

제 출 문

국토교통부장관(국토교통과학기술진흥원장) 귀하

'오픈소스 하드웨어 기반 독거노인 및 장애인 실내 환경 모니터링 시스템 개발 및 클라우드를 통한 지속적 자가 학습 딥러닝 시스템 구현'(연구개발 기간 : 2016 . 6. 24 ~ 2018 . 12. 23) 과제의 최종보고서를 제출합니다.

2019 . 1 . 23 .

주관연구기관명 : 동국대학교 산학협력단 (대표자) 이 용 규 (인)



주관연구기관책임자: 염 세 경 교수

국토교통부소관 연구개발사업 운영규정 제29조에 따라 최종보고서 열람에
동의합니다.

보고서 요약서

과제 고유 번호	16CTAP-C1 14867-01	해당 단계 연구 기간	2016. 6. 24 ~ 2018. 12. 31	단계구분	(해당단계)2/ (총단계)2
연구사업명	중사업명	국토교통기술촉진연구사업			
	세부사업명				
연구과제명	대과제명	오픈소스 하드웨어 기반 독거노인 및 장애인 실내 환경 모니터링 시스템 개발 및 클라우드를 통한 지속적 자가 학습 딥러닝 시스템 구현			
	세부과제명				
연구책임자	염세경	해당단계 참여연구원 수	총: 4 명 내부: 4 명 외부: 명	해당단계 연구비	정부: 50,000 천원 민간: 천원 정부외: 천원 상대국: 천원 계:50,000 천원
		총 연구기간 참여연구원 수	총: 6 명 내부: 6 명 외부: 6 명	총 연구비	정부: 100,000 천원 민간: 천원 정부외: 천원 상대국: 천원 계:100,000 천원
연구기관명 및 소속 부서명	동국대학교 산학협력단 산업시스템공학과			참여기업명	
국제공동연구	상대국명:			상대국 연구기관명:	
위탁연구	연구기관명:			연구책임자:	
※ 국내·외의 기술개발 현황은 연구개발계획서에 기재한 내용으로 같음					

연구개발성과의 보안등급 및 사유	
-------------------------	--

9대 성과 등록·기탁번호

구분	논문	특허	보고서 원문	연구시설 · 장비	기술요약 정보	소프트 웨어	화합물	생명자원		신품종	
								생명 정보	생물 자원	정보	실물
등록·기 탁 번호	2	2									

국가과학기술종합정보시스템에 등록된 연구시설·장비 현황

구입기관	연구시설 · 장비명	규격 (모델명)	수량	구입연월일	구입가격 (천원)	구입처 (전화)	비고 (설치장소)	NTIS 등록번호

(요약)

- 오픈 하드웨어 기반 환자 자세 분석 및 노인 활동 패턴 분석 알고리즘 개발
- 가속도 센서 및 심박측정 센서를 통한 움직임 모니터링 기술
- 딥러닝을 활용한 CNN(Convolution Neural Network)기반 정확도 향상 알고리즘 및 클라우드 솔루션의 구축

보고서 면수
79

요약문

연구의 목적 및 내용	<연구의 목적> • 오픈 하드웨어 기반 환자 자세 분석 및 노인 활동 패턴 분석 알고리즘 개발 • 가속도 센서 및 심박측정 센서를 통한 움직임 모니터링 기술 • 딥러닝을 활용한 CNN(Convolution Neural Network)기반 정확도 향상 알고리즘 및 클라우드 솔루션의 구축 <연구의 주요내용> • 저가형 카메라 모듈 다수를 오픈소스 하드웨어에 연동하여 효율적으로 다수의 카메라를 연동하는 시스템의 구현 • 기존 실루엣 이미지를 이용한 알고리즘에 기반을 둔 이미지 인식기법 구현 및 딥러닝을 이용한 상세 이미지 분석 및 구분 • 오픈소스 하드웨어에 기반을 둔 포터블 환자 패턴 분석용 모니터링 시스템의 개발 및 노인 활동 패턴 분석 알고리즘과의 결합 • 가속도 센서 및 심박측정 센서를 부착한 오픈소스 하드웨어 기술의 개발 구현 • 딥러닝을 활용한 CNN의 구축 및 이를 통한 자기 강화 학습 알고리즘의 구축 • 클라우드 시스템 구축을 통한 빅데이터 솔루션의 구현 및 이를 딥러닝에 활용				
연구개발성과	• 오픈소스 카메라를 이용한 모니터링 시스템 설계 및 구현 • Statics/Dynamic 활동 구분 및 기본 동작의 세분화 인식 구분을 통한 위급상황(낙상상황) 인식 • 개별 라즈베리파이 분석 시스템 모듈 개발 및 리눅스 서버 개발 • 카메라 영상 이미지와 연동한 바이오 센서의 낙상 상태 검출 • 자세분석 모델을 활용한 위급상황(낙상) 검출 시스템 구축 • 딥러닝 학습을 위한 이미지 수집 시스템 구축 • 딥러닝 학습을 위한 이미지 labeling 자동화 시스템 구축 • 정량성과: SCIE논문 2편 출간, 특허출원 2건				
연구개발성과의 활용계획 (기대효과)	• 오픈소스 하드웨어를 기반으로 하는 노인이나 환자, 어린이 등의 활동 패턴 분석 및 분류 연구를 통한 행동 위험 감지 시스템 • 실내공간조성(예. 1인 가구, 독거노인층, 실버타운, 장애인 가정)의 레퍼런스 자료 • 치매노인, 정신질환자들의 실내 행동 파악을 통한 관리 처방 가이드라인				
국문핵심어 (5개 이내)	활동모니터링	패턴인식	빅데이터분석	딥러닝	
영문핵심어 (5개 이내)	Activity monitoring	Pattern recognition	Big data analysis	Deep Learning	

〈 목 차 〉

1. 연구개발과제의 개요	5
2. 연구수행내용 및 성과	15
3. 목표 달성도 및 관련 분야 기여도	50
4. 연구개발성과의 활용 계획 등	52
붙임. 참고 문헌	54
붙임. 소스코드	56

<별첨> 주관연구기관의 자체 평가 의견서

1. 연구개발과제의 개요

1-1 연구개발의 개요

- 본 연구개발은 고령인구 증가에 따른 보건의료 수요 증가에 대응하고 지속가능한 의료서비스 제공을 위해 고령 노인이 활동하는 공간에 모니터링(monitring), 센싱(sensing)할 수 있는 장치를 설치하여 수집하고, 패턴인식 기법을 활용하여 해당 정보를 분석해 능동적으로 대응할 수 있는 시스템을 구현
- 이를 구현하기 위한 주요 핵심성과물 및 제안 연구내용
 - ① 오픈 하드웨어 기반 환자 자세 분석 및 노인 활동 패턴 분석 알고리즘 개발
 - ② 가속도 센서 및 심박측정 센서를 통한 움직임 모니터링 기술
 - ③ 딥러닝을 활용한 CNN 기반의 정확도 향상 알고리즘 및 클라우드 솔루션 구축하고 개발된 시스템의 적용을 위한 파일럿 테스트 실시

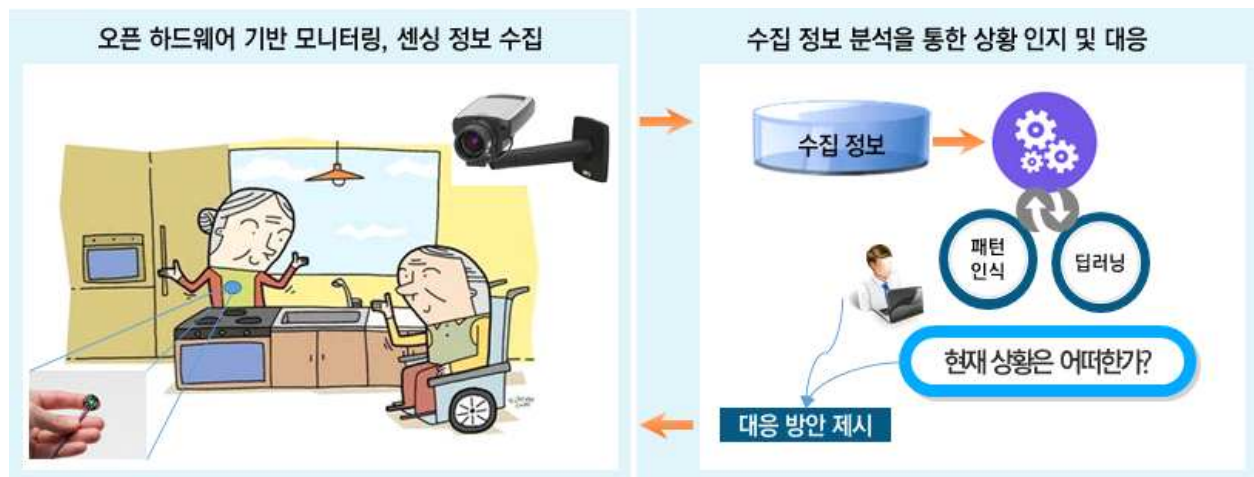


그림 1. 개략적 연구내용

- 본 연구개발을 통해 확보될 성과물
 - 저가형 카메라 모듈 다수를 오픈소스 하드웨어에 연동하여 효율적으로 다수의 카메라를 연동하는 시스템의 구현
 - Bayes Classifier 및 기존 알고리즘에 기반을 둔 이미지 인식기법 구현
 - 오픈소스 하드웨어에 기반을 둔 포터블 환자 패턴 분석용 모니터링 시스템의 개발 및 노인 활동 패턴 분석 알고리즘과의 결합
 - 가속도 센서 및 심박측정 센서를 부착한 오픈소스 하드웨어 기술의 개발 구현

- 딥러닝을 활용한 CNN(Convolution Neural Network)의 구축 및 이를 통한 자기 강화 학습 알고리즘의 구축
- 클라우드 시스템 구축을 통한 빅데이터 솔루션의 구현 및 이를 딥러닝에 활용
- 향후 초고령 사회로 진입하는 국내 상황과 그에 따른 전문 의료진 및 간호인 부족으로 인해 야기될 수 있는 한계를 해소하고 고령자의 건강한 삶의 질 향상을 도모
- 스마트 요양 공간을 구축하고, 행동 위험 감지를 통한 관리 처방 가이드라인 개발 촉진하는데 기여

1-2 연구개발 대상의 국내외 현황

○ 국내 현황

- 급속한 고령인구의 증가, 질병구조 변화 등은 보건·의료 분야의 새로운 수요를 발생시키고, 기존 보건의료 서비스의 제공 방식에 대한 전환을 촉구
 - OECD 국가 중 가장 빠른 고령화 추세를 보이는 우리나라의 고령인구는 2030년에는 24%로 급격하게 증가할 것으로 전망(통계청, 장래인구 추계, 2011)
- 지속적으로 증가하는 만성질환자의 수는 선진국형 질병구조로 변화하고 있으며, 국민 의료비 상승을 가져오고 있음
 - 우리나라 질병구조는 지난 20년간 순환기계, 신생물(암), 근골격계가 증가한 반면 소화기계 질환, 호흡기 질환 등이 감소한 선진국형 질환 구조로 전환(통계로 본 건강보험 시행 35년, 2012)
- 인구·질병 구조 등 새로운 보건·의료 환경은 국민 의료비 증대, 정부의 재정부담 가중 및 전문 의료진 부족 등의 현안을 야기
 - 고혈압, 당뇨, 비만 등 만성질환자 위험군 비율이 성인의 65.9%(2008년)
 - 고령인구 1인당 월평균 진료비 : 114,203원(2004년) → 235,305원(2010년)
- ICT 발달에 따른 보건·의료 정보화의 대상 및 서비스 확대
 - 2000년 이후 관계 부처 등은 단순 진료위주의 시범사업에서 ICT 발달에 따른 u-health 기반의 서비스 모델 발굴 등을 보건·의료 지원 사업으로 추진·확대
 - 원격진료 중심의 e-health 및 u-health 사업은 취약계층을 대상으로 하는 응급·안전관리, 건강증진 등 다양한 서비스 모델 발굴에 중점
- 신기술 융합·확산 기반의 신개념 보건·의료 정보화가 부상
 - ICT 등 신기술 융합·확산을 통한 보건·의료 정보화로 치료→예방, 병원→가정, 질병

관리→건강관리로 패러다임이 전환됨에 따라 웰니스 등 신개념 보건의료서비스 제공

- 새로운 보건의료 분야는 대상자 맞춤형의 원격의료(u-Medical), 건강서비스산업(u-Wellness), 고령친화산업(u-Silver) 등으로 구분
- 보건의료 디바이스 분야의 국내 기술력은 선진국 대비 60~70% 수준이며, ICT융합 기반 제품의 성장잠재력은 우수
 - 2013년 기준 우리나라는 OECD 국가 중 의료시설·장비 2위, 의료서비스 4위, 보건서비스 만족도 1위 등으로 각 부문에서 세계적 수준
- 정부에서도 보건의료서비스 패러다임 변화와 ICT 융복합 확산을 지원하기 위한 법제도 개선, 정책 확대 등을 추진
 - 의료기관 정보화 : 국내 의료기관 내·의료기관 간 업무 전산화, 전자의무기록·처방시스템 고도화 등으로 서비스 제공 기반 마련
 - 서비스 고도화 : 의사-환자간 원격의료 허용 기반 구축, 건강 생활서비스 시범사업 추진, 의료정보화 인프라 고도화 등 보건의료 서비스 경쟁력 제고를 위한 정책 지원
 - 법제도 개선 : 전자의무 기록, 전자처방 및 원격진료 등에 대한 법제화 추진

○ 국외 현황

추진방향	주요국의 추진전략 및 내용
건강정보 활용체계 구축	▶ 미국 : 의미 있는 정보 활용(Meaningful Use) 프로그램 - (목적) 의료 품질·안전성·효율성 증진을 위한 EHR 활용 - (내용) CMS와 ONC가 인증한 EHR 도입·활용 의료공급자에 인센티브 제공 - (기대효과) 우수한 의료정보 수집 기반 마련
	▶ 영국 : Care.data 프로그램 - (목적) 개인의 건강정보들을 수집·활용하여 효율적 질병예방 및 효과적 치료를 가능케 하여 의료의 안정성 및 품질 제고 - (내용) 의료인의 의료행위로 발생한 개인의료기록을 국가서버에 저장, 익명화하여 국가 보건의료 통계 및 임상자료 등으로 활용 - (기대효과) 국가적 보건의료 데이터의 효과적·효율적·경제적 활용을 통한 의료품질 제고 및 고도화된 보건의료 통계 제공
미래수요 맞춤서비스	▶ EU : AAL JP 공동 프로그램 - (목적) 고령자 수요에 대응하는 맞춤형 융합의료서비스 제공

발굴	- (내용) EU회원국 대다수와 민간기법의 긴밀한 협업 아래, 실제 수요 조사 등을 통한 수요반응형 신규서비스 발굴 및 확산 촉진 - (기대효과) 중소기업의 새로운 시장경쟁력을 강화하고, 다양한 협업관계를 통한 의료서비스 산업생태계 조성
	▶ 일본 : 어디든 My 병원 - (목적) 의료정보 수집·관리 일원화로 의료 효율화 및 국민 자가 건강 관리 의식 증대 - (기대효과) 의료정보 관리 주체 일원화, 정보 공유로 신산업 창출 기대

○ EU: AAL JP(Ambient Assisted Living Joint Program)

- 고령인구 증가로 유럽연합(EU) 회원국들의 노인연금, 의료비 등 정부재정 지출이 가중되며, 의료의 질이 떨어질 위험 도래
 - 2060년 유럽인구의 30%가 65세 이상의 고령자가 될 것이며, 은퇴한 사람 한 명당 근로자 수는 2명으로 하락할 전망
- 고령자의 새로운 수요가 늘어나는 데 반해, 전문 의료진 및 간호인이 부족해지면서 효과적인 의료서비스 제공에 한계
- EU는 고령자 중심의 새로운 보건·의료 패러다임 전환과 신기술 발달, ICT 융합·확산에 따른 고령자의 건강한 삶의 질 향상을 위해 유럽 내 공동 프로젝트를 추진
 - AAL JP는 지속가능한 보건·의료시스템에 대한 수요와 혁신적인 신규 상품·서비스 개발을 촉진시키며, EU의 미래 경제성장을 견인할 것으로 기대
 - 원격의료의 도입은 보건·의료서비스 효율성을 20% 이상 높이고, 고령 환자가 삶의 주체로서 독립적인 삶의 질 향상에 기여할 것으로 예측
- EU 회원국은 AAL JP를 통해 Well-aging 관련 혁신적인 ICT 서비스 응용연구 및 시범사업을 시장 출시 후 1~3년간 지
- EU는 AAL JP 추진으로 인구고령화의 사회적·기술적 해결과 ‘전방위 지원이 가능한 삶(Ambient Assisted Living)’ 과 관련된 상품·서비스의 유럽시장 경쟁력 향상을 도모

<HELP : 파킨슨병 환자들의 수요맞춤형으로 고안된 건강 모니터링 시스템>

- HELP는 2009년~2012년까지 약 500만 유로가 지원된 프로젝트로 주변 도움없이 스스로 자신을 돌보는 것이 어려운 파킨슨병 환자 지원을 위해 개발
- HELP 시스템은 원격지에서 임상전문가가 환자의 상태를 모니터링하고 통제할 수 있도록 지원
- 환자 신체에 부착된 웨어러블 디바이스 센서를 통한 건강정보(혈압, 움직임, 생체신호 등)를 가정용 디바이스로 전달, HELP 시스템을 통한 최종 송신
- 의료진은 이러한 정보를 실시간으로 확인하고, 적절한 약물투여량 등을 자동으로 조절 가능

<SOFTCARE : (잠재적)위험상황 시 발생 가능한 문제상황을 실시간 전달>

- SOFTCARE는 2009년~2012년까지 약 600만 유로가 지원된 프로젝트로 고령자의 위험상황을 인식하여 알림을 제공하고, 수집된 데이터를 분석하여 미래 문제에 대한 장기적 추세 정보를 제공
- SOFTCARE는 웨어러블 디바이스와 위치기반 정보에서 얻은 데이터를 기반으로 일상 활동을 인지·분석하는데 인공지능을 적용
- 3D 기반의 손목형 디바이스와 사물간 통신이 가능한 Zigbee 통신을 활용

○ 실내 활동 모니터링 관련 대표기술현황

연구 수행 기관	연구 개발의 내용	연구 개발 성과의 활용성
National Cheng Kung University	카메라를 이용해 간호센터에서의 고령자들의 이상 행위 모니터링	정확도 90%, 오경보율 0%
University of Missouri	비디오 카메라를 이용하여 실내에 있는 고령자의 행동을 모니터링	실루엣 기반으로 사생활 침해 정도를 줄임, 정확도 99.9%
Tohoku University	적외선 센서, 문열림 감지 센서, 워터벨브 센서, 전력계를 이용하여 생활 패턴 모니터링	거주자의 일상생활 패턴 특성들을 타입을 나누어 분류
Grenoble University Hospital	적외선 센서, 전력계를 이용한 실내 고령자 생활 패턴 모니터링	알츠하이머 질환자와 일반 고령자의 생활 패턴 구분 가능

서울대학교	적외선 센서를 이용하여 고령자들의 이상 행동 패턴 검출	정확도 최대 95.8%
University of Limerick	가속도 센서를 이용하여 일상 동작 및 낙상 검출	몸통과 허벅지에 센서 부착, 총 8개의 일상 동작 정의, 정확도 100%
건국대학교	허리에 차는 가속도 센서를 이용하여 동작 및 낙상 검출	낙상, 앉기, 서기, 눕기, 걷기, 뛰기 등 정확도 96.1%

○ 기존 기술의 한계점

- 단순 가속도 센서나 전력계를 이용한 모니터링으로 해당 정보를 토대로 하여 현재 수집 대상이 어떠한 상태인지를 확인할 수 있는 상황 인지가 떨어져 유사행동이나 상황에 대한 정확한 판단이 어려움
- 행동 인식 데이터를 이용한 응용서비스 부재(단순한 행동 패턴 분류)
- 활동 모니터링 기술에 관해 적외선센서, 영상카메라, 가속도 센서 각각을 이용한 연구는 다양하게 진행된 바 있지만, 이 정보들을 통합적으로 활용하여 정확한 상태의 인지 및 패턴 분류에 관한 연구는 아직 미비한 수준임

○ 본 연구의 차별성

- 생체신호 측정과 행동 영상을 결합하여 다양한 행동 인식뿐만 아니라 현재 정보를 수집하는 대상의 상황에 대해서 인지할 수 있는 방안 제시
- 동시에 활동량 및 운동량 측정과 건강관리 서비스에 응용할 수 있도록 다양한 측정 지표를 제공
- 생체신호와 결합한 복합 행동을 인식하여 개발을 통해 질병예방이나 사고 예방에 활용 가능한 시스템 개발

○ 기술개발 성숙도

- 단순 영상카메라를 이용해 거주자의 행태를 모니터링 하는 시스템은 이미 개발되었지만, 다양한 멀티센서와 결합하여 필요한 정보를 정확히 구분해내는 연구는 시도되지 않고 있음
- 특히, 데이터의 학습을 통해 자기 강화 알고리즘과 연계하고 솔루션을 제공하는 기술은 없음
- 최종 개발될 시스템(소프트웨어)와 이를 차이점을 고려하여 실험실 환경에서 개발된 기술 또는 모듈의 테스트 결과를 분석

구분		연구 착수	연구 종료	단계별 설명	
기초연구 단계	TRL 1	☐	☐	TRL 1	이론적 연구단계
	TRL 2	■	☐	TRL 2	기술적 개념정립 단계
실험 단계	TRL 3	☐	☐	TRL 3	실험적 증명 단계
	TRL 4	☐	■	TRL 4	실험실 성능평가 단계
시작품 단계	TRL 5	☐	☐	TRL 5	유사환경 시험 단계
	TRL 6	☐	☐	TRL 6	유사환경 기술시험 단계
실용화 단계	TRL 7	☐	☐	TRL 7	실제환경 시작품 데모 단계
	TRL 8	☐	☐	TRL 8	시제품 설치 및 성능시험 단계
사업화 단계	TRL 9	☐	☐	TRL 9	현장적용/사업화 단계

<표 1-1> 선행 연구

대분류	소분류	연구자 (연구년도)	장점	단점
접촉식 모니터링	착용형 센싱 모니터링	조위덕 외 (2015)	멀티 센싱 기술을 이용하여 개인의 삶에 최적화된 신뢰성 높은 시스템을 제안	상대적으로 많은 센서들을 이용하므로 비용이 큼
비접촉식 모니터링	설치형 센싱 모니터링	최경선 외 (2016)	전화 유선망을 사용하여 인터넷 기반이 취약한 고령자가 쉽게 사용할 수 있음	유선망의 성능의 정보 전달 속도가 느려 위급상황의 정확하고 빠른 판단이 어려움
	영상 모니터링	김상수 외 (2014)	비디오 카메라를 이용하여 목표대상의 낙상을 빠르게 파악 가능	- 낙상과 움직임의 경우만 판단이 가능함 - 가정 내의 물건의 위치 및 배경에 따라 움직임 벡터 값을 추출하지 못함
		Bian. Z 외 (2015)	- 모니터링 대상의 관절 부위를 추출하여 행동 변화 정도를 측정 가능 - 적외선을 이용해 어두운 환경에서 모니터링이 가능함	신체와 바닥사이의 거리가 멀거나 가구위에 누워있는 경우 낙상을 판단할 수 없음

○ 선행 연구

- 기술적 개념정립 단계인 TRL 2에 해당하는 선행연구는 <표 1-1>과 같음
- 실내 모니터링 시스템에 관한 선행 연구가 활발하게 진행되고 있음
- 실내 모니터링 시스템은 크게 접촉식 모니터링 시스템과 비접촉식 모니터링 시스템으로 나누어짐
- 접촉식과 비접촉식 모니터링 시스템 모두 가격이 비싼 센서를 사용하거나 생활에 있어 불편함을 주는 등 상용화에 한계점이 존재함
- TRL 2 단계에서 이미 선행 연구가 존재하지만 실제 주거 환경에 적용하기에 무리가 있어 상용화 되지 못했음
- 따라서 본 연구에서는 TRL 4 단계에서 실제 적용 가능한 모니터링 시스템을 구축하는 것을 목표로 함

1-3 연구개발의 중요성

○ 보건의료 수요의 변화

- 고령화로 인한 막대한 보건의료 비용의 증가
 - **고령화로 의료비 지속적 증가** : 고령화로 매년 9%씩 국가 전체 의료비 증가
 - 보건복지 예산은 정부예산의 10.8%로 정부예산 증가율 5.9%의 약 2.5배

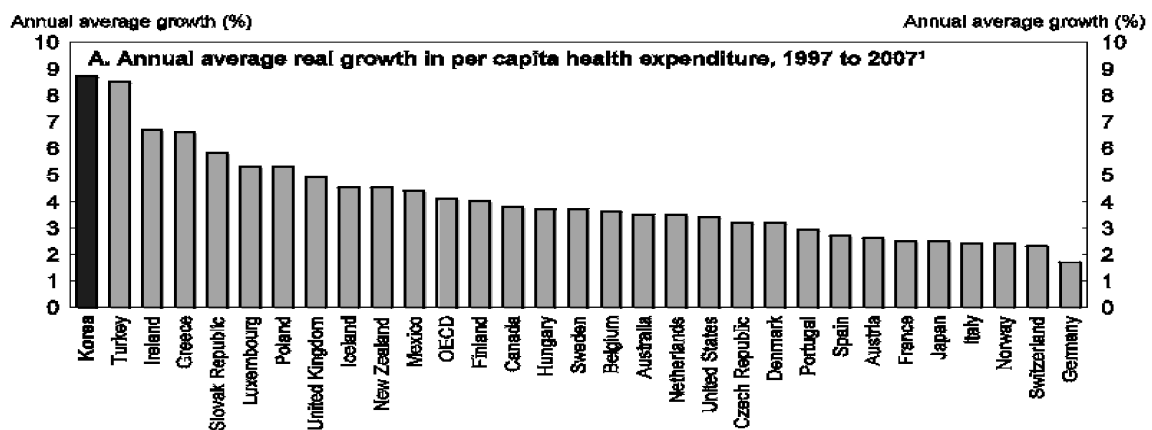


그림 2. 인당 진료비 지출 증가율 국제비교('97-'07년, OECD, 2010)>

- 고령화 및 핵가족화에 의한 독거노인수의 기하급수적 증가
 - **향후 10년간 급격한 고령화 전망** : 노인 인구 비율 : 7.2%('00) => 12.2%
 - 노인인구의 빠른 증가에 비해 노인인구를 도와줄 수 있는 인적, 물적 자원 부족
- 노인뿐만 아니라, 장애인을 돌봐줄 인력이 턱없이 부족 (상대적 소외계층을 위한 안전체계 구축 필요)

○ 실내 활동 모니터링 및 의도예측 기술의 필요

- 독거노인이나 장애인들과 같은 소외계층의 의료사각지대
 - 독거노인이나 장애인들의 응급 상황에 처한 경우 활동 모니터링이 되지 않아 혼자 사망하는 경우가 빈번히 발생
 - 노인의 실내 활동을 지속적으로 관찰하여 신체적, 정신적으로 건강하지 않아 나타나는 이상행동을 파악하는 서비스 필요
 - 무자각적인 기술을 기반으로 노인들의 일상생활을 자연스럽게 수집하는 기술이 필요
- 가속화된 기술발전으로 인한 대용량 데이터의 확보
 - 이를 정량화하고 분석하는 기술 필요
 - 상황 인지 및 판단을 위한 인공지능 기술개발의 확산

○ 개발 기술의 사회·경제적, 기술적, 정책적 타당성

- 사회 · 경제적인 타당성
 - **양극화된 노령화** 인구의 케어비용을 개선하여 **사회적 의료비용 절감**을 위한 효과적 수단
 - 오픈소스 하드웨어와 플랫폼을 이용한 제작비용의 절감으로 인해 소외계층이나 빈곤노인층 서비스 확산
- 기술적인 타당성
 - 기술의 발달로 인한 고성능 센서들의 저렴화와 오픈소스 하드웨어의 등장으로 쉽고 빠르게 기술개발이 가능해짐에 따라 실제 개발기술의 적용이 용이해짐
 - 단순 센서를 통한 측정보다는 다양한 센서로부터 수집되는 빅데이터의 학습을 통한 자기 강화 알고리즘 베이스의 콘텐츠 제공 기술적 이슈로 떠오르고 있음

- 정책적인 타당성

- 정부의 140대 국정과제와 #9, #48, #49 과도 부합

#9 보건 고령친화산업을 미래 성장산업으로 육성

#48 의료 보장성 강화 및 지속가능성 제고

#49 건강의 질을 높이는 보건의료서비스체계 구축

2. 연구수행내용 및 성과

2-1 연구개발 목표

- 오픈소스 하드웨어 기반 독거노인 및 장애인 실내 환경 모니터링 시스템 개발 및 클라우드를 통한 지속적 자가 학습 딥러닝 시스템 구현
 - ① 오픈 하드웨어 기반 환자 자세 분석 및 노인 활동 패턴 분석 알고리즘 개발
 - ② 가속도 센서 및 심박측정 센서를 통한 움직임 모니터링 기술 개발
 - ③ 딥러닝을 활용한 CNN(Convolution Neural Network)기반 정확도 향상 알고리즘 및 클라우드 솔루션의 구축
-

2-2. 연구개발 성과

- 논문 게재 성과
 - SCIE 논문 출판 1
 - 제목: A study on a fall detection monitoring system for falling elderly using open source hardware
 - 학술지명: Multimedia Tools and Applications (SCIE), 2017년 12월 출간, online first
 - 저자명: 최승현, 염세경
 - SCIE 논문 출판 2
 - 제목: Development of a methodology to predict and monitor emergency situations of the elderly based on object detection, Multimedia Tools and Applications,
 - 학술지명: Multimedia Tools and Applications (SCIE), 2018년 9월 출간, online first
 - 저자명: 강용신, 최승현, 김창균, 염세경
- 특허출원 1
 - 출원명: 디지털 영상 이미지를 이용한 위급상황 분석장치 및 방법
 - 출원번호:10-2018-0029180 , 2018년 , 대한민국
 - 발명자: 최승현, 염세경
- 특허출원 2
 - 출원명: 객체 인식을 통해 위급상황을 판단하는 장치 및 방법
 - 출원번호:10-2018-0106432 , 2018년, 대한민국
 - 발명자: 염세경, 최승현, 김창균, 원진영



A study on a fall detection monitoring system for falling elderly using open source hardware

Seunghyun Choi¹ · Sekyoung Youm^{1,2}

Received: 30 June 2017 / Revised: 2 November 2017 / Accepted: 21 November 2017
© Springer Science+Business Media, LLC, part of Springer Nature 2017

Abstract Due to the rapid increase in the elderly population and single-person households, it is absolutely necessary to observe the indoor activities of marginalized people, such as the elderly or the disabled, to detect abnormal conditions. At this time, contact sensors are have limited applicability due to their continuous wear, so there is a need to develop technology to collect daily life activities of the elderly. In this study, we tried to track the elderly living alone using images obtained from an open source camera. The system captures images by linking a low-cost camera module to open source hardware, and an algorithm is implemented to detect falls. This system can be used to reduce the cost of social care by improving the cost of care in the aging population, and this will contribute to promoting and developing management prescription guidelines through risk detection.

Keywords Fall detection · Elder · Open source hardware · Monitoring

1 Introduction

In recent years, the life expectancy has increased due to advances in medical technologies and changes in eating habits led by economic development. According to the UN's definition, a country with senior citizens aged over 65 that account for more than 7% of the total population is called an aging society. In addition, if the senior citizens are more than 14%, it called an aged society, and if the elderly population takes up more than 21%, it is called super-aged society [18]. In South Korea, the number of households of single elderly citizens is 7.1% of the total households in 2014, and it is predicted to reach 15.4% by 2035, which is double. One of the problems caused by an increase in elderly people is that the number of single elderly

✉ Sekyoung Youm
sekyoungyoum@gmail.com

¹ Department of Industrial and System Engineering, Dongguk University, Seoul, South Korea

² Department of Industrial and Systems Engineering, Dongguk University, 82-1, Pil-dong, Chung-gu, Seoul, South Korea

Published online: 15 December 2017



Development of a methodology to predict and monitor emergency situations of the elderly based on object detection

Sekyoung Youm¹ · Changgyun Kim¹ · Seunghyun Choi¹ · Yong-Shin Kang²

Received: 21 May 2018 / Revised: 10 August 2018 / Accepted: 5 September 2018
Published online: 14 September 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Because on the increase in the number of the elderly living alone and accidents occurring to them, the demand for a monitoring system capable of supporting fast response in case of an emergency situation by monitoring their everyday life in their residential spaces has been increasing. A framework and a system are presented to monitor the emergency situations of the elderly living alone using a low-cost device and open-source software. First, human pose recognition and emergency situations according to the pose change were defined using object recognition, and a procedure capable of detecting such situations was proposed. In addition, a pose recognition model was created using the TensorFlow Object Detection application programming interface (API) of Google to implement the procedure. Using a data preprocessing process and the created model, a system capable of detecting emergency situations and sounding an alarm was implemented. To verify the proposed system, the pose recognition success rate was examined, and an experiment on emergency situation recognition was performed while the angle and distance of the camera were varied in a setup similar to the residential environment. It is expected that the proposed framework for the emergency notification system for the elderly will be utilized for the analysis of various behavior patterns, such as the sudden abnormal behavior of the elderly, people with disabilities, and children.

Keywords TensorFlow · Poserecognition · The elderly · Emergency situation recognition · Object-detection

1 Introduction

According to an Organisation for Economic Co-operation and Development (OECD) demographic survey, the population proportion of the elderly will double within the next few

✉ Yong-Shin Kang
yskang7867@skku.edu

¹ Department of Industrial and System Engineering, Dongguk University, Seoul, South Korea

² Department of Systems Management Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangnan-gu, Suwon, Gyeonggi-do 16419, South Korea



논문 1

출원번호통지서

출원 일자 2018.03.13
특기사항 심사청구(무) 공개신청(무) 참조번호(NPF31684)
출원번호 10-2018-0029180 (접수번호 1-1-2018-0251129-25)
출원인명칭 동국대학교 산학협력단(2-2006-008178-3)
대리인성명 특허법인이지(9-2008-100041-4)
발명자성명 염세경 최승현
발명의명칭 디지털 영상 이미지를 이용한 위급상황 분석장치 및 방법

특 허 청 장

특허출원 1

논문 2

출원번호통지서

출원 일자 2018.09.06
특기사항 심사청구(유) 공개신청(무) 참조번호(NPF32284)
출원번호 10-2018-0106432 (접수번호 1-1-2018-0886801-15)
출원인명칭 동국대학교 산학협력단(2-2006-008178-3)
대리인성명 특허법인이지(9-2008-100041-4)
발명자성명 염세경 최승현 김창균 원진영
발명의명칭 객체 인식을 통해 위급 상황을 판단하는 장치 및 방법

특 허 청 장

특허출원 2

2-3. 연구 결과

(가) 1차 연도 연구개발 목표 및 결과

구분 (연도)	세부 과제명	세부 연구 목표	연구개발 수행 내용	연구 결과
1차 연도 (2016)	오픈 하드웨어 기반 환자 자세 분석 및 노인 활동 패턴 분석 알고리즘의 개발	* 저가형 카메라 모듈 다수를 오픈소스 하드웨어에 연동하여 효율적으로 다수의 카메라를 연동하는 시스템의 구현	* 시스템은 영상 처리 및 네트워크 연결을 위해 운영 체제를 탑제하여 서버 역할을 할 수 있는 라즈베리파이를 사용함	* 오픈소스 카메라를 이용한 모니터링 시스템 설계 및 구현 * 연구결과: A1 참조
		* Bayes Classifier 및 기존의 이미지 분석 알고리즘에 기반을 둔 이미지 인식 기법의 구현	* 낙상 검출 알고리즘 설계 및 개발 * 자세 분석 시스템 설계	* Statics/Dynamic 활동 구분 * 기본 동작의 세분화 인식 구분 * 위급상황(낙상) 인식 * 연구결과: A2 참조
	가속도 센서 및 심박측정 센서를 통한 움직임 모니터링 기술	* 오픈소스 하드웨어에 기반을 둔 포터블 환자 패턴 분석용 모니터링 시스템의 개발 및 노인 활동 패턴 분석 알고리즘과의 결합	* 라즈베리파이의 센싱 모듈 및 웨어러블 디바이스를 통한 생체 데이터를 수집 * 수집된 데이터를 영상 데이터와 연동한 분석 알고리즘을 구현	* 개별 라즈베리파이 분석 시스템 모듈 개발 * 리눅스 서버 개발 * 연구결과: A3 참조
		* 가속도 센서 및 심박측정 센서를 부착한 오픈소스 하드웨어 기술의 개발 구현	* 가속도 센서를 활용한 낙상 검출 검증 시스템	* 카메라 영상 이미지와 연동한 바이오 센서의 낙상 상태 검출 * 연구결과: A4 참조

○ 연구결과 A1:모니터링 시스템 설계

- 오픈소스 하드웨어는 기술에 대한 특허 라이선스가 없고 제품 개발에 필요한 리소스가 공개 되어 있어 제품 개발에 용이함
- 완성형, 표준형 제품에 비해 가격이 저렴하며 형태 변경을 통해 새로운 형태의 커넥티드 기기 개발 가능함

- 센서 제어나 조작에 용이하며 저렴한 가격으로 센서를 구할 수 있음
- 오픈소스 하드웨어는 작은 사이즈로 인하여 포터블 모니터링 시스템 제작 가능함
- 본 모니터링 시스템은 영상 처리 및 네트워크 연결을 위해 운영 체제를 탑재하여 서버 역할을 할 수 있는 라즈베리파이를 사용함
- 라즈베리파이는 다른 오픈소스 하드웨어들 보다 속도가 빠르고 리눅스 운영체제를 기반으로 하여 커스터마이징에 유리함
- 기존 웹캠 보다 저렴한 가격으로 구할 수 있는 라즈베리파이 전용 카메라 모듈을 사용하여 비용 절감됨

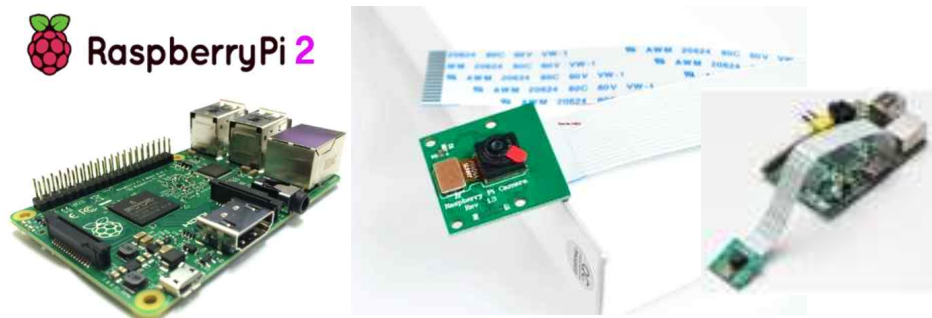


그림 3. 라즈베리파이 및 오픈소스 하드웨어용 일반 카메라 모듈

- 독거노인 모니터링을 위해 주거공간을 라즈베리파이와 카메라 모듈을 사용하여 촬영, 촬영된 영상을 통해 응급 상황 및 이상 탐지

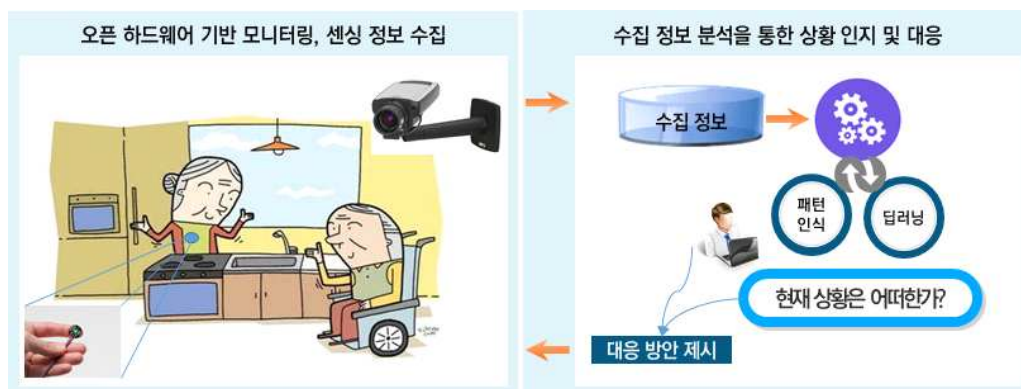


그림 4. 독거노인의 생활패턴 및 이상상황 감지 시스템

○ 연구결과 A2

(1) 낙상 검출 알고리즘 설계 및 개발

- 고령화로 인한 독거노인의 증가와 만성질환으로 인한 응급상황 검출이 필요함
- 대부분의 기존 연구는 1~2개의 센서를 이용하여 모니터링 대상이 센서를 착용해야 한다는 불편함이 있음
- 모니터링 대상이 항상 웨어러블 기기를 착용해야 한다는 기존의 연구의 한계점을 보완하기 위해 영상 속에서 낙상을 검출하는 알고리즘 구현 필요함
- 영상 속에서 모니터링 대상을 검출하기 위해 움직이는 물체를 검출함

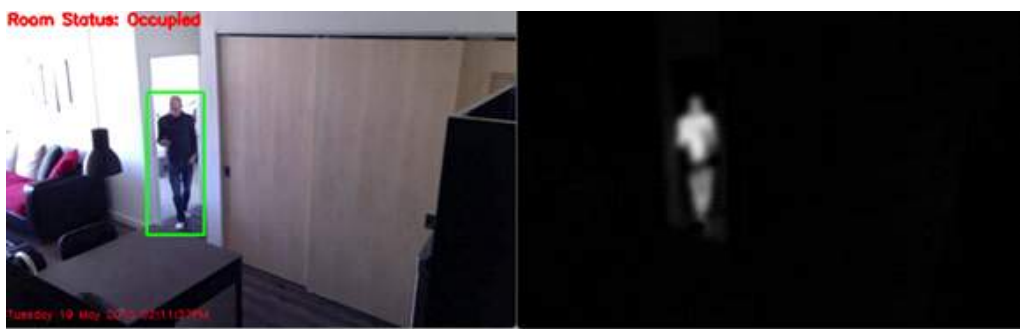


그림 5. 움직이는 이미지 검출 예시

- opencv의 threshold 함수를 사용하여 정적인 배경과 움직이는 물체를 분리하여 모니터링 대상을 추적, 움직이는 물체를 모니터링 대상이라 간주하고 영상 이진화를 통해 대상을 검출하여 네모 상자를 통해 추적함
- opencv의 contour함수를 사용하여 모니터링 대상의 검출 후 외곽선을 추출한 후 boundingRect 함수를 사용하여 영상 속 모니터링 대상의 가로, 세로 길이를 구함
- 모니터링 대상의 낙상을 검출하기 위하여 대상의 기울기의 변화를 사용, 영상 속 모니터링 대상의 가로와 세로의 길이를 구하여 대상을 추적하고 있는 상자의 기울기를 구한 후 저장
- 대상의 기울기의 변화량이 일정 수준 이상이 되면 모니터링 대상의 응급상황이라 판단함
- 기울기의 변화량을 이용할 경우 자기 위해 눕는 것과 쓰러지는 것의 차이를 구분하지 못하는 경우가 발생, 따라서 기울기의 변화량을 이용하기 위하여 시간의 개념을 추가하여 응급상황을 검출하는데 있어 정확도를 높임
- 노인의 경우 움직임의 빠르기가 일반 사람에 비해 현저하게 떨어짐 하지만 응급상황이 발생하여 낙상하는 경우 쓰러짐의 속도가 매우 빠르다는 것을 착안하여 기울기의 변화량이 매우 클 경우 시간을 측정하여 응급상황을 판단함

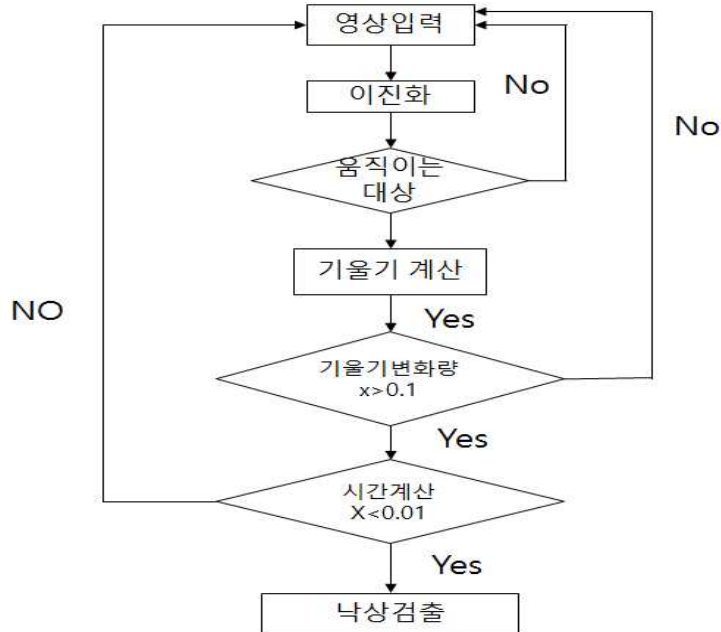


그림 6. 움직이는 이미지 검출 예시

(2) 자세 분석 시스템 설계

- 모니터링 영상 속 자세를 분석하기 위해 각 프레임의 이미지를 분석하여 모니터링 대상의 자세를 분석함
- 기존 목표였던 bayes classifier를 사용하는 대신 이미지 딥러닝 CNN 알고리즘을 기반으로 한 inception v3 module를 사용, inception v3 module는 Google사의 image classifier 모델로 정확도 약 97%를 가짐

※ Deeplearning model을 이용한 Image Classification방법의 강점(기존 방법과 비교하여)

- Convolution layer에서 Convolution feature들을 추출하여 Image의 특징들을 잡아줌
- Pooling layer에서 이미지의 많은 픽셀들의 feature들을 줄여서 분석을 용이하게 함
- Feedforward layer에서 Convolution layer와 pooling layer에서 나온 feature들을 이용해 Image를 분류 가능케 함
- 이러한 과정들을 통해 기존의 Image Classification 방법보다 더욱 Deep하게 학습을 하여 더욱 정확한 예측력을 가짐

- 서있는 자세, 누워있는 자세, 엎드려 있는 자세, 앉아있는 자세의 사진을 100장 이상 준비하여 inception v3 module를 재학습시킴
- 커스터마이징된 inception v3 module를 사용하여 영상 속 모니터링 대사의 자세를 분석

함

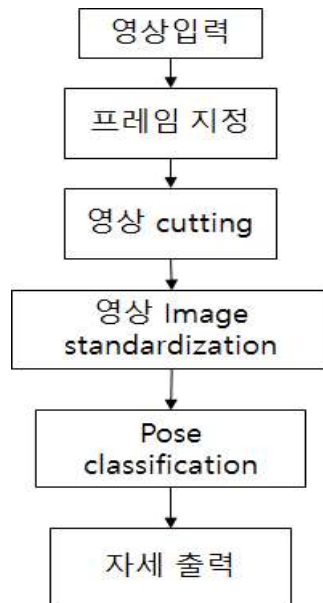


그림 7. 자세출력

○ 연구결과 A3:바이오 센서를 이용한 환자패턴 모니터링 설계

- 리벨리움사에서 제작한 e-Health kit을 사용, 저렴한 가격을 통해 구할 수 있으며 오픈소스 하드웨어로 IoT기반 의료기기 제작에 용의함
- 총 9가지 센서 중 산소포화도 센서, 호흡 센서, 온도 센서, 심전도 센서에 대한 개발을 진행함
- e-Health kit를 사용하여 건강모니터를 구현하고 이를 실시간으로 확인할 수 있는 웹페이지를 제작함

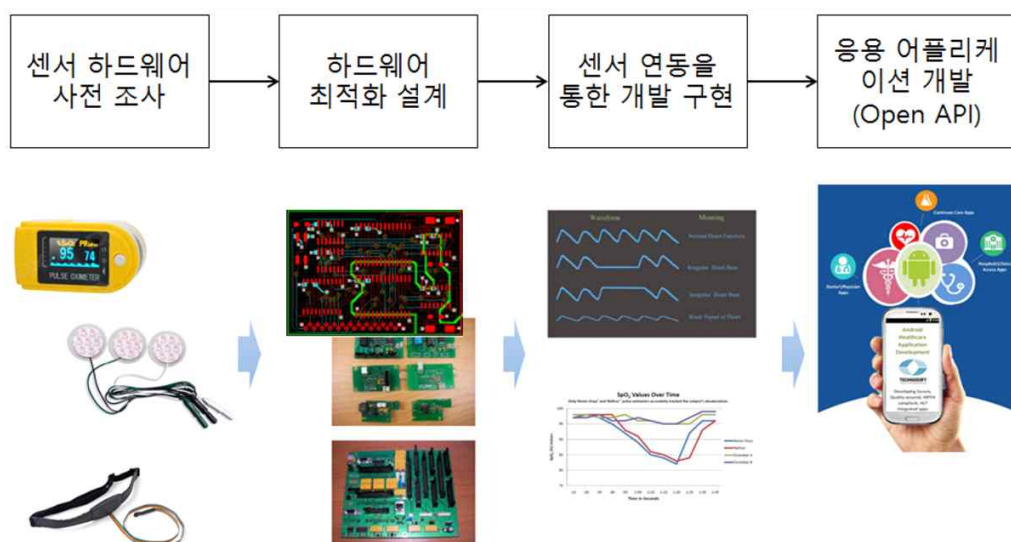


그림 8. 바이오센서 및 하드웨어 선정 및 API개발

- 하드웨어 최적화 설계
 - 오픈 소스 하드웨어에 대한 커스터마이징
 - 각각의 센서들을 결합했을 때 최적의 구성이 나올 수 있는 물리적인 디자인 제안
 - 각 센서에 대한 라이브러리 구성 개발
 - 개발된 라이브러리에 대한 API 모듈화
- 센서 연동을 통한 개발 구현
 - 각 센서 별로 데이터 취합 및 취합된 데이터에 기반을 둔 개별 응용 프로그램 개발
 - 취합된 데이터에 기반을 둔 응용 프로그램을 개발하는 것이 취지
- 응용 어플리케이션 개발
 - 클라우드 기반 데이터 콜렉션 및 콜렉팅 할 수 있는 의료 데이터 전송 툴 개발
 - 의료 표준을 가급적이면 준수하는 의료 데이터 표준에 따른 데이터 저장
 - 저장된 데이터의 클라우드화

센서종류	특징	실물
SP02 산소 포화도 센서	- 산소포화도 센서는 환자의 현재 상태를 검사하기 위한 가장 기초적인 감시장치로 착용이 간단하고 사용자의 안전이나 부작용에 대한 걱정 없이 사용할 수 있음 - 산소포화도 센서는 적외선 파장과 붉은 파장의 빛을 사용하여 혈중 산소 농도를 파악하기 때문에 산소포화도 센서에 달려 있는 LED의 세기를 잘 조절하여 의미 있는 신호를 받아야함 - LED의 세기를 잘 조절하여 의미 있는 신호를 받았음에도 잡음이 존재하기 때문에 이를 제거해야함	
EKG 심전도 센서	- 심전도 센서는 전기적 신호를 각 벡터별로 측정하여 심작박동의 비율과 일정함을 측정하는데 사용하며, 심장의 크기, 손상 부위 등을 판단하는데 사용함 3채널 심전도 센서의 경우 착용이 간편하고 단순한 알고리즘을 사용하여 24시간 환자 관찰에 사용하기 용이함 - 양쪽 가슴과 오른쪽 갈비뼈 아래에 센서를 착용함	
체온계	- 체온 센서는 접촉식 센서를 사용하고 있고 매우 저렴하지만 위급상황에 대한 1차 판단 등에 사용되는 기본적인 장비로 일정한 주기로 체온을 측정하여 생체 리듬을 파악할 수 있음 - 접촉식 센서는 가변 저항을 사용하여 전류-저항값을 측정해 체온을 측정함 - 체온의 소수점 이하 자리는 조정 가능함	
포지션 센서	- 포지션 센서는 환자의 특정 자세(누워있거나 엎드려 있거나 등등)에서 환자의 움직임을 관찰하는데 사용하는 센서 - 와상 환자나 장기요양(입원 요양) 서비스 대상자들의 낙상 예방 등에 활용가능 - 수면 진단 기능이 포함된 미밴드의 업그레이드(다운그레이드) 임	

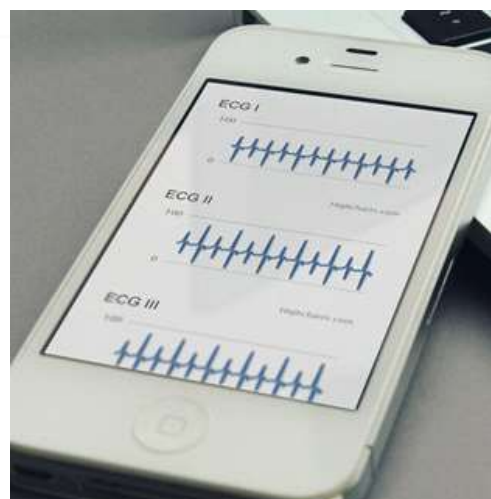


그림 9. 응용 어플리케이션 개발

○ 연구결과 A4:가속도 센서를 활용한 낙상 검출 검증 시스템

- 가속도 센서를 활용하여 낙상 검출 실험 진행



그림 10. 가속도 센서를 이용한 자세 실험

1	0	-	9	5	4	1	2	8	4	4	0	3	6	7	1	0	9	0	1	0	4	-	0	1	4	9	5	6	0	9	2	6	3	-	5	3			
1	0	-	9	0	7	5	6	3	0	2	5	2	1	1	1	9	-	0	1	0	8	-	0	1	4	9	5	6	0	9	2	6	4	-	0	9			
1	1	-	9	0	5	4	0	5	4	1	7	4	-	0	0	1	1	4	0	3	-	0	1	4	9	5	6	0	9	2	6	5	-	2	2	9			
1	1	-	6	7	1	0	5	2	6	3	1	5	8	7	6	-	0	0	1	2	0	3	-	0	1	4	9	5	6	0	9	2	6	5	-	7	7		
1	1	-	7	7	0	8	3	3	3	3	3	3	3	3	9	6	-	0	0	1	2	6	6	-	0	1	4	9	5	6	0	9	2	6	6	-	2	8	
1	1	-	2	1	8	4	8	7	3	9	4	9	6	1	1	9	-	0	0	1	2	6	4	-	0	1	1	4	9	5	6	0	9	2	6	6	-	7	
1	1	-	3	3	3	3	3	3	3	3	3	3	3	3	1	0	5	-	0	0	1	2	4	5	-	0	1	1	4	9	5	6	0	9	2	6	7	-	3
1	1	-	7	8	6	5	1	6	8	5	3	9	3	8	9	9	-	0	0	1	2	4	8	-	0	1	1	4	9	5	6	0	9	2	6	7	-	8	
1	1	-	1	5	4	4	7	1	5	4	4	7	2	1	2	3	-	0	0	1	2	6	5	-	0	1	1	4	9	5	6	0	9	2	6	8	-	3	
1	1	-	9	6	5	5	1	7	2	4	1	3	8	8	7	-	0	0	1	2	5	8	-	0	1	1	4	9	5	6	0	9	2	6	9	-	5		
1	1	-	4	8	9	1	3	0	4	3	4	7	8	9	2	-	0	0	1	2	2	9	-	0	1	1	4	9	5	6	0	9	2	7	0	-	0		
1	1	-	2	3	7	2	8	8	1	3	5	5	9	1	1	8	-	0	0	1	2	6	4	-	0	1	1	4	9	5	6	0	9	2	7	0	-	6	
1	3	-	0	4	7	6	1	9	0	4	7	6	2	8	4	-	0	0	1	2	5	6	-	0	1	1	4	9	5	6	0	9	2	7	1	-	2		
1	1	-	4	8	6	2	3	8	5	3	2	1	1	1	0	9	-	0	0	1	2	7	1	-	0	1	1	4	9	5	6	0	9	2	7	1	-	8	
1	1	-	7	2	7	2	7	2	7	2	7	2	7	8	8	-	0	0	1	2	4	0	-	0	1	1	1	4	9	5	6	0	9	2	7	2	-	3	
1	1	-	4	9	5	0	4	9	5	0	4	9	5	1	0	1	-	0	0	1	1	5	1	-	0	1	1	4	9	5	6	0	9	2	7	2	-	9	
1	1	-	0	1	4	5	9	8	5	4	0	1	5	1	3	7	-	0	0	1	1	3	9	-	0	1	1	4	9	5	6	0	9	2	7	3	-	4	
1	0	-	8	8	6	1	7	8	8	6	1	7	8	9	1	2	3	-	0	0	1	0	9	-	0	1	1	4	9	5	6	0	9	2	7	3	-	9	

그림 11. 낙상의 경우 데이터 변화 감지

- 가속도 센서를 이용하여 사람의 갑작스러운 행동 변화에 대한 속도 측정
- 그림 5의 빨간색 박스 친 속도데이터를 이용하여 속도가 급격히 변화하는 구간을 도출
- 기존 행동 패턴의 속도기준과 다르게 갑작스럽게 행동 속도의 기준치를 넘어서는 구간에 대해 낙상이라 판단함

(1) 1차 연도 연구 범위 및 연구 수행 방법

○ 오픈하드웨어 기반 움직임 패턴 분석 알고리즘 개발

연구 범위	연구 수행 방법 (이론적·실험적 접근방법)	구체적인 내용
오픈 하드웨어 기반의 환자 모니터링 영상 장치의 개발	* 5메가 픽셀 수준의 저렴한 일반 카메라 / 나이트비전 카메라 모듈을 집안 곳곳에 설치 * 설치된 다수의 카메라 모듈을 활용하여 환자가 어디에 있는지 확인	* 오픈소스하드웨어 및 카메라를 이용한 모니터링 시스템 구현 * 수행내용: B1 참조
Naive Bayes Classifier를 통한 노인 활동 패턴 인식 기법 개발	* Naive Bayes Classifier란 기계 학습 (Machine Learning) 분야에서 특성들 사이의 독립을 가정하는 베이즈 정리를 적용한 확률 분류기의 일종임 * 텍스트 분류 및 일반 영상의 이미지 분석에 사용됨으로써 이미지의 범주를 분류하는데 탁월한 효과가 있으며 대표적으로 개 vs 고양이, 환자의 자세 인식 등이 있음 * 자동 의료 진단 분야에서도 적절한 전처리를 통해 충분한 경쟁력을 보인 응용사례가 있으며 환자 동작 인식에서도 충분히 활용 가능할 것으로 보여짐	* Naive Bayes Classifier 대신 CNN을 이용하여 노인 활동에 대한 이미지 인식을 구현하였음 * 수행내용: B2 참조

○ 수행내용 B1: 오픈소스 하드웨어를 이용한 독거노인 모니터링 시스템

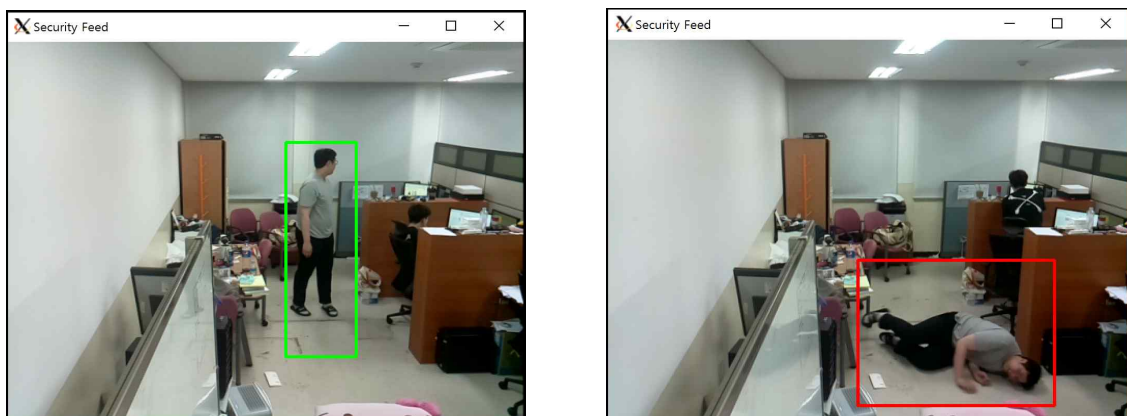


그림 12. 응급상황(낙상) 움직임 인식 시스템

- 독거노인 모니터링 시스템 개발 결과 낙상 발생 시 응급상황을 알리는 시스템 구축
- 초록색 박스로 신체의 범위를 잡아주어서 사람의 행동을 파악

- 갑작스럽게 사람이 쓰러지게 되면 위험을 알릴 수 있도록 초록색 박스의 색깔에서 빨간색 박스로 색깔이 변하게 되어 응급상황 알림
- 오픈소스 하드웨어인 라즈베리파이가 가지는 휴대성은 독거노인 모니터링 시스템을 원하는 장소에 구축할 수 있는 자유성 보장



그림 13. 낙상상황에서의 인식 데이터

※ 1차 년도까지 개발된 시스템의 한계점

- 독거노인 모니터링 시스템은 모니터링 대상의 움직임이 없을 경우 대상을 추적하는데 한계가 있어 모든 상황에서 응급상황을 검출하지 못함
- 추후 여러 상황의 응급상황을 검출하기 위하여 항상 모니터링 대상을 추적할 수 있는 알고리즘을 개발하는데 있어 딥러닝 활용 예정임
- inception v3 module를 사용하여 구축한 자세 분석 시스템의 경우 재학습된 inception v3 module의 정확도가 80%로 낮은 수준을 나타내어 추후 각 자세에 해당하는 이미지를 추가로 수집하여 정확도를 높여야 하는 문제점 존재함
- 이러한 문제점들을 해결한 후 모니터링 시스템과 자세 분석 시스템을 결합한다면 모니터링 대상의 응급상황을 판단하는데 있어 정확도가 크게 향상될 것으로 기대됨

○ 수행내용 B2: CNN을 이용하여 노인 활동에 대한 이미지 인식

(1) 딥러닝을 이용한 이미지 인식 방법론 선정

- CNN model의 layer 구성에 있어서 다양한 방법들이 존재하지만 layer 구성은 이미지의 픽셀 값과 data의 수에 따라 조절해야 되는 것이므로 확실화된 layer구성은 존재하지 않음 (따라서 data의 양과 구성에 따라 layer의 구성은 달라지게 됨)
- CNN layer 구성에서 초기에는 layer 층이 많아질수록 좋은 model을 가지고 예측력이 높다고 판단하였지만 적은 data에 많은 layer층을 쌓게 되면 overfitting이 일어나게 되어

좋은 정확도를 낼 수 없다고 판단

- 분석하고자 하는 data에 대한 정확한 이해가 필요하고 그에 대해 CNN layer을 구성하는 것이 가장 효과적이라고 판단

(2) 사람의 행동에 대한 분류방법을 deeplearning model중 Image 인식모델의 하나인 CNN(Convolution neural network)를 이용해 행동모델 분석

- CNN을 이용하여 사람의 행동에 대한 Image Classification 방법이용 사람의 행동에 대한 몇 가지의 label을 이용하여 행동 분석
- 초기CNN model을 이용하여 사람의 행동 패턴에 대한 label을 앓다, 높다, 서다 총 3가지로 분류
- 3가지의 label에 대해 총 100개의 Image 데이터를 이용하여 CNN의 Network 구축 CNN sample model 정의
- 사람의 행동에 대한 CNN sample model을 이용하여 정확도 예측

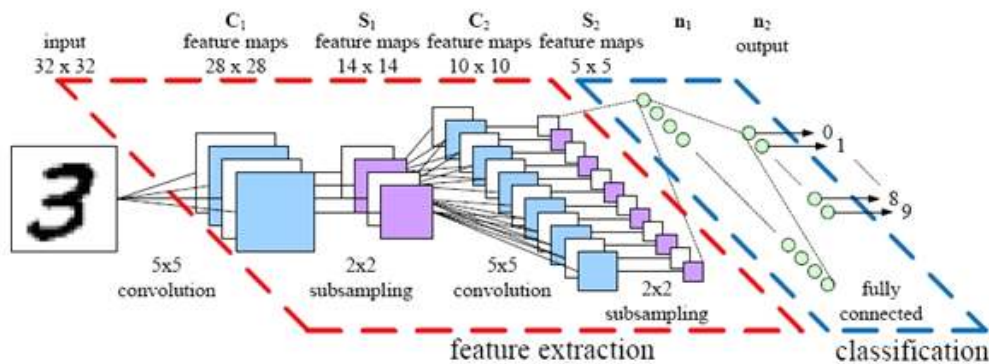


그림 14. CNN(Convolution neural network) 일반적인 구조

(3) CNN model 분석의 문제점과 이에 대한 해결방안

	문제점	해결방안
Image reshape	* Image reshape되지 않는 Image에 대한 overfitting * Image 색상에 대한 다양한 잡음이 존재	* 세 가지의 label에 대한 총 300개의 데이터를 128X128 구조로 reshape * 이미지의 색깔을 모두 gray로 바꾸어 이미지 인식에 대한 색깔의 잡음을 제거
CNN layer 정의	* 낮은 정확도로 인해 CNN Sample model의 layer정의 부족	* Image reshape를 통해 기존의 CNN Sample model 보다 높은 인식률 * fully connected layer의 구성단계에 대한 정확도 향상

- CNN을 이용하여 사람의 행동에 대한 Image Classification 방법이용 사람의 행동에 대한 몇 가지의 label을 이용하여 행동 분석
- 초기CNN model을 이용하여 사람의 행동 패턴에 대한 label을 서다, 앉다 총 2가지로 분류
- 2가지의 label에 대해 총 300개의 Image 데이터를 이용하여 CNN의 Network 구축 CNN sample model 정의
- 두 가지 행동에 대해 행동 동영상에 대해 초당 1장의 프레임으로 나누어서 총 9장의 이미지 test
- 서다[1,0] 앉다[0,1]로 임의로 label을 만들어주고 실험실시
- 첫 번째 실험은 128X128 이미지 데이터에 대해 color 수정을 하지 않고 다양한 색상의 이미지들이 그대로 들어가 있는 데이터를 이용하여 실험
- 두 개의 label을 이용해 실험을 하였으므로 5번째 칸의 사진은 앉음 이라고 임의로 지정하고 실험 실시
- 총 9개의 사진 중 4개의 사진을 CNN이 정확하게 인지

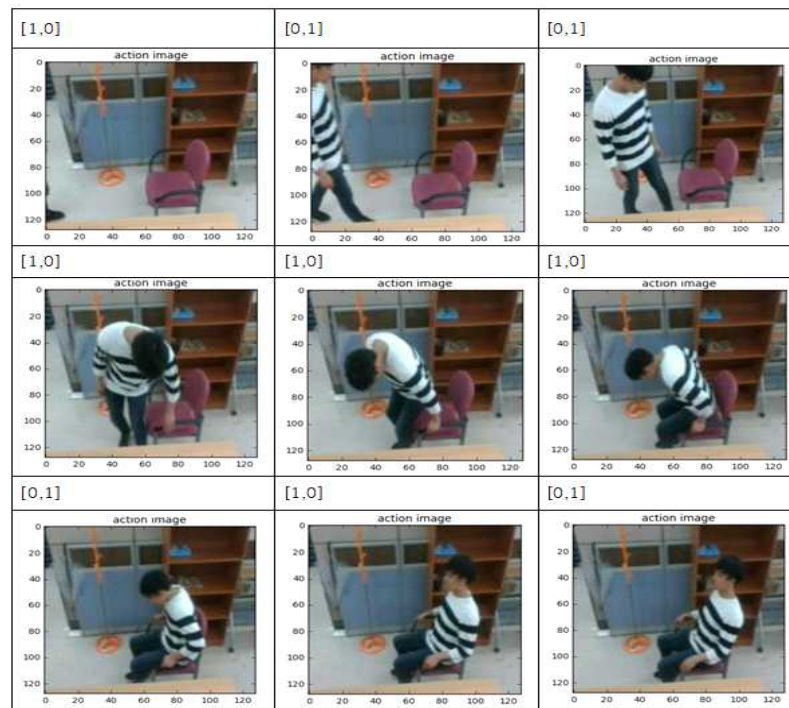


그림 15. 128X128 Color image classification

- 두 번째 실험은 128X128 이미지 데이터에 대해 color의 다양한 색상의 이미지들을 제외하고 사람에 대해 반응을 하도록 이미지를 임의 지정
- 두 개의 label을 이용해 실험을 하였으므로 5번째 칸의 사진은 앉음 이라고 임의로 지정하고 실험 실시

- 총 9개의 사진 중 6개의 사진을 CNN이 정확하게 인지
- 이미지 데이터에 대해서 불필요한 이미지 색상들을 제외시킨 것이 더욱더 높은 정확도를 가지는 것으로 판단
- 하지만 정확도가 낮게 나온 이유는 더욱 많은 데이터를 training시켜보지 못하였고 convolution layer에 대한 구성 및 이미지에 대한 많은 수정이 필요하다고 판단

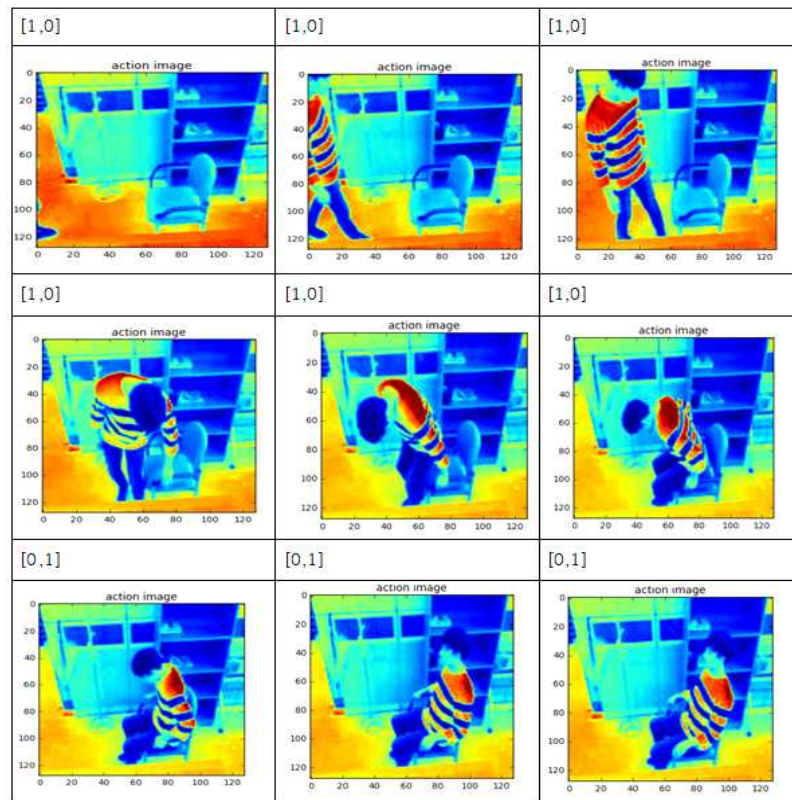


그림 16. 128X128 none color image classification

○ 센서를 이용한 움직임 모니터링

연구 범위	연구 수행 방법 (이론적·실험적 접근방법)	구체적인 내용
오픈소스 기반 웨어러블 센서의 연동 가능성 연구	* 활동 모니터링을 측정할 수 있는 오픈소스 기반의 웨어러블 센서들 중 2 ~ 3개 정도를 취사 선택하여 활동을 체크할 수 있는 웨어러블 디바이스를 개발하고자 함 * SPO2(산소포화도 센서), ECG(심전도 센서), EEG(뇌파 센서), 포지션 센서, 폐활량 센서, 혈당계, 혈압계, 운동량 센서, 피부 전도 센서 등 다양한 센서에 대한 사전 조사 필요 * 각 센서의 이론적인 활용 방안 및 결합 시의 시너지 조사	* 10개의 오픈소스 기반 웨어러블 센서 조사 및 24시간 모니터링에 필요한 기본적인 센서 선택 * 오픈소스 기반 산소포화도 센서, 심전도 센서, 온도 센서를 사용하여 노인의 일상생활 모니터링 가능 범위 및 효율성 연구
고령자의 활동 모니터링에 적합한 오픈소스 하드웨어 기반 웨어러블 솔루션 구축	* 센서 하드웨어 사전조사 * 하드웨어 최적화 설계 * 센서 연동을 통한 개발 구현 * 응용 어플리케이션 개발	* 오픈하드웨어와 오픈소스 기반 웨어러블 센서를 연동하여 저렴한 가격에 고령자의 신체 활동을 모니터링 * 오픈소스 기반 산소포화도 센서, 심전도 센서, 온도 센서를 사용하여 노인의 상태를 모니터링할 수 있는 시스템 개발
하드웨어 최적화 설계	* 오픈 소스 하드웨어에 대한 커스터마이징 * 센서들을 결합했을 때 최적의 구성이 나올 수 있는 물리적인 디자인 제안 * 각 센서에 대한 라이브러리 구성 개발 * 개발된 라이브러리에 대한 API 모듈화	* 오픈소스 하드웨어와 오픈소스 기반 웨어러블 센서를 결합하여 노인의 일상생활 전반에 걸친 모니터링 * 커스터마이징된 오픈소스 하드웨어를 통해 얻어지는 데이터를 통해 항상 모니터링 가능한 웹사이트 제작

생체 진단 신호에 기반을 둔 환자 패턴 분석 기법 개발	* 심박 변이율 및 가속 신호 특징에 근거한 Static/Dynamic 상태 분류 * Static 상태의 행위(서기, 눕기 등), Dynamic 상태의 행위(걷기, 뛰기, 계단오르기 등) 인식 * 오픈소스 하드웨어에 기반한 포터블 환자 패턴 분석용 모니터링 시스템의 개발 및 노인 활동 패턴 분석 알고리즘과의 결합	* 가속도 센서를 이용한 노인 낙상 검출 시스템 개발 * 가속도 센서를 통해 얻어진 값을 통해 오픈소스 하드웨어 기반 모니터링 시스템의 낙상 검출률 점검
센서 연동을 통한 개발 구현	* 각 센서 별 데이터 취합 및 취합된 데이터에 기반을 둔 개별 응용 프로그램 개발 * 취합된 데이터에 기반한 응용 프로그램을 개발하는 것이 취지	* 웨어러블 센서의 데이터 수집을 위한 리눅스 서버 구축 * 딥러닝 및 응용프로그램 제작을 위한 신체 리듬 데이터 베이스 구축
응용 어플리케이션 개발	* 클라우드 기반 데이터 콜렉션 및 콜렉팅 가능 의료 데이터 전송 룰 개발 * 의료 표준을 가급적이면 준수하는 의료 데이터 표준에 따른 데이터 저장	* 웨어러블 센서의 데이터 수집을 위한 리눅스 서버 구축 * 딥러닝 및 응용프로그램 제작을 위한 신체 리듬 데이터 베이스 구축

○ 수행내용: 분석 대상의 활동 모니터링에 적합한 웨어러블 및 하드웨어 솔루션 구축

- 현재까지의 연구에서는 서버 내에서 낙상을 구별하는데 한계가 있음
- 서버를 통한 분석은 시간이 많이 소요될 뿐 아니라 비용 면에서도 효율적이지 못함

(가) 개별 라즈베리파이 분석 시스템 모듈 개발

- 라즈베리파이는 소형 컴퓨터로써 라즈베리파이 내에서 분석 알고리즘을 통해 행동 패턴 분석 가능
- 서버를 거칠 필요 없이 라즈베리파이 내에서 행동 패턴을 인식, 위급상황에 보다 빠른 대처 가능
- 서버로부터 계속 학습된 분류 알고리즘을 자동으로 업데이트 하여 보다 정확한 패턴 인식 가능함
- 개개인별 행동 패턴을 분석하여 운동량 정보 및 생활 패턴을 분석한 측정 지표를 제공 하여 노약자의 건강관리를 도모
- 라즈베리파이를 이용해 심박 수, 체온, 심전도를 측정하여 신체정보를 수집하고 그에 따른 패턴을 인식하여 몸의 이상신호가 나타나게 되면 반응을 하여 빠른 대처 가능



그림 17. 라즈베리파이를 이용한 심박 수 측정

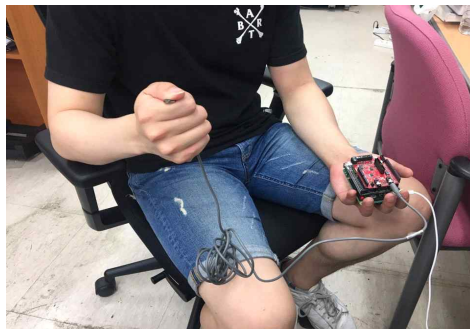


그림 18. 라즈베리파이를 이용한 체온 측정



그림 19. 라즈베리파이를 이용한 심전도 측정

(나) 리눅스 서버 개발

- 리눅스 서버의 역할은 영상을 수집하여, 해당 영상으로부터 노약자의 행동 패턴을 인식할 수 있도록 구현하였음
- 리눅스 서버에서는 라즈베리파이로부터 얻어진 영상 데이터와 센서데이터로부터 얻어진 데이터를 축적하고 축적된 데이터를 서버 내에서 Classifier하여 행동패턴 분석 알고리즘을 학습시키는 프로그램 개발에 대한 연구

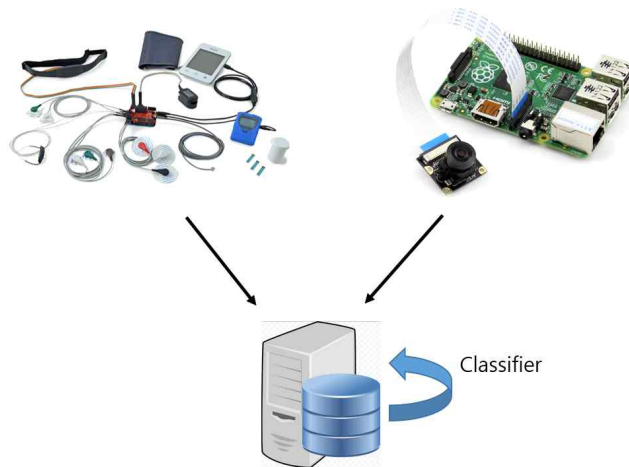


그림 20 . 리눅서 서버 개발

(다) 라즈베리파이 센서 연동 및 웨어러블 디바이스 개발

- 1차 년도의 영상을 통한 노약자의 행동패턴 인식기법 연동하여 사용될 라즈베리파이와 웨어러블 디바이스에 대한 조사 분석 완료 및 센서가 연동
- 여러 타입의 센서들은 배치가 랜덤하며, 센서간의 토폴로지를 예상하기 어려우므로 통합하는 시스템이 필요

(나) 2차 연도 연구개발 목표 및 결과

구분 (연도)	세부 과제명	세부 연구 목표	연구개발 수행 내용	연구 결과
2차 연도 (2017)	딥러닝을 활용한 CNN 기반의 정확도 향상 알고리즘 및 클라우드 솔루션의 구축	* 딥러닝을 활용한 CNN(Convolution Neural Network)의 구축 및 이를 통한 자기 강화 학습 알고리즘의 구축	* Fast R CNN 알고리즘을 사용한 자세 분석 모델 구축	* 자세분석 모델을 활용한 위급상황 (낙상) 검출 시스템 구축 * 연구결과: A5 참조
		* 클라우드 시스템의 구축을 통한 빅데이터 솔루션의 구현 및 이를 딥러닝에 활용	* 딥러닝 모델을 위한 이미지 전처리 자동화 * 딥러닝 모델을 위한 이미지 labeling 자동화	* 딥러닝 학습을 위한 이미지 수집 시스템 구축 * 딥러닝 학습을 위한 이미지 labeling 자동화 시스템 구축 * 연구결과: A6 참조

○ 연구결과 A5:자세 분석 모델 설계

- Fast R CNN은 이미지의 특징을 추출하고, 이미지의 노이즈를 최소화한 object detection을 위한 딥러닝 알고리즘임
- Convolution Neural Network 보다 더 높은 정확도를 가진다는 장점이 있음
- 또한 이미지 내에서 object 이외의 구간까지 함께 학습하여 객체의 특징을 정확히 인지할 수 있는 장점이 있음
- Region Proposal Network(RPN)개념을 추가하여 이전에 찾은 객체의 경계 영역 정보를 유지해 기존의 Convolution Neural Network보다 학습 시간을 획기적으로 단축하였음
- Fast R CNN은 Convolution feature map과 Region of interest from a proposal method(ROIs) feature vector로 이루어져 있음
- Feature를 입력 이미지로부터 뽑아내는 것이 아니라, CNN을 거친 Feature Map상에서 Spatial Pyramid Pooling의 특수한 형태인 RoI Pooling을 사용하여 Feature를 뽑아냄
- <그림 17>과 같은 구조를 통해 하나의 Region Proposal당 하나의 CNN이 돌아가면서 네트워크 구조가 획기적으로 개선됨

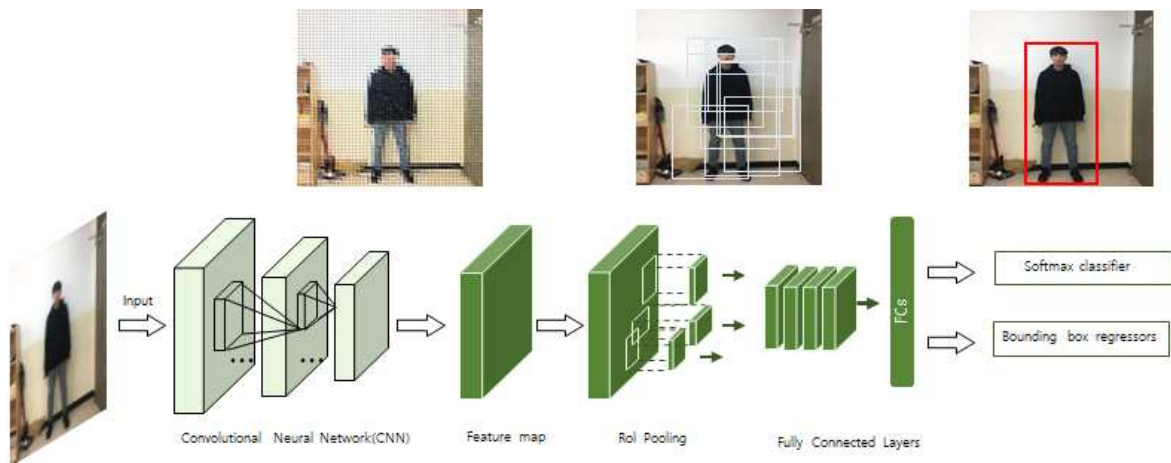


그림 21. Fast R-CNN의 기본구조

- Fast R-CNN으로 자세 분석 모델을 구축하기 위하여 먼저 서다, 앉다, 눕다 세 가지 자세에 대한 이미지를 각 350장씩 수집하여 트레이닝 셋을 생성하였음
- 수집한 트레이닝 셋을 TensorFlow 객체인식 API의 Fast R-CNN을 이용하여 학습을 진행하였음
- 최적의 손실률 값을 가지는 모델을 생성하고 자세 인식 학습 모델의 성능 검증 실험을 진행하였음

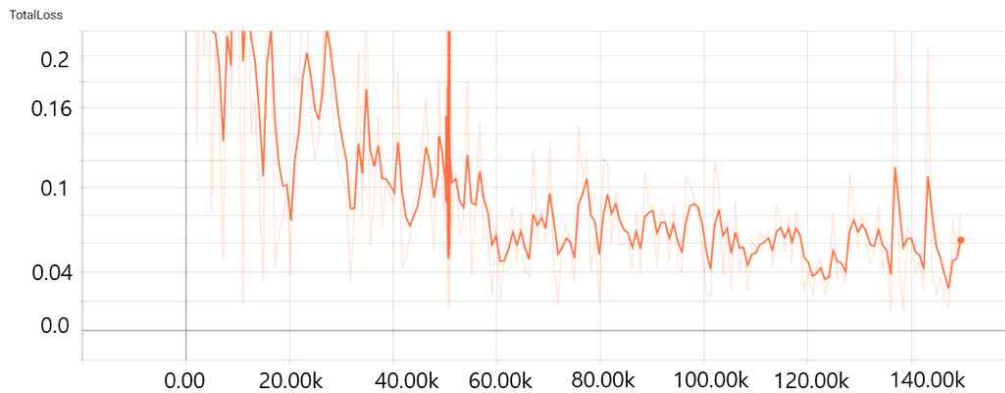


그림 22. 학습 횟수에 대한 손실률(Loss)값 그래프

- 자세 학습 분석 모델은 <그림 18>과 같이 학습 횟수가 약 10만 번에서 가장 좋은 성능을 보였으며, 학습 시간은 4시간이 소요되었음
- 따라서 10만 번 학습한 모델을 자세 분석 모델로 선정하였음

○ 연구결과 A6

(1) 이미지 수집 자동화 시스템 구축

- 자세 분석 모델을 학습하기 위해서 각 자세에 대한 이미지 데이터를 수집해야 함
- 이미지 수집에는 다양한 방법이 존재하나 대부분 사람이 프로그램을 사용하여 다운로드 해야 함
- 이러한 시스템은 이미지가 필요할 때 마다 프로그램을 실행해야 한다는 단점이 있음
- 따라서 크롤링 방법을 사용하여 서버가 자동으로 이미지를 다운로드할 수 있는 시스템을 구축
- 이미지 크롤링 시스템은 미리 저장된 키워드에 해당하는 이미지를 구글에서 검색하여 다운로드함
- 일정 주기로 자세 분석 모델의 3가지 자세인 서다, 눕다, 앉다에 해당하는 이미지를 검색하여 서버에 다운로드한 후 각 클래스별로 저장
- 저장된 이미지는 labeling 자동화 시스템에서 전처리 후 자세 분석 모델 학습에 사용됨

(2) 이미지 labeling 자동화 시스템 구축

- 자세 분석 모델을 학습하기 위해서 수집된 이미지의 labeling 작업이 필요함
- 이미지 labeling을 위해 수집된 이미지에 각각의 클래스를 지정해 주는 방법을 가장 많이 사용함

- 이미지 labeling이란 이미지에서 object의 영역과 클래스를 지정해주는 것임
- 기존에는 수작업으로 이미지에서 object의 영역을 지정해주었으나 사진이 많을 경우 시간이 오래 걸린다는 단점이 있음
- 이를 해결하기 위해 기존에 미리 학습된 object detection 모델을 사용하여 사람의 영역을 자동으로 검출함
- 미리 학습된 object detection 모델을 사용하여 사람의 영역을 추출하여 object의 영역을 지정함

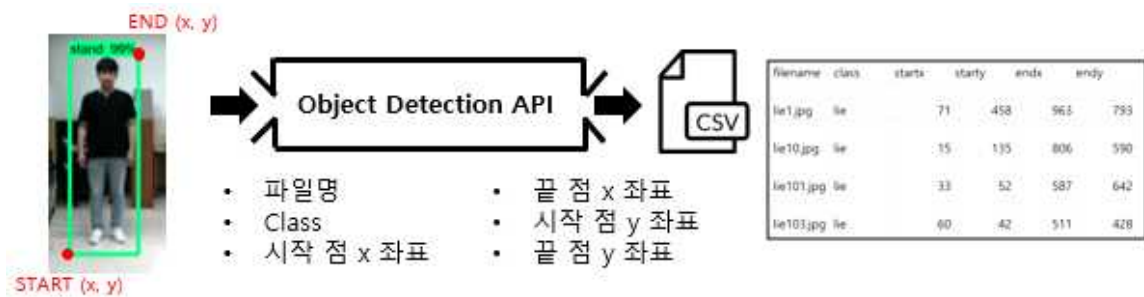


그림 23. 사람영역 추출 및 object 지정

- Single Shot Detector 알고리즘으로 학습된 object detection 모델을 사람만 인식하도록 수정하여 이미지 labeling 자동화 시스템 구축
- opencv의 Deep Neural Network 모듈과 수정된 object detection 모델을 사용하여 이미지 내에서 사람을 검출한 후 bounding box로 영역을 표시함
- 이 때 bounding box로 표시된 영역을 자세 분석 모델의 object 영역으로 지정함
- 검출된 사람의 영역은 csv파일로 저장하여 학습에 사용됨
- 검출된 사람의 영역을 표시하는 bounding box의 왼쪽 아래의 x좌표와 y좌표, 오른쪽 위의 x좌표와 y좌표를 csv 파일에 저장하여 자세 분석 모델 학습 시 사용함
- csv 파일은 이미지 파일명, class, bounding box의 좌표를 저장함

○ 2차 년도 연구 범위 및 개발 내용

연구 범위	연구 수행 방법 (이론적·실험적 접근방법)	구체적인 내용
딥러닝의 CNN(Convolutional Neural Network)를 활용한 이미지 분석 기법 활용	* Convolutional Neural Network에 기반한 3층 Layer 신경망에 Hidden Layer 를 약 5개에서 7개 정도 더 추가하는 딥러닝 시스템을 파이썬 기반으로 구현 * 여러 각도에서 찍은 사진들을 기반으로 복합적인 알고리즘을 활용하는 자가 학습 기반의 알고리즘의 개발을 통한 환자 패턴인식 알고리즘의 완성도 향상 * Static한 상태의 이미지 인식은 연구가 활발하나 (얼굴, 물건 식별), Dynamic 상태, 즉 움직임에 대한 이미지 분석에 대한 연구는 거의 수행되지 않고 있음 * 본 연구에서는 Dynamic 상태와 Static 상태 종합적인 이미지 인식 알고리즘을 개발 * 알고리즘의 성능 향상을 위한 공간 인식 알고리즘의 강화 요소 발굴	* cnn 기법 중 가장 인식률이 좋은 fast r c nn을 활용하여 자세 분석 모델을 구축하였음
클라우드 서버 구축을 통한 상시 성능 업데이트 및 컨설팅 시스템 구현	* 클라우드 기반 데이터 콜렉션 및 콜렉팅 할 수 있는 전송 기법의 개발 * 패턴 인식 및 다양한 알고리즘에 대한 딥러닝 기반 조합법 구현을 통한 성능 업데이트 * 인식된 패턴에 따른 노인 건강 상태에 대한 사전 경보 시스템 개발 * 하루 이상 거동이 없는 경우, 최근 부쩍 운동량이 줄어든 경우, 식사 시간이 오래 걸리는 경우 등에 대한 실시간 패턴링을 통한 경보 시스템 구현 가능	* 이미지 수집 자동화 시스템 구축 및 이미지 labeling 자동화 시스템을 구축하였음

개발된 시스템과 기 움직임 센서 기제품(Single sensor)을 이용한 데이터의 비교 검증	* 현재 기 상용화되어 있는 의료기기 중에 중저가 제품 및 필요시 신호 생성을 할 수 있는 시그널 제너레이터를 중심으로 데이터 비교 검증을 진행 * 의료 정보로서의 정확성이 중요한 SPO2, ECG, 혈당계, 혈압계 등 최대 4개 항목에 대해서는 기존에 시판되고 있는 의료기기들과의 교차 검증을 진행하여 정확성을 판단하고자 함 * 그 외 의료적 정확성이 크게 중요하지 않는 다른 주요 의료 센서에 대해서는 응용 어플리케이션의 동작 유무 및 주관적 사용 성을 중심으로 테스트 할 예정	* 기존 접촉식/비접촉식 센서를 사용한 위급상황 모니터링에 관한 선행 연구와 효율성 비교
개발된 시스템의 적용 파일럿 테스트	* 50대 이상 노인 가구를 대상으로 실제 파일럿 테스트 진행 * 테스트의 엄밀성을 위하여 모집단의 대표성을 나타낼 수 있도록 건강상태를 상중하에 맞추어 구분하여 대상을 확보하여 연령대 역시 80대, 70대, 60대로 구분하여 다양성이 있는 샘플을 구성하여 테스트 * 노인 요양 병원 1곳 대상 파일럿 테스트 진행 * 향후 서비스 추가 개발을 위한 설문 동시 실시	* 독거노인의 주거환경과 비슷한 26m²(8평)의 원룸형 주택을 기준으로 환경을 조성하였음
개발된 시스템의 건축물 내 적용 가능성 방안 구상	* 개발된 시스템의 건축물 내 설치에 대한 적용 방안에 대한 연구 * WiFi에 기반한 개발을 통해 설치 비용 및 설치에 따른 난이도 조절 * 파일럿 테스트 시에 일반 건축물 내에 적용 가능성에 대한 기준 구상 * 1개 가정집 및 요양 병원에 대한 샘플 스탠다드 모델을 디자인	* 실험 환경에 연령 별 사람을 대상으로 위급상황 검출 실험을 진행하였음
공간 설계 및 공간 제어 시스템과의 연계	* 공간 설계 시 데이터를 통해 수집된 정보를 활용할 수 있는 부분적 데이터 개방 고려 * 공간 제어 시스템과 관련된 가이드라인을 제공할 수 있는 방안 고려 * 공간분석방법이론을 활용하여 노인이나 장애인들의 공간설계에 활용	* 실험 환경에서 카메라의 위치에 따른 위급상황 정확도에 대한 실험을 진행하였음

○ 수행 내용 : fast r cnn을 활용한 자세 분석 모델 구축

- 위급상황 모니터링 시스템을 구축하기 위해 자세 분석 모델을 구축함
- 총 3가지 자세(앉다, 서다, 눕다)를 지정하고 이를 검출할 수 있는 자세 분석 모델을 구축함
- 각 자세 당 1000장씩 training set을 구성하여 학습을 진행함
- 이미지 전처리를 통해 생성된 이미지 메타데이터와 fast r cnn 알고리즘을 사용하여 자세 분석 모델 구축
- 학습에 사용되는 이미지를 Convolution feature map과 RoI feature vector 부분으로 나누어 학습을 진행함
- RoI feature vector 구간에서 추출된 이미지의 특징을 Fully Connected layers를 통해 노이즈를 제거
- Fully Connected layers를 통해 노이즈가 제거된 이미지를 확률 분포를 계산하는 softmax 함수를 사용하여 최적화 함
- 최종적으로 생성되는 이미지의 특징을 Bbox regressor를 사용하여 이미지의 class를 확정함

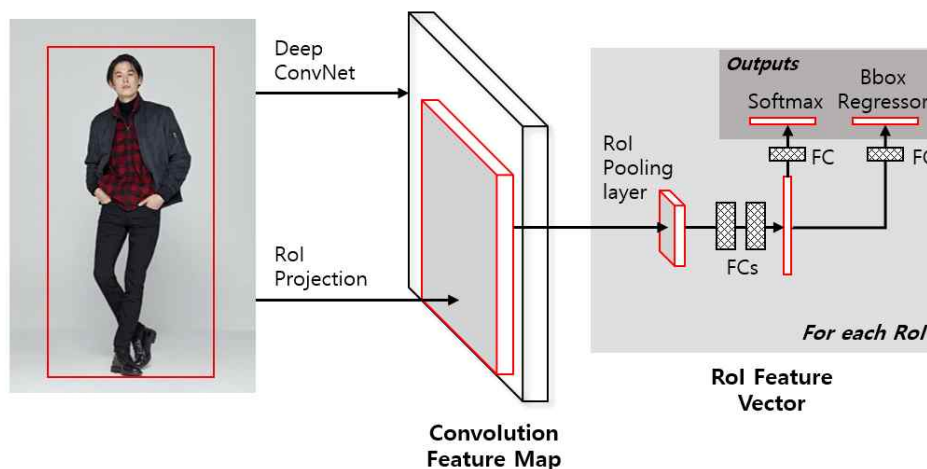


그림 24. Fast-RCNN 구조도

- 총 20만 번의 학습 횟수 중 약 10만 번째의 모델이 가장 높은 정확도를 보여 자세 분석 모델로 선정함
- 자세 분석 모델을 사용하여 실시간으로 입력되는 모니터링 영상에서 사람을 검출하고 자세를 분석하여 bounding box로 표시함



그림 25. 앉다/서다/눕다 의 실험 및 데이터 수집

○ 수행 내용 : 접촉식/비접촉식 센서를 사용한 위급상황 모니터링 시스템 구축에 관한 선행 연구 조사

- 위급상황 감지에 관한 선행 연구는 접촉식과 비접촉식 센서를 활용한 기술로 나눌 수 있음
- 직접 신체에 액세서리나 의복 형태로 제작된 생체 정보 수집용 웨어러블 디바이스 센서를 착용하는 것을 접촉식 센서 기술로 분류함
- 영상을 이용한 모니터링 방법과 거주환경 내에 여러 센서를 설치하여 스마트 홈과 같은 거주환경을 조성한 설치형 센싱 모니터링 방식 등이 비접촉식 기술에 해당함
- 접촉식 기술과 비접촉식 기술에 해당하는 선행연구를 조사하고 이를 분석하는 연구를 진행하였음

<표 2-1> 위급상황 모니터링에 관한 연구

대분류	소분류	연구자 (연구년도)	장점	단점
접촉식 모니터링	착용형 센싱 모니터링	장인훈 외 (2007)	• Zegbee를 이용한 원격 모니터링 • 맥박을 이용한 위급상황 감지	• 센서의 소형화가 덜 되어 착용감이 불편함 • 착용자의 신체조건에 따라 정확한 맥박을 측정하지 못함
		김남섭 (2010)	• 가속도 센서와 오디오 센서를 결합하여 낙상 검출률을 높임	• 주변환경에 따라 검출이 왜곡될 수 있음
		이용규 외 (2011)	• 센서 데이터의 전처리를 통해 정확도를 높임	• 신속하게 위급상황을 판단할 수 없음
		Wan-young Chung 외 (2008)	• 벨트와 팔찌형태의 심전도, 가속도계, 산소포화도 등을 무선네트워크를 통해 원격으로 생체정보를 수집함	• 착용이 불편함 • 센서로부터 얻는 데이터가 클수록 무선네트워크 수신에 제한됨

		Rita Paradiso 외 (2005)	• 압전 저항성 직물과 섬유 전극을 통해 의복형 센서 모니터링 시스템을 통해 착용감과 편의성을 증대시킴	• 각 센서들의 정보를 통합하고 서버로 보내는 상대적으로 큰 장치를 지니고 있어야 함 • 옷을 갈아입을시 센서들을 다른 옷에 교체시켜야 함
		조위덕 외 (2015)	• 멀티 센싱 기술을 이용하여 개인의 삶에 최적화된 신뢰성 높은 시스템을 제안	• 상대적으로 많은 센서들을 이용하므로 비용이 큼
		조위덕 외 (2017)	• SVM과 ANN을 사용하여 행동 패턴을 예측함	• 기존의 행동인식 구현 접근방법이 없어 학습모델을 직접 만들고 계산해야하기 때문에 기회비용이 큼
비접촉식 모니터링	설치형 센싱 모니터링	Zhu. L 외(2017)	• RFID를 이용하여 주파수 세기의 변화를 이용해 낙상을 빠르게 감지함	• 일반 행동들을 구분할 수 없음
		Mehta. R 외(2017)	• 가정내에 설치된 센서를 원격으로 수집하여 행동을 파악가능함	• 설치된 센서의 유지보수 문제가 존재
		Nag. D 외 (2017)	• 가정환경 내의 여러 정보를 수집 및 조작이 가능함	• 사용자의 직접적인 위급상황은 감지하기 어려움
		Iman Azimi 외(2017)	• 스마트 홈으로부터 얻어진 센서들을 중앙 서버가 관리 및 계산하여 정확하고 빠른 위급상황 대처가 가능함	• 중앙 서버 컴퓨터와 인터넷의 성능이 중요함 • 센서와 중앙서버의 비용이 큼
		최경선 외 (2016)	• 전화 유선망을 사용하여 인터넷 기반이 취약한 고령자가 쉽게 사용할 수 있음	• 유선망의 성능의 정보 전달 속도가 느려 위급상황의 정확하고 빠른 판단이 어려움
	영상 모니터링	김상수 외 (2014)	• 비디오 카메라를 이용하여 목표대상의 낙상을 빠르게 파악 가능	• 낙상과 움직임의 경우만 판단이 가능함 • 가정 내의 물건의 위치 및 배경에 따라 움직임 벡터 값을 추출하지 못함
		Bian. Z 외 (2015)	• 모니터링 대상의 관절 부위를 추출하여 행동 변화정도를 측정 가능	• 신체와 바닥사이의 거리가 멀거나 가구위에 누워있는 경우 낙상을 판단할 수 없음

			적외선을 이용해 어두운 환경에서 모니터링이 가능함	음
		Ahmed Elgammal 외 (2010)	커널 밀도를 추정하여 영상 간의 차이를 이용하여 객체 인식의 정확도를 높임	외부환경에 영향을 많이 받음
		Rogez. G 외(2018)	여러명의 사람에 대해서 자세 인식이 가능함	자세 인식율이 75%로 다른 모델에 비해 상대적으로 낮음
		Yang. Yi (2013)	다중 해상도의 영상으로부터 머리 인식이 가능함	정면이 아닌 방향에서의 인식 정확도가 낮음

○ 수행 내용 : 자세 분석 모델을 활용한 위급상황(낙상) 검출 시스템 구축

- 2차 년도 연구 결과인 자세 분석 모델을 활용하여 위급상황(낙상) 검출 시스템을 구축하였음
- 독거노인의 주거환경과 비슷한 26m²(8평)의 원룸형 주택을 기준으로 실험

(가) 자세 분석 모델을 활용한 위급상황(낙상) 검출 시스템 구축

- 2차 년도 연구에서 최종 선정된 자세 분석 모델과 웹캠을 이용해 위급상황(낙상) 검출 시스템을 구축하였음

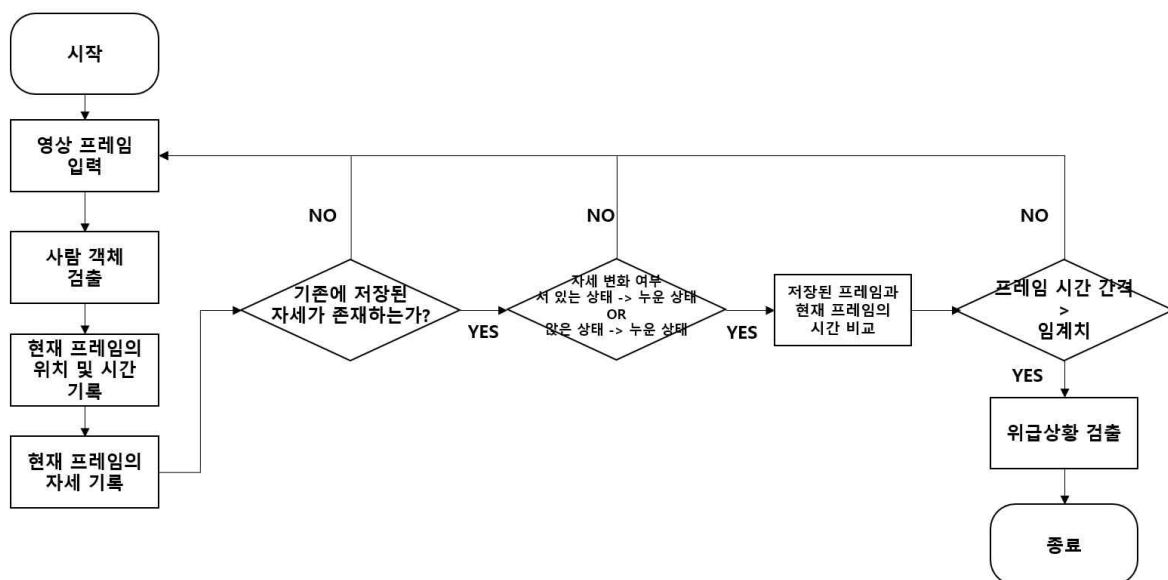


그림 26. 위급상황 검출 순서도

- 자세 분석 모델을 사용하여 실시간으로 촬영된 영상에서 사람을 검출하고 자세를 인식함
- 사람을 검출하고 자세를 인식 한 경우 검출된 사람의 가장 높은 부분의 화면 상 좌표를 계산 후 현재 시간과 함께 저장
- 실시간 모니터링 영상의 현재 프레임과 이전 프레임 간의 자세 변화가 일어나면 위급 상황(낙상) 알고리즘에 의해 위급상황(낙상) 판단



그림 27. 실시간 모니터링 영상의 위급상황 판단

(나) 위급상황(낙상) 검출 알고리즘

- 위급상황(낙상)을 검출하기 위해 특정 자세변화 조건(서다->눕다, 앉다->눕다)을 설정함
- 위급 상황(낙상) 조건을 만족할 경우 실시간 모니터링 영상의 현재 프레임과 이전 프레임에서 사람이 움직이는 속도를 계산함
- 속도 계산은 자세 인식 모델을 통해 인식된 모니터링 대상을 표기하는 영역인 bounding box를 사용하여 계산함
- bounding box의 가장 높은 중심지점에 대한 좌표를 추출하고 해당 좌표에 대해서 이전 프레임과 비교하여 픽셀의 이동 값을 이용해 속도를 계산함
- 위급상황 유형에 따라 속도 임계치를 지정함
- 계산 결과가 미리 설정해둔 속도의 임계값을 넘어갈 경우 낙상이라 판단함

(다) 독거노인 주거 환경 조성 및 실험

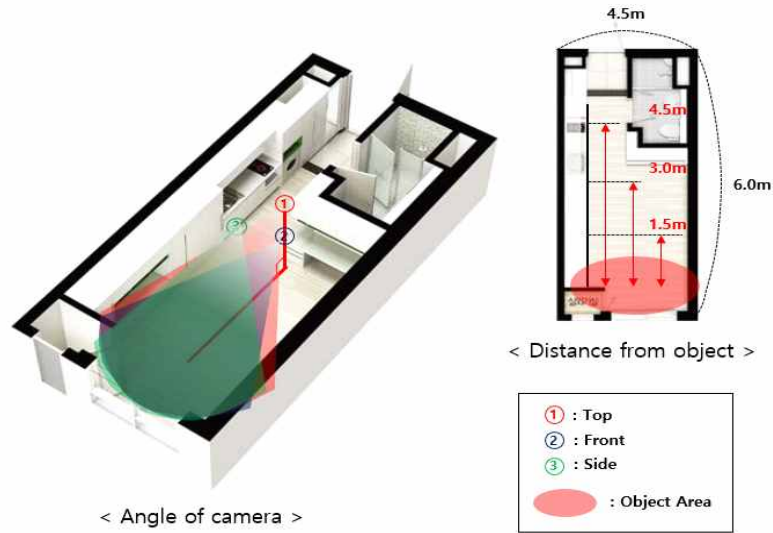


그림 28. 실험을 위한 독거노인 주거환경 조성

- 독거노인의 주거환경과 비슷한 $26m^2$ (8평)의 원룸형 주택을 기준으로 실험 환경 조성
- 위급상황 검출을 위해 카메라의 촬영 각도와 촬영 거리를 나누어 실험을 진행함
- 서다, 앉다, 눕다 세 가지 자세에 대해 피실험자를 통해 각 자세에 대한 상황을 연출하여 실험을 진행하였음
- 실험 환경은 촬영거리 (1.0~1.5m, 1.5~3.0m, 3.0~4.5m)와 방 전체를 촬영할 수 있는 위치에 서 카메라의 촬영 각도(상단, 정면, 측면) 세 가지로 구분하여 총 아홉 가지의 실험 조건을 구성하였음
- 촬영 거리를 달리하여 실험을 진행하였고 아래의 표와 같은 결과를 도출하였음

<표 2-2> 자세 분석 모델 실험 결과

촬영 거리 촬영 각도	1.0 ~ 1.5m			1.5 ~ 3.0m			3.0 ~ 4.5m		
	앉다	서다	눕다	앉다	서다	눕다	앉다	서다	눕다
상단	72/100	71/100	69/100	100/100	100/100	96/100	100/100	100/100	95/100
정면	74/100	76/100	71/100	100/100	100/100	97/100	100/100	100/100	97/100
측면	69/100	71/100	70/100	97/100	97/100	92/100	96/100	98/100	90/100

- 자세 Detecting 학습 모델을 이용하여 위급상황을 감지실험

<표 2-3> 낙상 상황 조건

	실험 조건	
	자세 변화	프레임 차이 임계치
위급상황 1	앉은 상태 → 누운 상태	0.3초
위급상황 2	서있는 상태 → 누운 상태	0.2초

- 속도 임계치는 낙상이 독거노인에게 치명상을 입힐 수 있는 속도인 2.5m/s로 설정
- 위급상황 1 유형에서는 앉은 상태로 시작하므로 상대적으로 낮은 속도인 1.5m/s를 임계치로 설정하여 실험을 진행

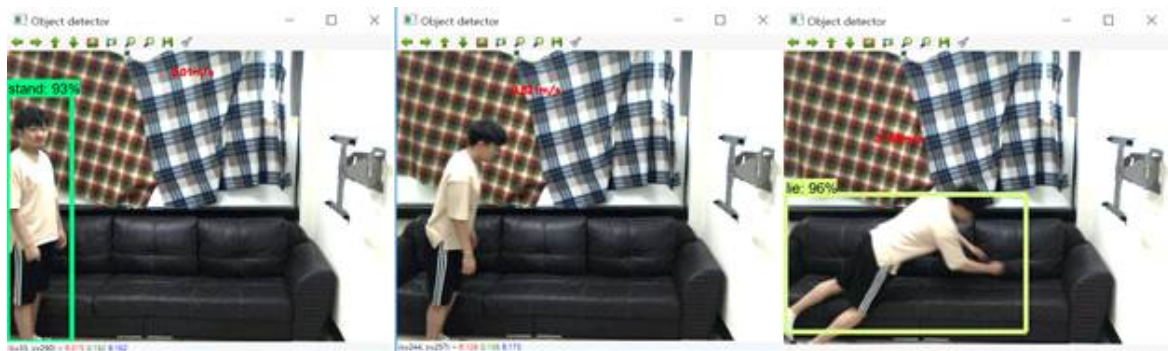


그림 29. 낙상 연출 실험

- 영상 내에서 신체의 일부만이 촬영되어 인식률이 상대적으로 낮게 나옴
- Detecting이 안되거나 다른 자세로 오인식하는 경우가 발생
- 1.5m 이상의 거리에서는 촬영 각도와 상관없이 90%이상에 가까운 Detecting 성공률을 나타냄
- 최적의 Detecting조건으로는 촬영 각도는 크게 상관이 없었으며 거리에 영향을 많이 받는 것으로 나타남
- 사람의 신체 대부분이 촬영될 수 있는 최소 2~3m 이상의 거리에서 촬영되어야 Detecting이 잘되는 것으로 나타났으며, 눕다의 자세인식 경우, 서다, 앉다에 비해 Detecting 성공률이 다소 낮은 것으로 나타남

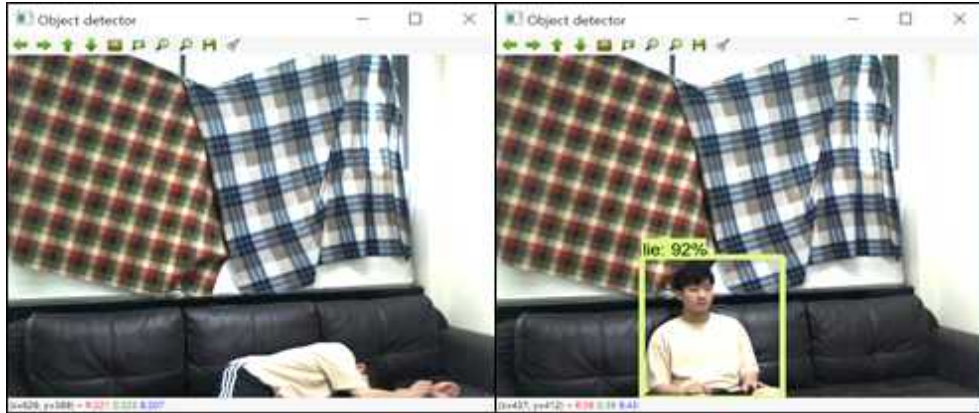


그림 30. 1.0~1.5m 거리의 오인식 화면

(라) 노인 및 장애인 등을 대상으로 적용 실험

- 노인 및 장애인의 경우 낙상 상황을 연출하는 것에 어려움이 존재
- 구축된 시스템을 평가하기 위해 인터넷 상의 동영상을 사용하여 실험을 진행함
- 노인 및 장애인의 일상생활이 녹화된 동영상을 대상으로 자세 분석 모델을 사용하여 노인의 현재 자세를 분석함
- 동영상의 경우 유튜브와 구글 검색을 통해 다운로드 하여 약 10초 정도의 동영상으로 잘라 실험을 진행함



그림 31. 노인 및 장애인 일상생활 동영상 예시

- 동영상이 진행됨에 따라 각 프레임에서 노인의 현재 자세를 분석하여 화면에 표시함
- 약 10개의 동영상을 수집하여 실험을 진행하였으며 실험 결과 일상생활에서의 자세 분석은 높은 정확도를 보임



그림 32. 동영상을 활용한 자세 분석 모델 실험 결과 화면

- 낙상의 경우 노인 및 장애인이 낙상을 연출하는 것이 현실적으로 어려움이 존재하여 일반 성인을 대상으로 낙상 상황을 연출하여 진행함
- 낙상의 경우 위의 (다)독거노인 주거 환경 조성 및 실험 결과로 도출된 임계치를 사용하여 낙상 상황을 정의하였음
- 일반적인 낙상 상황(걸다가 쓰러지는 경우, 침대에서 넘어지는 경우, 앉아 있다가 넘어지는 경우 등)을 정의하여 실험을 진행하였음



그림 33. 낙상 시나리오에 따른 성인 낙상 실험

- 일반 성인을 대상으로 낙상 상황을 연출하여 정확도를 측정하였음
- 실험 횟수는 시나리오 당 50번 진행하였음
- 카메라의 각도를 달리하여 실험을 진행하여 정확도를 측정함
- 낙상 실험 결과 약 80%이상의 정확도를 보여 유의미한 결과를 도출하였음

<표 2-4> 낙상 연출 시나리오

시나리오 촬영 각도	서있다 쓰러지는 경우	침대에서 넘어지는 경우	의자에 앉아 있다가 넘어지는 경우
상단	48/50	45/50	47/50
정면	49/50	48/50	48/50
측면	45/50	40/50	49/50

- 심박센서, 온도센서 등을 사용한 낙상 실험을 진행하기 위하여 중립 종합사회 복지관과 연계하여 실험을 진행함
- 노인 및 성인 각 3명에게 심박 센서와 온도 센서 등을 부착하여 하루 동안 데이터를 수집함
- 실험을 진행하는 동안 노인의 활동에 있어 응급상황이라 판단될 만한 근거가 되는 상황이 발생하지 않음
- 노인의 경우 일상생활이 안정적인 경우가 많기 때문에 센서 데이터를 통해 응급상황을 판단하기 어려움
- 따라서 심박 센서, 온도 센서 등을 연계한 응급상황 검출 실험은 센서 데이터 수집의 어려움으로 인해 추후 연구가 더 필요할 것으로 판단됨

(마) 실험 적용 결과

- 노인 및 장애인의 경우 낙상 상황의 연출이 어려움이 존재 낙상 실험을 진행해도 실제낙상이 아닌 이상 실험 환경구성에 어려움
- 실험대상을 이용해 낙상에 대한 실험 실시
- 촬영 거리와 촬영 각도에 따라 앉다 서다 눕다 세 가지 자세에 대해 판단 실험결과

<표 2-4> 촬영 각도 및 촬영 거리에 따른 자세 분석 실험 결과

촬영 거리 촬영 각도	1.0 ~ 1.5m			1.5 ~ 3.0m			3.0 ~ 4.5m		
	앉다	서다	눕다	앉다	서다	눕다	앉다	서다	눕다
상단	72/100	71/100	69/100	100/100	100/100	96/100	100/100	100/100	95/100
정면	74/100	76/100	71/100	100/100	100/100	97/100	100/100	100/100	97/100
측면	69/100	71/100	70/100	97/100	97/100	92/100	96/100	98/100	90/100

- 촬영 거리와 촬영 각도에 대한 최적의 결과에 대해 이 결과를 이용해 낙상 실험 실시
- 총 100회에 대한 낙상실험결과

<표 2-5> 실험 결과에 대한 precision 및 recall 결과

상황	정확도	Precision	Recall
앉다 - 상단	96	87.15	77.54
앉다 - 중단	97	88.14	78.48
앉다 - 하단	86	89.75	76.42
서다 - 상단	90	87.46	77.94
서다 - 중단	92	89.66	81.42
서다 - 하단	98	86.34	76.24
앉다 - 상단	97	88.96	78.74
앉다 - 중단	97	88.66	82.86
앉다 - 하단	98	89.34	80.59

3. 목표 달성도 및 관련 분야 기여도

3-1. 목표

구분	내용
최종 목표	오픈소스 하드웨어 기반 독거노인 및 장애인 실내 환경 모니터링 시스템 개발 및 클라우드를 통한 지속적 자가 학습 딥러닝 시스템 구현
세부목표	① 오픈 하드웨어 기반 환자 자세 분석 및 노인 활동 패턴 분석 알고리즘 개발 ② 가속도 센서 및 심박측정 센서를 통한 움직임 모니터링 기술 개발 ③ 딥러닝을 활용한 CNN(Convolution Neural Network)기반 정확도 향상 알고리즘 및 클라우드 솔루션의 구축

① 개발목표 1: 오픈 하드웨어 기반 환자 자세 분석 및 노인 활동 패턴 분석 알고리즘의 개발

- * 저가형 카메라 모듈 다수를 오픈소스 하드웨어에 연동하여 효율적으로 다수의 카메라를 연동하는 시스템의 구현
- * Bayes Classifier 및 기존의 이미지 분석 알고리즘에 기반을 둔 이미지 인식 기법의 구현

② 개발목표 2: 가속도 센서 및 심박측정 센서를 통한 움직임 모니터링 기술

- * 오픈소스 하드웨어에 기반을 둔 포터블 환자 패턴 분석용 모니터링 시스템의 개발 및 노인 활동 패턴 분석 알고리즘과의 결합
- * 가속도 센서 및 심박측정 센서를 부착한 오픈소스 하드웨어 기술의 개발 구현

③ 개발목표 3: 딥러닝을 활용한 CNN 기반의 정확도 향상 알고리즘 및 클라우드 솔루션의 구축

- * 딥러닝을 활용한 CNN(Convolution Neural Network)의 구축 및 이를 통한 자기 강화 학습 알고리즘의 구축
- * 클라우드 시스템의 구축을 통한 빅데이터 솔루션의 구현 및 이를 딥러닝에 활용

3-2. 목표 달성여부

구분	달성 여부	달성 도	내용
목표 1	달성	40%	○ 오픈소스를 사용한 노인 낙상 모니터링 시스템에 관한 연구 ○ 이미지 처리를 통해, 노인 자세의 slope을 계산하여 낙상을 감지하는 시스템 개발 (1) SCIE 논문 출간 A study on a fall detection monitoring system for falling elderly using open source hardware, Multimedia Tools and Applications, 2017. 12, (2) 특허 출원 - 출원번호:10-2018-0029180 - 발명자: 최승현, 염세경 - 영상 이미지를 이용한 위급상황 분석장치 및 방법
목표 2	달성	20%	○ 산업계에서 활용할 수 있는 오픈소스 기반 의료 서비스의 동향을 파악하며 6종의 의료용 센서를 실제로 다루고 이를 활용하여 웹 기반의 서비스를 만들어 보는 것을 목표로 함 (3) 교재개발(출간): 집에서 만드는 라즈베리파이 건강모니터 - 재직자 교육: 오픈소스 하드웨어를 활용한 헬스케어 디바이스 개발 20시간 코스워크
목표 3	달성	40%	○ 이미지 딥러닝 방법론을 이용한 노인 자세 판단 및 낙상 감지에 관한 연구 (4) SCIE 논문 출간 Development of a methodology to predict and monitor emergency situations of the elderly based on object detection, Multimedia Tools and Applications, 2018. 9, (5) 특허 출원 - 출원번호:10-2018-0106432 - 발명자: 염세경 최승현 김창균 원진영 - 객체 인식을 통해 위급 상황을 판단하는 장치 및 방법
		100%	

3-3. 목표 미달성 시 원인 및 차후대책(후속연구의 필요성 등)

- 해당사항 없음

- 4. 연구개발성과의 활용 계획 등

4-1 연구개발성과의 활용방안

○ 기술수요처 및 현장 적용방안

- 독거노인, 장애인, 환자, 소외계층의 행동 모니터링
- 실버타운, 복지센터, 병원 등(Reference Site)에 대한 적용

○ 활용분야 및 활용방안

- 노인이나 환자, 어린이 등의 활동 패턴분석 및 분류 연구를 통한 행동 위험 감지 시스템에 응용
- 실내공간조성(1인 가구, 독거노인층, 실버타운, 장애인 가정) 레퍼런스자료로 활용
- 치매노인, 정신질환자들의 행동 파악을 통한 관리 처방 가이드라인 응용
- 공공기관 및 위탁기관 등과 협력하여 솔루션 확장

○ 재개발 및 공급확대

- 관련 기술의 보고서를 통한 산업계 및 학계 노하우 제공
- 추후 상품화 단계에서의 개발 시 국내 SW 및 HW 업체, 건설시공사등과 협력하여 상업적인 틀의 형태로 제공
- 향후 해외시장 공급을 위해 Cooper 데이터를 이용해 해외 환경에 적합한 알고리즘을 동시에 개발할 예정
- 솔루션의 확장 및 최적화를 통해 고도화 후 공급 예정

○ 지적재산권 활용

- 과제수행 중 발생한 지적재산권, 보고서, 특허 등에 대한 기술료 수입 획득
- 추후 상용시스템 개발완료시 서비스권, 제품 판매권 등도 확보 예정
- 지적재산권을 확보하여 공공 및 위탁기관 등 수요가 있는 곳에 공급 예정

4-2 기대 효과

○ 기술적 측면

- 감시형태의 모니터링 시스템에서 능동적이고 적극적인 보건의료관리 서비스 제공 가능
- 이동궤적추적, 관리 대상자의 비정상행동 인식을 통한 상황 통제 가능으로 차별화된 새로운 응용 서비스 개발 기대
- 개발된 시스템을 어느 산업분야에서나 누구나 쉽게 접근할 수 있도록 범용성을 높여 개발
- 세대별, 환경별, 대상자 특징별, 시간대별 등의 다양한 데이터를 분석하고 제공하여, 향후 연구시 다양한 분야에서 사용할 것으로 기대됨

○ 경제적 · 산업적 측면

- 지능형 영상 모니터링 분야의 세계시장 주도 및 확대 기대
- 새로운 서비스 창출을 통해 유헤스, 유시티, 스마트 주거환경설계와 같은 관련 산업의 경쟁력 확보
- 독거노인 및 1인 가구에서 발생하는 위험상황의 자동 모니터링을 통해 케어 비용 절감가능
- 불필요한 의료 이용을 억제하고, 개인 맞춤형 의료 서비스 제공을 도와 스마트 헬스케어 주거의 기본 시스템으로
- 향후 u-city, 재택의료시스템등의 효율화 및 선진화를 견인, 신(新) 성장산업 발굴

○ 사회적 측면

- 고령화 시대 및 1인가구의 증가로 인한 서비스 필요 증가
- 소외계층 및 약자의 안전을 보장하는 선진국형 복지정책 확대

붙임. 참고문헌

1. Fougère M, Mérette M (1999). Population ageing and economic growth in seven OECD countries. *Economic Modelling* 16(3):411-427
2. 윤연옥 외. (2017). “2017 고령자 통계” , pp. 3, 5, 18, 통계청
3. 정기택. (2014). “보건산업 브리프 Vol.128 - 65세 이상 고령자의 개인지출 의료비 현황분석” , pp. 1, 8, 한국보건산업 진흥원
4. 최동진 외. (2017). “IoT 기반 Wearable Device 를 활용한 독거노인 Care 서비스에 관한 연구” , pp. 407-417, 한국경영정보학회 2017년 경영정보관련 추계학술대회
5. 박정훈. (2018). “헬스케어 웨어러블 디바이스의 기술 및 시장 동향” , 45(3), pp. 48-56, 전자공학회지
6. Azimi, Iman, et al. “Internet of things for remote elderly monitoring: a study from user-centered perspective.” *Journal of Ambient Intelligence and Humanized Computing* 8.2 (2017): 273-289.
7. 장인훈, & 심귀보. (2007). 센서 네트워크 응용을 위한 반지형 맥박센서와 모니터링 시스템. *한국지능시스템학회 논문지*, 17(5), 619-625.
8. 김남섭. (2010). 유비쿼터스 헬스케어를 위한 효율적인 낙상 감지 기법. *한국정보기술학회논문지*, 8(8), 133-140.
9. 이용규, 천대진, & 윤길원. (2011). 노인을 위한 원격 낙상 검출 시스템. *Journal of Sensor Science and Technology*, 20(6), 420-427.
10. Wan-Young Chung, et al(2008) A Wireless Sensor Network Compatible Wearable U-healthcare Monitoring System Using Integrated ECG, Accelerometer and SpO2 30th Annual International IEEE EMBS Conference Vancouver, British Columbia, Canada, August 20-24, 2008
11. Rita Paradiso, et al(2005) WEALTHY - a wearable healthcare system: new frontier on e-textile, *Journal of Telecommunications and Information Technology*, 4/2005
12. 조위덕, 양승국, 최선탍, 백재순, 민명기, 이영권, & 이규필. (2014). IoT 를 사용한 라이프로그 빅데이터기반 라이프스타일 (생활패턴) 분석 및 웰니스 예측케어 서비스시스템. *한국통신학회지 (정보와통신)*, 31(12), 17-24., 45(3), pp. 48-56, 전자공학회지
13. 조위덕, 성지훈, 최선탍, 김준호, 하정호, 백재순, & 이주영. (2017). IOT 멀티 센서 기반 빅데이터 추출을 위한 적응적 센싱 및 모니터링 기술. *정보과학회지*, 35(7), 26-34.
14. Zhu, L., Wang, R., Wang, Z., & Yang, H. (2017). TagCare: Using RFIDs to Monitor the Status of the Elderly Living Alone. *IEEE Access*, 5, 11364-11373.
15. Mehta R, Kale S, Utage AS (2017) The Internet Of Things (IOT) Intelligence Computing

Technology for Home Automation. International Journal of Current Engineering and Technology

16. Nag D, Majumder D, Raquib CM, Pramanik S, Basu A, Rana TK, Rana B (2017) Green energy powered smart healthy home. In Industrial Automation and Electromechanical Engineering Conference (IEMECON), pp 47-51
17. Iman Azimi, et al(2017), HiCH: Hierarchical Fog-assisted Computing Architecture for Healthcare IoT, International Conference on Embedded Software 2017 and appears as part of the ESWEEK-TECS special issue, 1539-9087
18. Choi, K. S., & Chun, J. C. (2016). Development of mobility and vitality signal monitoring system based on zigbee-pstn gateway for the elderly. The Journal of Korea Institute of Information, Electronics, and Communication Technology, 9(1), 9-14.
19. 김상수, 김선우, & 최연성. (2014). 센서와 영상을 이용한 이상 행동 모니터링 시스템. 정보과학회논문지, 41(12), 1152-1159.
20. Bian, Z. P., Hou, J., Chau, L. P., & Magnenat-Thalmann, N. (2015). Fall detection based on body part tracking using a depth camera. IEEE journal of biomedical and health informatics, 19(2), 430-439.
21. Ahmed Elgammal, et al (2000) Non-parametric Model for Background Subtraction, ECCV 2000, LNCS 1843, pp. 751-767, 2000.
22. Rogez G, Weinzaepfel P, Schmid C (2018) LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images. arXiv preprint arXiv:1803.00455
23. Yang Y (2013) Articulated Human Pose Estimation with Flexible Mixtures of Parts. University of California, Irvine
24. Girshick R (2015) Fast r-cnn. arXiv preprint arXiv:1504.08083
25. Gkioxari G, Girshick R, Malik J (2015) Contextual action recognition with r* cnn. In Proceedings of the IEEE international conference on computer vision, pp 1080-1088
26. https://chrome.google.com/webstore/detail/fatkun-batch-download_ima/nnjjahlikiabnchcpeh_cpkdeckfgnohf
27. https://download.cnet.com/Extreme-Picture-Finder/3000-2071_4-10070685.html
28. http://www.visipics.info/index.php?title=Main_Page
29. <https://opencv.org/opencv-3-3.html>
30. Lee, Tae Kyung, et al. "A Basic Study on Human Injuries according to Slip Velocity during Falling." Proceedings of the Korean Society of Precision Engineering Conference (2016): 217-218.

붙임. 소스코드

낙상 검출 시스템

```
from imutils.video import VideoStream
from imutils.video import FPS
import numpy as np
import argparse
import imutils
import time
import cv2
import datetime
import math

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--prototxt", required=True,
                help="path to Caffe 'deploy' prototxt file")
ap.add_argument("-m", "--model", required=True,
                help="path to Caffe pre-trained model")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
                help="minimum probability to filter weak detections")
args = vars(ap.parse_args())

# initialize the list of class labels MobileNet SSD was trained to
# detect, then generate a set of bounding box colors for each class
CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
            "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
            "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
            "sofa", "train", "tvmonitor"]
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))

# load our serialized model from disk
print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])
```

```

# initialize the video stream, allow the cammera sensor to warmup,
# and initialize the FPS counter
print("[INFO] starting video stream...")
video_path = "D:/fall4.mp4"
#vs = VideoStream(src=0).start()
vs = cv2.VideoCapture(video_path)

time.sleep(2.0)
fps = FPS().start()
f = open('asdfasdf.txt','w', encoding='UTF-8')
#variables for calculate object's speed
objectv = 0.0
objectd = 0.0
ptime = datetime.datetime.now()
ctime = datetime.datetime.now()
tempx = 0
tempy=0
flag = 0
# loop over the frames from the video stream
while True:
    # grab the frame from the threaded video stream and resize it
    # to have a maximum width of 400 pixels
    ret, frame = vs.read()
    frame = imutils.resize(frame, width=400)

    # grab the frame dimensions and convert it to a blob
    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 0.007843, (300, 300), 127.5)

    # pass the blob through the network and obtain the detections and
    # predictions
    net.setInput(blob)
    detections = net.forward()

    # loop over the detections

```

```

for i in np.arange(0, detections.shape[2]):
    # extract the confidence (i.e., probability) associated with
    # the prediction
    confidence = detections[0, 0, i, 2]

    # filter out weak detections by ensuring the 'confidence' is
    # greater than the minimum confidence
    if confidence > args["confidence"]:
        # extract the index of the class label from the
        # 'detections', then compute the (x, y)-coordinates of
        # the bounding box for the object
        idx = int(detections[0, 0, i, 1])
        if(idx != 15):
            continue
        else:
            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (startX, startY, endX, endY) = box.astype("int")

            # draw the prediction on the frame
            label = "{}: {:.2f}%".format(CLASSES[idx], confidence * 100)
            cv2.rectangle(frame, (startX, startY), (endX, endY), COLORS[i
dx], 2)

            y = startY - 15 if startY - 15 > 15 else startY + 15
            cv2.putText(frame, 'stand', (startX, y), cv2.FONT_HERSHEY_
SIMPLEX, 0.5, COLORS[idx], 2)

            #print(startX,startY,endX,endY)
            midx = int((startX+endX)/2)
            midy = int((startY+endY)/2)
            ctime = datetime.datetime.now()
            objectd = math.sqrt((midx-tempx)**2+(endY-tempy)**2)
            resultd = ((objectd*2.54)/96)/100
            tresult = ctime-ptime
            tempt = tresult.microseconds/1000000
            if(tempt<=0):
                continue

```

```

        objectv = resultd/tempt
        tv = round(objectv,5)
        tempv = str(tv)
        print(tempv)
        cv2.putText(frame,tempv+'m/s',(midx,midy),cv2.FONT_HERSHEY_
EY_SIMPLEX,0.5,COLORS[idx], 2)
        if tv > 0.2:
            f.write(tempv+'m/s\n'+warning'\n')
        tempx = midx
        tempy = endY
        ptime = ctime
        flag+=1

# show the output frame
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF

# if the 'q' key was pressed, break from the loop
if key == ord("q"):
    break

# update the FPS counter
fps.update()

# stop the timer and display FPS information
fps.stop()
print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))
f.close()

# do a bit of cleanup
cv2.destroyAllWindows()
vs.release()

```

이미지 자동 labeling 시스템

```
import os
import numpy as np
import argparse
import cv2
import csv

ap = argparse.ArgumentParser()
#ap.add_argument("-i", "--image", required=True,
#                help="path to input image")
ap.add_argument("-p", "--prototxt", required=True,
                help="path to Caffe 'deploy' prototxt file")
ap.add_argument("-m", "--model", required=True,
                help="path to Caffe pre-trained model")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
                help="minimum probability to filter weak detections")
args = vars(ap.parse_args())

CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
            "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
            "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
            "sofa", "train", "tvmonitor"]
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))

print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])

csvf = open('C:/Users/user/Desktop/dev/xmlcoordinate/automake.csv', 'w')
csv_writer = csv.writer(csvf, delimiter=',')
flag = 0

print("start")
csv_writer.writerow(['filename'] + ['class'] + ['xmin'] + ['ymin'] + ['xmax'] + ['ymax'])
for root, dirs, files in os.walk('./train/')
```

```

if(flag == 0):
    flag = 1
    continue
else:
    lists = root.split('/')
    dirname = lists[2]
    for file in files:
        image = cv2.imread(root+'/'+file)
        (h, w) = image.shape[:2]
        blob = cv2.dnn.blobFromImage(cv2.resize(image, (300, 300)), 0.007843,
(300, 300), 127.5)

        print("[INFO] computing object detections...")
        net.setInput(blob)
        detections = net.forward()

        for i in np.arange(0, detections.shape[2]):

            confidence = detections[0, 0, i, 2]

            if confidence > args["confidence"]:

                idx = int(detections[0, 0, i, 1])
                if(idx!=15):
                    continue
                else:
                    box = detections[0, 0, i, 3:7] * np.array([w, h,
w, h])

                    (startX, startY, endX, endY) = box.astype("int

")

                    print("aaa")
                    label = "{}: {:.2f}%".format(CLASSES[idx], conf

idence * 100)

                    print("[INFO] {}".format(label))
                    cv2.rectangle(image, (startX, startY), (endX,

```

```
endY),COLORS[idx], 2)
```

```
+ 15
```

```
[idx], 2)
```

```
[startY]+[endX]+[endY])
```

```
csvf.close()
```

```
print("done")
```

```
y = startY - 15 if startY - 15 > 15 else startY
```

```
cv2.putText(image, label, (startX, y),
```

```
cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS
```

```
print(startX,startY,endX,endY)
```

```
csv_writer.writerow([file]+[dirname]+[startX]+
```

```
print(file)
```

이미지 수집 시스템

```
import time          #Importing the time library to check the time of code execution
import sys           #Importing the System Library

import os
import os.path as osp

import fnmatch

import json

#import urllib2
from urllib import urlencode

from urllib2 import Request,urlopen
from urllib2 import URLError, HTTPError

import datetime

import hashlib

from collections import OrderedDict

##### CONFIGS #####
# Path to config_file
config_file = './config.json'

##### Default CONFIGS #####
CONFIGS = {}

# How many images you want to download for each class. Google will only return 100 images
at most for one search
CONFIGS[u'num_downloads_for_each_class'] = 200
```

```

# image type to search
CONFIGS[u'search_file_type'] = 'jpg'
#CONFIGS[u'search_file_type'] = 'bmp'
#CONFIGS[u'search_file_type'] = 'png'

# Because google only returns at most 100 results for each search query,
# we must send many search queries to get more than 100 results.
# We need to set cdr (date range, in the form of "tbs=cdr:1,cd_min:{start_date},cd_max:{end_d
ate}") to tell google
# we want to search images in some date range (start_date, end_date),
# so as to get different results for each search query.
# CONFIGS[u'search_cdr_days'] is the days between cd_min and cd_max.
CONFIGS[u'search_cdr_days'] = 60

#This dict is used to search keywords. You can edit this dict to search for google images of
your choice. You can simply add and remove elements of the list.
#{class1:[list of related keywords], class2:[list of related keywords]...}
CONFIGS[u'search_keywords_dict'] = {'animal':[u'猫', 'cat', 'dog'],
                                   'fruit':[u'apple', u'banana']}

#This list is used to further add suffix to your search term. Each element of the list will help
you download 100 images. First element is blank which denotes that no suffix is added to the
search keyword of the above list. You can edit the list by adding/deleting elements from it. So
if the first element of the search_keyword is 'Australia' and the second element of keywords
is 'high resolution', then it will search for 'Australia High Resolution'
#aux_keywords = [' high resolution']

CONFIGS[u'save_dir'] = './downloads'

CONFIGS[u'output_prefix'] = 'download_urls'
CONFIGS[u'output_suffix'] = 'google'

print '==>Default CONFIGS:'
print CONFIGS
##### End of Default CONFIGS #####

```

```

##### Load config.json if there is one #####
if osp.exists(config_file):
    print "Load CONFIGS from " + config_file
    fp = open(config_file, 'a+')
    CONFIGS_loaded = json.load(fp, object_pairs_hook=OrderedDict)

    print '==>Loaded CONFIGS:'
    print CONFIGS_loaded

    for k,v in CONFIGS_loaded.iteritems():
        if k in CONFIGS:
            CONFIGS[k] = v

    fp.close()

    print '==>CONFIGS after loading:'
    print CONFIGS
##### End of Load config.json #####

CONFIGS[u'output_prefix'] = CONFIGS[u'output_prefix'] + '_'
CONFIGS[u'output_suffix'] = '_' + CONFIGS[u'output_suffix']

CONFIGS[u'save_dir'] = CONFIGS[u'save_dir']+'/'
if not osp.exists(CONFIGS[u'save_dir']):
    os.mkdir(CONFIGS[u'save_dir'])

##### End of CONFIGS #####

##### Functions to Load downloaded urls #####
def load_url_files(_dir, file_name_prefix):
    url_list = []

    ttl_url_list_file_name = osp.join(_dir, file_name_prefix + '_all.txt')

```

```

if osp.exists(ttl_url_list_file_name):
    fp_urls = open(ttl_url_list_file_name, 'r')      #Open the text file called database.txt
    print 'load URLs from file: ' + ttl_url_list_file_name

    i = 0
    for line in fp_urls:
        line = line.strip()
        if len(line)>0:
            splits = line.split('\t')
            url_list.append(splits[0].strip())
            i=i+1

    print str(i) + ' URLs loaded'
    fp_urls.close()
else:
    url_list = load_all_url_files(_dir, file_name_prefix)

return url_list

def load_all_url_files(_dir, file_name_prefix):
    url_list = []

    for file_name in os.listdir(_dir):
        if fnmatch.fnmatch(file_name, file_name_prefix + '*.txt'):
            file_name = osp.join(_dir, file_name)
            fp_urls = open(file_name, 'r')          #Open the text file called database.txt
            print 'load URLs from file: ' + file_name

            i = 0
            for line in fp_urls:
                line = line.strip()
                if len(line)>0:
                    splits = line.split('\t')
                    url_list.append(splits[0].strip())
                    i=i+1

```

```

        print str(i) + ' URLs loaded'
        fp_urls.close()

    return url_list

##### End of Functions to Load downloaded urls #####

##### Functions to get date/time strings #####
def get_current_date():
    tm = time.gmtime()
    date = datetime.date(tm.tm_year, tm.tm_mon, tm.tm_mday)
    return date

def get_new_date_by_delta_days(date, delta_days):
    delta = datetime.timedelta(delta_days)
    new_date = date+delta
    return new_date

#Make a string from current GMT time
def get_gmtime_string():
    _str = time.strftime("GMT%Y%m%d_%H%M%S", time.gmtime())
    return _str

#Make a string from current local time
def get_localtime_string():
    _str = time.strftime("%Y%m%d_%H%M%S", time.localtime())
    return _str

##### End of Functions to get date/time strings #####

##### Google Image Search functions #####
# Get Image URL list form Google image search by keyword
def google_get_query_url(keyword, file_type, cdr):
    url = None

    # if keyword is unicode, we need to encode it into utf-8
    if isinstance(keyword, unicode):

```

```

keyword = keyword.encode('utf-8')

query = dict(q = keyword,
             tbm = 'isch',
             tbs=cdrt+',ift:'+file_type)

#url = 'https://www.google.com/search?q=' + keyword + '&espv=2&biw=1366&bih=667&site=webhp&source=lnms&tbm=isch&sa=X&ei=XosDVaCXD8TasATltgE&ved=0CAcQ_AUoAg'
#url = 'https://www.google.com/search?as_oq=' + keyword + '&as_st=y&tbm=isch&safe=images&tbs=ift:jpg'
url = 'https://www.google.com/search?'+urlencode(query)

print "\t==>Google Query URL is: " + url
return url

#Downloading entire Web Document (Raw Page Content)
def google_download_page(url):
    version = (3,0)
    cur_version = sys.version_info
    if cur_version >= version:      #If the Current Version of Python is 3.0 or above
        import urllib.request      #urllib library for Extracting web pages
        try:
            headers = {}
            headers['User-Agent'] = "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36"
            req = urllib.request.Request(url, headers = headers)
            resp = urllib.request.urlopen(req)
            respData = str(resp.read())
            return respData
        except Exception as e:
            print(str(e))
    else:                          #If the Current Version of Python is 2.x
        import urllib2
        try:
            headers = {}

```

```
headers['User-Agent'] = "Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.27 Safari/537.17"
```

```
req = urllib2.Request(url, headers = headers)
```

```
response = urllib2.urlopen(req)
```

```
page = response.read()
```

```
return page
```

```
except:
```

```
return "Page Not found"
```

```
#Finding 'Next Image' from the given raw page
```

```
def google_images_get_next_item(s):
```

```
    start_line = s.find('rg_di')
```

```
    if start_line == -1:    #If no links are found then give an error!
```

```
        end_quote = 0
```

```
        link = "no_links"
```

```
        return link, end_quote
```

```
    else:
```

```
        start_line = s.find('"class="rg_meta"')
```

```
        start_content = s.find('"ou"',start_line+1)
```

```
        end_content = s.find('"ow"',start_content+1)
```

```
        content_raw = str(s[start_content+6:end_content-1])
```

```
        return content_raw, end_content
```

```
#Getting all links with the help of '_images_get_next_image'
```

```
def google_images_get_all_items(page):
```

```
    items = []
```

```
    while True:
```

```
        item, end_content = google_images_get_next_item(page)
```

```
        if item == "no_links":
```

```
            break
```

```
        else:
```

```
            items.append(item)    #Append all the links in the list named 'Links'
```

```
            time.sleep(0.1)    #Timer could be used to slow down the request for image
```

```
downloads
```

```
        page = page[end_content:]
```

```

return items

def google_search_keyword(keyword, file_type, cdr):
    query_url = google_get_query_url(keyword, file_type, cdr)
    raw_html = (google_download_page(query_url))
    time.sleep(0.1)
    image_url_list = google_images_get_all_items(raw_html)
    return image_url_list

##### End of Google Image Search functions #####

##### Functions to get real urls and download images #####

#Get real url of a input url
def get_real_url(url, loaded_urls):
    real_url = None
    response = None
    try:
        req = Request(url, headers={"User-Agent": "Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.27 Safari/537.17"})
        response = urlopen(req)

        real_url = response.geturl()
        print 'Real_url is: ' + str(real_url)

        if real_url in loaded_urls:
            print 'URL had been downloaded in previous '
            real_url = None

    except IOError as e:    #If there is any IOError
        print("IOError on url "+str(url))
        print e
    except HTTPError as e:  #If there is any HTTPError
        print("HTTPError on url "+str(url))
        print e
    except URLError as e:

```

```

        print("URL error on url "+str(url))
        print e

    if response:
        response.close()

    return real_url

def download_image(url, save_dir, loaded_urls=None):
    real_url = None
    response = None
    save_image_name = None
    try:
        req = Request(url, headers={"User-Agent": "Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.27 Safari/537.17"})
        response = urlopen(req)

        real_url = response.geturl()

        if loaded_urls and real_url in loaded_urls:
            print 'URL had been downloaded in previous searching'
            real_url = None
        else:
            img_name = hashlib.md5(real_url).hexdigest()
            save_image_name = save_dir + '/' + img_name + '.' + CONFIGS[u'search_file_type']

            print 'Try to save image ' + real_url + ' into file: ' + save_image_name
            output_file = open(save_image_name,'wb')
            data = response.read()
            output_file.write(data)

            #response.close()
    except IOError as e: #If there is any IOError
        print("IOError on url "+str(url))
        print e

```

```

except HTTPError as e: #If there is any HTTPError
    print("HTTPError on url "+str(url))
    print e
except URLError as e:
    print("URLError on url "+str(url))
    print e

if response:
    response.close()

return real_url, save_image_name
##### End of Functions to get real urls and download images #####

##### Main Program #####
t0 = time.time() #start the timer

#Download Image Links
i= 0

cur_date = get_current_date()
print "Today is: " + cur_date.strftime("%Y/%m/%d")

time_str = get_gmtime_string()

for class_name,search_keywords in CONFIGS[u'search_keywords_dict'].iteritems():
    print "Class no.: " + str(i+1) + " -->" + " Class name = " + str(class_name)

    class_urls_file_prefix = CONFIGS[u'output_prefix'] + '_' + str(class_name).strip()

    items = load_url_files(CONFIGS[u'save_dir'], class_urls_file_prefix)
    loaded_urls_num = len(items)
    print 'Loaded URLs in total is: ', loaded_urls_num

# load pre-saved download parameters, actually cd_min for date range

```

```

cd_max = cur_date

params_file = osp.join(CONFIGS[u'save_dir'], class_urls_file_prefix + '_params_' + CONFIG
S[u'output_suffix'] + '.txt')
print 'Loaded pre-saved download parameters from: ' + params_file
params_list = []
fp_params = open(params_file, 'a+')
for line in fp_params:
    line = line.strip()
    if line!='':
        params_list.append(line)
        print "\t-->loaded parameters: ", line

if len(params_list)>0:
    splits = params_list[-1].split('/')
    if len(splits)==3:
        cd_max = datetime.date(int(splits[0]), int(splits[1]), int(splits[2]))

cd_min = get_new_date_by_delta_days(cd_max, -CONFIGS[u'search_cdr_days'])
print 'cd_max: ', cd_max
print 'cd_min: ', cd_min

print ("Crawling Images...")

class_save_dir = osp.join(CONFIGS[u'save_dir'], class_urls_file_prefix + '_' + time_str + '_'
+ CONFIGS[u'output_suffix'])
if not osp.exists(class_save_dir):
    os.mkdir(class_save_dir)

output_all_urls_file = osp.join(CONFIGS[u'save_dir'], class_urls_file_prefix + '_all.txt')

fp_all_urls = open(output_all_urls_file, 'a+')

output_urls_file = osp.join(CONFIGS[u'save_dir'], class_urls_file_prefix + '_' + time_str + '_'
+ CONFIGS[u'output_suffix'] + '.txt')

```

```

fp_urls = open(output_urls_file, 'a+')

# if osp.exists(output_urls_file):
#     fp_urls = open(output_urls_file, 'a+')          #Open the text file called database.txt
#     for line in fp_urls:
#         items.append(line.strip())
# else:
#     fp_urls = open(output_urls_file, 'w+')          #Open the text file called database.txt
#
cdr_enabled = False

while True:
    if cdr_enabled:
        cdr = 'cdr:1,cd_min:{},cd_max:{}'.format(cd_min.strftime('%m/%d/%Y'), cd_max.strf
time('%m/%d/%Y'))
        print "==>Search for Images between " + cd_min.strftime("%Y/%m/%d") + \
            ' and ' + cd_max.strftime("%Y/%m/%d")
    else:
        cdr = ""
        print "==>Search for Images in any time"

    j = 0

    # Google only return 100 images at most for one search. So we may need to try man
y times
    while j<len(search_keywords):
        print "\t==>Class name=" + str(class_name) + ', search keywords=' + search_key
words[j]

        keyword = search_keywords[j]#.replace(' ','%20')

#         # if keyword is unicode, we need to encode it into utf-8
#         if isinstance(keyword, unicode):
#             keyword = keyword.encode('utf-8')

#         query = dict(q = keyword,

```

```

#             tbm = 'isch',
#             tbs=tbs+',ift:'+CONFIGS[u'search_file_type'])
#
#         #url = 'https://www.google.com/search?q=' + keyword + '&espv=2&biw=1366&bih=6
67&site=webhp&source=lnms&tbm=isch&sa=X&ei=XosDVaCXD8TasATItgE&ved=0CAcQ_AUoA
g',
#         #url = 'https://www.google.com/search?as_oq=' + keyword + '&as_st=y&tbm=isch&s
afe=images&tbs=ift:jpg'
#         url = 'https://www.google.com/search?'+urlencode(query)
#
#         print "\t=>Query URL is: " + url
#
#         raw_html = (download_page(url))
#         time.sleep(0.1)
#         new_items = _images_get_all_items(raw_html)

new_items = google_search_keyword(keyword, CONFIGS[u'search_file_type'], cd
r)

for url in new_items:
    #real_url = get_real_url(url)
    real_url, save_name = download_image(url, class_save_dir, items)

    if real_url and real_url not in items:
        items.append(real_url)
        fp_all_urls.write(real_url + '\t' + save_name + "\n")
        fp_urls.write(real_url + '\t' + save_name + "\n")

fp_all_urls.flush()
fp_urls.flush()

print 'len(items)=', len(items)
j = j + 1

if cdr_enabled:

```

```

fp_params.write('{}{/}/{}\n'.format( cd_min.year, cd_min.month, cd_min.day))
cd_max = cd_min
cd_min = get_new_date_by_delta_days(cd_max, -CONFIGS[u'search_cdr_days'])

else:
    fp_params.write('{}{/}/{}\n'.format( cd_max.year, cd_max.month, cd_max.day))
    cdr_enabled = True

fp_params.flush()

print 'len(items)=', len(items)
if len(items) >= loaded_urls_num + CONFIGS[u'num_downloads_for_each_class']:

    break

fp_params.close()
fp_all_urls.close()
fp_urls.close()

#print ("Image Links = "+str(items))
print ("Total New Image Links = " + str(len(items) - loaded_urls_num))
print ("\n")
i = i+1

t1 = time.time()    #stop the timer
total_time = t1-t0   #Calculating the total time required to crawl, find and download all
the links of 60,000 images
print("Total time taken: "+str(total_time)+" Seconds")

print("\n")
print("===All are downloaded")
#----End of the main program ----#

```

주관연구기관 자체평가의견서

1. 과제 현황

		과제번호			16CTAP-C114867-01			
사업 구분		국토교통기술촉진연구사업						
연구 분야		창의도전연구분야			과제 구분		단위	
사업명		국토교통기술촉진연구사업					주관	
주관과제		오픈소스 하드웨어 기반 독거노인 및 장애인 실내 환경 모니터링 시스템 개발 및 클라우드를 통한 지속적 자가 학습 딥러닝 시스템 구현			주관연구책임자		염세경	
세부과제명					과제 유형		(기초, 응용, 개발)	
연구기관					연구책임자			
연구 기간 연구비 (천원)		연차	연구개발기간	정부출연금	민간부담금	정부 외 출연금	상대국 부담금	계
		1차 연도	12개월	50,000			50,000	
		2차 연도	12개월	50,000			50,000	
		3차 연도						
		4차 연도						
		5차 연도						
		계	24개월	100,000			100,000	
참여기업								
상대국					상대국 연구기관			

※ 총연구 기간이 5차 연도 이상인 경우 셀을 추가하여 작성

2. 평가일: 2018년 12월 22일

3. 평가자(연구책임자):

소속	직위	성명
동국대학교	조교수	염세경

4. 평가자(연구책임자) 확인 :

본인은 평가 대상 과제에 대한 연구 결과에 대하여 객관적으로 기술하였으며, 공정하게 평가하였음을 확약하며, 이 자료가 전문가 및 전문기관 평가 시에 기초 자료로 활용되기를 바랍니다.

확약	염세경 
----	---

I. 연구 개발 실적

1. 연구 개발 결과의 우수성/창의성

■ 등급: (아주 우수, 우수, 보통, 미흡, 불량)

- 단순히 낙상의 상태를 평가하는 것이 아니며 노인의 상태(서기/앉기/눕기)를 딥러닝 방법을 적용하여 인식률을 높였음
- 생체신호를 동시에 측정하여 상태에 따른 신체의 변화 데이터값을 이용하여 이미지에 의한 상태 판단결과를 검증하였음
- 연구책임자를 포함한 전문가 5인 평가 결과: 4.15로 우수성/창의성에 있어 아주 우수라고 판단

2. 연구 개발 결과의 파급 효과

■ 등급 : (아주 우수, 우수, 보통, 미흡, 불량)

- 딥러닝 방법론을 이용하여 사람의 자세 자체를 판단하는 것은 처음으로 시도된 연구이며 노인이나 장애인의 움직임을 자동으로 파악하고 이상이나 위급사항을 감지하는 것 뿐만 아니라, 움직이는 영상에 대한 실시간 이미지 판독기술의 구현에 더 큰 의미가 있으며 유사한 서비스를 개발할 경우 적용이 가능할 것으로 판단됨
- 무선 생체시그널 동시 측정시스템을 동시에 구현하여 영상이미지 판독의 결과를 보완할 수 있는 가능성을 높였다고 판단됨
- 전문가 5인 평가 결과: 4.82로 연구개발성과의 파급효과는 '아주 우수' 하다고 판단됨

3. 연구 개발 결과에 대한 활용 가능성

■ 등급 : (아주 우수, 우수, 보통, 미흡, 불량)

- 처음엔 개인정보보호를 위해 개인의 별도 서버를 두지 않는 것으로 개발하였으나, 딥러닝을 적용한 후 이미지 처리 때문에, 별도의 서버를 두어 개발을 진행하였음
- 쿠킹헥스라는 바이오 센서의 데이터를 수집하고 분석할 수 있는 시스템이 동시에 개발되어 접촉식 센서를 활용한 기타 연구의 플랫폼으로 사용 가능함
- 전문가 5인 평가 결과: 4.62로 연구개발성과의 우수성/창의성에 있어 '매우 우수' 라고 판단됨

4. 연구 개발 수행 노력의 성실도

■ 등급: (아주 우수, 우수, 보통, 미흡, 불량)

- 실루엣을 이용한 기존의 classification 방식의 한계를 먼저 파악하고 2차년도에 적용할 딥러닝 방법론을 앞서 적용하여 1차년도 기술구현의 수준을 높였다고 판단됨
- 특히 바이오센서는 개별 측정 센서가 대부분이지만, 동시에 무선 측정 가능하도록 구현하기 위해 해외 연구조사를 성실히 수행하였고 연구에 가장 적절한 방식인 오픈소스를 이용해 진행하였음
- 전문가 5인 평가 결과: 3.74로 연구개발성과의 수행의 성실도는 ‘우수’ 라고 판단됨

5. 공개 발표된 연구 개발 성과(논문, 지적소유권, 발표회 개최 등)

■ 등급: (아주 우수, 우수, 보통, 미흡, 불량)

- 본 연구 결과물에 대하여 SCIE 저널 두편을 출간하였으며, 2018년 컴퓨터 종합 학술대회에서 위 논문을 공개 발표하였음
- 2건의 특허를 출원중에 있음
- 개발된 내용의 저서와 함께 재직자들의 교육활동을 진행하여 결과를 확산하는데 노력하고 있음
- 전문가 5인 평가 결과: 4.23로 연구개발성과는 아주 우수라고 판단됨

II. 연구 목표 달성도

세부 연구 목표 (연구 계획서의 목표)	비중 (%)	달성도 (%)	자체 평가
오픈 하드웨어 기반 환자 자세 분석 및 노인 활동 패턴 분석 알고리즘 개발	40%	100%	• 기존 실루엣 이미지를 이용한 판독방식의 한계를 넘을 수 있는 딥러닝을 이용한 이미지 판독 기술을 구현하였음 • 이미지만 가지고 자세를 평가하는 것은 구현이 되었으나 이를 사양이 낮은 오픈소스 하드웨어와 무선 네트워크를 통해 실시간 전송할 수 있는 시스템 개발이 진행되어야 그 의미가 있다고 할 수 있음
가속도 센서 및 심박측정 센서를 통한 움직임 모니터링 기술 개발	20%	100%	• 가속도센서(움직임)를 이용하여 이미지 판독 결과를 동시에 검증하기는 했지만, 심박, 심전도, 온도센서의 데이터는 상황에 따라 크게 변하지 않아 이미지 판독결과를 보완하는데 적용기 어려웠음 • 그러나 시스템상으로 구현되어 있으므로 데이터가 함께 입력될 경우 언제든 활용 가능하도록 구현하였음
딥러닝을 활용한 CNN 기반의 정확도 향상 알고리즘 및 클라우드 솔루션의 구축	40%	100%	• 촬영된 이미지 자체를 딥러닝 방법론으로 구현하여 인식 정확도를 높였음 • 클라우드 시스템의 구축을 통해 빅데이터 솔루션을 구현하고 이를 딥러닝에 활용하였음
합계	100%		

III. 종합 의견

1. 연구 개발 성과에 대한 종합의견

- 본 기술의 개발을 통해 감시형태의 모니터링 시스템에서 능동적이고 적극적인 서비스 제공 가능하다고 판단됨
- 특히 이동궤적추적, 관리 대상자의 비정상행동 인식을 통한 상황 통제 가능으로 차별화된 새로운 응용 서비스 개발 가능할 것이라고 기대됨
- 딥러닝 방법론을 통해 어느 산업분야에서나 누구나 쉽게 접근할 수 있도록 범용성을 높여 개발하였음
- 노인이나 장애인 뿐만 아니라, 사무실이나 학교, 일반인들의 데이터 수집을 통해 이를 분석하고 제공하여, 향후 연구시 다양한 분야에서 사용할 수 있도록 개발되어야 함

2. 평가 시 고려할 사항 또는 요구사항

- 시스템 고도화를 위해 직접 노인자택에 설치하여 실험하는 것이 필요하지만, 개인 정보보호의 문제등으로 파일럿 테스트만 실시하였음
- 현재 유 씨티 실용화 사업에서 본 연구를 활용할 수 있는 기회를 국토교통부에서 제공해 준다면 결과의 가치가 더욱 높아질 것으로 판단됩니다.

3. 연구 결과의 활용 방안 및 향후 조치에 대한 의견

- 기술수요처 및 현장 적용방안
 - 독거노인, 장애인, 환자, 소외계층의 행동 모니터링
 - 실버타운, 복지센터, 병원 등(Reference Site)에 대한 적용
- 활용분야 및 활용방안
 - 노인이나 환자, 어린이 활동 패턴분석 및 분류 연구를 통한 행동 위험 감지 시스템에 응용
 - 실내공간조성(1인 가구, 독거노인층, 실버타운, 장애인 가정) 레퍼런스자료로 활용
 - 치매노인, 정신질환자들의 행동 파악을 통한 관리 처방 가이드라인 응용
- 향후조치 방안
 - 추후 상품화 단계에서의 국내 SW및 HW 업체와 협력하여 상업적인 톨의 형태로 재개발
 - 과제수행 중 발생한 지적재산권, 보고서, 특허 등에 대한 기술료 수입 획득
 - 추후 상용시스템 개발완료시 서비스권, 제품 판매권 등도 확보 예정

IV. 보안성 검토

- 보안성이 필요하지 않은 과제임