적인 비교는 불가하다[61].

5개의 서브루틴 중 침입자 검출의 알고리즘 처리 시간 변화폭이 가장 적은 이유는 얼굴 검출을 위한 Haar-like feature 기반 학습파일의 입력영상과 얼굴인식을 위한 딥 러닝 학습파일의 입력영상의 최적화가 수행되지 않았기 때문이다. 향후에는 지능형 영상감시 시스템의 처리 시간 감소를 위해 학습파일의 최적화를 수행하며 TCP/IP 기반 임베디드 모듈 병렬처리에 관해 연구할 계획이다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

참고문헌

[1] e-나라지표, http://www.index.go.kr/

[2] 박화진, “객체 추적을 통한 이상 행동 감시 시스템 연구,” 디지털콘텐츠 학회논문지, Vol. 14, No. 9, pp.589-596, 2013.

[3] 이윤, 홍서아, 임보영, “우리나라 지역경찰의 치안활동 연구,” 형사정책 연구원 연구총서, pp. 19-513, 2011.

[4] 김원제, “CCTV 시장동향 및 전망,” 전자정보센터, 2011.

[5] 장일식, 차현희, 박구만, 이광직, 김성권, 차재상, “지능형 감시 카메라 동향 및 시나리오 연구,” 한국ITS학회논문지, Vol. 8, No. 4, pp. 93-101, 2009.

[6] 강주형, 곽수영, “지능형 감시시스템을 위한 배회, 도주, 침입자 검출,” 한국정보과학회 학술발표 논문집, Vol. 31, No. 1, pp. 353-355, 2012.

[7] KOSIS, http://kosis.kr/

[8] Shuai Zhang, Shing Chow Chan, Yeung-Sam Hung, “On the design and implementation of a high definition multi-view intelligent video surveillance system,” Signal Processing, Communication and Computing, pp. 353-357, 2012.

[9] Bor-Shing Lin, Jhe-Shin Su, Hao Chen, Ching Yuh Jan, “A fall detection system based on human body silhouette,” Intelligent Information Hiding and Multimedia Signal Processing, pp. 49-50, 2013.

[10] 에스원, https://www.s1.co.kr/

[11] DSN, http://www.daeshin-it.net/

[12] CRIFAS, http://www.crifas.co.kr/

[13] NUUO, “Intelligent video surveillance(IVS),” 2012.

[14] Maria Tsourma, Minas Dasygenis, “Development of a hybrid defensive embedded system with face recognition,” Computational Science and Engineering and Embedded and Ubiquitous Computing and Distributed Computing and Applications for Business Engineering, pp. 154-157,

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

2016.

[15] Virendra, Shete, Niraj Ukunde, "Intelligent embedded video monitoring system for home surveillance," Inventive Computation Technologies, Vol. 1, pp. 1-4, 2016.

[16] Chi Yuan, Khaled Ghamry, Zhixiang Liu, Youmin Zhang, "Unmanned aerial vehicle based forest fire monitoring and detection using image processing technique," Chinese Guidance, Navigation and Control Conference, pp. 1870-1875, 2016.

[17] EasyN, https://www.easyn.com/

[18] KT, https://product.kt.com/

[19] LGU+, https://www.uplus.co.kr/

[20] Matthew Turk, Alex Pentland, "Face recognition using eigenfaces," Computer Society Conference on Computer Vision and Pattern Recognition, pp. 586-591, 1991.

[21] Jia zhong He, Qing huan Zhu, Ming hui Du, "Face recognition using PCA on enhanced image for single training images," Machine Learning and Cybernetics, pp. 3218-3221, 2006.

[22] Daoqiang Zhang, Zhi Hua Zhou, Songcan Chen, "Diagonal principal component analysis for face recognition," Pattern Recognition, Vol. 39, No. 1, pp. 140-142, 2006.

[23] Hong Duan, Ruohe Yan, Kunhui Lin, "Research on face recognition based on PAC," Future Information Technology and Management Engineering, pp. 29-32, 2008.

[24] Jianke Li, Baojun Zhao, Hui Zhnag, "Face recognition based on PCA and LDA combination feature extraction," Information Science and Engineering, pp. 1240-1243, 2009.

[25] Hae Min Moon, Dong Jin Choi, Pan Koo Kim, Sung Bum Pan, "LDA-based face recognition using multiple distance training face images with low user cooperation," Consumer Electronics, pp. 7-8, 2015.

[26] Hongshuai Zhang, Zhiyi Qu, Liping Yuan, Gang Li, "A face recognition method based on LBP feature for CNN," Advanced

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Information Technology, Electronic and Automation Control Conference, pp. 544-547, 2017.

[27] Junging Zeng, Yikui Zhai, Junying Gan, "A novel sparse representation classification face recognition based on deep learning," UIC-ATC-ScalCom, pp. 1520-1523, 2015.

[28] Gyu Jin Jang, Jin Pyung Kim, Moon Hyun Kim, Jae Yong Jung, Tae Ki Ahn, "Recognition of fire levels using fuzzy reasoning," Industrial Mechatronics and Automation, Vol. 2, pp. 557-560, 2010.

[29] Solorzano, Javier Rodriguez Fonollosa, Liliana Fernandez Samaca, Eichmann, Santiago Marco, "Fire detection using a gas sensor array with sensor fusion algorithms," Olfaction and Electronic Nose, pp. 1-3, 2017.

[30] Wenhao Wang, Hong Zhou, "Fire detection based on flame color and area," Computer Science and Automation Engineering, Vol. 3, pp. 222-226, 2012.

[31] Teng Wang, Leping Bu, Qingfeng Zhou, Zhonglin Yang, "A new fire recognition model based on the dispersion of color component," Progress in Informatics and Computing, pp. 138-141, 2015.

[32] 김민구, 반성범, "라즈베리파이 기반 색상정보를 이용한 화재 검출 시스템에 관한 연구," 한국정보기술학회논문지, Vol. 14, No. 6, pp. 87-93, 2016.

[33] Lin Wang, Aiguo Li, "Early fire recognition based on multi-feature fusion of video smoke," Chinese Control Conference, pp. 5318-5323, 2017.

[34] Nathaniel Bird, Osama Masoud, Mikolaos Papanikolopoulos, Aaron Isaacs, "Detection of loitering individuals in public transportation areas," Transactions on Intelligent Transportation Systems, Vol. 6, No. 2, pp. 167-177, 2005.

[35] Mohannad Elhamod, Martin Levine, Life Fellow, "Automated real-time detection of potentially suspicious behavior in pubilc transport areas," Transactions on Intelligent Transportation Systems, Vol.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

14, No. 2, pp. 688-699, 2013.

[36] 이승원, 김태경, 유장희, 백준기, “지능형 비디오 분석을 위한 적응형 배경 생성 기반의 이상행위 검출,” 전자공학회논문지, Vol. 48, No. 1, pp. 111-121, 2011.

[37] 장태우, 신용태, 김종배, “지능형 감시를 위한 객체 추출 및 추적시스템 설계 및 구현,” 한국통신학회논문지, Vol. 38, No. 7, pp. 589-595, 2013.

[38] Jong Gook Ko, Jang Hee Yoo, “Rectified trajectory analysis based abnormal loitering detection for video surveillance,” Modeling and Simulation, Artificial Intelligence, pp. 289-293, 2013.

[39] 박은수, 이형호, 윤명규, 김민규, 곽종훈, 김학일, “그림자 제거와 색도 히스토그램 비교를 이용한 배회행위 검출,” 정보보호학회논문지, Vol. 21, No. 6, pp. 159-169, 2011.

[40] Wenting Li, Dongping Zhang, Min Sun, Yibo Yin, Ye Shen, “Loitering detection based on trajectory analysis,” Intelligent Computation Techonology and Automation, pp. 530-533, 2015.

[41] 강주형, 곽수영, “방범용 CCTV를 위한 배회행위 탐지 솔루션,” 한국멀티미디어학회논문지, Vol. 17, No. 1, pp. 15-25, 2014.

[42] Lindemann, Hock, Stuber, Keck, Becker, “Evaluation of a fall detector based on accelerometers: A pilot study,” Medical and Biological Engineering and Computing, Vol. 43, No. 5, pp. 548-551, 2005.

[43] Medrano, Igual, Plaza, Castro, Fardoun, “Personalizable smartphone application for detecting falls,” Biomedical and Health Informatics, pp. 169-172, 2014.

[44] Zhigang Liu, Yanan Song , Ye Shang, Jinkuan Wang, “Posture recognition algorithm for the elderly based on BP neural networks,” Chinese Control and Decision Conference, pp. 1446-1449, 2015.

[45] Foroughi, Rezvanian, Paziraee, “Robust fall detection using human shpae and multi-class support vector machine,” Computer Vision, Graphics and Image Processing, pp. 413-420, 2008.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[46] Bevilacqua, Nuzzoloese, Barone, Pantaleo, Suma, Ambruosos, Volpe, Loconsole, Stroppa, "Fall detection in indoor environment with kinect sensor," Innovation in Intelligent Systems and Applications Proceedings, pp. 319-324, 2014.

[47] Jia Dongyao, Viocu Groza, Liu XiaoHui, Liu Xu, Zhu Huaihua, Qingsheng Zeng, "A method of detecting human body falling action in a complex background," Jubilee International Conference on Intelligent Engineering Systems, pp. 51-56, 2016.

[48] Viola, Jones, "Rapid object detection using a boosted cascade of simple features," Electrical and Electronics Conference on Computer Vision and Pattern Recognition, Vol. 1, pp. 511-518, 2001.

[49] Viola, Jones, "Robust real-time face detection," International Journal of Computer Vision, Vol. 57, No. 2, pp. 137-154, 2004.

[50] Smit Trambadia, Hemant Mayatra, "Food detection on plate based on the HSV color model," Green Engineering and Technologies, pp. 1-6, 2016.

[51] Wen chen, Yun Shi, Guorong Xuan, "Identifying computer graphics using HSV color model and statistical moments of characteristic functions," International Conference on Multimedia and Expo, pp. 1123-1126, 2007.

[52] Berthold Horn, Brian Schunck, "Determining optical flow," Artificial Intelligence, Vol. 17, pp. 185-203, 1981.

[53] Jin Su Kim, Min Gu Kim, Byung Rae Cha, Sung Bum Pan, "A study of determining abnormal behaviors by using system for preventing agricultural product theft," International Conference on Computer Science and its Applications, Springer Sigapore, 2016.

[54] 황세현, 반성범, "오픈소스 하드웨어와 RGB 카메라를 이용한 낙상 검출 시스템," 한국정보기술학회논문지, Vol. 14, No. 4, pp. 19-24, 2016.

[55] Wikipedia, https://en.wikipedia.org/wiki/Raspberry_Pi

[56] Wikipedia, https://en.wikipedia.org/wiki/Cython

[57] Marzetta, Ambros, "ZRAM: A library of parallel search algorithms

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

and its use in enumeration and combinatorial optimization," Hartung Gorre, 1998.

[58] Gigglehd, http://gigglehd.com/zbxe/10355190

[59] NIST, https://www.nist.gov/

[60] PETS, http://www.cvg.reading.ac.uk/PETS2007/

[61] 강주원, "Front-End 최적화를 통한 기업용 모바일 애플리케이션 처리속도의 개선 연구," 석사학위논문, 서울과학기술대학교 컴퓨터공학과, 2016.

Collection @ chosun