

박사학위논문

인공지능 기반의 자세 추정을 통한
스마트 헬스케어에 관한 연구

2020년

한 성 대 학 교 일 반 대 학 원

스마트융합컨설팅학과

스마트융합제품전공

하 태 용

박사학위논문
지도교수 이후진

인공지능 기반의 자세 추정을 통한 스마트 헬스케어에 관한 연구

A Study on the Smart Healthcare through Human
Pose Estimation Based on Artificial Intelligence

2020년 6월 일

한 성 대 학 교 일 반 대 학 원

스마트융합건설팅학과

스마트융합제품전공

하 태 용

박사학위논문
지도교수 이후진

인공지능 기반의 자세 추정을 통한 스마트 헬스케어에 관한 연구

A Study on the Smart Healthcare through Human
Pose Estimation Based on Artificial Intelligence

위 논문을 공학 박사학위논문으로 제출함

2020년 6월 일

한 성 대 학 교 일 반 대 학 원

스마트융합건설팅학과

스마트융합제품전공

하 태 용

하태용의 공학 박사학위논문을 인준함

2020년 6월 일

심사위원장 _____(인)

심 사 위 원 _____(인)

심 사 위 원 _____(인)

심 사 위 원 _____(인)

심 사 위 원 _____(인)

국 문 초 록

인공지능 기반의 자세 추정을 통한 스마트 헬스케어에 관한 연구

한 성 대 학 교 일 반 대 학 원
스 마 트 융 합 컨 설 텅 학 과
스 마 트 융 합 제 품 전 공
하 태 용

각종 첨단 정보통신 기술을 활용하여 언제 어디서나 건강관리 서비스를 받을 수 있는 스마트 헬스케어가 부상하고 있다. 이미 오래전부터 헬스케어 영역에서는 IT 기술과의 융합적인 시도가 있었으며 최근의 변화가 새로운 것은 아니다. 스마트폰이 등장하면서 시간적, 공간적인 제약을 벗어난 건강관리 서비스 제공이 가능해지고 있고, 인공지능(Artificial Intelligence: AI)과 빅데이터(Big Data) 등의 첨단 분석 기술과 융합으로 새로운 건강관리 서비스의 질적 수준이 향상되고 있다.

인공지능 기술의 발전을 통해 미래의 헬스케어 건강관리 서비스는 신체와 관련된 많은 건강 정보를 스스로 분석하고 학습하여 질병으로 인한 발현 시기를 예측할 수 있다. 그리고, 개인 맞춤형 진단 및 생활습관 정보 제공을 통해 질병 발현 예방에 도움을 줄 수 있을 것으로 기대된다.

인공지능 기반의 스마트 헬스케어는 건강 의료 서비스 제공 공간이 가정, 피트니스센터, 길거리 등 실생활 전 영역으로 확장되어 개인의 건강 정보가 평생 동안 축적됨에 따라 기존 의료 시스템과의 호환을 통해 스마트 헬스케

어가 가능하리라 본다. 또한, 의료 서비스가 단발성 질병 치료에서 이제는 평생 치료의 개념으로 확장하는 연구가 필요하리라 판단된다.

본 연구의 인공지능 자세 추정 기반의 스마트 헬스케어 시스템 개발을 통하여 “언제나, 어디서나” 건강관리가 가능하도록 하기 위하여, 기존 헬스케어 관련 운동 Application의 단순히 따라 하기 형이나 기존 제품의 단순히 데이터를 보여주는 형태를 취하다 보니 실사용자의 흥미 및 몰입에 도움 되지 않는다. 이를 개선하고자 운동 형태와 사용자의 움직임 감지하여 운동량에 비례한 소모 칼로리를 계산하여 보여주는 기능이 있다면 사용자 입장에서 보다 실제적인 모니터링을 통한 운동 효과를 볼 수 있을 것이다.

본 연구에서는 자세 추정을 통한 딥러닝(Deep Learning: DL) 학습으로 LSP DataSet을 사용하여 PoseNet으로 자세 추정 학습을 해 보았으며, 다른 자세 추정 주요 모델들과 비교하여 눈, 코, 귀 KeyPoint들에 대한 결과가 타 모델과 해석의 차이로 다소 차이가 나지만, 나머지 모든 부위에서는 성능이 크게 떨어지지 않음을 확인하였다. 이에 따른 최적 모델로 PoseNet을 선택하고, 이를 기존의 자세 추정 알고리즘들과 비교 분석을 통해 입증하였으며, 인식을 위한 기본 운동 동작인 손뼉(박수)치기라는 최적의 운동 모델을 설정하여 자바스크립트로 동작 알고리즘을 구현하였다. 사용자의 동작과 운동량을 파악하고 운동량에 대한 신체활동 목록 상의 해당 칼로리를 이용하여 스마트 헬스케어 운동관리 시스템을 구현하였다. 또한, 스마트 헬스케어 시스템 구현을 통해 사용자의 소모 칼로리를 알 수 있었고 기존 디바이스를 활용하여 시간과 비용을 절감할 수 있다는 것을 확인하였다.

본 연구의 성과를 바탕으로 인공지능 동작 인식을 활용한 스마트 헬스케어의 다양한 건강관리 서비스와 시스템에 접목하는 활용이 가능할 것으로 기대된다.

【주제어】 AI, 헬스케어, CNN, MET, 딥러닝

목 차

제 1 장 서 론	1
제 1 절 연구 배경	1
제 2 절 연구 필요성 및 목적	6
1) 연구 필요성	6
2) 연구 목적	10
제 3 절 연구 방법과 범위	11
1) 연구 방법	11
2) 연구 범위	12
제 2 장 이론적 배경 및 선행 연구	13
제 1 절 스마트 헬스케어 연구	13
1) 스마트 헬스케어 정의	13
2) 스마트 헬스케어 구조	16
제 2 절 자세 추정(Pose Estimation) 연구	20
1) 인공지능(AI) 정의	20
2) 합성곱 신경망(CNN)	21
3) PoseNet Model	24
4) Human Pose Estimation 주요 모델	33
제 3 절 신체활동목록(MET) 연구	53
1) 신체활동에 대한 연구	53
2) 신체활동의 측정 및 수준	54
3) 신체활동목록에 대한 연구	57
제 3 장 스마트 헬스케어 모델 설계	59
제 1 절 자세 추정 모델 설계	59
1) 전체 구성도	59
2) 자세 추정 모델	59

3) 자세 추정 방법	60
제 2 절 동작 인식 모델 설계	61
1) PoseNet 동작 인식 모델 범위	61
2) PoseNet 동작 인식 모델 설정	61
3) PoseNet 동작 인식 모델 방법	62
제 3 절 신체활동 목록과 MET 설계	64
1) 운동량 측정 범위	64
2) 운동량 측정 방법	65
제 4 절 스마트 헬스케어 시스템 설계	66
1) 애플리케이션 구성도	66
2) 시스템 흐름도	67
3) 데이터베이스 구조	67
4) 스마트 헬스케어 개발 환경	68
5) 스마트 헬스케어 시나리오	70
제 4 장 스마트 헬스케어 구현 및 고찰	72
제 1 절 스마트 헬스케어 시스템 구현	72
1) 자세 추정 학습 결과	72
2) 동작 인식 개발 구현	89
3) 신체활동 목록과 MET 연계 구현	94
4) 스마트 헬스케어 시스템	97
5) 스마트 헬스케어 추가 운동	100
제 2 절 스마트 헬스케어 운동관리 효용성	101
1) 일반적 기대효과	101
2) 기술적 측면	101
3) 연구의 성과	102
제 5 장 결 론	104
참 고 문 헌	106
ABSTRACT	117

표 목 차

[표 1-1] 헬스케어 패러다임의 변화	2
[표 1-2] 의료 + IT 융합트렌드 변화	3
[표 1-3] 인공지능 헬스케어 세계 및 국내 시장 규모 전망	5
[표 1-4] 세계 인공지능 헬스케어 시장 규모	5
[표 1-5] 기술별 세계 인공지능 헬스케어 시장 규모	6
[표 1-6] 우리나라 국민의 건강위험요인의 질병 비용에의 기여도	10
[표 2-1] 스마트 헬스케어의 정의	14
[표 2-2] 재택 개인생활 및 건강관리 시스템 분야 연구 동향의 특징	19
[표 2-3] Float당 특징	27
[표 2-4] 각 신체와 ID	32
[표 2-5] DeepCut Branch and Cut Algorithm	50
[표 2-6] Deepcut 단계별 인식 관절 후보군	51
[표 2-7] DeeperCut MPII DataSet	52
[표 2-8] DeeperCut LSP DataSet	52
[표 2-9] 연령별 신체활동별 권장량	54
[표 2-10] 신체활동과 에너지 소비량 평가 방법과 단위	55
[표 2-11] 신체활동 내용과 범위	58
[표 3-1] 자세 추정 모델 목록	60
[표 3-2] 디바이스(스마트폰) 시스템 사양	61
[표 3-3] MET 기준정보 데이터베이스 저장 내용	65
[표 3-4] 운동 동작 데이터베이스 저장 내용	65
[표 3-5] 메뉴 구성도	67
[표 3-6] PC서버 개발 사양	68
[표 4-1] 모델 학습 환경 비교	74
[표 4-2] PoseNet 모델 구조에 따른 파라미터 정보	75
[표 4-3] LSP & MPII DataSet에서 PCK 결과	76
[표 4-4] LSP DataSet에서 PCK 결과(Head 제외)	77
[표 4-5] PoseNet 관점에서의 결과 분석	89

그 림 목 차

[그림 1-1] 스마트 헬스케어 부상 배경	1
[그림 1-2] 헬스케어 분야의 기술혁신, 변화, 시장요구의 결합	7
[그림 1-3] 스마트 헬스케어 분야의 진화 방향	8
[그림 1-4] 의료산업 생태계	8
[그림 1-5] 인공지능 헬스케어의 개념	9
[그림 2-1] 스마트 헬스케어 산업구조	14
[그림 2-2] 스마트 헬스케어 개념도	15
[그림 2-3] 시대별 디지털 헬스케어의 기술 개념 등장	16
[그림 2-4] 모니터링 기반 헬스케어 정보시스템의 3 계층 서비스 체계	17
[그림 2-5] 헬스케어 4계층 서비스 영역 구조	18
[그림 2-6] 헬스케어 연구 구조 개념도	19
[그림 2-7] 딥러닝, 기계학습, 인공지능 비교	21
[그림 2-8] 인공 신경망의 구조	22
[그림 2-9] 심층 신경망의 구조	22
[그림 2-10] 합성곱 신경망의 구조	23
[그림 2-11] Convolutional and Pooling Layers	24
[그림 2-12] PoseNet을 사용한 단일 인물 자세 감지기 파이프 라인	28
[그림 2-13] PoseNet이 반환한 17개의 자세 KeyPoint	30
[그림 2-14] Architecture and Receptive Fields of CPM	37
[그림 2-15] Convolutional Pose Machines 구조	37
[그림 2-16] CPM에서 생성되는 단계별 신뢰지도	38
[그림 2-17] Comparisons on 3-stage Architectures on the LSP	38
[그림 2-18] Quantitative Results on the MPII DataSet	39
[그림 2-19] Quantitative Results on the LSP DataSet	39
[그림 2-20] Pose Estimation Stacked Hourglass Modules	40
[그림 2-21] Residual Module들로 구성된 “Hourglass” Module	41
[그림 2-22] Residual Module	42
[그림 2-23] Residual Module Algorithm	43

[그림 2-24] 다수의 Hourglass Network를 쌓아놓은 구조	43
[그림 2-25] Results on MPII & FLIC	45
[그림 2-26] Comparison of Validation Accuracy as Training Progresses	46
[그림 2-27] Left: Example Validation, Right: Validation Accuracy	46
[그림 2-28] DeeperCut이 인식한 사람들의 자세 예시	47
[그림 2-29] Deeper Cut Model Architecture	48
[그림 2-30] Deeper Cut Model Architecture 수식	49
[그림 2-31] 신체활동 관련 개념도	53
[그림 2-32] 신체활동과 에너지 소비량 평가를 위한 개념	55
[그림 2-33] 신체활동 수준 분류	56
[그림 2-34] 신체활동 목록	57
[그림 2-35] 신체활동 목록 분류	58
[그림 3-1] 자세 추정에 따른 동작 반영 스마트 헬스케어 모형	59
[그림 3-2] PCK 0.2	60
[그림 3-3] PoseNet can Detect Human Figures	62
[그림 3-4] 눈, 코, 귀 포인트로 머리 끝 포인트 구한 방법	63
[그림 3-5] 손뺌(박수) 횟수 카운트 순서도	64
[그림 3-6] 스마트 헬스케어 MET 인터페이스 시스템 환경	66
[그림 3-7] 시스템 구성도	66
[그림 3-8] 시스템 흐름도	67
[그림 3-9] 데이터베이스 구조	68
[그림 4-1] Convolutional Pose Machine 모델 학습 과정	73
[그림 4-2] Stacked Hourglass Networks 모델 학습 과정	73
[그림 4-3] PoseNet의 Single-Person Pose Estimation	74
[그림 4-4] LSP DataSet에서 PCK@0.2 결과(전체)	75
[그림 4-5] LSP DataSet에서 PCK@0.2 결과(부분)	76
[그림 4-6] LSP DataSet에서 PCK@0.2 결과(Head 제외)	77
[그림 4-7] Matlab 코드	78
[그림 4-8] Visual Code Setup	79

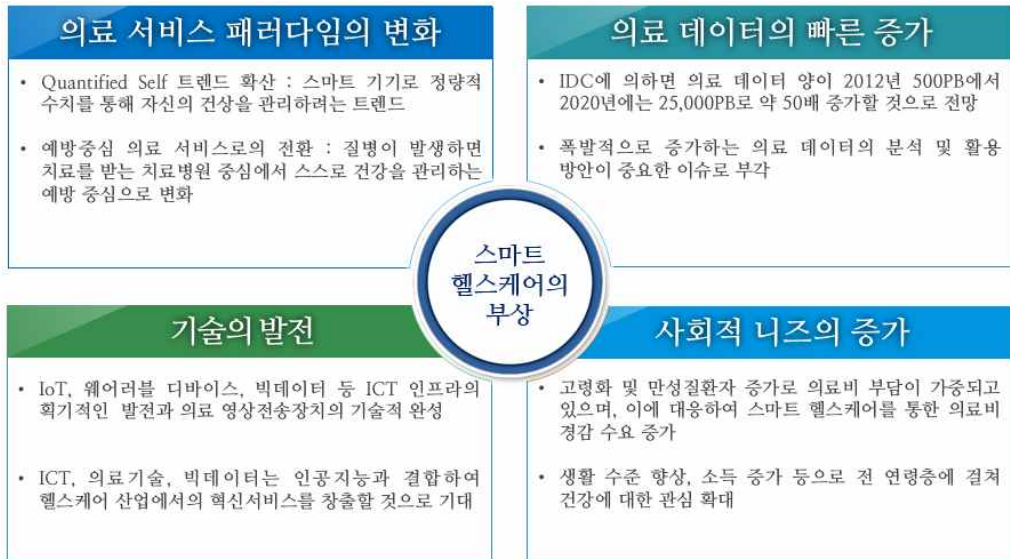
[그림 4-9] 다운로드 한 LSP	79
[그림 4-10] Via Script Tag 코드	80
[그림 4-11] Gram기준 MobileNet & ResNet 이미지 넣기	81
[그림 4-12] Script 부분의 Array에 이미지 넣기	81
[그림 4-13] PoseNet Architecture 설정	82
[그림 4-14] Pose에 대한 KeyPoint x, y 좌표	82
[그림 4-15] Yarn Watch 이미지	83
[그림 4-16] Console 창 좌표	83
[그림 4-17] KeyPoint 값 출력 순서	84
[그림 4-18] PoseNet의 출력 값 리다이렉트	84
[그림 4-19] PoseNet의 출력 값 .json 파일	85
[그림 4-20] Matlab 실행 화면	85
[그림 4-21] PoseNet KeyPoint과 LSP DataSet KeyPoint	86
[그림 4-22] Reshape 함수 사용	86
[그림 4-23] LSP DataSet의 Annotation 포맷 .mat	87
[그림 4-24] NECK.m 파일	88
[그림 4-25] MobileNet & ResNet 수행한 모델	88
[그림 4-26] PoseNet의 ResNet & MobileNet 성능 비교	89
[그림 4-27] DataSet분석	91
[그림 4-28] 손목 관점으로 DataSet분석	92
[그림 4-29] 손뺍(박수) 친 횟수를 세는 코드	93
[그림 4-30] 손뺍(박수) 동작 인식 화면	94
[그림 4-31] MET 모델 설계에 따른 구현 결과	94
[그림 4-32] 운동 등록 화면	95
[그림 4-33] 운동 항목 선택 화면	95
[그림 4-34] 데이터베이스에 저장된 운동 종목	96
[그림 4-35] 손뺍(박수)치기 인식 화면	96
[그림 4-36] 손뺍(박수)치기 소모 열량	97
[그림 4-37] 홈 화면	98

[그림 4-38] 대사 관리 화면	98
[그림 4-39] 식사 관리 화면	99
[그림 4-40] 운동 관리 화면	99
[그림 4-41] 추가 운동 동작 관리	100

제 1 장 서 론

제 1 절 연구 배경

최근 경제성장과 시대적 흐름의 변화에 따른 고령 인구의 증가, 질병 구조의 변화, 건강증진 및 신체단련으로 인한 욕구 증대로 건강관리에 대해서 현대인의 주요 관심 분야로 자리매김하고 있으며, 시간과 공간에 구애받지 않고 건강관리를 할 수 있는 스마트 헬스케어가 미래의 고부가가치 산업으로 부상하고 있다(박선주, 2014). 즉, 바꾸어 말하면 각종 첨단 정보통신기술(ICT) 기술을 활용하여 언제 어디서라도 건강관리 서비스를 받을 수 있는 스마트 헬스케어에 대한 관심이 현재 확대되고 있으며, 배경으로는 [그림 1-1]에서 네 가지 정도로 구분할 수 있다(삼정 KPMG, 2018).



[그림 1-1] 스마트 헬스케어 부상 배경 (IDC; Worldwide Black Book)

21세기에 접어들면서 헬스케어 시스템은 3.0의 시대를 맞이하게 되었다. 단순히 수명만을 연장하는데 만족하지 않고, 일생 동안 건강하게 유지하고 싶

어 한다. 헬스케어 산업 패러다임의 변화로 [표 1-1]처럼 헬스케어 2.0 시대를 지나면서 질병을 치료하는 기술이 발전하고 기대수명이 증가하였다면, 현재 헬스케어 3.0 시대에는 증가한 기대수명과 건강수명의 격차를 좁히는 것이 중요하게 되었다. 이 때문에 지속적인 일상 관리와 예방을 통해 발병을 사전에 방지하는 것이 중요해지기 시작했다(유선길, 2018).

[표 1-1] 헬스케어 패러다임의 변화(하나금융경영연구소, 2016)

구 분	1.0 공중보건 시대	2.0 질병치료 시대	3.0 건강수명 시대
시 대	• 18 ~ 20C 초	• 20C 초 ~ 말	• 21C 이후
기 술	• 인두접종 개발	• 페니실린 발견	• 인공지능 프로젝트
목 적	• 전염병 예방과 확산방지	• 질병의 치료 및 치유	• 질병 예방 및 관리로 건강한 삶 영위
지 표	• 전염병 사망률	• 기대수명, 질병 사망률	• 건강수명, 의료비 절감
공 급	• 국가	• 제약, 의료기기, 회사, 병원	• 기존 공급자 + IT, 전자, 건설, 자동차 회사 등
수 요	• 전 국민	• 환자	• 환자 + 정상인
산업 변화	• 예방접종, 상하수도 보급 • 청진기, 엑스레이 발명 • 의사 양성체계 확립	• 제약, 기기, 병원 산업화 • 신약 및 치료법개발	• 유전자 조기 진단 맞춤 치료제 등장 • u-헬스 보급

현재 헬스케어에 대한 영역은 이미 오래전부터 바이오기술(BT)과 정보기술(IT) 과의 융합 시도가 됐기 때문에 최근의 변화적 시도가 아주 새로운 것은 아니라고 본다. 하지만 스마트폰이라는 새로운 디바이스의 등장으로 말미암아 시간적, 공간적 제약을 벗어난 새로운 건강관리 서비스 제공이 가능해지고 있고, 인공지능과 빅데이터 등 새로운 첨단 분석 기술과의 융합으로 인해 의료의 질적 수준이 향상되고 있다. 이와 함께, IT와 헬스케어 관련 기술의 융합 속도가 빨라지면서 관련된 시장도 계속적으로 성장하고 있다. 헬스케어 IT 시장은 병원정보시스템 등 기존의 시장에 더해, 인공지능, 빅데이터, 클라

우드, 사물인터넷(Internet of Things: IoT) 기술 등과 접목된 분야로 계속 확대되고 있다(고은지, 2018).

의료산업의 패러다임을 변화시키는 핵심으로 [표 1-2]에서 하나는 BT와 ICT와 같은 기술의 발전과 융합에 따른 착용 컴퓨터, 사물인터넷, 클라우드, 빅데이터와 같은 기술의 발전과 융합은 의료 혁신을 더욱 가속하는 원동력이 될 것이다(하나금융경영연구소, 2016). 또한 4차 산업혁명의 핵심기술인 인공지능, 사물인터넷, 웨어러블 디바이스 등이 의료분야 밖에서 빠르게 발전하고 있는 디지털 기술과 의료분야에 접목하면서 더욱 광범위한 변화가 일어나고 있으며, 이에 대한 정책적, 기술적, 사회적 대응 전략이 필요한 시점이다(스마트 헬스케어 의료기기 기술표준 전략 보고서, 2018).¹⁾

[표 1-2] 의료 + IT 융합 트렌드 변화(하나금융경영연구소, 2016)

구분	e - 헬스	u - 헬스케어	스마트 헬스케어	IT 헬스
서비스	• 디지털병원, 의료정보학	• e-헬스 + 원격의료, 만성질환자 관리	• u-헬스 + 운동, 식사량 등 건강관리	• 스마트 헬스 + 개인 맞춤형 관리, 근거 중심 의학, 예방 중심, 자가 관리
Player	• 병원	• 병원, IT 기업	• 병원, IT기업, 서비스 업체	• 보험사, 서비스 업체 등 이해관계자
이용자	• 의료인	• 의료인, 환자	• 의료인, 환자, 일반인	• 의료인, 환자, 일반인, 정부, 기업
시스템	• 병원운영시스템	• 의무기록(EMR), 건강기록(EHR)	• 건강기록(EHR)	• IoT 기반의 PHR, 클라우드, 빅데이터, 인공지능

스마트 헬스케어 서비스는 모바일 및 웨어러블 디바이스를 통해 측정된 생체신호 및 생활 습관 환경 정보에 따라 개인별 건강 상태 평가와 건강위험도 등을 분석함으로써 원격 모니터링을 통한 의약 및 운동, 영양 처방 등 개인 맞춤형 보건 의료 서비스를 제공하게 된다(배장은, 2015). 또한, 인공지능 기술의 발전을 통해 미래 헬스케어 서비스는 많은 양의 유전자 정보를 스스로 분석하고 학습하여 질환 발현 시기를 예측하거나, 개인 맞춤형 진단 및 생활 습관 정보 제공을 통해 질병 발현 예방에 도움을 줄 수 있을 것으로 기대

1) u-헬스(Ubiquitous Health), m-헬스(Mobile Health), e-헬스(Electronic Health)

되며, 헬스케어 분야의 인공지능 역할로는 웨어러블 기기를 통한 실시간 건강 관리, 영상인식 및 진단보조, 의료 통계 결과의 효율적 관리를 할 수 있다 (KISTI, 2016; Frost & Sullivan, 2015).

스마트 헬스케어로의 변화는 질병에 걸리지 않았더라도 본인의 운동량과 식사량, 건강행태 등을 각종 스마트 기기를 통해 일상적으로 점검하고 질병 발현 이전에 문제가 되는 행태를 수정할 수 있는 ‘소비자 중심의 서비스 제공’으로 옮겨가는 것이다(이진수, 2014; 2018).

국내의 경우 고령화에 따라 2060년 국민 의료비 지출은 2016년 대비 35.6% 이상 증가할 것으로 예상되기에, 스마트 헬스케어 기술을 이용한 헬스케어 서비스 분야의 효율성과 효과성을 개선하고 이를 통해 정부와 개인의 부담을 줄일 필요성이 제기되고 있다(김수범, 2018). 따라서, 헬스케어 중 스마트 헬스 데이터 분석 관련 국내 시장은 연평균 23.0%로 세계시장보다 빠르게 성장할 것으로 예상되며, 2020년에는 3조 4,994억 원에 달할 것으로 전망한다(전성표, 2018).

글로벌 스마트 헬스케어 시장규모는 지속적인 성장을 보일 전망이다. 한국 보건산업진흥원에 따르면 2014년 기준 210억 달러에 머물렀던 글로벌 스마트 헬스케어 시장규모가 2020년에는 1,015억 달러 규모가 되면서 약 4.8배의 성장을 보일 것으로 전망했으며, 스마트 헬스케어의 주요 기술 분야로는 빅데이터 45.9%, 인공지능 35.3%, 사물인터넷 14.8% 등으로 나타났다(삼정 KPMG, 2018).

Markets and Markets(2016) 보고서에 따르면 [표 1-3]에 2015년에 인공지능 헬스케어 세계 시장규모는 71.3백만 달러에서 2020년에는 754.7백만 달러로 성장할 것으로 크게 예상된다. 특히 전체 인공지능 시장의 응용 분야 중 국내 시장은 연평균 성장률(CAGR)이 2015년 17.9억 원에서 2020년 256.4억 원으로, 세계 인공지능 헬스케어 시장의 CAGR보다 높은 70.4%의 성장률을 보이며 빠르게 성장할 것으로 전망된다.

[표 1-3] 인공지능 헬스케어 세계 및 국내 시장 규모 전망(Markets and Markets(2016) 및 KISTI 재작성) *환율: 2016년 4월 기준 적용 1\$=1,144.20원

개요	시장 규모 단위 : (억달러, 억원)						CAGR (%)
	2015년	2016년	2017년	2018년	2019년	2020년	
세계시장	71.3	108.7	185.2	273.4	438.32	754.7	60.3
국내시장	17.9	29.1	46.7	83.0	141.4	256.4	70.4

세계적인 시장분석 전문 업체인 Allied Market Research(2018)는 세계 인공지능 헬스케어 시장이 [표 1-4]에 의거 2016년 14.4억 달러에서 연평균 성장률(CAGR) 48.7%를 기록하며 2023년 227.9억 달러까지 성장할 것으로 예측하며, 크게 하드웨어, 소프트웨어, 서비스 세 부문으로 구분하였을 때 소프트웨어 부문에서의 성장이 2016년 9.4억 달러에서 연평균 성장률 47.6%로 2023년까지 14.5억 달러 규모까지 성장함으로써 인공지능 헬스케어 시장 성장에 가장 크게 이바지할 것으로 전망한다.

[표 1-4] 세계 인공지능 헬스케어 시장 규모(Markets and Markets, 2016)
(단위 : \$million)

부 문	2016년	2018년	2020년	2022년	2023년	CAGR (%)
하드웨어	191.63	441.89	999.87	2219.65	3283.41	49.7
소프트웨어	940.91	2107.34	4633.51	9999.99	14587.77	47.6
서비스	308.25	696.04	1543.04	3357.77	4918.56	48.2
합계	1440.79	3245.28	7176.42	15577.40	22789.74	48.7

인공지능 헬스케어 시장을 인공지능 기술별로 나누어 보면 [표 1-5]에서 딥러닝(Deep Learning) 부분의 경우 가장 높은 연평균 성장률인 52%로 2023년까지 28.5억 달러 규모의 시장으로 성장할 것으로 예상된다(한국보건산업진흥원, 2018).

[표 1-5] 기술별 세계 인공지능 헬스케어 시장 규모(Markets and Markets, 2016)
(단위 : \$million)

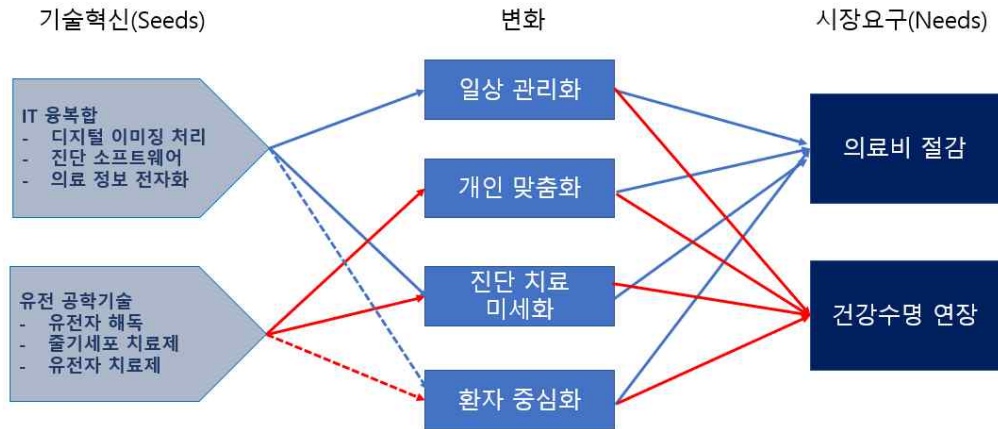
부 문	2016년	2018년	2020년	2022년	2023년	CAGR (%)
Deep Learning	149.10	355.99	831.80	1,902.32	2,853.89	52.0
Querying Method	298.14	667.62	1,467.68	3,166.98	4,619.52	47.6
Natural Language Processing	713.14	1,586.52	3,464.60	7,425.46	10,794.00	47.1
Content Aware Processing	280.41	635.14	1,412.33	3,082.65	4,522.34	48.4
Total	1,440.79	3,245.28	7,716.42	15,577.40	22,789.74	48.7

제 2 절 연구 필요성과 목적

1) 연구 필요성

4차 산업혁명 등의 기술 발달로 ICT 융합 기술 및 의료 빅데이터를 활용한 인공지능 기반 의료기기 기술 개발이 활발히 진행 중이다. 이와 함께, 인공지능을 이용한 영상, 음성 그리고 데이터 인식 기능을 의료기기에 접목하면 판독 정확성을 높이고, 질병 예측 및 예방 등 새로운 가치 창출하는데 공헌할 수 있을 것이라 기대된다. 이에 인공지능 기반 의료기기는 기존 의료기기보다 성능, 효율 그리고 질적 수준을 크게 향상할 수 있다(KHIDI, 2018).

헬스케어 분야는 [그림 1-2]에서 과거 치료·병원 중심에서 다양한 ICT 융합 기술 및 바이오 분야의 기술혁신으로 개인 스스로 일상에서 관리하는 예방·소비자 중심으로 의료 패러다임이 크게 변화되면서 고령화, 만성질환자 증가에 따른 의료비 급증을 해결하기 위한 시장의 요구가 반영되어 발전되어 가고 있다(식품의약품안전처, 2018).



[그림 1-2] 헬스케어 분야의 기술혁신, 변화, 시장요구의 결합

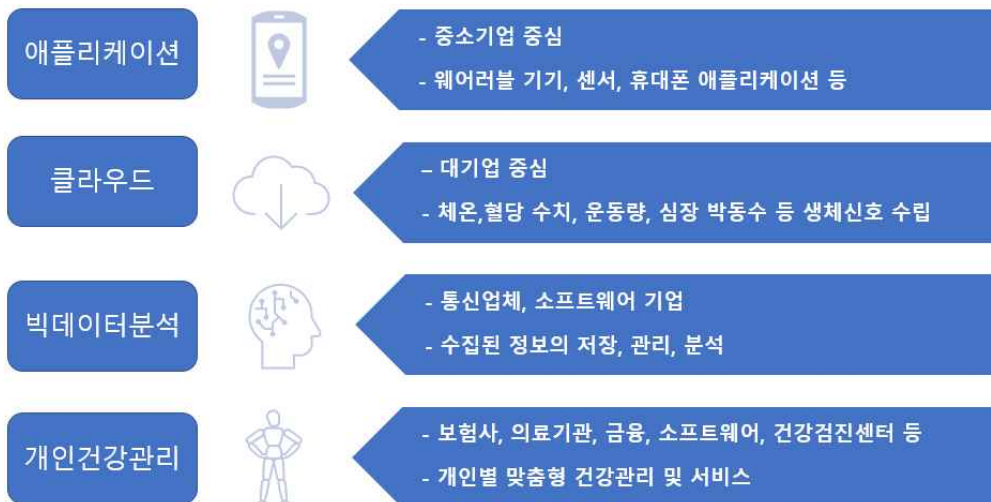
지금까지의 헬스케어는 의사와 의료기관을 중심으로 이끌어져 왔다. 보건 의료 패러다임의 변화로 헬스케어 가치사슬에서 헬스케어는 사회변혁의 중심에 있으며, 현재의 대응적, 사후적 헬스케어에서 미래 예측(Predictive), 예방(Preventive) 의학으로 변화하고 있으며, 환자 개개인의 고유한 특성에 적합한 맞춤 의학(Personalized), 사전진단과 사전 진료 즉 환자가 적극적으로 참여하는 참여 의학(Participatory)의 새로운 현상으로 나타났다(IT 동아, 2019; 정성희, 2015).

스마트 헬스케어는 [그림 1-3]에서 기존 의료용 제품 중심에서 벗어나 개인의 의료와 연관되는 다양한 데이터와 서비스를 통합하는 시스템 모델을 거쳐, 궁극적으로 수많은 시스템을 실시간 유기적 결합을 통해서 한 발 더 진화시키는 지능형 건강 의료 솔루션 방향으로 발전하고 있다.



[그림 1-3] 스마트 헬스케어 분야의 진화 방향(Frost & Sullivan, 2016)

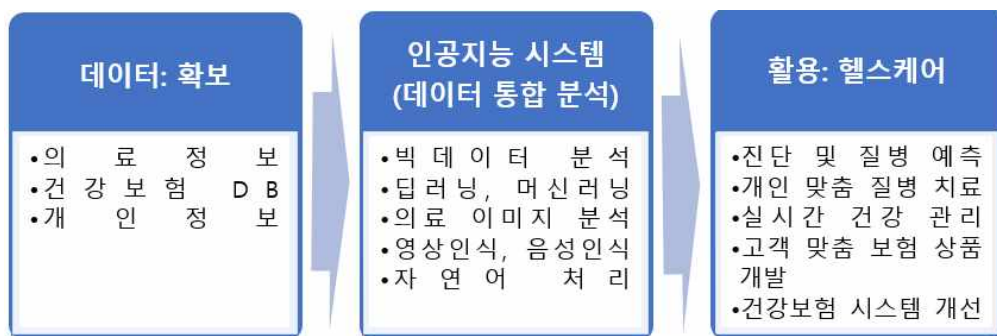
모바일, 인공지능, 빅데이터, 사물인터넷 등과 같은 기술의 발전과 보건 의료의 융합되고 의료 혁신이 가속화되면서, 미래 혁신적 예방 및 질병 예측 중심으로 의료 형태가 변화하고 있다. 이를 바탕으로, ICT 융합 의료산업 생태계는 의료기기·정보와 관련된 소프트웨어부터 맞춤형 건강관리 서비스, 보험 금융 등까지 확장되고 필요성이 주목받고 있다(김정곤, 2016). 아울러, [그림 1-4]의 의료산업 생태계에서 개인 맞춤형 보건 의료 서비스의 등장은 스마트 헬스케어를 기존의 헬스케어와 구분 짓는 또 하나의 중요한 특징이다(한국보건사회연구원, 2016).



[그림 1-4] 의료산업 생태계(한국보건사회연구원, 2018)

스마트 헬스케어를 위한 핵심기술로는 인공지능, 빅데이터, 가상현실, 유전체 분석, 정밀 의료, 재생의료 등이 거론되어, 가까운 미래에는 규제가 개선되고 기술이 더욱 발전함으로써 예측의학, 맞춤 의학이 의료의 핵심 영역으로 자리잡게 될 것으로 보인다(IT 동아, 2019).

핵심기술 중 인공지능 헬스케어는 [그림 1-5]과 같이 다양하고 많은 양의 데이터를 인공지능 기술을 활용하여 학습·분석하고 헬스케어 산업에 적용함으로써 질병 예측, 예방의학, 환자 맞춤형 치료 등 혁신적인 고부가가치형 의료 서비스의 제공을 의미한다(김문구, 2018).



[그림 1-5] 인공지능 헬스케어의 개념(김문구, 2018)

신체활동 권고기준의 변화 추이로 인해 과거에는 심폐기능의 향상을 위한 격렬한 신체활동을 강조하였으나 최근에는 다양한 건강상의 이득을 위한 적정 신체 활동량 실천을 권고하는 것으로 변화하였다.

우리나라 국민의 건강위험요인의 질병 비용에 대한 신체활동의 기여도는 [표 1-6]에서 흡연, 음주에 이어 운동 부족(신체 비활동)이 세 번째로 높게 나타났으며, 약 1조 2천억 원에 육박한다(보건복지부, 2014).

[표 1-6] 우리나라 국민의 건강위험요인의 질병 비용에의 기여도(정영호 외, 2012)

구 분	위험요인	비용(단위: 백만원)	기여도
건강생활 실천	흡연	2,967,554	9.12%
	음주	2,791,697	8.58%
	운동부족(신체비활동)	1,221,318	3.75%
	영양	120,064	0.37%
	소계		21.82%
생물학적 요인	체중(과체중,비만)	2,161,851	6.63%
	고혈압	723,184	2.22%
	고콜레스테롤	77,166	0.24%
	소계		9.09%
환경요인	대기오염	1,167,873	3.59%
계(20세 이상 질병 비용 대비)			34.49%

앞으로는 건강관리 서비스가 가정 등 실생활 전 영역에서 평생에 걸쳐서 제공될 것이며, 이로 인해 기존 의료시스템과의 호환성 및 저렴한 비용으로 해당 의료기관의 스마트 헬스케어 시스템을 구축할 수 있으리라 본다. 또한, 인공지능 자세 추정 기반의 스마트 헬스케어 시스템은 건강관리 의료 서비스 제공 공간이 가정, 피트니스 센터, 길거리 등 실생활 영역으로 확장되어 개인의 생체 및 건강 정보가 전 생애에 걸쳐 축적됨에 따라 건강관리 서비스를 일시성이 아닌 일생 동안 케어를 할 수 있는 서비스로 확장하는 연구가 필요하다고 본다(최동운, 2009).

2) 연구 목적

본 연구에서 개발하고자 하는 인공지능 자세 추정 기반의 스마트 헬스케어 시스템은 CNN 딥러닝 기술을 이용하여 더 정확한 운동 동작의 정보를 획득하여 사용자의 현 운동 상태와 앞으로의 운동으로 건강 상태와 관련된 정보들을 시간과 장소와 관계없이 감지함은 물론, 미래 동작 상황에 대해 인

공지능을 통해 예측하여 신속하게 대처하고자 하며, 이를 통해 신체활동을 통한 개인별 열량을 계산하여 개인에게 적합한 운동 방법을 제시하고자 한다.

본 연구의 목적은 스마트 헬스케어의 특성을 파악하고 국내·외 헬스케어 주요 관심 분야와 인공지능 기반의 자세 추정 기술을 반영하여 현대인의 건강한 생활 만족도를 향상시킬 수 있는 새로운 형태의 스마트 헬스케어 운동 관리 시스템을 구현하고자 한다. 이를 통해, 구현된 스마트 헬스케어 시스템은 향후 실생활에 적용하여 운동을 통한 건강관리 서비스를 제공하는 비즈니스 모델로 활용될 수 있도록 정보를 제공하는데 부가적인 목적이 있다.

제 3 절 연구의 방법과 범위

1) 연구 방법

본 연구의 목적을 위하여 스마트 헬스케어와 관련된 문헌 및 선행연구를 분석하였으며, 4차 산업혁명 시대에 ICT 분야인 인공지능과 신체활동 목록에 따른 운동량 측정 그리고 스마트 헬스케어 동향에 관한 국내·외 연구 자료를 수집 및 정리하여 본 연구에 대한 이론적 배경을 구축하고자 한다. 서비스 개발 구현을 위해 개발 방법론에 근거하여 스마트 헬스케어 시스템을 구현하기 위한 연구 방법으로 4단계의 과정을 수행하였다.

첫 번째, 분석 단계에서는 스마트 헬스케어와 인공지능 그리고 ICT 정보 통신 자료와 선행연구를 통한 조사를 시행하여 건강·운동 생활의 현황과 문제점을 파악하였으며, 시장 조사 리포트 미래 예측 방법을 통해 보완하였다.

두 번째, 설계 단계에서는 딥러닝을 활용하여 각각 모델 학습에 대한 신체 유형별 정확도를 분류하였으며, 각각의 모델이 향후 사용자 동작 지도를 설계하는데 사용자의 요구를 파악하고 구체적으로 표현할 수 있도록 하였다.

세 번째, 구현 단계에서는 앞에서 발견된 문제점을 개선하고 새로운 방향으로 발전하도록 서비스 관념화를 하였으며, 자세 추정을 통한 동작 인식을 개발하고, 그에 따른 어떤 동작인지를 판별할 수 있는지 기능 검증을 구성하였다.

네 번째, 적용 단계에서는 모든 아이디어를 종합하여 건강관리 서비스인 ‘스마트 헬스케어’를 시각화하였으며, 본 서비스의 일환인 스마트 헬스케어 시스템 건강 관리체계를 마련하고, 사용자의 손뼉(박수) 치기를 통한 건강 정보와 대사 관리를 통한 스마트 헬스케어 시스템을 구현하였다.

2) 연구 범위

본 연구의 범위로는 총 5장으로 구성하였으며, 1장에서는 연구의 배경, 필요성, 연구 목적, 연구의 방법과 범위에 관해 기술하였으며, 2장에서는 스마트 헬스케어, 인공지능, 신체활동 목록과 MET(Metabolic Equivalent of Task)에 관하여 이론적 배경을 기술하였다. 3장은 자세 추정 학습 방법과 동작 인식 방법을 제시하고, 스마트 헬스케어 시스템 모델 설계에 관한 연구로 시스템 범위, 시스템 구조, 시스템 흐름도를 구성하였으며, 4장에서는 자세 추정 학습에 대한 정량적 결과를 비교하고 이를 통해 손뼉(박수) 치기라는 운동 동작 인식을 통해 운동량을 측정하고 측정값을 스마트 헬스케어 시스템에 연계하여 시스템 구현에 관한 연구 결과를 분석하고자 한다. 이를 바탕으로 일반적, 기술적, 연구적 측면의 기대효과를 기술하였다. 마지막 5장은 본 연구에 관한 결과를 요약하고 본 연구를 진행하다가 나온 미해결 과제와 이를 극복하기 위해서 향후 해야 할 연구 방향에 대해 기술하였다.

제 2 장 이론적 배경 및 선행연구

제 1 절 스마트 헬스케어 연구

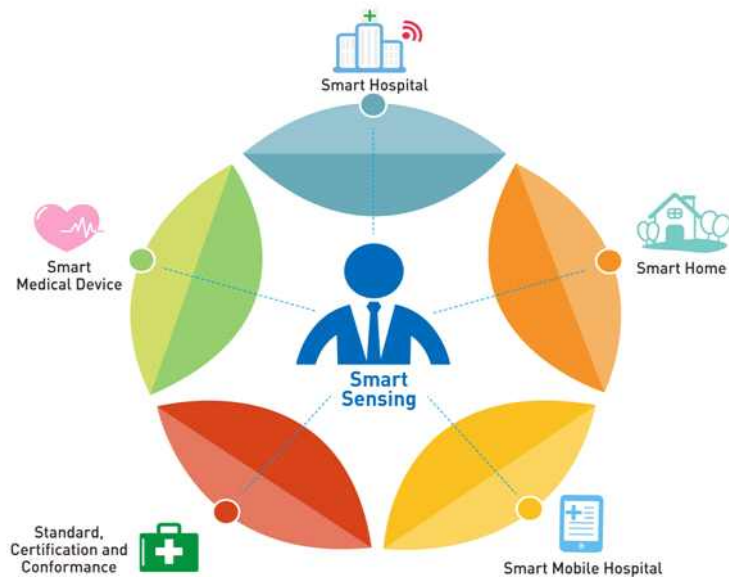
1) 스마트 헬스케어의 정의

최근의 ICT와 보건 의료 서비스의 결합은 한 걸음 더 나아가 ‘스마트 헬스케어’라는 새로운 형태로 변모하고 있다(한국보건사회연구원, 2016; 2018).

스마트 헬스케어는 의료와 기술이 융합된 형태로 의료 데이터 기반 지능화된 서비스를 의미하며, 환자 사용자의 개인별 건강 상태를 시간과 장소의 제약 없이 실시간으로 모니터링 및 관리하고 건강 정보 및 질병 상태 등을 분석하여 최적화된 맞춤형 진료를 제공하는 서비스 또는 시스템으로 정의한다(정보통신산업진흥원, 2017). 또는, 헬스케어와 인공지능, 빅데이터, 사물인터넷, 클라우드, 나노 등의 기술들이 융합된 새로운 개념으로 기존의 헬스케어 영역에서 더 나아가 언제 어디서나 예방, 진단, 치료, 사후 관리의 보건 의료 서비스로 각종 정보 기술을 활용하여 언제 어디서나 건강관리를 받을 수 있는 원격 의료 서비스이다(Wikipedia, 2018).

스마트 헬스케어는 4차 산업혁명의 핵심인 ICT 기술로 인공지능 및 빅데이터, 사물인터넷, 클라우드 컴퓨팅을 헬스케어와 접목한 분야로, [그림 2-1]에서 기본적인 산업구조를 살펴보면, 소비자가 일상생활이나 의료기관 등 전문기관에서 생성해낸 데이터를 데이터 전문 기업이 수집 및 분석하여, 이를 의료 및 건강관리 기업이 다시 활용하여 소비자에게 자문 및 치료해 주는 구조이다(삼정 KPMG, 2018).

스마트 헬스케어에 대해서 요약하자면 다음의 [표 2-1]와 같다.

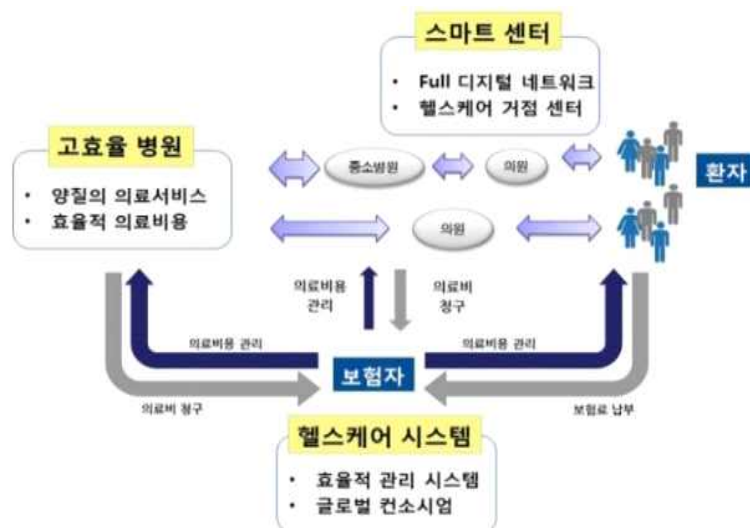


[그림 2-1] 스마트 헬스케어 산업구조(국가기술표준원, 2018)

[표 2-1] 스마트 헬스케어의 정의(식약처 기술표준 전략보고서, 2018)

〈국가기술표준원〉 2016년 표준기반R&D 로드맵	- 개인의 건강과 의료에 관한 정보로서 기기와 시스템 추가로 시스템 모델을 다루는 산업 분야로 건강 관련 서비스와 의료 IT가 융합된 분야. - 개인 맞춤형 건강관리 서비스를 제공하여 개인이 소유한 휴대형 또는 착용형 기기나 클라우드 정보시스템 등에서 확보된 생활 습관, 신체 검진, 의료이용정보와 그리고 인공지능, 가상현실, 유전체 정보 등의 분석을 통해 제공되는 개인 중심의 건강관리 생태계.
〈NIPA〉 스마트 헬스케어서비스 분야 도입 사례 분석집 (2017)	- 의료와 ICT 기술이 융합된 형태로 의료 데이터 기반 지능화된 서비스를 의미. - 사용자의 개인별 건강 상태를 시간과 장소의 제약 없이 실시간으로 모니터링 및 관리하고 사용자 건강 정보 및 질병 상태 등을 분석하여 최적화된 개인별 맞춤형 진료를 제공하는 서비스 또는 시스템.
〈TTA〉 2018 표준화 전략맵 - 스마트헬스	- 스마트 헬스는 언제 어디서나 질병에 대한 예방, 상태 파악, 진단, 치료, 예후, 건강 및 생활 관리 등의 개인 맞춤형 보건의로 건강관리 서비스를 제공하는 기술. - 유무선 통신망을 활용한 건강 관련 정보기기에서 필요로 하는 정보를 측정 및 분석 관리하는 기술과 해당 서비스를 위한 기기, 용어, 시스템 모델, 시험 및 인증 등 포함.

Medical 요소의 의료 서비스와 Care 적인 건강관리 서비스가 모두 제공되어 보건의료 관련 서비스를 요구하는 사용자는 물론 건강에 관심이 있는 일반인을 대상으로 상시적인 건강관리 케어 서비스와 필요에 따라 제공되는 건강관리 의료 서비스를 포함한다. 즉 스마트 헬스는 스마트, 빅데이터, 소셜 네트워크를 모두 포함하는 개념으로 가야 할 것이며 개념도는 [그림 2-2]와 같다(한국정보화진흥원, 2011).



[그림 2-2] 스마트 헬스케어 개념도(한국정보화진흥원, 2011)

기존 연구에서는 [그림 2-3]처럼 Tele-헬스²⁾, e-헬스³⁾, u-헬스⁴⁾, s-헬스의 순으로 스마트 헬스케어 분야의 새로운 주요 기술 개념이 등장하여 왔다(안무업, 2009; 김주원, 2011; L. V. Dyk, 2014). 또한, 스마트 헬스케어(Smart Healthcare)는 u-헬스, s-헬스와 같은 ICT 기술이 적용된 모든 헬스케어를 포괄하는 개념이다(김옥남, 2009; 문세연 외, 2017; M. Volk, et al.,

2) Tele-헬스: Tele-health : 통신서비스를 통하여 원거리에 건강관련 (Health-Related) 정보나 서비스를 전달하는 것(Wikipedia).

3) e-헬스: 정보통신기술의 활용을 통하여 보건의료시스템의 접근성, 품질 및 효율성 향상을 달성하는 것(WHO).

4) u-헬스: Ubiquitous Healthcare의 의미로 보건의료 혹은 건강관리로써 IT(Information Technology), BT(Biology Technology), NT(Nano Technology) 등과 연관한 인본주의적 의료.

2015; B. M. Choi and T. R. Kim, 2015).



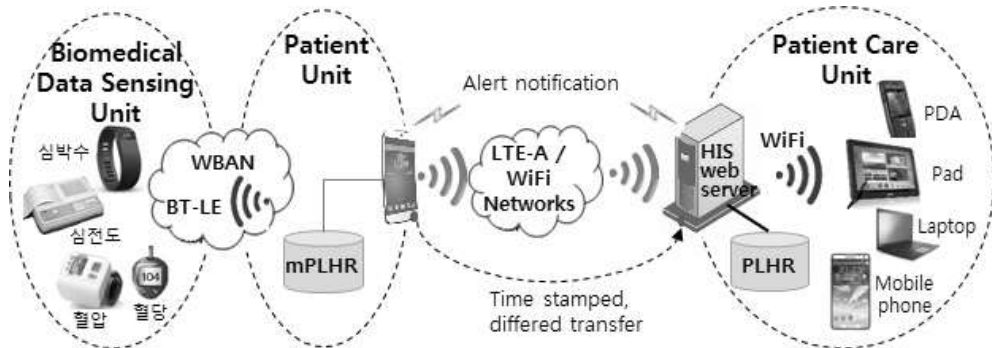
[그림 2-3] 시대별 디지털 헬스케어의 기술 개념 등장(김주원, 2011)

2) 스마트 헬스케어 구조

이난경 외(2014)의 연구 [그림 2-4]에서 헬스케어 정보시스템의 서비스 계층을 1) 재택에 있는 환자의 생체신호를 취득하는 ‘생체 자료 센싱 영역(Biomedical Data Sensing Unit: BDSU)’, 2) 스마트폰을 활용하여 생체 신호를 수집-처리-전송하는 ‘환자 영역(Patient Unit: PU)’, 3) e-헬스케어와 전통적 병원 영역인 ‘환자 치료 영역(Patient Care Unit: PCU)’ 등으로 제시하였다.

Mettler, et al.(2012)의 연구 구조에 추가해야 할 또 다른 최근 연구 진행 상황으로는 예방적 사회치료에 대한 관점이다. 즉, 최근 정보통신 기술에 대한 편재성과 헬스케어 범위 확대로 인해 헬스케어 분야는 이제는 개인의 치료 영역에 머무르지 않고, 공공 또는 사회 전반의 질병 예방 관리로 그 폭이 넓어지고 있다(A. M. McDaniel, et al., 2008; C. C. Poon, et al., 2008; G. Thomase, et al., 2013; Mohammadzadeh, et al., 2014; A. K.

Triantafyllidis, et al., 2014).



[그림 2-4] 모니터링 기반 헬스케어 정보시스템의 3계층 서비스 체계(이난경 외, 2014)

즉, 네트워크화되는 공동체, 개인과 사회의 변화, 이를 통한 협업적 질병 관리로 인해 이제 새로운 시도와 체계는 개인의 단위를 넘어 사회 전 구성원을 대상으로 한 질병의 치료뿐만 아니라 질병에 대한 예방과 관리에 주안점을 두는 공공영역으로 헬스케어가 전개되고 있다.

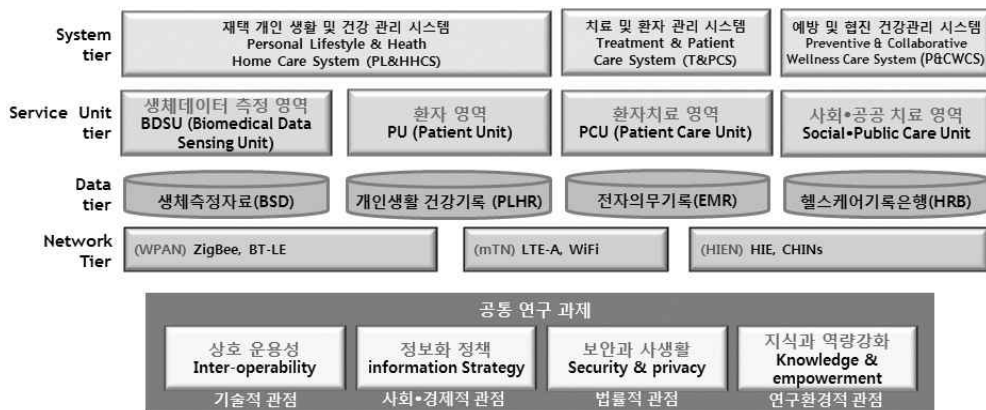
Mettler, et al.(2012)의 개인 환자 영역과 연결 시스템 그리고 병원 정보 시스템으로 이루어지는 3단계 개념을 수정 확대하여 이난경 외(2014)의 연구 결과에서 사회·공공 치료 분야의 개념을 추가 적용하여 [그림 2-5]와 같은 ‘헬스케어 4계층 서비스 영역 구조’를 도출하였다. 요약하면, 생체 데이터를 측정하는 영역과 환자 영역은 핵심 개체와 기술이 다른 배타적 서비스 영역이지만, 결과적으로 헬스케어 서비스를 독립적 제공보다 상호 연계하여 하나의 시스템적 관점으로 서비스를 제공하는 것이다. 스마트폰 디바이스를 통해 수집한 데이터 기반으로 건강관리 상태를 관찰하거나 자기관리를 위해 정보 서비스를 하는 형태로 연계성이 매우 높아서 헬스케어 서비스를 창출하고 이를 전달하는 정보 시스템적 관점으로 통합되어야 할 필요성이 제기된다(이난경 외, 2014).

Service Unit Tier(서비스 계층 영역)



[그림 2-5] 헬스케어 4계층 서비스 영역 구조(이난경 외, 2014; I. Cubic, et al., 2010)

헬스케어 연구 구조 [그림 2-6]에서 최상단 계층인 System Tier는 1) 모바일 Health 기술을 적용하여 병원 밖 사용자들의 개인 생활과 건강 상태를 지켜보며 이를 통해 의료전문가나 사용자 자신이 직접 건강 개입을 통해 생활 습관을 관리하는 ‘재택 개인 생활 및 건강관리 시스템’, 2) 전통적인 의료기관의 역할과 타 시스템과의 연계를 통한 새로운 보건의료 건강관리 서비스를 제공하기 위한 ‘치료 및 환자 관리 시스템’, 3) 의료기관 간, 지역 간 의료정보의 교환과 협업을 통한 사회와 공공의 질병 예방과 관리를 위해 ‘예방 및 협진 건강관리 시스템’으로 정립한다(이난경 외, 2014; M. ElHelw, 2009; O. Postolache, 2011).



[그림 2-6] 헬스케어 연구 구조 개념도(이난경 외, 2014)

최근의 헬스케어 서비스에 관한 연구는 [표 2-2]에 의해서 전통적으로 행

해지는 치료 및 환자 관리 시스템에 관한 연구보다 이제는 재택 개인 생활을 위한 건강관리 시스템, 예방 및 협진을 통한 건강관리 시스템에 관한 연구가 활발하게 진행되고 있다. 헬스케어 연구 구조의 체계에 따라 재택 개인 생활 및 건강관리 시스템, 치료 및 환자 관리 시스템, 예방 및 협진 건강관리 시스템 등의 3가지 정보시스템 관점으로 구분하여 제시하고자 한다(이난경 외, 2014).

[표 2-2] 재택 개인 생활 및 건강관리 시스템 분야 연구 동향의 특징(이난경 외, 2014)

구분	연구 동향 요약 및 특징
서비스	- 착용형 무선 인체 센서와 스마트폰을 이용하여 재택 환자의 건강 상태 모니터링하고, 이를 바탕으로 스마트폰을 환자 스스로, 생활 습관에 대한 자기관리와 건강 개입의 수단으로 활용하고자 하는 방안에 관한 연구가 전개되고 있음. - 모니터링의 효과성 제고를 위해 복수의 생체 센서로부터 수집된 기본 데이터와 보건 건강지수 등의 외부환경정보 및 스마트폰 자체 센서 등을 결합하여 환자의 정확한 상태 추론을 위한 상황인지 영역에 관한 연구가 활발히 진행되고 있음. - 궁극적으로 센서와 스마트폰 및 의료시스템을 연계하여 일상생활의 자기관리와 의료인의 적절한 개입을 통해 노령화와 만성질환에 대비하고, 환자 자신의 독립적인 생활을 영위하기 위한 시스템을 구축하기 위해 연구하고 있음.
데이터	- 전통적 헬스케어 시스템에서는 존재하지 않았던 스트리밍 형의 실시간 생체 헬스케어 데이터의 활용에 관한 적극적인 연구가 전개되고 있음. - 생체 헬스케어 정보를 스마트폰에 저장하고, 이를 스마트폰이 수집한 외부정보와 결합해 적절한 자기관리와 건강 개입의 기본 데이터 베이스로 활용하고자 하는 연구가 진행 중임.
네트워크	- 건강 모니터링을 위해서는 스마트폰을 전통적 의료시스템 영역과 착용형 생체 센서 영역을 연결하는 게이트웨이의 역할로 활용하는 것이 보편적 추세이며, 특수한 네트워크를 활용하기보다는 상용화된 네트워크를 활용하는 추세임. - 개인의 경우는 이동통신의 LTE-A와 BT-LE의 프로토콜을 활용하고 있음. - 특정 지역의 다수의 환자에 대한 모니터링은 WBAN의 Zig Bee를 이용하는 경향이 있음.

제 2 절 자세 추정(Pose Estimation) 연구

1) 인공지능(AI) 정의

S. Russell and P. Norvig의 “Artificial Intelligence: A Modern Approach”에서는 인공지능의 4개의 카테고리를 고려하여 인공지능을 정의하고 있다(S. Russell, et al., 2009). 구체적으로는 “사람처럼 vs 합리적으로”, “생각 vs 행동”을 기준으로 해서 4개의 카테고리로 해서 인공지능을 정의하고 카테고리별로 관점을 소개한다.⁵⁾ 물론, 이 네 가지 카테고리가 모두 중요한 관점이지만 “사람처럼 행동하는(Acting Humanly)”과 관련된 기준이 공학적 판단으로 쉽고, 공학과 직접 관련된 연구 활동과 밀접한 관계가 있다. 이러한 관점에서 정리하자면 첫 번째로 인공지능은 사람에 의해서 수행될 때 지능이 필요한 일을 수행하는 기계를 만드는 기술이며, 두 번째로 인공지능은 컴퓨터가 사람이 잘하는 일을 할 수 있도록 하는 방법을 연구하는 학문이다(구형일, 2018).

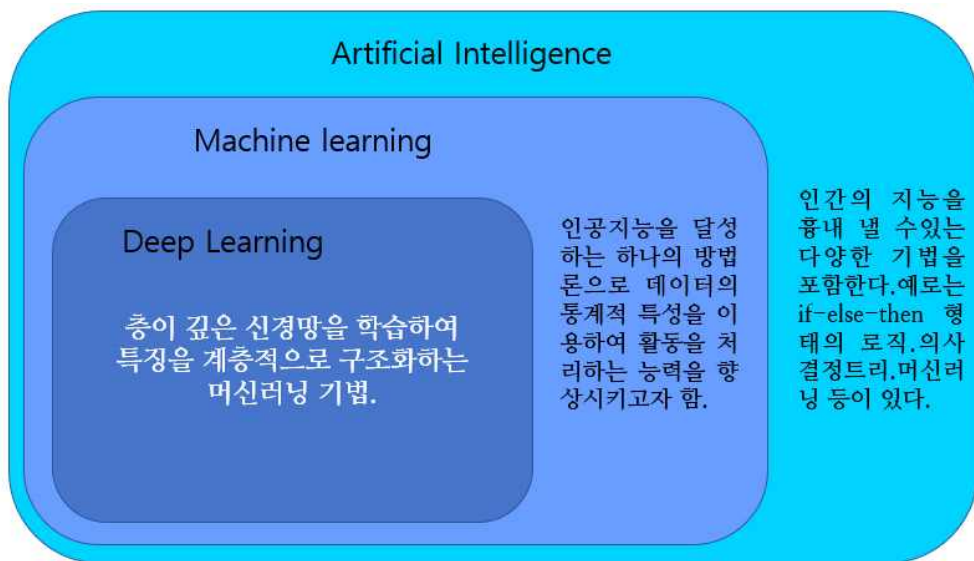
인공지능, 기계학습, 딥러닝 간의 관계를 정리하면 [그림 2-7]과 같이 된다. 여기서 딥러닝은 규모에 있어 큰 신경망을 사용하고 있다는 점과 방대한 데이터 양을 학습한다는 특징을 가지고 있는 기계학습으로 기계학습은 인공지능 구현 방식의 하나이다. 하지만 딥러닝은 기존의 인공지능, 기계학습 방법과 비교하여 보면 결정적 특징으로 세 가지를 논할 수 있다. 먼저 딥러닝은 데이터의 양에 비례하여 성능이 향상되는 경향성을 보인다, 그리고 또 딥러닝 학습의 결과로부터 얻어진 모델을 분석하면 층이 높으면 높을수록 점진적으로 복잡한 특징을 구성한다는 것을 확인할 수 있다. 마지막으로 딥러닝의 또 다른 특징으로 End-to-End 학습이 가능하다는 점이다(구형일, 2018; 한석민, 2018; 하태용, 2019).

최근 들어 뛰어난 성능을 보이며 많은 관심을 받는 딥러닝 역시 기계학습과 유사한 방법이다. 딥러닝과 기계학습은 데이터를 이용하여서 모델을 학습한다는 공통점이 있다지만, 데이터를 학습하는 과정에서 큰 차이가 있다. 기

5) 4개의 카테고리: “이성적으로 생각하는(Thinking rationally)”, “사람처럼 생각하는(thinking humanly)”, “이성적으로 행동하는(Acting rationally)”, “사람처럼 행동하는(Acting humanly)”

계학습을 통해 이미지를 인식하기 위해서는 먼저 사진을 그대로 사용하는 것이 아니라 사진 속에 있는 객체를 가장 잘 구분할 수 있는 특성 인자(Feature)를 찾아내야 한다. 반면 딥러닝의 경우에는 사람이 특성 인자를 선정하는 것이 아니라 데이터에서 모델을 학습하는 과정에서 목표를 잘 달성할 수 있는 특성 인자를 스스로 찾는다(이승철 외, 2017).

딥러닝이 데이터로부터 패턴을 학습하는 방법이기도 하지만 아직은 개발자가 경험이나 직관을 바탕으로 결정해야 할 변수(하이퍼-파라미터)가 많으며 이 변수들을 변경해가며 학습하는 과정에는 많은 시간과 비용이 필요하다(J. Clark, and D. Amodei, 2016)

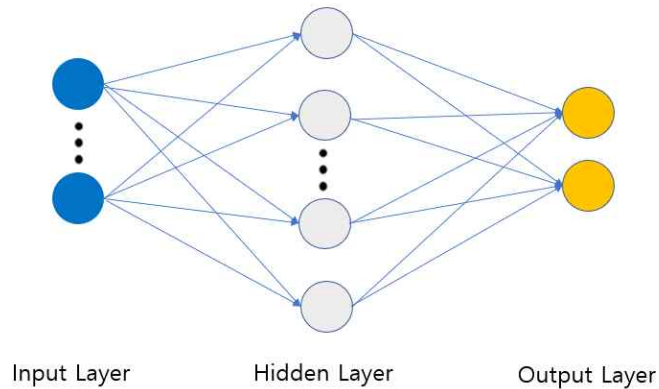


[그림 2-7] 딥러닝, 기계학습, 인공지능 비교(하태용 외, 2019)

2) 합성곱 신경망(Convolutional Neural Networks: CNN)

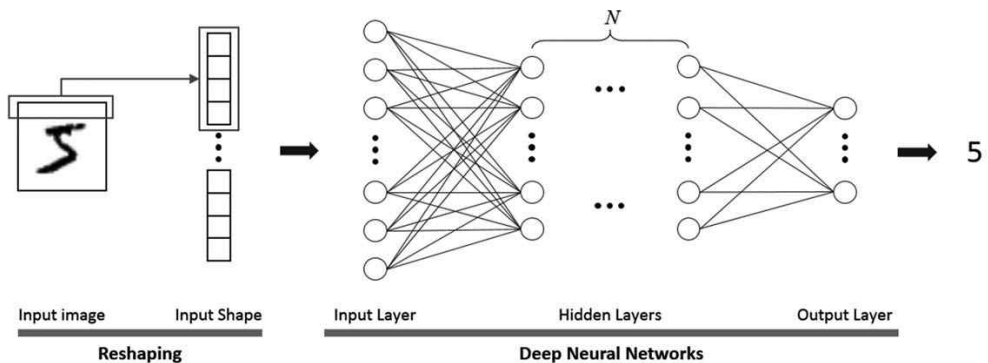
자세 추정 알고리즘으로 다양한 모델이 있고, 대표적으로 활용되고 있는 CNN에 대해서 전반적인 구조를 파악하고자 한다. 먼저, 인공 신경망(Neural Networks)은 [그림 2-8]과 같이 일반적으로 입력층(Input Layer), 은닉층(Hidden Layers)과 출력층(Output Layer)으로 구성되어 있으며 각 층에 포함

된 뉴런들이 가중치를 통해 연결되어 있다.



[그림 2-8] 인공 신경망의 구조(이승철 외, 2017)

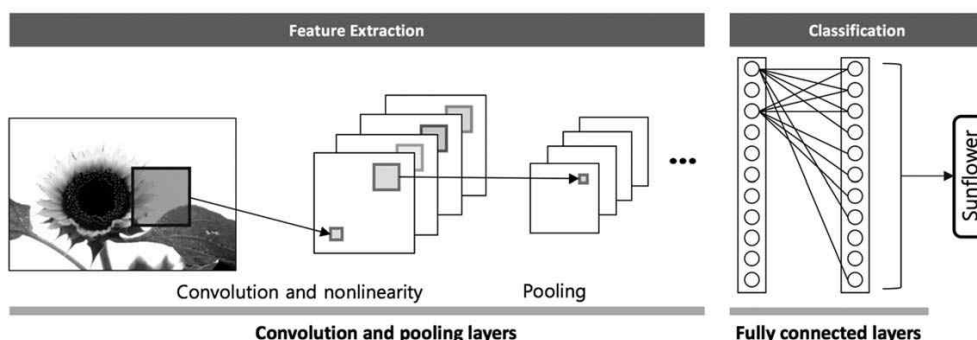
심층 신경망(Deep Neural Networks)은 입력층과 출력층 사이에 여러 개의 은닉층으로 이루어진 인공 신경망이다. 심층 신경망은 많은 은닉층을 통해서 복잡한 비선형 관계들을 모델링하고 있다. 이처럼 은닉층의 개수를 늘림으로써 고도화된 추상화가 가능해졌다. 이런 신경망 구조를 딥러닝이라고 부른다(Y. LeCun, et al., 2015). 하지만 단순히 은닉층의 개수만 늘어난 것뿐만 아니라, 최근 관련 연구에 의해서 다양한 문제를 해결하기 위한 독특한 구조가 많이 개발되었으며, [그림 2-9]에서 심층 신경망의 구조를 알 수 있다.



[그림 2-9] 심층 신경망의 구조(이승철 외, 2017)

최근 딥러닝 중 합성곱 신경망의 등장은 이미지 인식의 패러다임을 바꾸어 놓았다. 인간은 물체를 인지할 때에, 이미지에 대한 단위 요소를 수치로 이해하기보다 물체를 종합적 또는 전체적으로 받아들여려고 한다. 결국, 합성곱 신경망은 이러한 인간의 인지 모델을 수학적으로 표현하고자 한 구조이다 (Y. LeCun, et al., 1995).

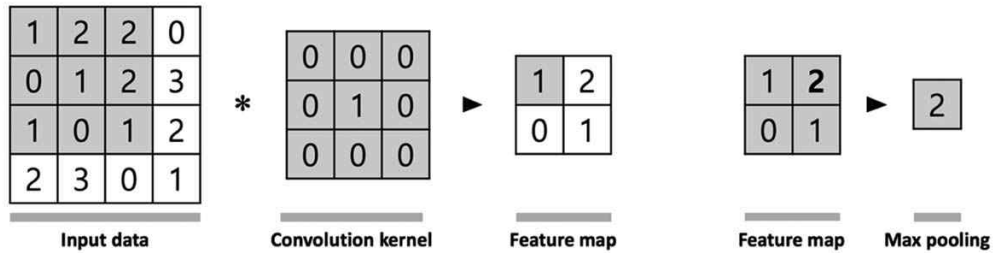
특성 인자를 추출하기 위해서 [그림 2-10]에서 합성곱 계층(Convolutional Layer)과 통합 계층(Pooling Layer)을 사용하면서, 이와 같은 특성 인자를 일반적인 인공 신경망(Fully - Connected Layers)의 입력을 이용해 이미지를 분류하고 인지한다.



[그림 2-10] 합성곱 신경망의 구조(이승철 외, 2017)

합성곱 계층은 [그림 2-11]에서 커널을 일정한 간격으로 이동해가며 합성곱 연산을 수행하며, 이는 영상처리 분야에서 필터링과 비슷한 과정이다. 합성곱에 사용되는 커널 값은 인공 신경망의 가중치에 해당하기 때문에 데이터로부터 최적의 커널을 학습한다고 볼 수 있다. 통합 계층은 합성곱 계층에서 출력된 데이터의 크기를 줄이기 위해서 주로 사용한다. 통합 계층에는 Max-pooling과 Average Pooling 등이 있는데, Max-Pooling은 해당 영역에서 최댓값을 찾고자 하는 연산이며, Average Pooling은 해당 영역의 평균값을 계산하고자 하는 연산이다. 합성곱 연산과 마찬가지로 창을 이동하면서 해당 Pooling 연산을 수행하며, 이는 특성 인자에 공간 불변성(Spatial Invariance)을 더하는 기능으로도 해석한다(이승철 외, 2017; H. Jeong, et

al., 2013).



[그림 2-11] Convolutional and Pooling Layers(이승철 외, 2017)

3) PoseNet Model

(가) 소개

PoseNet은 키바디 조인트의 위치를 추정하여 이미지 또는 비디오에서 사람의 자세를 추정하는 데 사용할 수 있는 비전 모델로 Google Creative Lab(GCP)에서 출시했다. PoseNet의 경우 Tensorflow.js로 개발되어 크롬과 같은 브라우저에서 실시간으로 구동할 수 있어서 다른 모델들에 비해 활용 가능성이 크다.

PoseNet은 ResNet 모델과 MobileNet 모델이 있지만, ResNet 모델의 정확도는 높지만 크기가 크고 단계가 많다. 그래서, 실시간 응용 프로그램에서 페이지 로드 시간과 추론 시간이 길어 바람직하지 않다. 결과적으로, 같은 자세에 추정하는 알고리즘에 의해 MobileNet과 ResNet 아키텍처는 모두 일인칭 자세 추정을 지원한다. 아울러, 모바일 기기에서 실행되도록 설계된 MobileNet 모델이 단일 자세 추정에서 두 알고리즘 중 ResNet 보다 간단하고 빠르다. 하지만, 이상적인 자세 추정은 이미지에 한 사람만 있는 경우이다. 단점으로 이미지에 여러 사람이 있는 경우 두 사람의 KeyPoint가 같은 단일 자세의 일부로 추정될 수 있다는 것이다.

실제로 MobileNet 버전 모델은 PoseNet이 모바일 디바이스에서도 구동하

기 위해 고안된 모델이다⁶⁾. 별도의 복잡한 드라이버나 라이브러리 설치 과정 없이, Tensorflow.js의 특성상 Keras API 또는 Tensorflow에서 학습한 모델을 로드해서 사용하거나 재학습시킬 수 있다. 특히 MobileNet 모델의 가중치 파일은 <https://storage.googleapis.com/tfjs-models/7>)에서 다운로드해 적당한 포맷의 Annotation 파일로 만들 수 있으며, MobileNet 모델을 Leeds Sports Pose(LSP) DataSet이나 MPII(Multiple People) DataSet⁸⁾ 등의 학습 데이터로 조정하여 인식률을 향상하는 것도 가능하다.

또한, PoseNet은 입력 영상 속 이미지로 사람의 위치에 영향을 받지 않는다. 사람의 운동 동작을 실시간으로 추정을 해야 하면, PoseNet은 운동하고 있는 동안 사람은 입력 영상 구조의 중앙에 계속해서 있을 필요 없이 프레임을 벗어나지 않는 범위에서 자연스럽게 활동할 수 있다. 그러므로, PoseNet은 동작을 끊김 없이 인식하기는 한다. 하지만, 각각의 입력 구조가 한 동작의 연속으로 보는 것이 아니다. 그러므로, 예측되는 관절 포인트가 일관되게 유지되지 않는 것이다.

즉, 새로운 구조가 들어올 때마다 KeyPoint들이 새롭게 예측되기 때문에 결국 결과값이 안정적이지 못한 것이다. 이로 인해, 사람이 어떤 동작을 할 때 그 동작이 어떤 동작인지 인식한 다음 올바른 자세로 수행되지 않을 때 피드백을 줄 수 있어야 한다. 그리고 해당 성능을 기대하기 위해서는 궁극적으로 순환신경망(Recurrent Neural Network: RNN)이나 RNN의 일종인 장단기 메모리(Long Short-Term Memory Models: LSTM) 모델로의 발전이 불가피하다고 본다. 그래서, CNN 기반의 자세 인식 모델의 한계성을 인식한 사람들이 제시하는 새로운 모델 LSTM Pose Machines에 대해 발표되고 있다.

(나) 방법

6) <https://medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow>

7) <https://github.com/rwightman/posenet-python/blob/master/posenet/converter/wget.py>

8) MPII Human Pose: 25000개 이미지와 라벨링(Annotations) ; multiple people 포함 40000개의 라벨링(28000 trainset, 11000 testset). Validation 3000개.

아래의 순서대로 PoseNet 자세 추정하는 방법을 제시한다.

① 아키텍처-MobileNet 또는 ResNet 50 일 수 있다. 로드할 PoseNet 아키텍처를 결정한다.

② OutputStride-8, 16, 32중 하나일 수 있다. ResNet 아키텍처에서는 Stride 16, 32가 지원되고 MobileNet 아키텍처에서는 Stride 8, 16, 32가 지원되며, PoseNet 모델의 출력 보폭을 지정한다. 값이 작을수록 출력 해상도가 커지고 속도를 낮추어서 모델이 더 정확해진다. 정확도를 높이면서 속도를 높이려면 이 값을 더 큰 값으로 설정한다.

③ InputResolution-{width : number, height : number} 유형의 숫자 또는 객체이다. 기본값은 257이다. 이미지가 PoseNet 모델로 공급되기 전에 크기가 조정하고 채워지는 크기를 표시한다. 값이 클수록 속도에 따른 비용도 커지면서 모델이 더 정확해진다. 그러므로, 정확도를 높이면서 속도를 높이려면 이 값을 더 작은 값으로 설정하여야 한다. 예로서, 숫자를 입력하면 이미지의 크기와 너비가 같은 정사각형으로 채워진다. 이것을 이용하여 해당 이미지의 크기를 조정하면, 조정된 크기와 너비로 채워진다.

④ 승수 - 1.01, 1.0, 0.75 또는 0.50 중 하나를 선택한다(값은 ResNet 아키텍처가 아닌 MobileNet 아키텍처에서만 사용된다). 모든 컨볼루션 연산의 깊이(채널 수)에 대한 부동 승수이다. 값이 클수록 레이어의 크기가 커지고 속도에 따른 비용도 커지면서 모델이 더 정확해진다. 그러므로, 정확도를 높이면서 속도를 높이려면 이 값을 더 작은 값으로 설정하면 된다.

⑤ QuantBytes - 이 인수는 가중치 양자화에 사용되는 바이트를 제어한다. 사용 가능한 옵션은 [표 2-3]에서 특징과 같다.

[표 2-3] Float당 특징

구 분	내 용
Float 당 4 바이트	최고의 정확도와 원본 모델 크기 (~ 90MB)로 이어진다.
Float 당 2 바이트	정확도가 약간 낮아지고 모델 크기가 2배 작아진다.
Float 당 1 바이트	정확도를 낮추고 모델 크기를 4배로 (~ 22MB) 줄인다.

⑥ ModelUrl- 모델의 사용자 정의 URL을 지정하는 선택적 문자열이다. 이는 GCP에서 호스팅 되는 모델에 액세스할 수 없는 지역, 개발 또는 국가에 유용하다. 그래서, 기본적으로 PoseNet은 0.75 배율로 MobileNet 아키텍처를 로드하며, 미드 레인지 / 하위 GPU가 장착된 컴퓨터에 권장한다. 모바일에는 배율이 0.50인 모델이 권장하므로, 더 강력한 GPU를 갖춘 컴퓨터에는 ResNet 구조를 권장한다.

(다) 결과

자세 추정의 제1단계에서, 정사각형 이미지가 사전 훈련된 모델을 통해 공급된다. 사용자는 MobileNet 과 ResNet 사이에서 모델 아키텍처를 선택할 수 있다. MobileNet 모델이 더 빠르므로 실시간 응용 프로그램에 더 적합하지만 ResNet 모델보다 정확도가 떨어진다. [그림 2-12]에서 로드된 모델은 입력 이미지에서 사람을 감지하고 17개의 KeyPoint(눈, 코, 귀, 왼쪽 어깨, 오른쪽 무릎 등)에 대한 신뢰도 점수 및 위치뿐만 아니라 자세 신뢰도 점수를 반환한다.

PoseNet은 이미지를 처리할 때 사용자가 선택한 출력 단계에 따라 원본 이미지를 축소하게 된다. 확장된 출력 단계는 모델이 더 큰 출력 해상도를 만들어야 할 때 사용하게 된다. 이때, 모델 출력은 KeyPoint HeatMap⁹⁾ (KeyPoint 당 하나의 채널) 및 Offset Vector(KeyPoint 당 두 개의 채널 (x, y))이다. Offset Vector¹⁰⁾는 KeyPoint를 근접하게 하여 정확도를 향상하게 된다.

9) HeatMap: 다양한 정보를 일정한 이미지 위에 열분포 형태의 색상 그래픽으로 출력하는 것

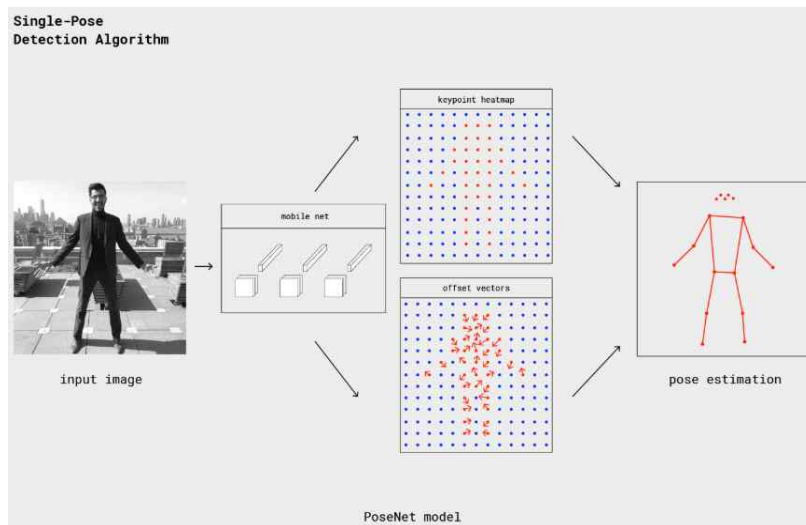
10) Offset Vector: 사각형을 이동할 가로 및 세로 거리를 지정하는 벡터.

KeyPoint를 얻기 위해 각 신체의 최고 점수 HeatMap x 및 y 지수에 출력 보폭을 곱한 다음 해당 Offset Vector에 추가하게 된다. 마지막으로 각 KeyPoint 신뢰 점수는 HeatMap 위치의 신뢰 점수가 된다. 자세 신뢰도 점수는 KeyPoint 점수의 평균이다.

① 모델 입력 처리 : 출력 보폭에 대한 설명

먼저 출력 보폭을 논의하여 PoseNet 모델 출력(주로 HeatMap 및 Offset Vector)을 얻는 방법을 설명하면 다음 [그림 2-12]와 같다.

편리하게 PoseNet 모델은 이미지 크기가 변하지 않기 때문에 이미지의 축소 여부와 관계없이 원본 이미지와 같은 배율로 자세 위치를 예측할 수 있다. 즉, 위에서 언급한 출력 보폭을 실행 시간에 설정하여 성능을 낮추면서 PoseNet을 더 높은 정확도로 구성할 수 있다.



[그림 2-12] PoseNet을 사용한 단일 인물 자세 감지기
파이프라인

출력 보폭은 입력 이미지 크기를 기준으로 출력 크기를 얼마나 줄일지 결정하여야 하며, 레이어의 크기와 모델 출력에 영향을 준다. 결국, 출력 보폭이

높을수록 네트워크 및 출력 레이어의 해상도가 낮아지고 그에 따라 정확도가 떨어진다. 예컨대, 이 구현에서 출력 보폭은 8, 16 또는 32 값을 가질 수 있다. 즉, 출력 보폭 32는 성능은 가장 빠르지만 정확도는 가장 낮고, 8은 정확도가 가장 높지만 성능은 가장 느리다. 예컨대, 16부터 시작하는 것이 좋다.

Hood 아래에서 출력 보폭을 8 또는 16으로 설정하면 레이어의 입력 보폭이 줄어들어 더 큰 출력 해상도를 생성한다. 그런 다음 후속 레이어의 컨볼루션 필터가 더 넓은 시야를 가질 수 있게 하려면 강력한 컨볼루션이 사용된다. 그러나 출력 보폭이 32일 때는 강력한 컨볼루션이 적용되지 않는다. Tensorflow는 강력한 컨볼루션을 지원했지만 TensorFlow.js는 지원하지 않았으므로 PR(Performance Reasons)을 추가하도록 하였다.

② 모델 출력 : HeatMap 및 Offset Vector

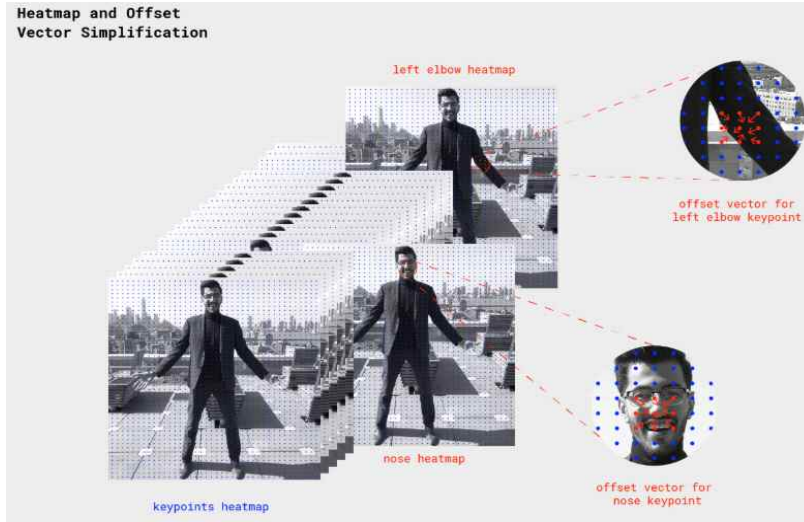
PoseNet이 이미지를 처리할 때 실제로 반환되는 것은 자세 KeyPoint에 해당하는 이미지에서 높은 신뢰도 영역을 찾기 위해 디코딩 할 수 있는 Offset Vector와 함께 HeatMap이다. [그림 2-13]는 각 자세 KeyPoint가 하나의 HeatMap 텐서(Tensor)¹¹⁾와 Offset Vector 텐서와 어떻게 관련되어 있는지를 개략적으로 보여준다.

PoseNet이 반환한 17개의 자세 KeyPoint 각각은 KeyPoint의 정확한 위치를 결정하는 데 사용되는 하나의 HeatMap 텐서인 선형 관계를 나타내는 다중 선형대수의 대상과 하나의 Offset Vector 텐서와 연결된다.

이 두 출력은 모두 높이와 너비가 3D 텐서이며 해상도라고 할 수 있다. 해상도는 입력 이미지 크기와 출력 보폭에 의해 결정되며, 만약 너비가 225 픽셀 입력 이미지와 출력 16의 보폭일 경우 식(1)에 의해 $((225-1) / 16) + 1 = 15$ 의 출력 해상도를 나타낸다.

$$\text{해상도} = ((\text{InputImageSize} - 1) / \text{OutputStride}) + 1 \quad (1)$$

11) Tensor : 벡터 계산을 단순화하기 위해 같은 성질의 여러 벡터를 한 행렬 안에 표기하고 단순화한 것



[그림 2-13] PoseNet이 반환한 17개의 자세 KeyPoint

[그림 2-13]의 각 HeatMap은 크기 해상도 $\times$ 해상도 $\times$ 17의 3D 텐서이다. 17은 PoseNet에서 감지한 KeyPoint 수이다. 예를 들어, 이미지 크기가 225이고 출력 보폭이 16이면 $15 \times 15 \times 17$ 이다. 3차원의 각 슬라이스는 특정 KeyPoint에 대한 HeatMap에 해당한다. 해당 HeatMap의 각 위치에는 신뢰 점수가 있으며, 이는 해당 KeyPoint 유형의 일부가 해당 위치에 존재할 확률이다. 열화상 점수는 각 KeyPoint가 각 그리드 정사각형에 존재할 가능성의 분류를 제공하는 원래 이미지가 15×15 그리드로 분리된 것으로 생각할 수 있다.

각 Offset Vector는 크기 해상도 $\times$ 해상도 $\times$ 34의 3D 텐서이며 여기서 34는 KeyPoint 수 $\times$ 2이다. 이미지 크기가 225이고 출력 보폭이 16이면 $15 \times 15 \times 34$ 이다. HeatMap은 KeyPoint가 있는 위치의 근사치이므로 Offset Vector는 HeatMap 포인트에 해당하는 위치에 해당하며 해당 HeatMap 포인트에서 벡터를 따라 이동할 때 KeyPoint의 정확한 위치를 예측하는 데 사용된다. Offset Vector의 처음 17개 슬라이스는 벡터의 x 와 마지막 17개의 y 를 포함한다. Offset Vector 크기는 원본 이미지와 크기가 같다.

③ 모델의 출력에서 자세 추정

이미지가 모델을 통해 공급된 후, 출력으로부터 자세를 추정하기 위해 몇 가지 계산을 수행한다. 예를 들어, 단일 자세 추정 알고리즘은 각각 자체 신뢰 점수 및 x, y 위치를 갖는 KeyPoint(신체 ID로 색인화됨)의 배열을 포함하는 자세 신뢰 점수를 반환한다.

자세의 KeyPoint를 얻으려면 HeatMap에서 S 자형 활성화가 수행되어 점수를 얻을 수 있으며, 이것을 식(2)으로 표현하면 다음과 같다.

$$\text{점수} = \text{HeatMap.Sigmoid}() \quad (2)$$

다차원 배열의 경우에 차원에 따라 가장 큰 값의 인덱스들을 반환해주는 함수 Argmax_{2d} 는 핵심 신체 신뢰도 점수에 대해 수행되어 각 신체에 대해 가장 높은 점수를 갖는 HeatMap에서 x 및 y 지수를 얻는다. 이렇게 하면 크기가 17×2 인 텐서가 생성되며 각 행은 HeatMap에서 각 신체의 점수가 가장 높은 y 및 x 인덱스로 표현할 수 있으며, 이것을 식(3)으로 표현하면 다음과 같다.

$$\text{HeatMapPositions} = \text{Scores.Argmax}(y, x) \quad (3)$$

각 신체의 Offset Vector는 해당 신체에 대한 HeatMap의 x 및 y 색인에 해당하는 Offset에서 x 및 y 를 가져와서 검색된다. 그러면 크기가 17×2 인 텐서가 생성되며 각 행은 해당 KeyPoint의 Offset Vector이며, 인덱스 k 의 신체에서 HeatMap 위치가 y 및 d 였으면 Offset Vector는 다음의 식(4)과 같이 된다.

$$\text{OffsetVector} = [\text{Offsets.get}(y, x, k), \text{Offset.get}(y, x, 17 + k)] \quad (4)$$

다음으로 KeyPoint를 얻기 위해 식(5)에서 각 신체의 HeatMap x 및 y 에 출력 보폭을 곱한 다음 해당 Offset Vector에 원래 이미지와 같은 스케일로

추가한다.

Keypoint Positions = HeatMap Positions x OutputStride + Offset 벡터 (5)

마지막으로 각 Keypoint 신뢰 점수는 HeatMap 위치의 신뢰 점수이며, 자세 신뢰도 점수는 Keypoint 점수의 평균이다.

(라) 결론

단일 자세에는 신뢰도 점수와 각 신체 ID 별로 색인된 Keypoint 배열이 있으며 각각 점수와 위치가 있으며, 모든 핵심 사항은 신체 일부로 색인 된다. 각 신체와 그 ID는 아래와 [표 2-4]과 같다¹²⁾

[표 2-4] 각 신체와 ID

Id	Part
0	Nose
1	Left Eye
2	Right Eye
3	Left Ear
4	Right Ear
5	Left Shoulder
6	Right Shoulder
7	Left Elbow
8	Right Elbow
9	Left Wrist
10	Right Wrist
11	Left Hip
12	Right Hip
13	Left Knee
14	Right Knee
15	Left Ankle
16	Right Ankle

12) <https://medium.com/tensorflow/>

4) Human Pose Estimation 주요 모델

가) Convolutional Pose Machines(CPM)

(1) 소개

본 연구에서는 Articulated Pose를 추정하는 Convolutional Pose Machines (CPM)을 제안하고 있다. CPM은 Pose Machine 구조가 가지고 있는 장점을 활용하고 있다. 이러한 장점으로 CPM은 각 파트의 신호와 이미지 사이의 종속성을 학습하고, 학습과 추론 과정의 통합을 거쳐서 순서대로 실행되는 모듈 디자인 등이 있으며, 이러한 부분을 컨볼루션 구조를 이용하여 하나로 통합할 수 있다(L. Pishchulin, et al., 2015; J. Tompson, et al., 2014; 2015; A. Toshev, et al., 2013; S. E. Wei, et al., 2016).

CPM은 컨볼루션 네트워크의 연속으로 이루어져 있으며, 각 파트에 대한 2D 신뢰지도 생성을 반복적으로 수행한다. CPM의 각 단계에서는 이전 단계에서 생성된 특징 맵과 이미지 특징이 입력으로 사용되며, 이전 단계에서 넘어온 신뢰지도를 이용하면 현재 단계의 모호성을 제거할 수 있다. 이와 같은, 단계가 거듭할수록 추정된 파트의 위치는 점점 더 정교화된다. 각 파트들간의 상호작용을 위해서 이미지와 신뢰지도 전체에서 넓은 Receptive Field를 사용하며, 관련된 연구의 기여는 다음과 같이 정리된다.

첫 번째, 컨볼루션 구조의 순서로 구성된 모델로 지역적 정보를 활용하도록 학습하였다(J. Carreira, et al., 2015; X. Chen and A. Yuille, 2014).

두 번째, 그래픽 모델 없이 이미지 특징과 이미지에 종속적인 지역 모델을 학습하도록 하는 시스템적 디자인과 학습법을 사용하였다(W. Ouyang, 2014; T. Pfister, et al., 2015; S. E. Wei, et al., 2016).

(2) 모델 아키텍처

CPM의 아키텍처에 살펴보면, p 번째 랜드마크의 픽셀 위치를 $Y_p \in Z \subset \mathbb{R}^2$

이라고 하고, Z 는 이미지 내의 모든 (u, v) 위치를 나타낸다. 우리의 목적인 각 P 파트의 이미지 위치 $Y = (Y_1, \dots, Y_p)$ 을 추정하는 것이다. Pose Machine은 멀티 클래스 추정기 $g^t(\cdot)$ 의 연속으로 이루어져 있고, 이것들은 각 파트의 위치를 추정 하도록 학습되어 있다. 단계 $t \in \{1, \dots, T\}$ 에서 분류기 g^t 는 이미지의 Z 위치에서 추출된 특징과 이전 단계의 분류기에서 계산된 z 주변의 맥락적 정보를 이용하여 Z 위치에서의 각 파트의 신뢰도를 추정한다.

첫 번째 단계 $t=1$ 에서 신뢰도는 다음의 수식(6)과 같이 계산된다.

$$g_1(Xz) \rightarrow \left\{ b_1^p(Y_p = z) \right\}_{p \in \{0, \dots, p\}} \quad (6)$$

이렇게 계산된 이미지 내의 모든 파트에 대한 신뢰도를 $b_t^p \in R^{w \times h}$ 로 표현 한다면, $b_t^p[u, v] = b_t^p(Y_p = z)$ 로 생각할 수 있다. P 개의 각 파트와 배경에 대한 것을 생각했을 때 편의상 $b_t \in R^{w \times h \times (P+1)}$ 로 생각하면 된다.

다음 단계에서는, 이미지로부터 계산된 특징 $X_z^t \in R^d$ 와 각 파트 위치 주변의 맥락 정보를 이용하여 신뢰도를 다시 다음의 수식(7)과 같이 계산한다.

$$g_t(X_z, \psi_t(z, b_{t-1})) \rightarrow \left\{ b_t^p(Y_p = z) \right\}_{p \in \{0, \dots, p+1\}} \quad (7)$$

여기서 $\psi_t > 1(\cdot)$ 는 신뢰지도 b_{t-1} 을 맥락 특징으로 변환하는 함수이다. 또한, 각 단계에서 사용되는 특징 $X'z$ 은 첫 번째에서 사용되는 X 와 다른 특징을 생성하여 사용할 수도 있다.

실제 Pose Machine은 g^t 에는 Boosted Random Forests를 사용하였고, 모든 단계에서의 이미지 특징은 동일한 것을 사용하였으며($X' = XX' = X$), 이미지의 특징과 맥락의 특징은 사람이 직접 설계한 것을 사용하였다.

(3) 방법

① KeyPoint Localization Using Local Image Evidence

CPM의 첫 번째 단계는 이미지의 지역적 정보만을 사용하여 신뢰도를 추정한다. 이 때문에 첫 번째 단계에서의 Receptive Field는 작은 패치로 제한된다. 네트워크는 5개의 컨볼루션 레이어와 마지막에는 2개의 1×1 컨볼루션 레이어로 구성되어 있다. 구현에서는 이미지를 Normalize 한 뒤, 368×368 로 잘라내어 입력하였다. 이 경우 Receptive Field는 160×160 이 된다. 출력으로는 160×160 의 영역에 대하여 각 파트에 해당하는 $P+1$ 크기의 출력을 내보내게 되어있다(S. E. Wei, et al., 2016).

② Sequential Prediction with Learned Spatial Context Feature

얼굴과 어깨와 같은 부분에서는 검출률이 높지만, 팔이나 다른 같은 부분에서는 자세나 모양새가 큰 변화를 있으므로 검출률이 떨어지게 된다. 이러한 문제점으로 각 파트 위치 주변의 신뢰지도는 많은 정보를 가지고 있다. 예를 들어 오른쪽 팔꿈치를 검출하고자 할 때 오른쪽 어깨에 대한 신뢰지도는 좋은 신호로 사용될 수 있다. 2단계 이후에서의 추정기 $g_t > 1$ 는 이전 단계의 신뢰지도의 지역 맥락 $\psi_t > 1(\cdot)$ 특징을 이용하면 추정 성능을 향상할 수 있다. Pose Machine의 두 번째 단계는 분류기 g_2 는 이미지 특징 X^2Z 와 이전 단계의 각 파트를 ψ 을 이용하여 만든 그 특징을 입력으로 한다. 함수 ψ 는 위치 Z 주변 영역을 입력으로 하여 인코딩하게 된다. CPM에서는 특별한 특징을 계산하지 않고, 이전 단계 특징 맵의 Receptive Field 영역을 직접 사용하였다(S. E. Wei, et al., 2016).

네트워크를 디자인할 때에는, [그림 2-13]의 두 번째 단계의 출력이 가지는 Receptive Field는 멀리 떨어져 있는 파트까지 포함할 수 있도록 적당히 넓어야 한다. 이전 단계의 출력을 별도로 처리하지 않고 그대로 제공하게 되면, 컨볼루션 레이어는 자연스럽게 가장 그럴듯한 특징을 골라서 해당 정보

를 이용하게 된다. 첫 번째 레이어는 작은 Receptive Field를 가지지만, 여러 컨볼루션 레이어를 거침으로써 Receptive Field를 대폭 늘리도록 한다. 여기에서는 8x로 다운 스케일 된 히트맵을 출력하도록 설계하였다. 실험에서는 넓은 Receptive Field를 가질수록 성능이 높아졌으며, 304×304 이미지에서, 250픽셀 주변 영역 사용하게 되면 성능 향상이 멈추었다(S. E. Wei, et al., 2016).

실험적으로 368×368 크기로 짧게 잘랐을 때 가장 성능이 좋았으며, 두 번째 단계에서는 첫 번째 단계에서의 신뢰지도 31×31 크기를 가져오게 하였다. 이 크기는 실제 이미지의 400×400 영역과 같은 것이다. 이어지는 단계에서는 더욱 Receptive Field가 커지며, 6단계까지 계산을 수행하였다.

③ Learning in Convolutional Pose Machines

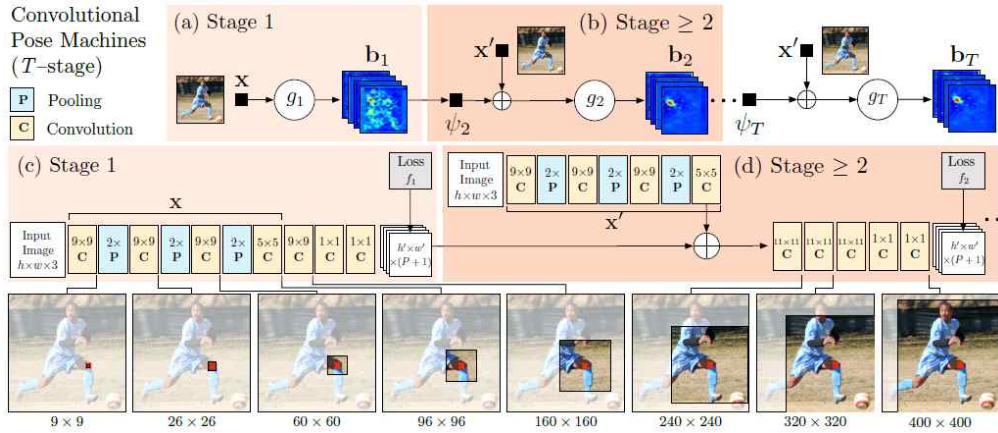
[그림 2-14]에서 각 단계의 Pose Machine은 반복적으로 학습이 된다. 각 단계에서의 로스 함수는 추정된 신뢰지도와 이상적인 경우의 신뢰지도 사이의 거리를 최소화하도록 하였다. 이상적인 신뢰지도 $b_p^*(Y_p = z)$ 은 가우시안의 정점을 정답 위치에 놓도록 생성하였다. 단계 t에서의 로스 함수는 다음의 수식(8)과 같다.

$$f_t = \sum_{p=1}^{P+1} \sum_{z \in Z} \| b_t^p(z) - b_p^*(z) \|_2^2. \quad (8)$$

전체 목표 함수는 모든 단계에서의 로스를 더해서 수식(9)와 같이 계산한다.

$$F = \sum_{t=1}^T f_t. \quad (9)$$

모든 후속 단계에서 이미지 특징 x' 을 공유하기 위해, $t \geq 2$ 단계에서 해당 컨볼루션 레이어의 가중치를 공유합니다.(S. E. Wei, et al., 2016).



[그림 2-14] Architecture and Receptive Fields of CPM(S. E. Wei, et al., 2016)

(4) 결과

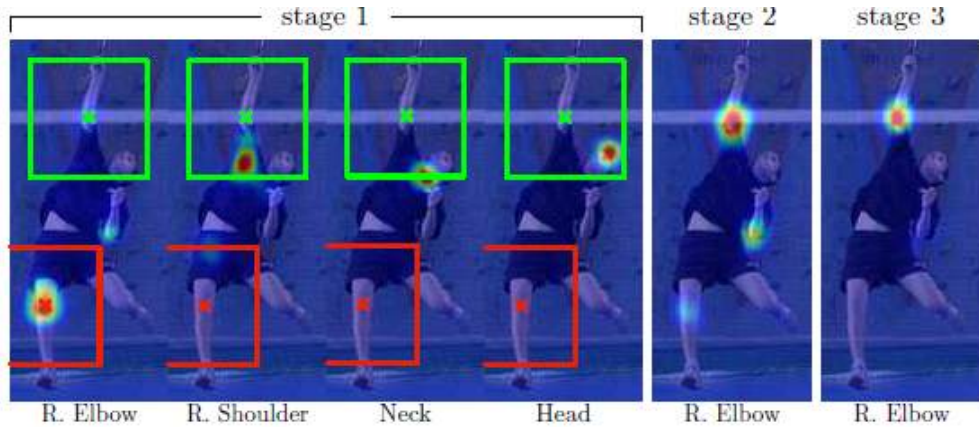
CNN을 이용해 각 관절의 HeatMap을 출력하는 네트워크 구조이다.



[그림 2-15] Convolutional Pose Machines 구조(S. E. Wei, et al., 2016)

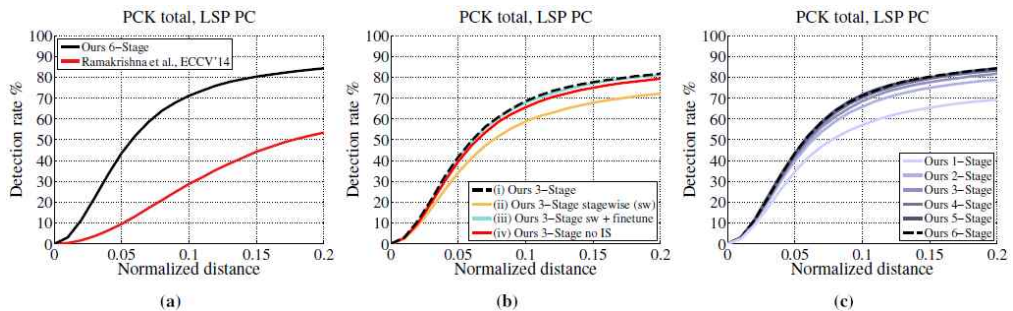
[그림 2-15] Stage 1의 결과는 이미지 Feature와 함께 Stage 2의 입력으로 다시 들어가게 된다. 반복적으로 같은 HeatMap 결과를 여러 Stage에 걸쳐서 처리한 최종적인 결과가 네트워크의 Output이 된다(S. E. Wei, et al., 2016). 이렇게 하는 이유로 초반의 Stage에서 Local 한 이미지 영역을 통해 관절을 찾아내고, 후반의 Stage에서 더욱 커진 Receptive Field의 영향으로 서로 다른 관절 간의 관계까지 고려해야 하므로 정확한 HeatMap 결과를 얻을

수 있다.



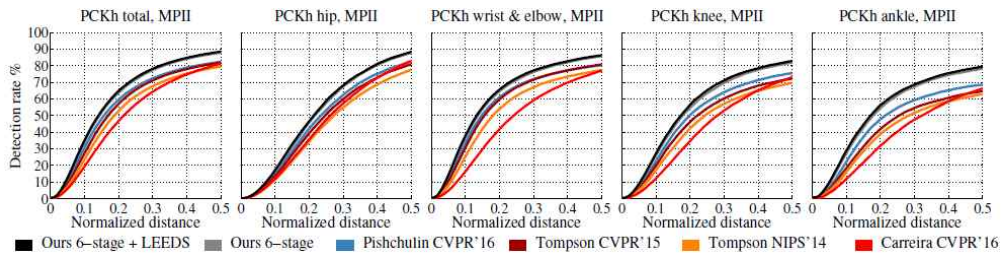
[그림 2-16] CPM에서 생성되는 단계별 신뢰지도(S. E. Wei, et al., 2016)

[그림 2-16]에서 보이는 것처럼 Stage 초반에는 '오른쪽 팔꿈치'에 대해 무릎 영역에 HeatMap이 생성되었지만, 여러 Stage를 거치면서 이는 제거되는 것을 볼 수 있다. 그리고 Receptive Field 즉 받아들이는 필드의 크기가 커질수록 정확도 역시도 증가하는 것을 볼 수 있다. 또한 중간 HeatMap에 대해 Loss를 계산하도록 설계되어 있어서 Gradient가 Vanishing이라는 문제 역시도 상대적으로 덜하다고 말할 수 있다. 요약하면, 이들 결과 비교를 아래의 [그림 2-17], [그림 2-18], [그림 2-19]로 평가한다.

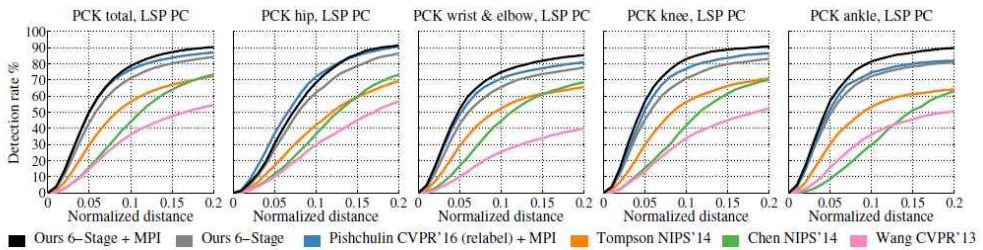


[그림 2-17] Comparisons on 3-stage Architectures on the LSP DataSet(PC)(S. E. Wei, et al., 2016)

Extended LSP DataSet에서 학습용 11,000개 이미지와 테스트용 1,000개 이미지로 구성하여, 백분을 올바른 KeyPoint 값으로 Percentage Correct KeyPoint(PCK)를 사용하여 평가했다. PCK 측정항목으로는 MPII DataSet과 같은 기능 체계를 사용하여 모델 정확도는 84.32%이며, MPII 학습 데이터를 추가할 때 90.5%로 좀 더 높은 정확도를 달성하였다. MPII 데이터를 여기에 추가하면 라벨 품질이 향상되어 LSP DataSet 보다 성능이 크게 향상될 것이다.



[그림 2-18] Quantitative Results on the MPII Dataset using the PCKh Metric(S. E. Wei, et al., 2016)



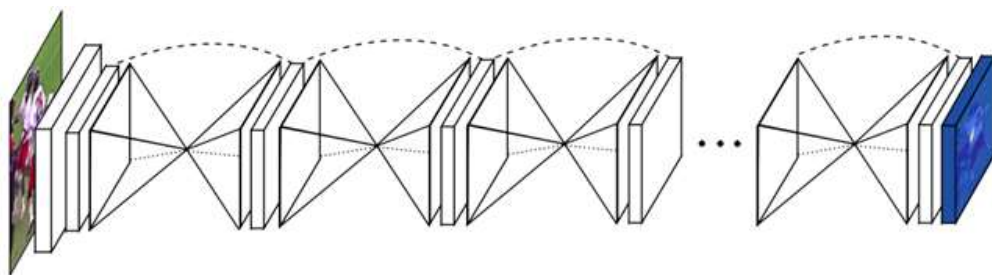
[그림 2-19] Quantitative Results on the LSP DataSet using the PCK Metric(S. E. Wei, et al., 2016)

나) Stacked Hourglass Networks for Human Pose Estimation

(1) 소개

사진 속에 담긴 사람들의 행동에 대한 이해를 높이기 위해서는 그들이 취하고 있는 자세에 대해 먼저 이해하는 것이 중요하다. 이러한 연구를 Human Pose

Estimation이라 하며, 이미 오래 전 Computer Vision에서 다루어지고 있는 분야이다(A. Newell, et al., 2016; Computerphile, 2016; Michigan Vision and Learning Lab, 2016). 현재의 Pose Estimation System들은 대부분 수작업으로 생성한 Feature들이나 Graphical Model을 사용하는 것에서 ConvNets을 주요 구조로 반영하는 것으로 옮겨갔다. 이 ConvNets을 이용하는 것의 연장선으로, 이미지에 대한 모든 Scale 정보를 Down Sampling 과정에서 추출하고 이를 Up Sampling 과정에 반영하여 Pixel-Wise Output을 생성하는 것을 목표로 한다. 또한, 해당 아이디어(Single Hourglass)를 확장하고 여러 Hourglass Module을 연속하여 처리하는 방식으로 Stacked Hourglass Networks 구조를 소개한다. [그림 2-20]에서 이 구조는 여러 Scale에 대해서 반복적으로 Bottom-Up, Top-Down Inference를 가능하게 하고, 결과적으로 표준 Pose Estimation Benchmarks(FLIC and MPII Human Pose)에서 확연한 성능 향상을 보여준다(Y. LeCun, et al., 1998; A. Krizhevsky, et al., 2012; C. Szegedy, et al., ; 2015; S. Ioffe, et al., 2015; K. He, et al.,;2016; J. Tompson, 2014; S. E. Wei, et al., 2016; M. Andriluka, et al., 2014; S. Johnson, et al., 2010; J. Long, et al., 2015; A. Newell, et al., 2016).



[그림 2-20] Pose Estimation Stacked Hourglass Modules(A. Newell, et al., 2016)

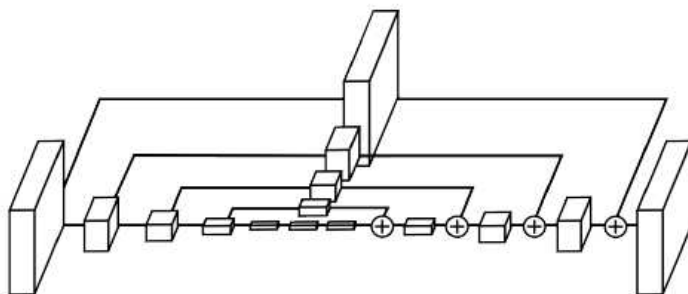
(2) 모델 아키텍처

(가) Hourglass Design

얼굴이나 손과 같은 Feature들을 식별하는 것에는 Local Evidence가 중요

하지만, 최종적인 자세의 추정하기 위해서는 Full Body에 대한 이해가 필요하다. 그리고, 이를 위해서 여러 Scale에 걸쳐서 필요한 정보를 포착해낼 수 있어야 한다.

Hourglass는 이러한 모든 Feature를 잡아내어 네트워크의 출력인 픽셀 단위의 예측에 반영하도록 한다.



[그림 2-21] Residual Module들로 구성된 “Hourglass” Module(A. Newell, et al., 2016)

기존의 몇몇 방식들이 여러 크기에 대한 Feature를 뽑아내기 위해 분리된 다수의 파이프라인을 사용하지만, [그림 2-21]에서 Residual Module들로 구성된 “Hourglass” Module이 연구에서는 Skip Layer를 이용하여 단 하나의 파이프라인만으로 Spatial Information을 유지하는 방식을 채택하였다(J. Tompson, et al., 2014; S. E. Wei, 2016; A. Newell, et al., 2016). 따라서, Hourglass Network는 다음과 같은 구조를 보인다.

① Feature의 추출과 저차원으로의 DownSampling을 위해 Convolutional Layer와 Maxpooling Layer를 사용한다.

② Max Pooling 단계에서의 입력을 통해 별도의 Branch로 내보내고 여기에 Convolution 연산을 적용한다. 이를 통해서 Scale마다 Feature가 추출된다.

③ 가장 낮은 Resolution에 도달하면은 UpSampling 과정의 Scale 별로 추출한 Feature들을 조합한다. UpSampling으로는 Nearest Neighbor Up Sampling 방식을, Feature와의 조합에는 Element Wise Addition 연산을 이용한다.

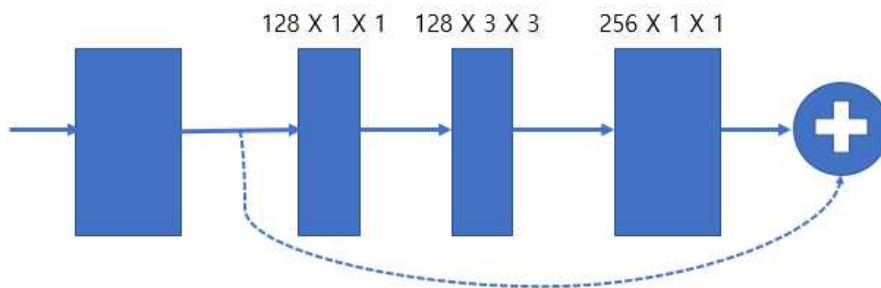
④ 네트워크는 대칭적인 구조를 띈다.

⑤ Output Resolution에 도달하면 두 번의 연속된 1×1 Convolution 연산을 적용하여 최종적인 예측값을 출력한다.

⑥ 네트워크의 출력은 각 관절에 대한 추정 값이 담긴 HeatMap들이다.

(나) Layer Implementation

각 Box는 [그림 2-22]와 같은 Residual Module이다. 3×3 보다 큰 Filter는 사용하지 않고 메모리 사용량을 줄이기 위해 Bottle Neck 구조를 이용한다. 64×64 크기의 입력으로 성능에는 지장이 없다고 한다(A. Newell, et al., 2016).



[그림 2-22] Residual Module(A. Newell, et al., 2016)

눈여겨볼 점은 해상도가 256×256 일 때 굉장히 높은 GPU 메모리 사용량을 요구하기 때문에 다음의 [그림 2-23]처럼 이미지 전처리 과정을 거친다

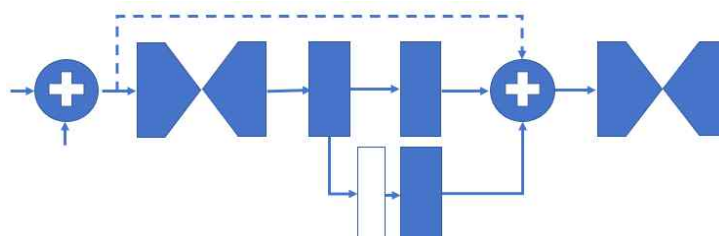
는 것이다.

구분	알고리즘
Stacked HG	Input size: (256, 256, 3) Initial processing of the image <code>localcnv1=nnlib.SpatialConvolution(3,64,7,7,2,2,3,3)(inp)</code> 128 <code>localcnv1=nnlib.ReLU(true)(nn.SpatialBatchNormalization(64)(cnv1_))</code> <code>localr1=Residual(64,128)(cnv1)</code> <code>localpool=nnlib.SpatialMaxPooling(2,2,2,2)(r1)</code> 64 <code>localr4=Residual(128,128)(pool)</code> <code>localr5=Residual(128,opt.nFeat3)(r4)</code> 1. 7x7 convolutional layer with stride2(w&h size: 256 -> 128, output channel: 64) 2. a residual module(w&h size: 128, output channel: 128) 3. max pooling layer(w&h size: 128 -> 64, output channel: 128) 4. Two subsequent residual modules(output channel: 256)

[그림 2-23] Residual Module Algorithm

참고로 Hourglass Network 내부의 모든 Residual module은 Output Channel의 크기가 256이다.

(다) Stacked Hourglass with Intermediate Supervision



[그림 2-24] 다수의 Hourglass Network를 쌓아놓은 구조(A. Newell, et al., 2016)

[그림 2-24]에서 Stacked Hourglass Networks는 여러 개의 Hourglass Network를 쌓아놓은 구조다. 해당 구조는 반복적인 Bottom-Up, Top-down Inference를 가능하게 하며, 이를 통하여 Initial Estimates와 이미지 전반에 대한 Feature를 다시 추정(Reevaluation)할 수 있게 한다. 여기서 중요한 점은 중간마다 얻어지는 예측값(The Prediction of Intermediate HeatMaps)들에 대해서도 Ground Truth의 Loss를 적용할 수 있다는 것이다(Intermediate Supervision). 반복적인 예측값의 조정으로 좀 더 세밀하게 결과를 도출할 수 있으며, 이를 통해 중간중간 적용되는 Loss를 좀 더 깊고 안정되게 학습이 가능하리라 예상한다(C. Szegedy, et al., 2015; K. He, et al., 2016; A. Newell, et al., 2016).

Intermediate Predictions에 1×1 Convolutional Filter를 적용하여 그 차원수를 증가시키고, 이를 이번 Hourglass Stage의 Intermediate Features와 이전 Hourglass Stage에서의 Features Output과 합산한다. 그리고 이 합산 결과는 고스란히 연이어 등장하는 Hourglass Module의 입력이 된다. 최종적인 Network Design에서는 8개의 Hourglass Module이 사용되었으며, 주의할 점으로는 아래와 같다(A. Newell, et al., 2016).

- ① Hourglass Module 간에 가중치를 공유하지 않는다.
- ② Loss를 계산할 때, 모든 Hour Glass들의 Prediction에 대해 같은 Ground Truth를 사용한다.

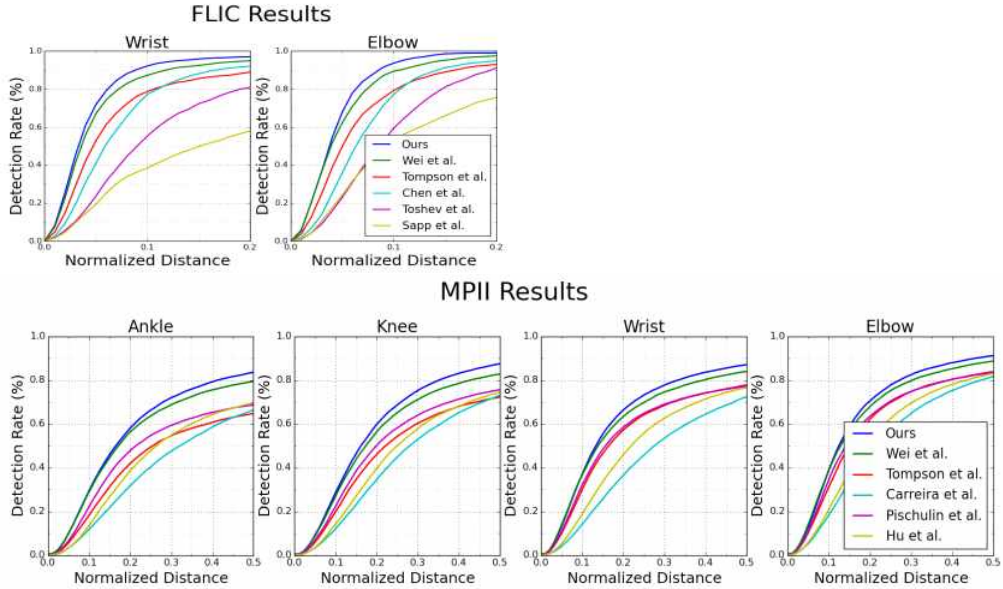
(3) 결과

MPII와 FLIC¹³⁾ DataSet에서 Standard Percentage of Correct KeyPoint(PCK)를 이용한 성능 평가. [그림 2-25]에서 모든 방면에서 월등히 우수한 성능을 보인다(A. Newell, et al., 2016).

DeepCut, IEF, Tompson CVPR'15, CPM 등이 있는데 가장 좋은 성능을 보

13) FLIC: 5003개 이미지 (3987: trainset, 1016: testset)

였다. 손목, 팔꿈치, 무릎과 발목부분의 정확도는 타 논문 대비 3.5% 상승되었다(PCKh@0.5:문턱값 성능지표). 추가로 평균 에러율은 12.8%이다.



[그림 2-25] Results on MPII & FLIC(A. Newell, et al., 2016)

[그림 2-26]는 5가지 서로 다른 네트워크 구조들의 성능을 비교한 결과이며, 정확성은 손목, 팔꿈치, 무릎 및 발목 전체에 걸쳐 평균과 다른 네트워크 디자인, 오른쪽에 설명된 원은 손실이 적용되는 위치를 나타내는 데 사용된다(A. Newell, et al., 2016).

HG : 한 개의 HG 네트워크(long version),

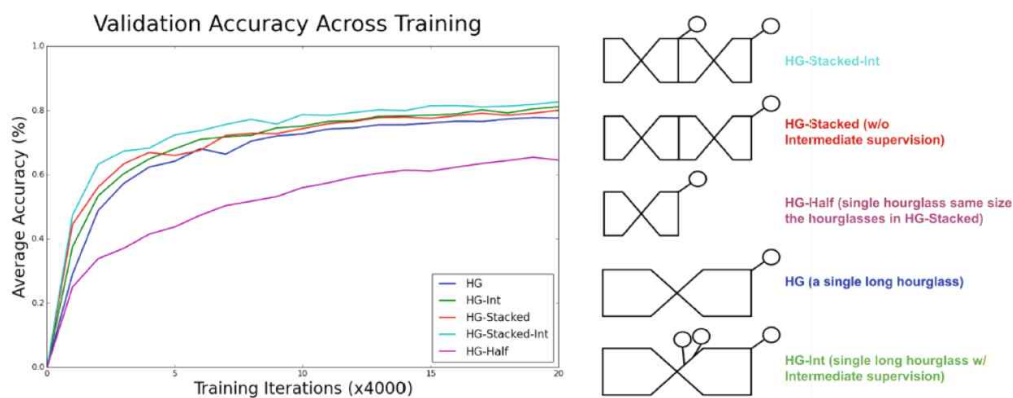
HG-Int : 한 개의 HG + Intermediate Supervision,

HG-Stacked : 두 개의 HG 네트워크,

HG-Stacked-Int : 두 개의 HG + Intermediate Supervision,

HG-Half : 한 개의 HG 네트워크,

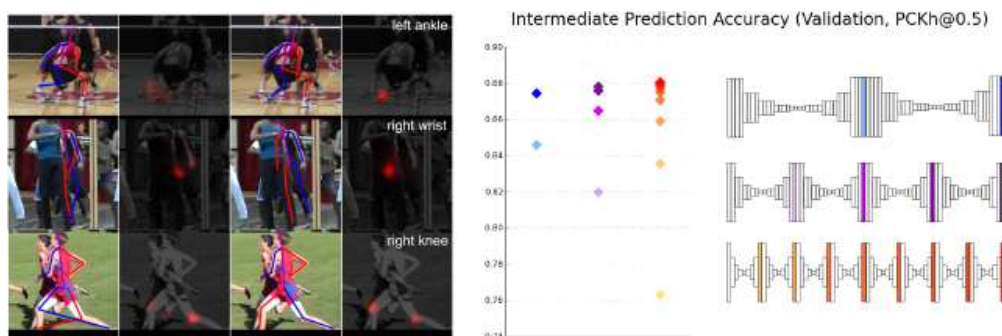
HG 네트워크의 갯수가 정확도에 어떤 영향을 주는지 알 수 있는 그림이다. 모래시계 모양의 네트워크들이 많이 쌓을수록 여기서는 빨간색으로 표시되는 부분에 정확도가 가장 높은 것을 알 수 있다. 반면, 적게 쌓은 네트워크



[그림 2-26] Comparison of Validation Accuracy as Training Progresses(A. Newell, et al., 2016)

는 성능이 그다지 좋지 않았다. 87.4% - > 87.8% -> 88.1% 로 개선된다.

[그림 2-27]에서는 네트워크가 깊어질수록 Intermediate Prediction Accuracy가 서서히 높아지는 것을 볼 수 있다. 다수의 사람이 한 사진에 담겨 있을 때, 네트워크는 사진의 중앙에 있는 사람에 대해 동작하는 경향성을 보인다(A. Newell, et al., 2016).



[그림 2-27] Left: Example validation, Right: Validation accuracy(A. Newell, et al., 2016)

(4) 결론

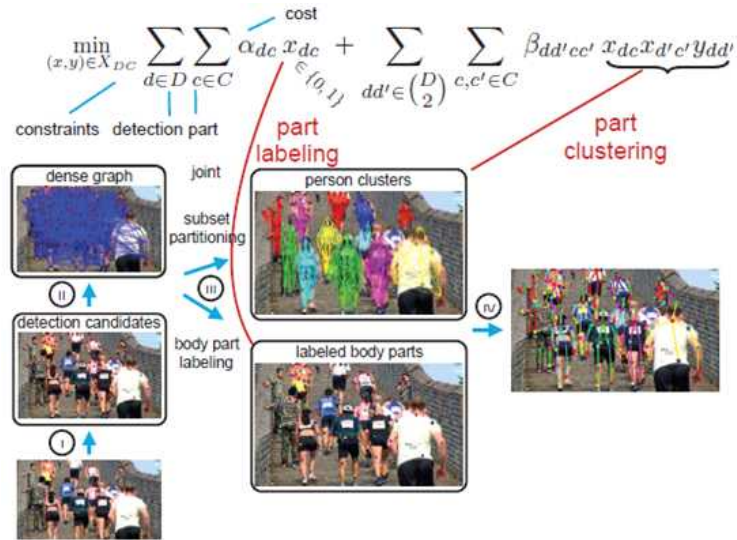
Hourglass 구조로 다양한 Scale에 대응할 수 있도록 네트워크 구조를 만

들고, 이를 Stacking 해 사람 자세에 대한 논리적인 Refine이 가능하도록 설계한 네트워크 구조가 핵심인 논문이다. 1명의 사람에 대해 관절 위치를 HeatMap으로 출력하는 단순한 구조로 좋은 성능을 Report 하였고, 반복되는 구조가 성능에 미치는 영향에 대해 분석한 점이다.

다) DeeperCut

(1) 소개

DeeperCut은 [그림 2-28]에서 DeepCut의 더 깊고 강력하며 빠른 버전이며 여러 사람이 있는 장면에서 명료한 자세 추정을 생성할 수 있다. 신체 부위 탐지기로 매우 깊은 잔류 네트워크(ResNet)를 사용하여 심화한다.



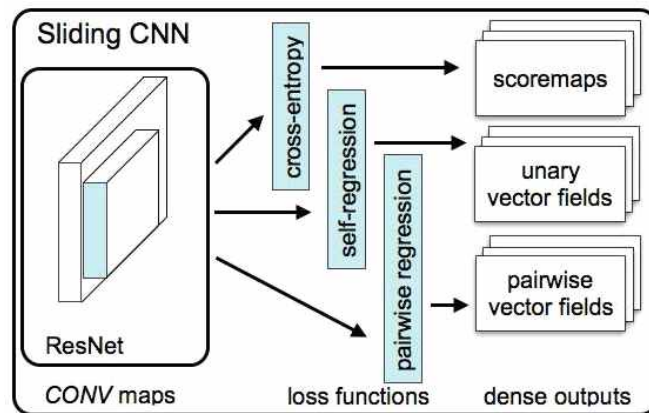
[그림 2-28] DeeperCut이 인식한 사람들의 자세 예시(E. Insafutdinov, et al., 2016)

신체 부위 간에 새로운 이미지 조건 쌍조합 용어를 사용하기 때문에 더 강해지는 것이다. 다자간 자세 추정을 해결하는 데 도움이 되므로, 이 두 기능은 상당한 시너지를 발생시키고 실행 시간을 2~3배 줄임으로써 더 빨라진

다. 새로운 증분 최적화 방법은 인간 자세 추정 정확도를 향상하면서 4배 더 실행 시간을 단축하는 데 기여하고 있다(M. Andriluka, et al., 2014; M. Eichner and V. Ferrari, 2010; K. He, et al., 2015; S. Johnson and M. Everingham, 2015; L. Pishchulin, et al., 2016; S. Ren, et al., 2015; E. Insafutdinov, et al., 2016)

(2) 모델 아키텍처

각 단계별로 설명하면 [그림 2-29]과 같다(E. Insafutdinov, et al., 2016).



[그림 2-29] Deeper Cut Model Architecture(E. Insafutdinov, et al., 2016)

① Part Detectors

DeeperCut은 ResNet을 조정하지만 약간 변환한다. [그림 2-30]에서 먼저, 최종 분류 층뿐만 아니라 평균 풀링층이 제거된다. 다운 샘플링을 방지하기 위해 Conv5 블록의 첫 번째 컨볼루션 레이어의 보폭이 2 픽셀(PX)에서 1 픽셀(PX)로 줄었다. 다음으로 Conv5의 모든 3×3 컨볼루션에 구멍이 추가되고 $2 \times$ 업 샘플링을 위한 디컨볼루션 레이어가 추가되며, 신체손실 레이어는 Conv4 블록 안에 추가된다. 최종 출력은 Conv3 블록의 출력에 연결이

된다. ResNet의 심한 깊이는 큰 수용 필드 크기를 허용하므로 개별 신체 부위의 위치를 예측할 때 컨텍스트를 통합하는 데 도움이 된다. 중간 감독도 제공된다. 최상의 성능 설정은 중간 감독과 함께 ResNet-152(16 PX + 2 × 업 샘플링의 보폭)이다. 입력 치수는 $688 \times 688 \times 3$ 이다.

- uniqueness

$$\forall d \in D: \sum_{c \in C} x_{dc} \leq 1$$



- consistency

$$\forall dd' \in \binom{D}{2}: y_{dd'} \leq \sum_{c \in C} x_{dc}$$

$$\forall dd' \in \binom{D}{2}: y_{dd'} \leq \sum_{c \in C} x_{d'c}$$



- transitivity

$$\forall dd'd'' \in \binom{D}{3}: y_{dd'} + y_{d'd''} - 1 \leq y_{dd''}$$



[그림 2-30] Deeper Cut Model Architecture 수식(E. Insafutdinov, et al., 2016)

② Image-Conditioned Pairwise Terms

DeeperCut은 회귀 예측을 수행하며 이는 다중 인물 자세 추정에서 성능을 향상한다. 신체 탐지기는 신체 부위의 존재를 예측하고 모델은 다른 신체의 위치에 대한 근거로 현재 위치를 증거로 사용한다. 이를 위해 상대 위치를 예측하는 계층(Layer)이 추가된다. 이 레이어에서의 예측은 손실 함수를 계산하는 데 사용되며, 조정된 손실 함수 사용으로 여러 사람의 신체 부위 감지를 필터링하여 개인 간에 예측을 명확하게 해결할 수 있게 된다.

③ Incremental Optimization

첫째, Branch & Cut 알고리즘에 의해 상수 값을 계산하는 데 걸리는 시간은 신체 부위 후보의 수에 기하급수적으로 늘어날 수 있다. 이와 같은 이유로, 실제로 이 알고리즘으로 처리할 수 있는 후보의 수를 제한한다. 이 제한으로 인해 신체 부위와 많은 사람을 보여주는 이미지의 경우 전체에 포함되지 않을 수도 있기 때문에 놓친 경우가 있다.

둘째, 전체 이미지에 대한 최적화 문제를 해결한다는 것은 정확하게 감지된 신체 등급 간에는 구분이 없다는 것을 의미한다. 즉 머리와 어깨의 신체 등급이 덜 안정적으로 감지된다. 추가로, 손목과 팔꿈치 그리고 발목 등을 들 수 있다. 따라서, 신뢰할 수 없는 탐지로 인해 솔루션이 손상될 수 있다. 이런 문제를 해결하기 위해 DeepCut ILP는 가장 신뢰할 수 있는 신체 부위 클래스부터 시작하여 덜 신뢰할 수 있는 신체 부위 클래스까지 고려하여 최적화 접근법으로 문제를 해결한다.

이 실행 가능한 해결 방법은 신체 후보의 어떤 집합에 대해서, 그 집합의 개별적인 요소로 어떤 클래스에 속하는 각 객체를 인스턴스라고 한다. 즉 이 인스턴스에서 고려되는 신체 클래스를 선택, 레이블 지정 및 클러스터링을 한다. 다음 인스턴스의 경우 이전 인스턴스와 같은 클래스의 신체 부분 후보 클러스터가 신체 클래스 후보로 된다. 따라서 다음 인스턴스는 이전 인스턴스에 의해 결정되지 않은 신체 부위를 선택, 라벨링 및 클러스터링 하기 위한 최적화 문제이다. 전반적으로, 이것은 [표 2-5]에 의해 더 많은 신체 부분 후보로 일관되게 시작하여 자세 추정 결과를 크게 향상할 수 있게 한다.

[표 2-5] DeepCut Branch and Cut Algorithm(E. Insafutdinov, et al., 2016)

	Stage 1	Stage 2	Stage 3
2-Stage	Head, Shoulders, Elbows, Wrists	Hips, Knees, Ankles	
3-Stage	Head, Shoulders	Elbows, Wrists	Hips, Knees, Ankles

[표 2-6]에서 DeeperCut 자세 추정 결과는 DeepCut ILP의 여러 인스턴스를 해결하여 크게 개선되었다. 예를 들어 확실하게 감지된 신체 부위 등급 간에 명확한 구분이 이루어진다. 머리와 어깨 부위 등급이 덜 안정적으로 감지되고, 손목, 팔꿈치 및 발목은 Branch-and-Cut 알고리즘으로 더 많은 신체 부위를 처리할 수 있다.

[표 2-6] Deepcut 단계별 인식 관절 후보군 (E. Insafutdinov, et al., 2016)

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank
1-stage Optimize, 100 det, nms 1x	70.3	61.6	52.1	43.7	50.6	47.0	40.6
1-stage Optimize, 100 det, nms 2x	71.3	64.1	55.8	44.1	53.8	48.7	41.3
1-stage Optimize, 150 det, nms 2x	74.1	65.6	56.0	44.3	54.4	49.2	39.8
2-stage Optimize	75.9	66.8	58.8	46.1	54.1	48.7	42.4
3-stage Optimize	78.3	69.3	58.4	47.5	55.1	49.6	42.5
+split Detections	78.5	70.5	59.7	48.7	55.4	50.6	44.4
DeepCut	50.1	44.1	33.5	26.5	33.0	28.5	14.4

(3) 방법

따라서 3개의 손실 기능을 모두 선형으로 결합하여 신체 부위 검출기(시그모이드 활성화 및 교차 엔트로피 손실, 위치 회귀(L1 Loss) 및 손실 회귀(L1 Loss)의 공동 훈련을 수행한다.

(4) 결과

각 개인이 충분히 분리된 사람에 대해서만 평가가 수행하며, Single Person의 하위 집합의 전반적인 성능인 PCK 평가 측정하였다.

아래 [표 2-7]과 [표 2-8]은 결과값을 보여주고 있다. DeeperCut의 MPII DataSet에서는 머리, 어깨, 팔꿈치 등의 신체에 대한 PCK 점수가 88점으로 나타났다.

[표 2-7] DeepCut MPII DataSet(L. Pishchulin, et al., In *CVPR'16*)

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	PCK	AUC
DeepCut	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4	56.5
Deepcut	96.6	94.6	88.5	84.4	87.6	83.9	79.4	88.3	60.7
A. Newell et al., ECCV'16	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9	62.9

아래 [표 2-8]는 결과값이다. DeepCut의 LSP DataSet에서는 머리, 어깨, 팔꿈치 등의 신체에 대한 PCK 점수가 90.1점으로 나타났다.

[표 2-8] DeepCut LSP DataSet(L. Pishchulin, et al., In *CVPR'16*)

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	PCK	AUC
DeepCut	97.0	91.0	83.8	78.1	91.0	86.7	82.0	87.1	63.5
Deepcut	97.4	92.7	87.5	84.4	91.5	89.9	87.2	90.1	66.1
Bulat&Tzimir., ECCV'16	97.2	92.1	88.1	85.2	92.2	91.4	88.7	90.7	63.4

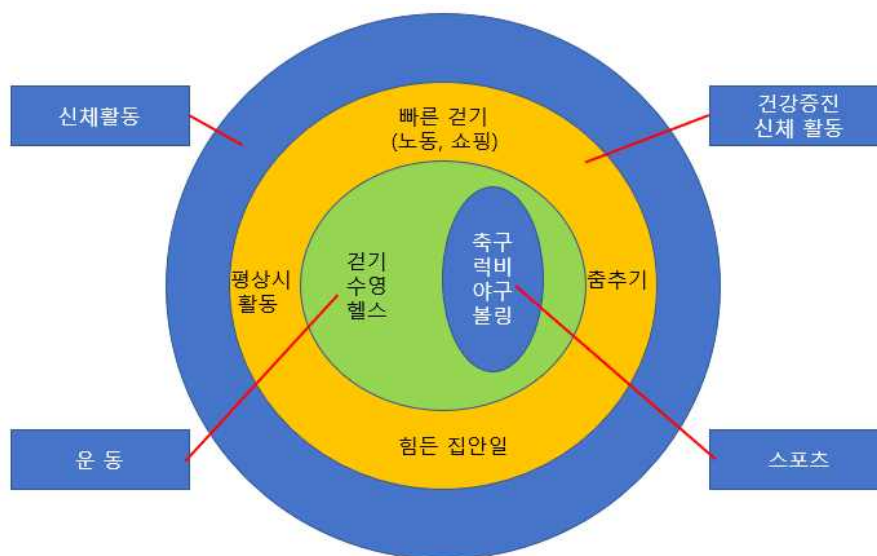
(5) 결론

DeepCut에서는 여러 사람 자세 추정으로 최첨단 기술을 크게 발전시켰다. 첫 번째, 최근 딥러닝의 발전을 바탕으로 표준 자세 추정 벤치마크에서 최첨단 성능을 얻을 수 있는 매우 깊고 강한 신체 부위 검출기를 제안했다. 두 번째, 신체 부위 사이에 새로운 이미지 조건 변수로 손실 함수를 도입하여 Multi Person 자세 추정의 경우에 성능을 크게 향상하고, 연결된 공간 모델에서 추론의 실행 시간을 크게 줄인다. 세 번째, 실행 시간을 더욱 줄이고 인간 자세 추정 정확도를 향상하기 위해 새로운 증분 최적화 전략을 도입했다. 전반적으로, 제안된 개선은 까다로운 멀티 자세일 때 추정 정확도를 거의 두 배로 늘리는 동시에 실행 시간을 3배 줄였다.

제 3 절 신체활동목록(MET) 연구

1) 신체활동에 대한 연구

신체활동(Physical Activity)은 “골격근 수축으로 에너지 소비를 일으키는 신체의 움직임”으로 [그림 2-31] 넓은 범위에서 운동스포츠를 포함하며, 이외에도 일상생활이나 업무 수행 중에 발생하는 모든 신체의 움직임을 포함하는 것이다(국립암센터, 2010).



[그림 2-31] 신체활동 관련 개념도(B. C. Corbin, 1996)

신체활동은 인간의 기초대사에서 일상생활 정도의 신체 움직임을 의미하며 에너지를 소모하는 모든 신체적 움직임을 말하는 것으로(S. N. Yun, et al., 2002). 1996년 미국은 신체활동에 대한 참여 중요성을 Surgeon General Report on Physical Activity를 통해 발표 이후 일상생활에 대한 신체활동 습관과 건강 간의 관계 규명이 활발히 이루어지기 시작했다.

그 결과를 토대로 미국 신체활동 지침(W. L. Haskell, et al., 2007)이 개발되었고, 이후 브라질, 스페인, 호주 등 여러 나라에서 국가별 신체활동 지침

을 발표하고 있다. 우리나라는 2010년 한국인 신체활동 지침을 제시하였으나 현재까지 한국인을 위한 신체활동과 건강에 대한 과학적인 근거는 미미한 수준이다(J. G. Park, 2010).

미국 신체활동 지침에는 신체활동을 크게 두 가지 분류로 하였다. 하나는 기본적인 활동(Baseline Activity)이고, 또 다른 하나는 건강증진 활동(Health-Enhancing Physical Activity)이다. 기본적인 활동으로는 서기, 가벼운 물건 들기, 천천히 걷기 등과 같은 가벼운 활동이며, 중강도 또는 고강도 신체활동도 드물게 수행한다. 반면, 건강증진 활동으로는 기본적인 활동뿐만 아니라 빠르게 걷기, 근력 운동, 줄넘기, 댄스, 등산 등과 같이 2008년 제시한 지침에는 중강도 이상의 신체활동을 의미한다. 보다 구체적으로 어린이, 성인, 노인을 대상으로 유산소, 근력, 근지구력과 관련된 신체활동 향상에 초점을 맞춘 신체활동 권장량은 [표 2-9]과 같다(U.S. Department of Health and Human Services, 2008).

[표 2-9] 연령별 신체활동별 권장량(이미영, 2011)

어린이	성인	노인
• 매일 60분(1시간) 이상의 중 강도 또는 고강도 신체 활동 권장 • 적어도 주3일 이상 고강도 시 신체 활동 • 근력과 골밀도를 증가시킬 수 있는 활동을 적어도 3일 이상 실시	• 적절한 건강 수준을 유지하기 위해 주당 150분(2시간 30분)의 중강도 활동이나 75분 이상(1시간 15분)의 고강도 활동 또는 중강도와 고강도를 혼합 • 유산소 관련 신체활동을 실시할 경우 한번에 적어도 10분 이상 유지 • 신체활동을 주중에 나누어서 실시하는 것이 보다 효과적	• 성인의 권장량을 따르나, 만약 질병이나 신체적 장애로 인해서 주당 150분의 중강도 신체활동에 참여할 수 없을 경우 체력과 신체적 건강상태에 따라 변형하여 실시 • 낙상을 예방하기 위해 평형성을 향성을 유지시킬 수 있는 활동을 증가

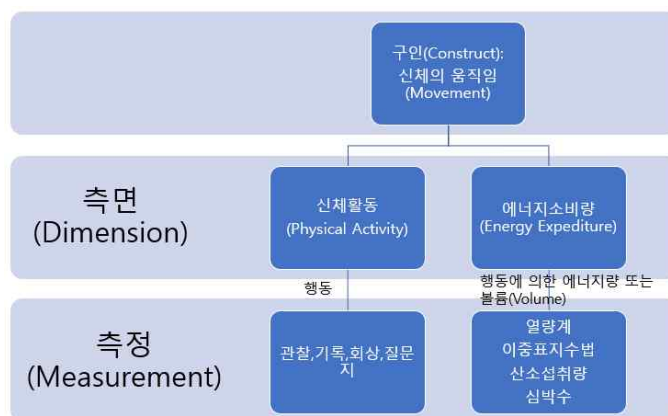
2) 신체활동의 측정 및 수준

신체활동 권장량을 개발하기 위해서는 먼저 신체 활동량을 정확하게 측정해야 한다. 이때 “신체활동”을 측정하기 위해서 [표 2-10]에 측정 단위(Measurement Units)가 있다.

[표 2-10] 신체활동과 에너지 소비량 평가 방법과 단위(이미영, 2011)

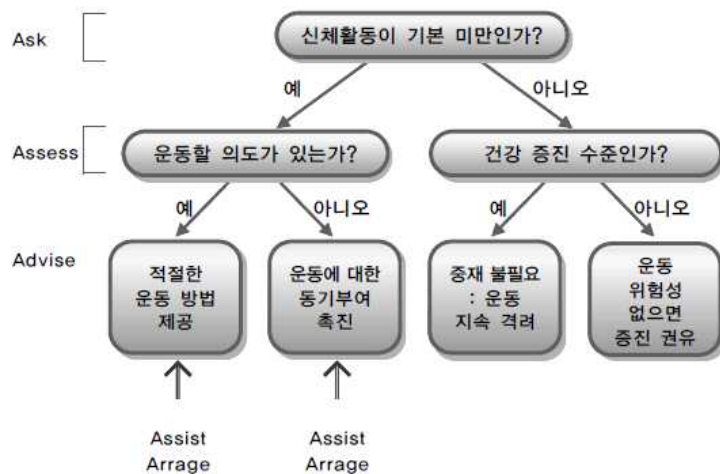
측정 항목	객관적 검사 도구					주관적 검사 도구		
	이중 표지물	간접 열량계	심박수 측정형	가속도 계형	보행 계수계	기록법	관찰법	질문지
신체활동량(Kcal) 계산 정확성	높음	높음	비교적 높음	비교적 높음	낮음	비교적 높음	비교적 높음	낮음
신체 활동 패턴이나 유형	불가능	불가능	불가능	불가능	건기	다양함	다양함	다양함
신체 활동 빈도나 기간	불가능	가능	가능	가능	기계에 따라 다름	가능	가능	가능
신체 활동 강도	가능	가능	가능	가능	불가능	가능	가능	가능
사용의 편리성	아주 낮음	아주 낮음	높음	비교적 높음	높음	낮음	낮음	아주 높음
비 용	아주 높음	아주 높음	높음	높음	낮음	높음	높음	아주 낮음

신체 활동과 에너지 소비량 평가 방법과 단위로 [그림 2-32]에서 LaMonte와 Ainsworth (2001)에 의하면 신체활동(Physical Activity)을 골격근 수축에 의한 신체 움직임을 만들어 내는 행동 과정(Behavioral Process)이라 정의하고, 에너지 소비량(Energy Expenditure)을 골격근 수축을 위해 요구되는 에너지 소비 값(Values)으로 신체활동에 따른 직접적인 결과(Results)를 의미한다.



[그림 2-32] 신체활동과 에너지 소비량 평가 개념(M. J. Lamonte and B. E. Ainsworth, 2001)

모든 성인에게 신체활동 정도로 최소한 유산소 신체활동이 ‘기본 신체활동’으로 근력 운동을 주 2일 이상 시행해야 한다. 이 활동이 1주일 동안 중강도 신체활동이 150분 이상이라면 ‘기본 신체활동’ 수준이며, 1주일 동안 중강도 신체활동이 300분 이상이라면 ‘건강증진 신체활동’ 수준이라 한다. 그러므로 근력 운동을 1주일 동안에 2일 이상을 시행한다면 ‘만족’ 수준이다. 그리고, 유산소 신체활동을 ‘기본 신체활동’보다 낮게 하고, 근력 운동을 1주일에 1일 이하면 ‘기본 미만’으로 판단하고 있으며 신체활동 증진을 권유하고 있다.



[그림 2-33] 신체활동 수준 분류(양윤준, 2016)

신체활동 수준은 [그림 2-33]에서 우리나라의 신체활동 지침을 기준으로 삼아서, 지침에 만족하면 ‘기본 신체활동’ 수준이라 분류한다. 유산소 신체활동이 지침보다 2배이면 ‘건강증진 신체활동’ 수준으로 분류한다(양윤준, 2016). 또한, 신체활동에 있어 전통적으로 형태(Mode; 걷기, 달리기, 물건 등 옮기기), 빈도(Frequency; 해당 활동을 통한 횟수), 기간(Duration; 활동한 시간), 강도(일의 최대 능력에 대한 상대적 또는 절대적 에너지의 소비량)로 요약된다. 이때 개개인의 에너지 소비량은 빈도, 기간, 절대 강도(Absolute Intensity; MET)의 곱으로 나타낸다. 즉, 총 에너지 소비량은 각각의 신체활동에서 소비된 에너지량을 더한 값이 된다(T. M. Wood & W. Zhu, 2006).

3) 신체활동목록에 대한 연구

MET 값은 미국 스탠퍼드 대학에서 함께 제시한 신체활동을 기반으로 표준화된 목록이 1993년 Ainsworth, W. L. Haskell, Leon, Jacobs, Montoye 등에 의해서 개발되어 2003년도 Ainsworth, W. L. Haskell, Whitt, Irwin, Swartz 등에 의해 개정되었다. 2011년 개정된 신체활동 목록(Compendium of Physical Activity)들은 21개 카테고리로 활동 유형별 605개로 구성되어 있으며, 활동 내용에 대한 MET(신진대사 해당치, Metabolic Equivalent of Task) 점수가 0.9 MET부터 18 MET까지 다양하게 제시되고 있다. 또한, MET는 신체활동 에너지 비용을 평가하는 척도이며, 체중과는 별도로 MET를 “정규화된 에너지”라고도 한다. 그리고, 측정법은 캘리포니아의 스탠퍼드 대학에서 도입했으며 전 세계적으로 30년 이상 사용되었다.

[그림 2-34]에는 참조를 포함하여 추가 정보가 있는 각 범주에 대한 문서를 보여준다¹⁴⁾.

Complete Reference List						
	2011CompendiumReferences.pdf View Download	2011 Compendium References List	267k	v. 2	Feb 23, 2011, 5:24 AM	Stephen Herrmann
References by Category						
	01-Bicycling-2011CompendiumPA.pdf View Download	Bicycling References	32k	v. 2	Apr 11, 2011, 7:33 AM	Stephen Herrmann
	02-ConditioningExercise-2011CompendiumPA.pdf View Download	Conditioning Exercise References	220k	v. 2	Apr 12, 2011, 4:13 AM	Stephen Herrmann
	03-Dancing-2011CompendiumPA.pdf View Download	Dancing References	164k	v. 2	Apr 11, 2011, 7:33 AM	Stephen Herrmann
	04-Fishingandhunting-2011CompendiumPA.pdf View Download	Hunting & Fishing References	77k	v. 2	Apr 11, 2011, 7:33 AM	Stephen Herrmann
	05-HomeActivities-2011Compendium.pdf View Download	Home Activity References	302k	v. 2	Apr 11, 2011, 7:38 AM	Stephen Herrmann

[그림 2-34] 신체활동 목록

[그림 2-35]에서 5자리 코드의 처음 두 자리는 일반 범주를 나타내며, 파

14) <https://sites.google.com/site/compendiumofphysicalactivities/Activity-Categories/home-activity>

란색으로 표시된 것은 2011년 출판된 MET 값을 가진 활동이며, 2011년 붉은색으로 표시되는 범주는 MET 값을 추정한다.

1993 Compendium		2000 Compendium		2011 Compendium		Home Activities
Codes	METs	Codes	METs	Codes	METs	Description
05010	2.5	05010	3.3	05010	3.3	cleaning, sweeping carpet or floors, general
				05011	2.3	cleaning, sweeping, slow, light effort
				05012	3.8	cleaning, sweeping, slow, moderate effort
05020	4.5	05020	3.0	05020	3.5	cleaning, heavy or major (e.g. wash car, wash windows, clean garage), moderate effort

[그림 2-35] 신체활동 목록 분류

[표 2-11]에서 MET는 일반인들에게 신체활동의 중요성을 인식시키면서 신체활동의 내용과 범위를 정해주는 데 도움이 된다.

[표 2-11] 신체활동 내용과 범위(이미영, 2011)

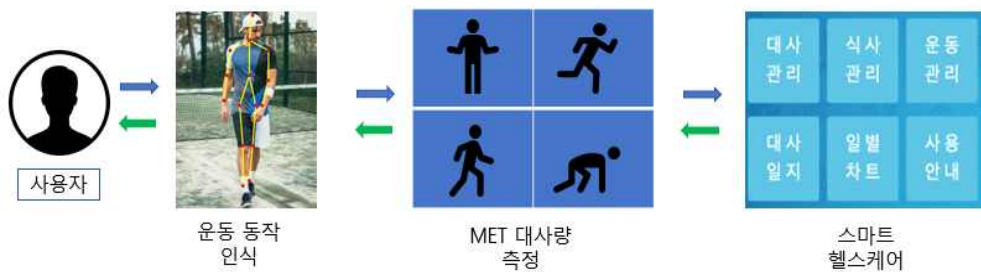
활동	저강도 < 3.0 MET	중강도 3.0~5.9 MET	고강도 > 6.0 MET
걷기	집, 사무실, 건물 주위를 천천히 걷기=2.0MET	보통 걷기(3.0mph)=3.3MET 빠르게 걷기(4mph)=5.0MET	가벼운 등산=7.0MET 조깅/달리기 8.0kph=8.0MET 달리기 11.0kph=11.5MET
집안일 또는 사무실 작업	컴퓨터나 책상에 앉아서 작업=1.5MET 가벼운 집안일(침대정리, 설거지, 다 린질, 요리 등)=2.0~2.5MET	청소;창문 닦기,차 청소=3.0MET 바닥 닦기,청소기 돌리기, 밀대 걸레질 하기=3.0~3.5MET 목공작업=3.6MET 물건나르기(종비박스)=5.5MET	논 밟을 때 삽질하기=7.0MET 무거운 벽돌 옮기기=7.5MET 농작물 옮기기= 8.5MET 삽질 등 땅 파기= 8.5MET
레저 또는 스포츠 활동	미술 만들기=1.5MET 당구=2.5MET 앉아서 하는 낚시 2.5MET 악기 연주=2.0~2.5MET	레크레이션 배드민턴=4.5MET 가벼운 농구 연습=5.5MET 경사없는 곳 자전거 타기=6.0MET 골프=4.3MET 레크레이션 수영=6.0MET 탁구=4.0MET	농구 게임=8.0MET 빠르게 자전거 타기=8.0MET 축구 게임=10.0MET 수영 경기=10.0MET 싱글 테니스=8~11MET 배구 게임=8.0MET

제 3 장 스마트 헬스케어 모델 설계

제 1 절 자세 추정 모델 설계

1) 전체 구성도

본 연구에서는 [그림 3-1]과 같이 ① LSP DataSet에서 사용되는 데이터를 기반으로 딥러닝 중 선행 연구의 CNN 5개 모델을 이용하여 Pose Estimation 비교를 하고 선정된 모델을 통한 운동 동작 및 운동량 알고리즘을 개발하고, ② 신체활동 목록(MET) 상의 대사량과 연계하여 소모된 칼로리를 계산하여, ③ 최종 운동량 결과값을 스마트 헬스케어의 운동 관리와의 연계를 구현하고자 한다.



[그림 3-1] 자세 추정에 따른 동작 반영 스마트 헬스케어 모형

2) 자세 추정 모델

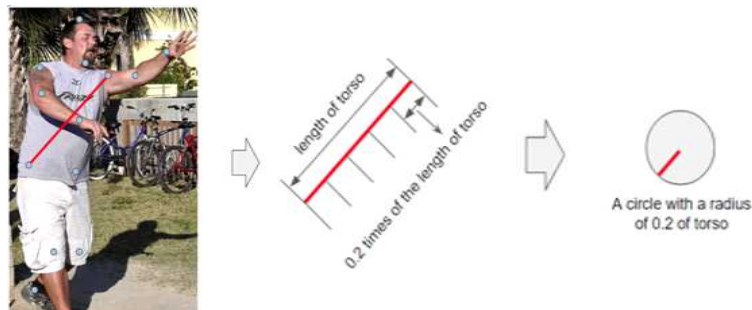
자세 추정 선행 연구의 5개 모델을 선정하여 학습 비교를 하고자 한다. [표 3-1]은 자세 추정 학습 비교를 위해 선정된 모델이다.

[표 3-1] 자세 추정 모델 목록(하태용 외, 2019)

제 목	출 처
Convolutional Pose Machines	S. E. Wei, V. Ramakrishna, and T. Kanade. (2016).
Stacked Hourglass Networks for Human Pose Estimation	A. Newell, K. Yang, and J. Deng. (2016).
DeeperCut: A Deeper, Stronger, and Faster Multi -Person Pose Estimation Model	E. Insafutdinov, L. Pishchulin, and B. Andres. (2016).
PoseNet : ResNet	https://medium.com/tensorflow/real-time-human-pose-estimation
PoseNet : MobileNet	

3) 자세 추정 방법

자세 추정 학습 방법으로 [그림 3-2]에서 PCK를 사용하고자 한다. 요약하면, LSP DataSet에서 사용하는 Evaluation Metric을 이용하여 오른쪽 어깨부터 왼쪽 골반까지 거리를 Length Of Torso라 하고, 기준 거리에 0.2를 곱한 것을 PCK 0.2라 표현한다. 다음으로, 이를 측정 기준으로 사용하여 결과 값이 해당 원의 반지름에 들어오는 확률을 의미한다. 그러므로, PCK 0.2는 고정된 값이 아니라 이미지별로 나타나는 사람의 크기에 따라 비례하는 유동적인 값이다. 그리하여, 본 연구에서는 Pose Estimation Evaluation 결과로 LSP DataSet을 대상으로 PCK 0.2 기준 80% 이상을 제시하고자 한다.



[그림 3-2]. PCK 0.2(하태용 외, 2019)

제 2 절 동작 인식 모델 설계

1) PoseNet 동작 인식 모델 범위

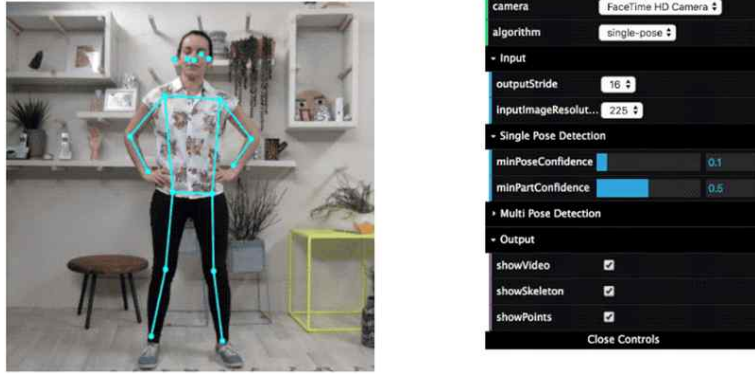
PoseNet 동작 인식을 위해서 몸에 디바이스를 착용하지 않는 전제로 구성하며, 현재 누구나 가지고 있는 디바이스를 사용하고자 한다. [표 3-2]는 본 연구에 사용한 디바이스 사양이다.

[표 3-2] 디바이스(스마트폰) 시스템 사양

OS	Android 8.0 or higher, 64bit
CPU	ARMv8 instruction Set Based Architecture Combination ARM Cortex-A57 + ARM
RAM	4GB or more
GPU	Adreno 530, Mali-G71 MP20, Mali-G72 or higher (Constitutional)
CAMERA	HAL 3.2 or higher

2) PoseNet 동작 인식 모델 설정

PoseNet 모델을 사용하여 사람의 자세 추정을 위한 TensorFlow Lite 애플리케이션을 사용하였다. PoseNet은 주요 신체 부위의 위치를 감지하여 이미지나 동영상으로 부터 사람의 포즈를 추정하는 비전 모델이다. 이 모델은 이미지에서 사람의 팔꿈치 및 팔목 또는 무릎 위치 등을 추정할 수 있으며, 자세 추정 모델은 이미지에 나와 있는 사람이 누군지 식별하는 것이 아니라, 주요 신체 부위의 위치만을 식별하므로 PoseNet 모델로는 각 관절의 좌표 값을 알아낼 수 있다. 하지만, 직접 운동 인식 알고리즘을 구현하기에는 수많은 체형에 따른 차이를 고려하는 것에 한계가 있을 수 있다. 이에 스마트 헬스케어 동작 인식 모델 학습을 위한 설정은 [그림 3-3]과 같다.



[그림 3-3] PoseNet can Detect Human Figures

TensorFlow Lite는 디바이스의 카메라를 활용하여 사람의 주요 신체 부위를 실시간으로 감지하고 표시하는 Android 애플리케이션을 제공한다. 먼저, TensorFlow.js 및 PoseNet 라이브러리 가져오기를 하다. 그리고, PoseNet 프로젝트 설정 방법의 기본 사항으로 설치에 따른 라이브러리는 `npm install @tensorflow-models/posenet` 명령어를 사용하여 설치할 수 있다.

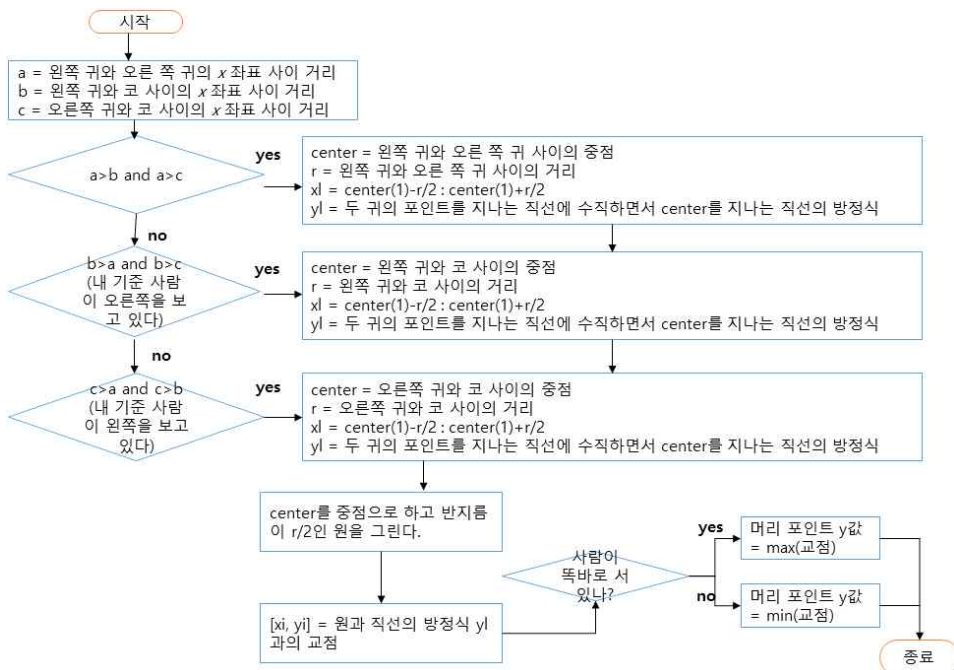
그리고 나서, Single-Person Pose Estimation 단일 자세 추정 알고리즘의 입력을 검토한다. 요약하면 첫 번째로, 입력 이미지 요소로는 자세를 예측할 이미지 또는 비디오 요소는 정사각형이어야 한다. 두 번째는, 이미지 스케일 팩터로는 0.2와 1.0 사이의 숫자이며, 기본값은 0.50으로 되어 있다. 아울러, 네트워크를 통해 이미지를 공급하기 전에 이미지 크기 조정값을 낮게 설정하면 이미지가 축소되어 정확도에 떨어진다. 세 번째로, 화면 표현되는 자세를 올바른 방향으로 자세를 반환하려는 경우 True로 설정하면 된다. 네 번째 출력 보폭으로는 32, 16 또는 8이 있으며, 기본값은 16이다.

3) PoseNet 동작 인식 모델 방법

가) 눈, 코, 귀 포인트로 머리 끝 포인트 구한 방법

PoseNet의 단일 자세 추정 알고리즘의 출력값을 검토해 보면, 자세 신뢰 점수와 17개의 KeyPoint 배열을 모두 포함하는 자세에서 각 KeyPoint마다

Keypoint 위치와 Keypoint 신뢰 점수가 포함된다. 또한, 모든 Keypoint 위치에는 입력 이미지 공간에 맞는 x 및 y 좌표를 가지고 있으며 해당 이미지에 직접 대응할 수 있다. [그림 3-4]와 같이 알고리즘을 사용하여 눈, 코, 귀 포인트를 머리끝 포인트로 변환하는 작업하여 17개 포인트를 14개 포인트로 변환하여 자세 추정 학습을 비교하고자 한다. 왜냐하면, 기존 선행 연구 모델들이 14개의 Keypoint 위치와 그에 따른 신뢰 점수를 바탕으로 연구되어 있기 때문에, 적절한 비교 분석을 위해서는 동일한 Keypoint 인식 환경으로 맞추어 필요하기 때문이다.

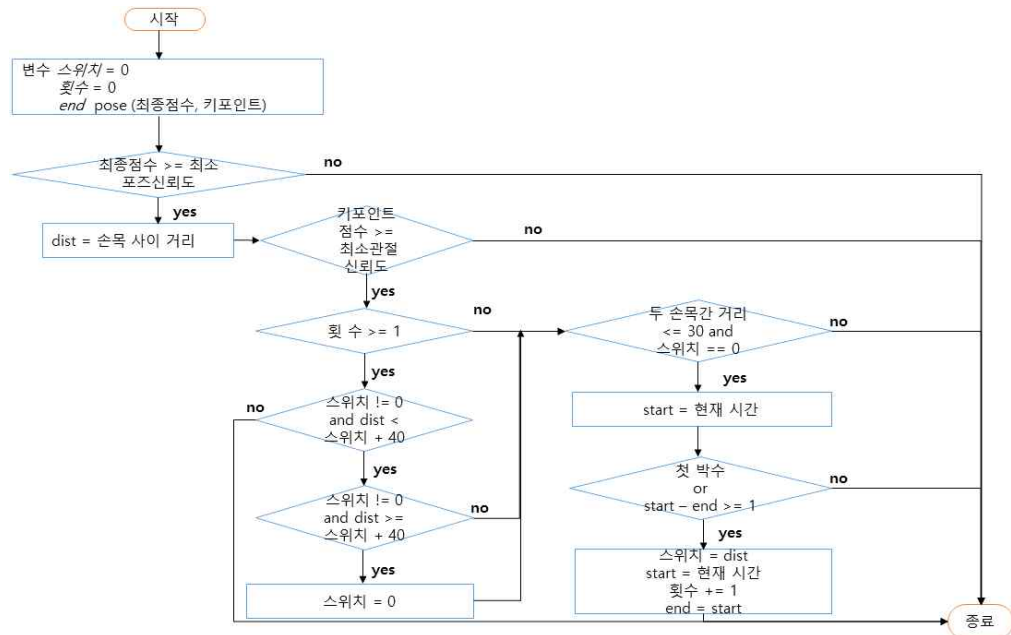


[그림 3-4] 눈, 코, 귀 Keypoint를 머리 끝 Keypoint로 변환(하태용 외, 2020)

나) 손뺑 치기 횟수 방법

PoseNet을 활용하여 사람의 운동 동작 중 가장 기본적인 ‘손뺑(박수) 치기’ 동작을 인식해 보았다. 사람이 카메라 앞에서 손뺑(박수)을 치면 PoseNet을 통해 얻은 관절들의 위치 정보들을 분석하여 손뺑 친 횟수를 파악하고 이

를 화면에 표시해 주는 구조로 아래의 [그림 3-5]와 같은 알고리즘으로 구성된다.



[그림 3-5] 손뺍(박수) 횟수 카운트 순서도(하태용 외, 2020)

제 3 절 신체활동 목록과 MET 설계

1) 운동량 측정 범위

신체활동과 에너지 소비량을 측정하는 기준 단위로는 다양하게 나타낼 수 있다. 그 중 MET가 측정 단위들 중에서 산소 섭취량과 신체 활동 강도를 잘 반영한 효과적인 단위이다. 이를 통해, MET는 휴식상태에서 몸이 소비하는 산소량 또는 에너지 또는 운동의 상대적인 세기를 나타내는 대사량의 단위이며, 산소 200ml는 약 1Kcal의 열량을 낸다. 이를, MET 공식으로 표현하면 $O_2 200ml = 1Kcal = 1,000cal$ 같이 된다. 이것을 산소 1ml 당 칼로리로 계산하면 5 cal/ml 이다. 그러므로, 1MET = 3.5ml/min/kg로 3.5ml가 사용되는 산소량을 뜻한다. 그리고 에너지 소비량은 Kcal.min 혹은 Kcal.hour로 정리

할 수 있다. $1MET = 3.5ml/min/kg$ 에 O_2 칼로리 $5cal/ml$ 로 대입하면 $17.5cal/min/kg$ 가 된다. 이 값을 이용하여 분당 O_2 소모량을 시간당 소모량으로 변환하면 $17.5 \times 60cal/hr/kg = 1,050cal/hr/kg$ 가 되며 대략 $1Kcal/hr/kg$ 가 된다. 따라서 운동량으로 1시간 동안 1 MET의 운동을 하면 $1,050cal/hr/kg$ 가 되며 대략 $1Kcal/hr/kg$ 소모함을 예상할 수 있다.

[표 3-3]에서 2011 Compendium 카테고리별로 현재 신체활동 목록에 포함된 카테고리 01 - 21까지 있으며, 5자리 코드로 605개 신체활동 목록과 MET를 생성한다.

[표 3-3] MET 기준정보 데이터베이스 저장 내용(하태용 외, 2020)

Category	Code	MET	Description
05- Home Activity	05011	2.3	Cleaning, Sweeping, Slow, Light Effort
13- Self Care	13040	2.0	Grooming, Washing Hands, Shaving, Brushing Teeth,
17- Walking	17082	5.3	hiking or walking at a normal pace through fields and hillsides
11- Occupation	11792	3.5	Walking on Job, 3.0 mph, in Office,

2) 운동량 측정 방법

PoseNet의 손뼉(박수) 치기 동작을 [표 3-4] 같은 형식으로 데이터베이스에 저장되도록 한다. 신체활동 목록에 손뼉(박수) 치기 MET 코드가 없으나, 유사한 운동 분류로 13- Self Care 집안 활동의 설거지, 칫솔질 등 2.0 MET를 적용하고자 한다.

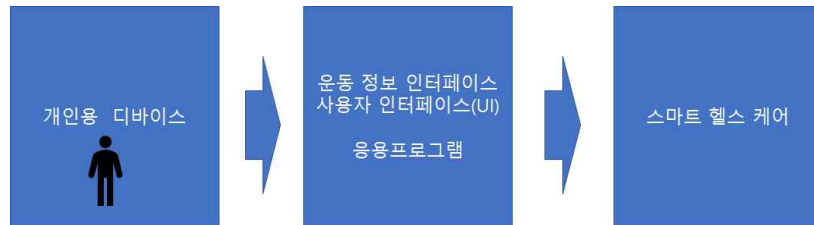
[표 3-4] 운동 동작 데이터베이스 저장 내용(하태용 외, 2020)

사용자ID	운동 날짜	운동 시작	운동 종료	MET Code	운동량 (횟수)	운동시간 (분)
1 User	2020.01.01	14:00:00	14:10:00	3040	10	10min

제 4 절 스마트 헬스케어 시스템 설계

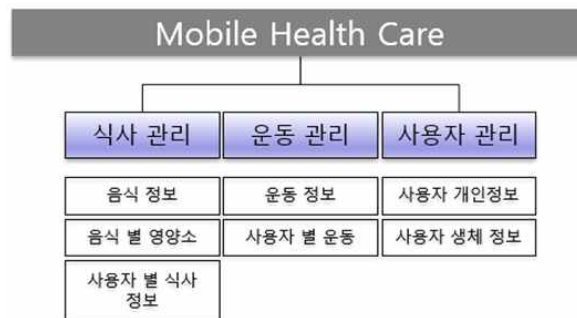
1) 애플리케이션 구성도

본 연구에서는 운동 정보를 스마트 헬스케어 시스템에 연계하는 방식으로 스마트폰 디바이스로부터 추정한 운동 동작을 통해 운동 자세와 횟수 정보를 파악하고, [그림 3-6]처럼 해당 운동 정보의 MET 값을 분석하여 운동에 따른 소모 칼로리를 계산한다. 나아가, 운동에 맞게 칼로리를 소비했는지를 스마트 헬스케어 대시보드에 표현한다.



[그림 3-6] 스마트 헬스케어 MET 연계 시스템 환경

아울러, [그림 3-7]과 [표 3-5]는 각각 스마트 헬스케어 시스템 구성도와 메뉴 구성도이다.



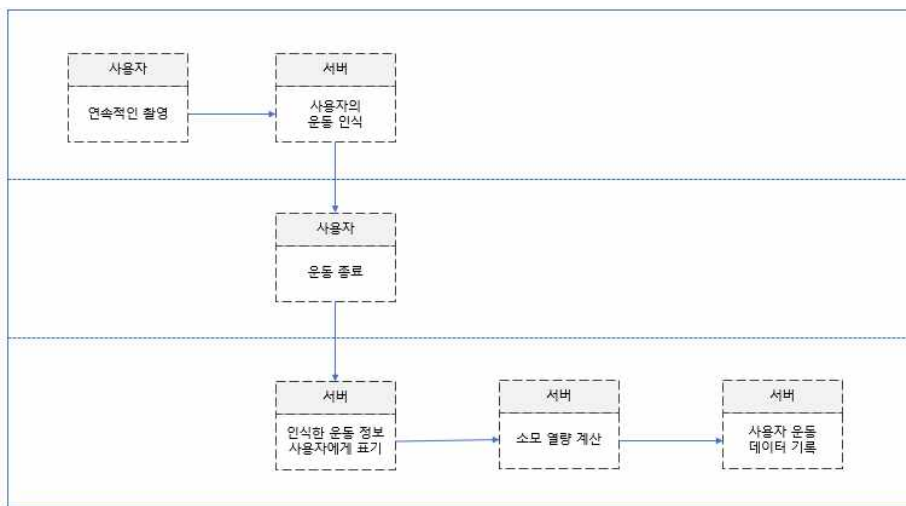
[그림 3-7] 시스템 구성도

[표 3-5] 메뉴 구성도

화 면	설 명
초기 화면	로그인 시 나오는 첫 화면
대사 관리	사용자의 건강 정보를 분석하여 비정상 여부를 판단
식사 관리	일별 식사 기록을 입력할 수 있는 화면
운동 관리	일별 운동 기록을 입력할 수 있는 화면

2) 시스템 흐름도

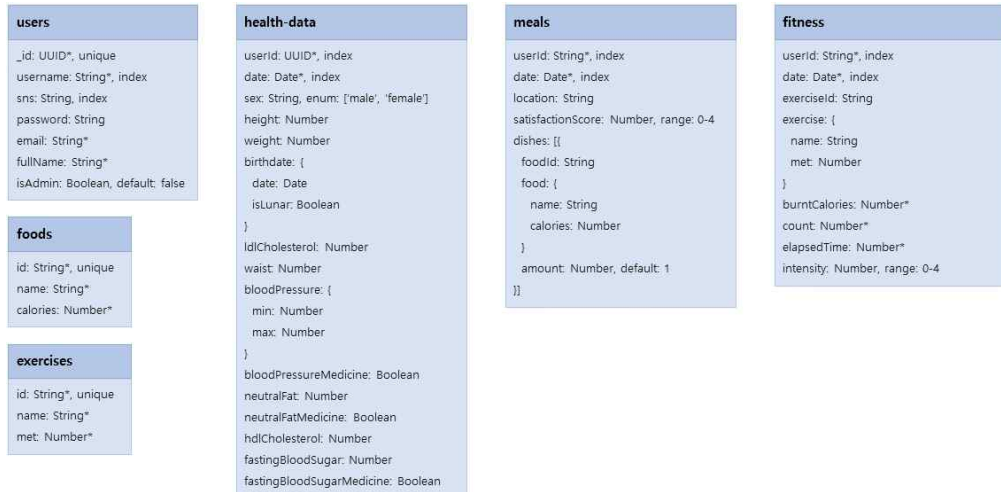
스마트 헬스케어 시스템 흐름도는 [그림 3-8]과 같이 6개의 프로세스를 가지고 있다.



[그림 3-8] 시스템 흐름도(하태용 외, 2020)

3) 데이터베이스 구조

스마트 헬스케어 데이터베이스 구조는 [그림 3-9]와 같다.



[그림 3-9] 데이터베이스 구조

4) 스마트 헬스케어 개발 환경

효율적이고 실질적인 개발 환경 구현을 위해 [표 3-6]과 같이 일반 사양 PC서버를 구성하여 소프트웨어를 개발하였으며, 시제품 및 테스트 단계에서 Google Cloud Platform을 활용하여 서비스 운용 테스트 및 사용화를 진행하였다.

[표 3-6] PC서버 개발 사양

구 분	사 양
하드웨어	Intel i5-6500 CPU
소프트웨어	Node.js v12.16.0

웹 애플리케이션 개발 측면에서는 앞으로 여러 기능이 추가될 것에 대비하여 소스 코드의 품질을 높이고 더욱 변화에 유연한 구조를 확보하는 것에 초점을 두었다. 음식 인식과 운동 인식 서비스가 성공적으로 이식될 수 있어야 하며, 추후 관리도 용이한 환경을 제공해야 하기 때문이다.

가) 웹 서비스 구조

Frontend 서버와 Backend 서버, NoSQL DB로 구성된 구조를 설계했다. Frontend 서버에서 Backend 서버로 HTTP 요청을, Backend 서버에서는 DB와의 상호작용을 진행한다. 이러한 구조는 분리를 통해 더 체계적인 개발을 진행할 수 있게 하고, 서비스의 확장성도 보장할 수 있도록 한다.

나) Frontend 서버

Frontend 서버는 사용자와 직접적으로 상호 작용하는 역할을 수행한다. 구현에는 Vue.js 프레임워크를 활용하여 컴포넌트 기반 개발 방법을 지향하고, 웹 UI 디자인은 Bootstrap Vue 모듈을 통해 반응형 웹 페이지를 구축한다. Backend 서버의 Web API를 통해 여러 데이터에 접근하며, 이때의 인증 절차는 API로부터 JWT를 발급받아 클라이언트에 저장해 두는 토큰 기반 인증 시스템으로 구현한다.

다) Backend 서버

Backend 서버는 DB와 통신하여 비즈니스 로직을 처리하는 API를 제공한다. 구현에는 Node.js Express를 활용하며, 컨트롤러-서비스-모델의 3계층 설계를 기반으로 한다. 그리고, Web API 라우터에서 컨트롤러를 호출하며 컨트롤러에서는 클라이언트 인증 검사와 인자 유효성을 확인한 뒤 서비스를 호출한다. 서비스는 비즈니스 로직으로 구성되어 모델과 상호작용하고, 결과적으로 DB와의 통신이 이루어진다. 이러한 구조를 통해 소스 코드의 재사용성과 유지 보수성을 확보한다.

라) DB 서버

DB 서버는 NoSQL인 MongoDB로 구성한다. MongoDB는 RDB에 비해 성능이 100배 이상 빠르고, 유연한 스키마를 가져 새로운 필드를 쉽게 추가

할 수 있다. 데이터가 많아질 경우 수평적 확장 또한 가능하다.

5) 스마트 헬스케어 시나리오

스마트 헬스케어 시스템에 대한 단계별 시나리오는 다음과 같다.

1 단계 : PoseNet의 손뼉(박수)치기 동작을 데이터베이스에 저장하기

사용자가 운동을 시작할 때 스마트 헬스케어 애플리케이션 중 운동 관리를 사용하여 손뼉(박수) 치기 운동 화면에서 손뼉(박수) 치기 버튼을 클릭하면 MET 데이터베이스를 조회해서 손뼉(박수) 치기에 대한 MET Code를 획득하고 사용자가 박수 치기를 마치면 운동 종료 시간으로 운동 시간을 계산하고 [사용자 ID, 운동 날짜, 운동 시작 시간, 운동 종료 시간, MET Code, 운동량, 운동 시간]에 대한 정보를 데이터베이스에 저장한다.

2 단계 : 605개 MET 목록 테이블 생성

먼저 신체 활동 목록의 MET 대사량 테이블을 생성한다. MET 한글 파일을 엑셀 파일로 변환하여 Python Pandas 라이브러리를 사용해 엑셀 파일을 읽어서 PyOdbc 라이브러리를 사용해서 데이터베이스에 저장한다.

3 단계 : 손뼉(박수) 치기 동작 MET 칼로리 계산

사용자가 운동 정보를 입력할 때 운동 정보 입력 페이지의 운동 정보에서 운동 종목, 운동 시간을 입력 후 확인을 누르면 웹 서버에 칼로리 계산을 요청하고 서버는 칼로리 계산 후 계산된 칼로리를 전달해 화면에 표시한다. 그 후 저장 버튼을 누르면 사용자 데이터베이스에 저장한다.

4 단계 : MET 칼로리 계산 및 스마트 헬스케어 운동관리 Insert

사용자가 운동 정보를 입력하거나 측정할 때 스마트 헬스케어의 운동관리로 연계하여 지금 운동하기 버튼을 누르면 어떤 운동을 할 것인지 선택하는 창을 띄우고 운동 정보 입력을 누르면 MET 칼로리 계산 화면으로 이동한다.

예를 들어, 60Kg인 사용자가 시간당 8METs로 운동한다고 가정한다면 $1,050\text{cal/hr/kg} \times 8(\text{MET}) \times 60\text{Kg} = 504\text{Kcal/hr/kg}$ 로, 시간당 504Kcal를 소모하게 된다. 이처럼, 신체활동 목록을 사용하여 에너지 소비량을 계산하는 방법으로 스마트 헬스케어 시스템의 데이터베이스에 저장하고자 한다.

제 4 장 스마트 헬스케어 시스템 구현 및 고찰

제 1 절 스마트 헬스케어 시스템 구현

1) 자세 추정 학습 결과

가) LSP DataSet

LSP DataSet¹⁵⁾은 운동하는 사람의 이미지 2,000개로 이루어져 있다. LSP Extended Training DataSet은 이보다 좀 더 많은 10,000개의 이미지로 이루어져 있다. 이미지는 사람의 크기가 대략 150픽셀 정도 되도록 스케일링 된 상태이다. 양 발목, 무릎, 엉덩이, 손목, 팔꿈치, 어깨, 그리고 목과 머리끝까지 총 14개의 관절이 라벨링 된 것이 특징이다. LSP Extended Training DataSet 전체와 LSP DataSet의 처음 1,000개의 이미지는 훈련 데이터 셋으로 쓰였고, LSP DataSet의 나머지 1,000개의 이미지가 모델 성능 평가에 사용되었다.

CPM과 DeeperCut은 공식적으로 Caffe 라이브러리로 구현되었고, 모든 코드를 실행시키기 위해서 GPU와 MatLab이 동시에 사용 가능해야 한다. 하지만 본 연구에서는 완전한 개발 환경을 갖출 수 없어 공식 소스 코드는 이용하지 못했고, [그림 4-1]에서 CPM은 동일한 모델을 PyTorch 라이브러리로 대체하여 구현한 소스코드¹⁶⁾가 있어 이를 이용해 직접 학습을 시켜볼 수 있었다. 하지만, 코드에서 별도의 Ground Truth 값을 생성하여 사용하면 기존 LSP DataSet의 Annotation을 Ground Truth 값으로 사용하는 다른 모델과 동일한 방법으로 PCK@0.2 결과를 낼 수가 없었다¹⁷⁾.

15) <http://sam.johnson.io/research/lsp.html>

16) <https://github.com/frankchen121212/MobilePose-master>

17) <https://github.com/farrajota/human-pose-benchmark>

```
python3 train.py --pos=0 --lr=5 --model=cpe --log_freq=1000 --debug --dataset=LSP --weight_decay=0e-5 --step_size=50000 --gamma=0.333 --max_epochs=30 --batch_size=
***
Checkpoint prefix will be 2019-08-28-04-45...
/usr/local/lib/python3.6/dist-packages/torch/nn/reduction.py:46: UserWarning: size_average and reduce args will be deprecated, please use reduction='s
warnings.warn(warning.format(re))
[2019-08-28 04:45:53.339][line: 38] ==> creating /content/drive/My Drive/MobilePose-master/output/com/LSP/2019-08-28-04-45.log
[2019-08-28 04:45:53.339][line: 194] ==>
[epoch 24]
0% 2/368T [00:54:36:12.21, 35.56s/it, acc=0.0008, acc_avg=20.0008, loss=0.068, loss_avg=0.062, time_avg=1753ms]
```

[그림 4-1] Convolutional Pose Machine 모델 학습 과정

결과로, S. E. Wei, et al.의 CPM과 Insafudinov et al.의 DeeperCut에 대한 PCK@0.2의 결과는 기존 연구에서 발표된 공식적인 수치를 이용하였다. 아울러, 예측 결과에 대한 수치로 CPM과 DeeperCut 모델이 LSP DataSet와 LSP Extended DataSet뿐만 아니라, MPII Human Pose DataSet에 대한 결과값도 추가로 테스트 학습을 진행한 상태에서 얻은 수치임을 밝힌다.

```
python3 train.py --pos=0 --lr=5 --model=cpe --log_freq=1000 --debug --dataset=LSP --weight_decay=0e-5 --step_size=50000 --gamma=0.333 --max_epochs=30 --batch_size=
***
Checkpoint prefix will be 2019-08-28-04-45...
/usr/local/lib/python3.6/dist-packages/torch/nn/reduction.py:46: UserWarning: size_average and reduce args will be deprecated, please use reduction='s
warnings.warn(warning.format(re))
[2019-08-28 04:45:53.339][line: 38] ==> creating /content/drive/My Drive/MobilePose-master/output/com/LSP/2019-08-28-04-45.log
[2019-08-28 04:45:53.339][line: 194] ==>
[epoch 24]
0% 2/368T [00:54:36:12.21, 35.56s/it, acc=0.0008, acc_avg=20.0008, loss=0.068, loss_avg=0.062, time_avg=1753ms]
```

[그림 4-2] Stacked Hourglass Networks 모델 학습 과정

A. Newell, et al.의 Stacked Hourglass Networks에 대한 PCK@0.2 결과는 GitHub에 공개한 모델 학습 소스 코드¹⁸⁾를 PyTorch로 재구현한 코드¹⁹⁾를 사용하여 직접 학습시켜 얻은 것이다. 학습은 12GB NVIDIA Tesla K80의 GPU를 무료로 제공하는 Google Colaboratory 환경에서 이루어졌고, 학습

18) <https://github.com/princeton-vl/pose-hg-train>

19) <https://github.com/bearpaw/pytorch-pose>

과정으로 입력 이미지는 모두 256×256 픽셀로 조정하고, 회전, 스케일링 등의 Data Augmentation 기법을 활용하였다. Learning Rate는 $2.5e^{-4}$ 에서 시작하고, RMSprop과 Batch Normalization 기법을 사용하여 100 Epoch 정도 학습하였다. LSP DataSet와 LSP Extended DataSet만을 활용하여 학습하였다. [그림 4-2]에서 모델 학습 과정을 나타낸다. 다음으로, [표 4-1]은 공개된 부분을 정리한 모델별 학습 환경 비교한 것이다.

[표 4-1] 모델 학습 환경 비교

구 분	CPM	Stacked Hourglass	DeeperCut	PoseNet	
				MobileNet	ResNet
Input	368×368	256×256	$850 \times 850(\text{max})$	API	API
Optimizer	SGD	RMSProp	SGD	API	API
Loss Function	L2	MSE	Cross Entropy	API	API
Activation	ReLU	ReLU	Sigmoid	API	API
Metric	Accuracy	Accuracy	Accuracy	API	API

CPM, Stacked Hourglass Networks, DeeperCut은 소스에 대한 해석이 복잡하여 학습 시 여러 가지 오류가 발생하였다. 그래서 본 연구에서 고려하고 있는 환경에서 구현이 가능할 수 있도록 PoseNet을 선택하게 되었다. PoseNet은 다른 모델들과 성능을 비교하기 위해 LSP DataSet으로 [그림 4-3]처럼 Single-Person Pose Estimation 알고리즘을 사용하였고, 모델 구조에 따른 설정값은 아래 [표 4-2]와 같다.

```

35  const net = await posenet.load()
36
37  var files = fs.readdirSync('./lsp');
38
39
40  files.forEach(async x => {
41
42    const img = new Image()
43    img.src = './lsp/' + x
44
45    const canvas = createCanvas(img.width, img.height);
46    const ctx = canvas.getContext('2d');
47    ctx.drawImage(img, 0, 0);
48
49    const input = tf.browser.fromPixels(canvas);
50    const pose = await net.estimateSinglePose(input, imageScaleFactor, flipHorizontal, outputStride);

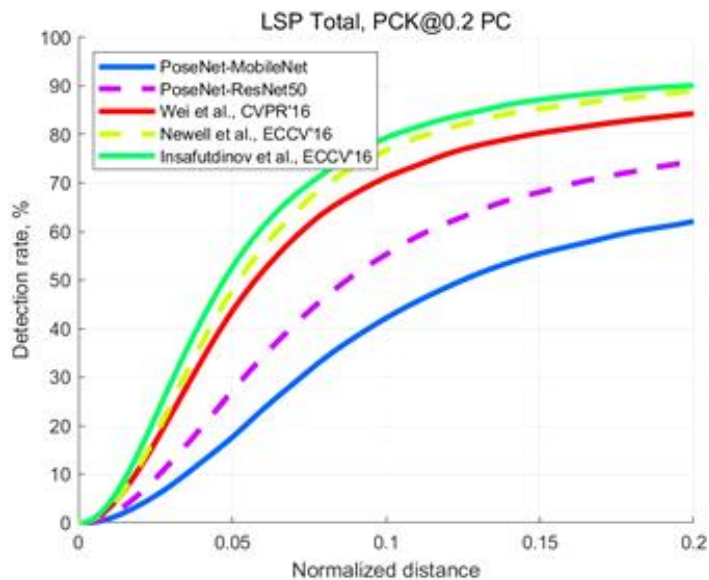
```

[그림 4-3] PoseNet의 Single-Person Pose Estimation(하태용 외, 2020)

[표 4-2] PoseNet 모델 구조에 따른 파라미터 정보(하태용 외, 2020)

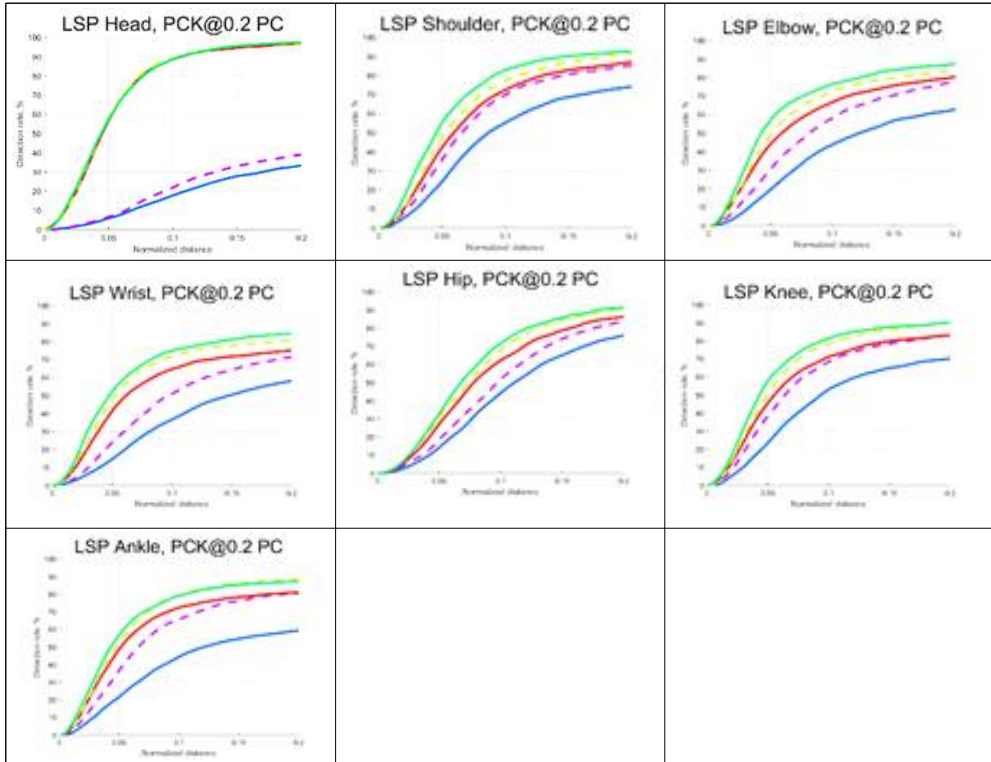
Architecture	MobileNet	ResNet
Output Stride	16	32
Input Resolution	513	257
Multiplier	0.75	-
QuantBytes	2	2

PoseNet은 목과 머리끝 포인트 대신 코, 눈, 귀 포인트를 예측하여 결과값으로 17개의 KeyPoint를 출력한다. 추가적으로, 행렬로 표현해 보면, $2(x, y \text{ 좌표}) \times 17(\text{KeyPoint}) \times 1,000(\text{테스트 이미지 개수})$ 이다. 그러므로, PoseNet의 성능을 LSP DataSet으로 보기 위해서는 타 모델과 같은 14개의 KeyPoint로 추가적인 변환 처리가 필요하다. 이에 대한, 데이터 포맷 변환 결과는 [그림 4-4]와 같다. 결론적으로, PoseNet의 두 모델이 다른 세 모델에 비해 낮은 PCK 수치를 보인다. 특히, [그림 4-5] 와 [표 4-3]의 Head 부분에서 다른 부분과 비교하여 PCK 결과값이 좀 더 낮게 보이게 된다. 이에 대한, 원인은 추후 PoseNet 관점에서의 결과 분석에서 논하고자 한다.



[그림 4-4] LSP DataSet에서 PCK@0.2 결과(전체)

[표 4-3]은 다른 모델은 14개 포인트를 가지고 정확도를 측정한 반면, PoseNet은 17개 포인트를 기준으로 정확도를 측정하였다. 그래서, 측정에 대한 타당도 기준을 마련하고자 눈, 코, 귀로 이루어지는 수평적 5개 포인트를 정수리로 통하는 머리꼭지 부분과 턱 부분의 두 점으로 이루어지는 수직적 2개 포인트로 변환하는 작업을 하여야 한다.



[그림 4-5] LSP DataSet에서 PCK@0.2 결과(부분)

[표 4-3] LSP & MPII DataSet에서 PCK 결과(하태용 외, 2020)

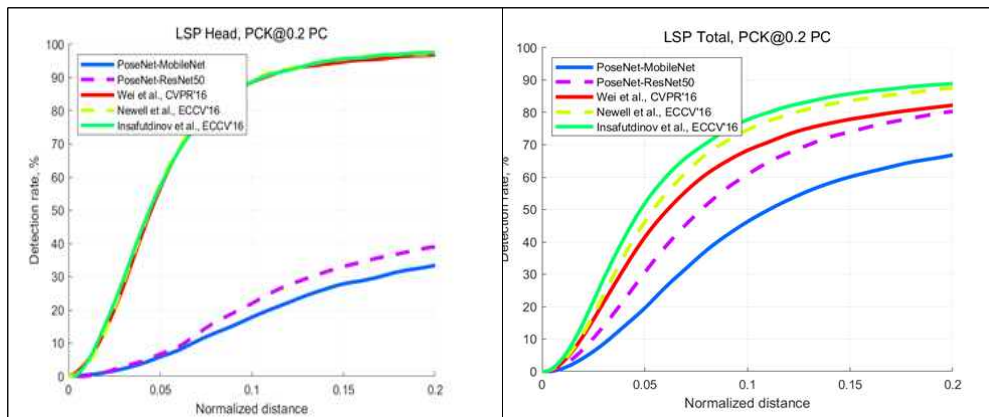
Method	Head	Sho	Elb	Wri	Hip	Knee	Ank	PCK mean	AUC
PoseNet – MobileNet	33.45	74.20	63.10	58.30	76.10	70.30	59.40	62.12	36.53
PoseNet – ResNet	39.00	85.20	77.65	71.35	83.80	83.25	80.65	74.41	46.90
S. E. Wei, et al, CVPR'16	97.80	92.50	87.00	83.90	91.50	90.80	89.90	90.50	65.40
A. Newell, et al., ECCV'16	98.20	96.3	91.20	87.10	90.10	87.40	83.6	90.90	65.20
E. Insafutdinov, et al., ECCV'16	97.45	92.70	87.55	84.45	91.45	89.90	87.20	90.10	66.08

[표 4-4]와 [그림 4-6]에서 PoseNet 17개 포인트 중 Headdp 해당하는 수치를 제외하면 ResNet이 타 모델과 비슷하게 나왔다.

[표 4-4] LSP DataSet에서 PCK 결과(Head 제외)

Method	Sho	Elb	Wri	Hip	Knee	Ank	PCK mean	AUC
PoseNet-MobileNet	74.20	63.10	58.30	76.10	70.30	59.40	66.9	39.80
PoseNet-ResNet	85.20	77.65	71.35	83.80	83.25	80.65	80.32	51.33
S. E. Wei, et al., CVPR'16	92.50	87.00	83.90	91.50	90.80	89.90	90.50	65.40
A. Newell, et al., ECCV'16	91.10	83.90	80.60	90.75	90.05	88.50	87.48	61.99
E. Insafutdinov, et al., ECCV'16	92.70	87.55	84.45	91.45	89.90	87.20	88.88	64.98

PoseNet에서 목과 머리끝의 두 포인트는 정식으로 예측되는 부분이 아니기 때문에 PoseNet이 PCK의 Head 부분에서 받은 낮은 점수에는 큰 의미를 둘 필요가 없다고 본다. 자세 추정을 위한 운동 동작에서 머리 부분을 활용한 동작이 없는 관계로 비교를 통한 그래프 작성으로 의미가 있다고 본다. [그림 4-6]에서 Head를 수치를 제외하면 ResNet이 타 모델과 비슷하게 나왔다.



[그림 4-6] LSP DataSet에서 PCK 결과(Head 제외)

본 연구에서는 손뼉(박수) 치기에 대한 운동 파악이다. 요약하면, [그림

4-7과 같이 목 포인트는 왼쪽과 오른쪽 어깨 포인트의 중간점으로 지정했다. 추가로, 머리끝 포인트를 정하기 위해서는 눈과 귀, 코 포인트를 이용한다. 머리끝 포인트는 얼굴의 다섯 포인트에 대해 예측된 값을 최대한 활용하여 계산하였는데, 이미지에서 사람이 앞과 뒤를 보고 있을 경우와 왼쪽을 보고 있을 경우 또는 오른쪽을 보고 있을 경우로 나누어 계산해 주었다.

```

1 - nose = a(:,1,:);      lefteye = a(:,2,:);      righteye = a(:,3,:);
2 - leftear = a(:,4,:);   rightear = a(:,5,:);   leftshoulder = a(:,6,:);
3 - rightshoulder = a(:,7,:); leftthip = a(:,12,:); righthip = a(:,13,:);
4 -
5 - theta=linspace(0,2*pi,100);
6 - x1 = 0; y1 = 0;
7 -
8 - for i=1:1000
9 -     a_ear_to_ear = abs(rightear(1,:,i) - leftear(1,:,i));
10 -    a_leftear_to_nose = abs(nose(1,:,i) - leftear(1,:,i));
11 -    a_rightear_to_nose = abs(rightear(1,:,i) - nose(1,:,i));
12 -
13 -    if (a_ear_to_ear > a_leftear_to_nose && a_ear_to_ear > a_rightear_to_nose)
14 -
15 -        center = [(rightear(1,:,i)+leftear(1,:,i))/2 (rightear(2,:,i)+leftear(2,:,i))/2];
16 -        r = norm(rightear(:,i)-leftear(:,i));
17 -
18 -        x1 = linspace(center(1)-r/2,center(1)+r/2);
19 -        y1 = -(rightear(1,:,i)-leftear(1,:,i))/(rightear(2,:,i)-leftear(2,:,i))*(x1-center(1))+center(2);
20 -
21 -    elseif(a_leftear_to_nose > a_ear_to_ear && a_leftear_to_nose > a_rightear_to_nose)
22 -
23 -        center = [(nose(1,:,i)+leftear(1,:,i))/2 (nose(2,:,i)+leftear(2,:,i))/2];
24 -        r = norm(leftear(:,i)-nose(:,i));
25 -
26 -        x1 = linspace(center(1)-r/2,center(1)+r/2);
27 -        y1 = -(nose(1,:,i)-leftear(1,:,i))/(nose(2,:,i)-leftear(2,:,i))*(x1-center(1))+center(2);
28 -    elseif (a_rightear_to_nose > a_ear_to_ear && a_rightear_to_nose > a_leftear_to_nose)
29 -
30 -        center = [(nose(1,:,i)+rightear(1,:,i))/2 (nose(2,:,i)+rightear(2,:,i))/2];

```

[그림 4-7] MatLab 코드

MatLab 코드에서 a는 LSP DataSet의 테스트 이미지 1,000개에 대해 각각 17개의 관절 포인트를 예측한 값을 가지고 있는 크기가 $2 \times 17 \times 1,000$ 인 변수이고, Pred는 a를 LSP DataSet의 Annotation 포맷의 크기 $2 \times 14 \times 1,000$ 에 맞춘 값을 가진 변수이다.

나) PoseNet 관점에서의 결과 분석

먼저 PoseNet의 평가를 위한 LSP DataSet을 다운로드한다. 그리고 Single

Pose를 인식하기 위해선 URL에 접속하여 PoseNet 모델을 다운로드한다.²⁰⁾ 그리고 나서, PoseNet 파일로 이동 후 Demos 파일로 들어가서 index.html 파일을 코드 에디터를 사용하여 작업한다. 여기서는, Visual Studio Code에서 작업하였고, [그림 4-8]에서 Visual Code Terminal의 Setup 부분을 따라 해주면 된다.²¹⁾

```
Setup

cd into the demos folder:

cd posenet/demos

Install dependencies and prepare the build directory:

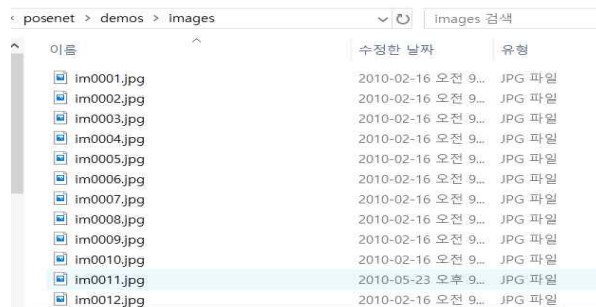
yarn

To watch files for changes, and launch a dev server:

yarn watch
```

[그림 4-8] Visual Code Setup

다음으로 LSP DataSet을 Input으로 넣고, PoseNet으로 17개 KeyPoint를 인식하여 Output을 콘솔 창에 띄우고자 한다. 그러기 위해, Demos 파일 안에 Input이 들어갈 Images 파일을 만들어 그 안에 [그림 4-9]처럼 다운로드 한 LSP DataSet 사진 1,000장을 넣으면 된다.



[그림 4-9] 다운로드 한 LSP

20) <https://sam.johnson.io/research/lsp.html>

21) <https://github.com/tensorflow/tfjs-models/tree/master/posenet/demos>

GitHub 페이지의 사용 방법 중에 Via Script Tag의 코드를 복사하여 index.html에 복사하여 넣는다.²²⁾ 다음으로, [그림 4-10]처럼 코드에 대해 부연 설명을 하면 <head> 부분은 tensorflow.js와 PoseNet을 불러오는 코드이다. 추가적으로, <body>에선 이미지를 불러온다. 마지막으로, <script>부분은 불러온 이미지를 처리할 코드로 이 부분을 주로 코딩한다.

Example Usage

via Script Tag

```
<html>
<head>
  <!-- Load TensorFlow.js -->
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs"></script>
  <!-- Load Posenet -->
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/posenet"></script>
</head>

<body>
  <img id='cat' src='/images/cat.jpg' />
</body>
<!-- Place your code in the script tag below. You can also use an external .js file -->
<script>
  var flipHorizontal = false;

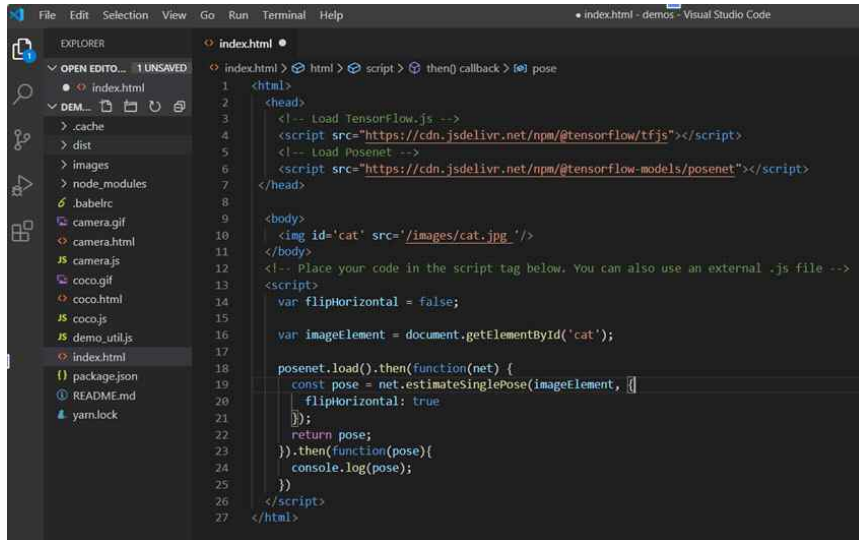
  var imageElement = document.getElementById('cat');

  posenet.load().then(function(net) {
    const pose = net.estimateSinglePose(imageElement, {
      flipHorizontal: true
    });
    return pose;
  }).then(function(pose){
    console.log(pose);
  })
</script>
</html>
```

[그림 4-10] Via Script Tag 코드

[그림 4-11]에서 <head> 부분은 놔두고, <body> 부분에 이미지를 넣어주는 작업할 것이다. 본 연구에서는 Gram 기준으로 MobileNet은 한 번에 100장 ResNet은 50장씩 넣어서 돌렸으며, ResNet은 50장을 넣어도 GPU 부족하다고 나왔으며, GPU가 충분한 환경에서 하고 있다면 1,000장 한 번에 돌려도 될 듯하다.

22) <https://github.com/tensorflow/tfjs-models/tree/master/posenet>



[그림 4-11] Gram기준 MobileNet & ResNet 이미지 넣기

[그림 4-12]에서 <body>부분에 PoseNet으로 인식할 이미지를 넣어줬다면, <script> 부분에서 Array를 만들어 그 안에 이미지 파일의 주소를 넣어준다. String으로 들어가기 때문에 For 문을 계속하여 바꾸어야 한다.



[그림 4-12] Script 부분의 Array에 이미지 넣기

[그림 4-13]에서 PoseNet을 Load 하고 Architecture를 설정한다. 그리고 Architecture는 PoseNet의 MobileNet 버전으로 한 번, ResNet 버전으로 한 번 하여 총 두 번의 작업을 한다.

```
console.log(["  
|  
posenet.load({  
  architecture: 'ResNet50',  
  outputStride: 32,  
  inputResolution: { width: 257, height: 200 },  
  quantBytes: 2  
}).then(function(net) {  
  const pose = net.estimateSinglePose(k, {  
    flipHorizontal: true  
  });  
  return pose;  
}).then(function(pose){
```

[그림 4-13] PoseNet Architecture 설정

Pose를 인식했으면 [그림 4-14]처럼 For 문이 돌면서 Pose에 대한 KeyPoint x 좌표, y 좌표를 Console.log로 출력하게 된다.

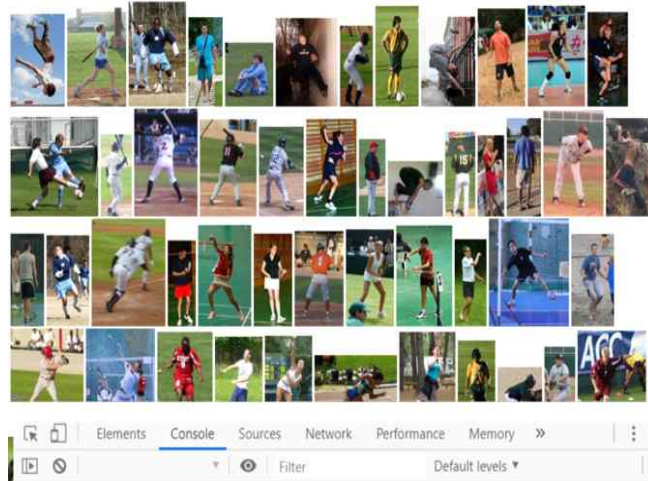
```

    }).then(function(net) {
      const pose = net.estimateSinglePose(k, {
        flipHorizontal: true
      });
      return pose;
    }).then(function(pose){
var poseArray = new Array
console.log("[")
console.log("{ keypoint: ")
//console.log("{ keypoint: ")%o
//console.log("[")%o
var aaa =0;
  for(const keypoint of pose.keypoints) {
    // console.log(' ${keypoint.part}: (${keypoint.position.x},${keypoint.position.y})');
    aaa++;
    if(aaa==17)
    {
      console.log([keypoint.position.x,keypoint.position.y])
      //console.log([keypoint.position.x,keypoint.position.y])%o
    }
    else
    {
      console.log([keypoint.position.x,keypoint.position.y],",")
      //console.log([keypoint.position.x,keypoint.position.y],",")%o
    }
  }
  console.log("] ],")
})
})();
}
</script>
</html>

```

[그림 4-14] Pose에 대한 KeyPoint x, y 좌표

코드를 저장하고 [그림 4-15]처럼 Yarn Watch를 하면 크롬 창이 열리고 그 안에 이미지가 들어 있는 것을 확인할 수 있다.



[그림 4-15] Yarn Watch 이미지

F12를 눌러 [그림 4-16]처럼 Console 창에 좌표가 찍히는 것을 볼 수 있다.

```
{ keypoint: (index):128
  ▶ (2) [43.68001986848705, 137.87112403039487] ", " (index):140
  ▶ (2) [44.633776712974225, 138.8982546317145] ", " (index):140
  ▶ (2) [38.43750486856305, 141.02986247798938] ", " (index):140
  ▶ (2) [45.42092004917009, 142.39807057751275] ", " (index):140
  ▶ (2) [30.720162714501754, 135.8041153349407] ", " (index):140
  ▶ (2) [54.88500703447987, 133.20115116099618] ", " (index):140
  ▶ (2) [35.29588228812014, 131.71186923486582] ", " (index):140
  ▶ (2) [64.09900988864527, 125.17734302263803] ", " (index):140
  ▶ (2) [47.07593937142812, 119.67860531189281] ", " (index):140
  ▶ (2) [61.821637699112344, 21.02875551411525] ", " (index):140
  ▶ (2) [71.79555437258709, 28.017412867570787] ", " (index):140
  ▶ (2) [49.10276313514561, 106.6568499154995] ", " (index):140
  ▶ (2) [47.3994025441923, 97.53087604107634] ", " (index):140
  ▶ (2) [76.51998479820874, 93.96658404256395] ", " (index):140
  ▶ (2) [35.61418359659987, 85.18581829169871] ", " (index):140
  ▶ (2) [97.3071741482627, 91.33237346589874] ", " (index):140
  ▶ (2) [63.37968771374179, 54.428699770121995] (index):138
```

[그림 4-16] Console 창 좌표

마지막으로 [그림 4-17]에서 각 사진에 대한 KeyPoint 값이 출력된다. 출력이 되는 순서는 0부터 16까지 17개 KeyPoint이다.

Id	Part
0	nose
1	leftEye
2	rightEye
3	leftEar
4	rightEar
5	leftShoulder
6	rightShoulder
7	leftElbow
8	rightElbow
9	leftWrist
10	rightWrist
11	leftHip
12	rightHip
13	leftKnee
14	rightKnee
15	leftAnkle
16	rightAnkle

[그림 4-17] KeyPoint 값 출력 순서

KeyPoint 값을 다운로드하여 로그를 확인하면 불필요한 문자들이 섞여 있다. 이것을 Visual Studio Code나 Visual Studio로 열어 전체 바꾸기로 양식을 맞춰야 한다. 추가로, BackSpace도 글자를 인식하기에 json 파일을 가로줄에 나열하는 것이 중요하며, 나중에 Matlab 변수를 가져올 때 편하다. [그림 4-18]은 PoseNet의 KeyPoint 출력값을 .json 파일로 리다이렉트하기이다.

```
[ { score: 0.18704429268836975,
  part: 'nose',
  position: { x: 50.825444796669814, y: 33.31685185710744 } },
{ score: 0.1975477933883667,
  part: 'leftEye',
  position: { x: 45.305119941206755, y: 27.55495869974218 } },
{ score: 0.21696946024894714,
  part: 'rightEye',
  position: { x: 54.463458406322204, y: 28.66240763385936 } },
{ score: 0.6581169962882996,
  part: 'leftEar',
  position: { x: 44.40706782953285, y: 29.414613196821993 } },
```

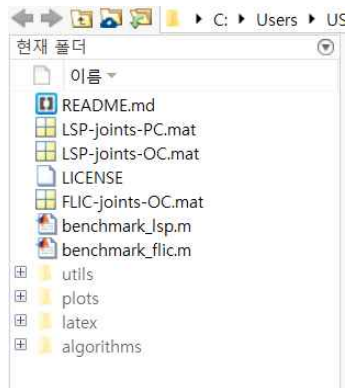
[그림 4-18] PoseNet의 출력 값 리다이렉트(하태용 외, 2020)

[그림 4-19]에서 PoseNet의 출력값을 콘솔 창으로 프린트하는 대신에 임의의 json 파일로 리다이렉트 한다. 그다음 json 파일을 열어 파일의 가장 앞과 뒤에 각각 ‘[와]’를 추가해 주고, 각 이미지에 대한 KeyPoint 값이 ‘,’로 구별되도록 ’}’ 뒤에 ‘,’를 추가해 주면 MatLab에서 열 수 있는 json 파일 형식이 완성된다. 그리고 같은 방식으로 문자를 맞춰 .json 파일로 저장한다.

```
[{"keypoint": [{"x": 50.825444796669814, "y": 33.31685185710744}, {"x": 45.305119941206755, "y": 27.55495869974218}, {"x": 54.463458466322264, "y": 28.66240763385936}, {"x": 56.077515954037335, "y": 31.92623369981342}, {"x": 46.78921725499491, "y": 44.21837779241777}, {"x": 54.80848717596744, "y": 44.62795820077479}, {"x": 49.12008797612171, "y": 57.89891586600575}, {"x": 77.27875124065236}, {"x": 60.72728684718507, "y": 78.87097693417323}, {"x": 46.12986874116534, "y": 84.65845676526021}, {"x": 49.78223095608129, "y": 81.87809973097033}, {"x": 112.71612085720908}, {"x": 38.07283589997644, "y": 142.56991927447487}, {"x": 20.601923961119084, "y": 148.1250771251634}, {"x": 127.76033509773225, "y": 50.67996672990257}, {"x": 123.02816974328185, "y": 50.69085292593514}, {"x": 124.99558528202510, "y": 51.10558211508428}, {"x": 110.8270992768871, "y": 95.68328263694673}, {"x": 62.997578186747624}, {"x": 140.33966905474828, "y": 91.49693993000669}, {"x": 79.17219596892478, "y": 82.66892886625654}, {"x": 134.7115321771644, "y": 110.19262481570706}, {"x": 112.85476833002112}, {"x": 89.40216597857642, "y": 104.68172766540705}, {"x": 146.80411847185067, "y": 106.04198589399167}, {"x": 58.901451905240, "y": 47.18329558019972}, {"x": 128.2690556745121}, {"x": 64.28528621688427, "y": 44.98365401687325}, {"x": 63.5292881768965, "y": 43.143403257377415}, {"x": 60.7072795467154, "y": 41.96233050275869}, {"x": 55.25903257971143, "y": 54.74030912525459}, {"x": 83.8263164475734, "y": 51.81773263656675}, {"x": 44.09573987671373, "y": 55.63679047224586}, {"x": 80.49761541344311}, {"x": 103.313973749658, "y": 81.24557756357156}, {"x": 69.6238177110249, "y": 96.11142555563366}, {"x": 90.3499259503435, "y": 88.4186.81957105347153, "y": 102.13446400530915}, {"x": 64.80403006587846, "y": 130.441884141024}, {"x": 112.09170365982946, "y": 130.9684455960641}], {"keypoint":
```

[그림 4-19] PoseNet의 출력 값 .json 파일(하태용 외, 2020)

1,000장의 이미지가 변환이 완료되었다면, [그림 4-20]처럼 Matlab에 변수로 넣는다. 이전에 사용한 Benchmark²³⁾ 코드를 다운로드한다. 그리고, 이 Benchmark는 같은 LSP DataSet에 대한 다양한 모델로 측정했을 때의 정확도를 PCK 그래프로 산출해내는 코드이다. 다음으로 Matlab을 실행하여, 현재 폴더를 다운로드한 Benchmark 파일로 이동시킨다. 이때, 사용법은 명령창에 Benchmak_lsp를 입력하면 된다. 하지만, 그전에 벤치마크의 변수를 입력하고 json 파일을 수정해야 한다.



[그림 4-20] Matlab 실행 화면

23) <https://github.com/farrajota/human-pose-benchmark>

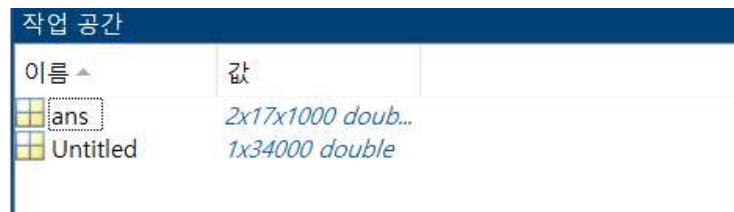
[그림 4-21]에서 PoseNet KeyPoint과 LSP DataSet KeyPoint를 산출해낸 json 파일은 PoseNet 기준 키값을 산출해 낸 것이고, PCK 그래프를 산출하기 위해선 LSP DataSet 기준의 값으로 바뀌야 비교할 수 있다. LSP DataSet 중에서도 1~12번까지는 PoseNet KeyPoint에 있지만 13(Neck)과 14번(Head Top)은 따로 값을 구해서 추가해야 한다. 따라서, Matlab에서 사용하기 위해서 변수를 먼저 가져와야 하며, 이때 산출해 낸 json 파일을 가져오게 된다.

Id	Part
0	nose
1	leftEye
2	rightEye
3	leftEar
4	rightEar
5	leftShoulder
6	rightShoulder
7	leftElbow
8	rightElbow
9	leftWrist
10	rightWrist
11	leftHip
12	rightHip
13	leftKnee
14	rightKnee
15	leftAnkle
16	rightAnkle

1. Right ankle
2. Right knee
3. Right hip
4. Left hip
5. Left knee
6. Left ankle
7. Right wrist
8. Right elbow
9. Right shoulder
10. Left shoulder
11. Left elbow
12. Left wrist
13. Neck
14. Head top

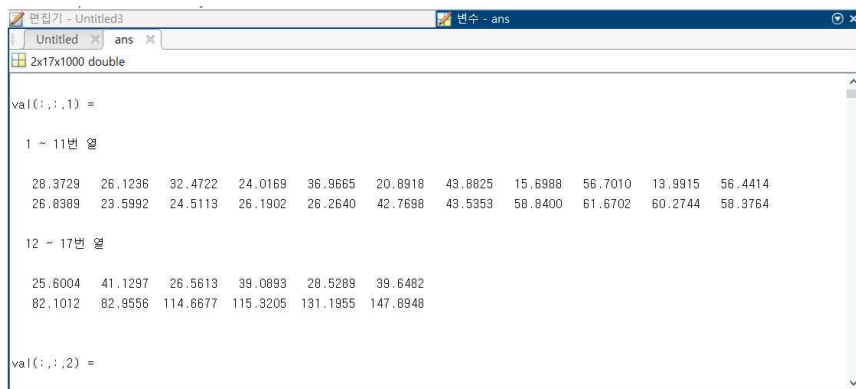
[그림 4-21] PoseNet KeyPoint과 LSP DataSet KeyPoint

[그림 4-22]에서 json 파일이 있는 파일로 이동하여 데이터 가져오기 -> 숫자형 행렬 -> 선택 항목 가져오기 하여 Mmatlab 변수와 데이터 포맷을 맞춘다. 그리고, 명령어 창에서 Reshape 함수를 사용하여 포맷을 $2 \times 17 \times 1,000$ 으로 맞춰준다. 그러면, Ans라는 $2 \times 17 \times 1000$ 변수가 생성된다.



[그림 4-22] Reshape 함수 사용

[그림 4-23]에서 PoseNet의 LSP DataSet에 대한 출력값을 LSP DataSet의 Annotation 포맷인 .mat와 같이 MatLab에서 읽을 수 있는 변수로 만들기 위해 고안한 방법으로 출력값을 모두 .json 형식으로 변환시켜 저장하는 것이다. MatLab은 json 형식 텍스트를 디코딩 하는 함수 'jsondecode'를 지원하기 때문이다. 이제, Ans를 클릭하면 1번 ~ 17번 열은 PoseNet의 키값 들이고 윗줄은 x 좌표 아랫줄은 y 좌표값이다. 끝으로, 변수 불러오기는 끝이며 LSP DataSet 형식으로 맞춰주면 된다.



[그림 4-23] LSP DataSet의 Annotation 포맷 .mat(하태용 외, 2020)

[그림 4-24]에서 Neck.m 파일을 열어 명령어 창에서 Neck을 쳐서 코드를 수행한다. 이 작업은, PoseNet의 키 순서를 LSP 순서로 바꿔주는 코드이다. 또한, 13번인 Neck은 양쪽 어깨의 중간지점으로 설정하여 13번에 삽입해주는 코드이다. 완료가 되면, 오른쪽 변수 창에 Neck에 대한 변수들이 생긴다. 이제, Head Top을 구해주기 위해 GetHead.m 파일을 열고 실행한다. Neck.m 파일 실행 후에 변수 창을 보면 Neck1 변수가 $2 \times 13 \times 1000$ 에서 $2 \times 14 \times 1000$ 의 형태로 바뀐 것을 확인할 수 있다. Neck1을 제외한 다른 변수는 다 지워줍니다. 클릭하여 잘 들어갔는지 체크해 준다. 그리고 Neck1의 변수의 이름을 Pred로 바꿔 준 후 홈 -> 작업공간에 저장한다. 이제, Mat 형식의 파일로 저장이 된 것이다.

```

1  R_ankle = ans(:,17,:); R_knee = ans(:,15,:); R_hip = ans(:,13,:); L_hip = ans(:,12,:);
2  L_knee = ans(:,14,:); L_ankle = ans(:,16,:); R_wrist = ans(:,11,:); R_elbow = ans(:,9,:);
3  R_shoulder = ans(:,7,:); L_shoulder = ans(:,6,:); L_elbow = ans(:,8,:); L_wrist = ans(:,10,:);
4  for i=1:1000
5      neck_1= (L_shoulder(1,:,i)+R_shoulder(1,:,i))/2;
6      neck_2= (L_shoulder(2,:,i)+R_shoulder(2,:,i))/2;
7
8      NECK1(1,1,i) = R_ankle(1,:,i); NECK1(2,1,i)=R_ankle(2,:,i);
9      NECK1(1,2,i) = R_knee(1,:,i); NECK1(2,2,i)=R_knee(2,:,i);
10     NECK1(1,3,i) = R_hip(1,:,i); NECK1(2,3,i)=R_hip(2,:,i);
11     NECK1(1,4,i) = L_hip(1,:,i); NECK1(2,4,i)=L_hip(2,:,i);
12     NECK1(1,5,i) = L_knee(1,:,i); NECK1(2,5,i)=L_knee(2,:,i);
13     NECK1(1,6,i) = L_ankle(1,:,i); NECK1(2,6,i)=L_ankle(2,:,i);
14     NECK1(1,7,i) = R_wrist(1,:,i); NECK1(2,7,i)=R_wrist(2,:,i);
15     NECK1(1,8,i) = R_elbow(1,:,i); NECK1(2,8,i)=R_elbow(2,:,i);
16     NECK1(1,9,i) = R_shoulder(1,:,i); NECK1(2,9,i)=R_shoulder(2,:,i);
17     NECK1(1,10,i) = L_shoulder(1,:,i); NECK1(2,10,i)=L_shoulder(2,:,i);
18     NECK1(1,11,i) = L_elbow(1,:,i); NECK1(2,11,i)=L_elbow(2,:,i);
19     NECK1(1,12,i) = L_wrist(1,:,i); NECK1(2,12,i)=L_wrist(2,:,i);
20     NECK1(1,13,i) = neck_1; NECK1(2,13,i)=neck_2;
21 end

```

[그림 4-24] NECK.m 파일

마지막으로, [그림 4-25]에서 MobileNet 또는 ResNet 수행한 모델이다.

```

val(:,1,1) =

1 ~ 11번 열

    39.6482    39.0893    41.1297    25.6004    26.5613    28.5289    56.4414    56.7010    43.8825    20.8918    15.6988
   147.8948   115.3205    82.9556    82.1012   114.6677   131.1955    58.3764    61.6702    43.5353    42.7698    58.8400

12 ~ 14번 열

    13.9915    32.3871    30.5286
    60.2744    43.1526    19.7553

val(:,1,2) =

```

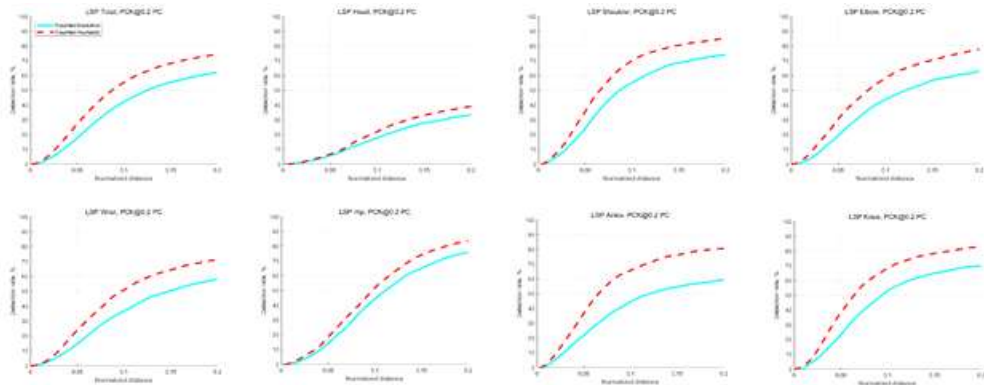
[그림 4-25] MobileNet & ResNet 수행한 모델

두 모델의 MatLab의 Benchmark 파일 중 Algorithms 폴더 안에 들어가 Benchmark_lsp 명령어를 실행하면 그래프가 산출된다. [표 4-5]와 [그림 4-26]에서 PCK@0.2 값들을 구체적으로 살펴보면, ResNet 버전 모델이 MobileNet 버전 모델보다 전체적으로 10% 이상 높은 성능을 보인다. 레이어가 많아 모델 자체가 무겁고 느린 대신에 정확도는 상대적으로 높은 ResNet

구조 자체의 특성이 잘 보이는 결과값이다.

[표 4-5] PoseNet 관점에서의 결과 분석

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	PCK mean	AUC
ResNet	39.00	85.20	77.65	71.35	83.80	83.25	80.65	74.41	46.90
MobileNet	33.45	74.20	63.10	58.30	76.10	70.30	59.40	62.12	36.53



[그림 4-26] PoseNet의 ResNet & MobileNet 성능 비교(LSP DataSet , PCK@0.2)(하태용 외, 2020)

관절 부분에 따른 인식 정확도의 양상을 보면 다른 모델들과 크게 다르지 않다. 다른 모델들이 상대적으로 높은 정확도로 인식한 어깨나 골반 부분은 PoseNet도 다른 부분들에 비해 잘 인식했다. 무릎과 발목 부분에 대한 성능은 CPM과 거의 차이 나지 않거나 조금 높다(1% 안팎).

2) 동작 인식 개발 구현

PoseNet 모델을 활용하여 각 관절의 좌표 값들은 얻어낼 수 있으나, 해당 좌표 값을 가지고 직접 운동 인식 알고리즘을 구현하기에 어려움이 있었다. 운동하는 각 자세의 좌표 값들을 학습 데이터로 적절히 구성하고, 자세를 분류하는 딥러닝 모델을 구현하여, 규칙 기반 알고리즘으로 운동을 인식하기로

결정하였다. 이에 본 연구에서는 PoseNet을 활용하여 사람의 운동 동작 중 가장 단순한 편에 속하는 ‘손뽕(박수) 치기’ 동작을 인식해 보았다. 사람이 카메라 앞에서 손뽕(박수)를 치면 PoseNet을 통해 얻은 관절들의 위치 정보들을 분석하여 손뽕(박수) 친 횟수를 세고 이를 화면에 표시해 주는 구조이다.

PoseNet 모델과 Node.js express 모듈을 활용하여 좌표 값을 저장하는 웹 페이지를 구성하였으며, 인공지능 모델 학습을 위한 좌표를 저장하기 위해 다음과 같은 제약 사항을 두었다.

첫 번째로, 많은 데이터의 좌표 값들을 빠른 시간 내에 얻기 위해 버튼 한 번 클릭하여, 0.4초의 간격으로 좌표들을 연속적으로 저장하였다.

두 번째로, 좌표 값들을 얻을 때 모든 관절의 좌표(17개)를 얻도록 하였다. 아울러 모든 관절의 좌표가 표시되지 않으면 데이터 저장을 하지 않았다.

세 번째로, 모든 관절을 추정한 정확도가 70% 정확도 이상이어야 좌표를 저장하였다.

가) DataSet 분석

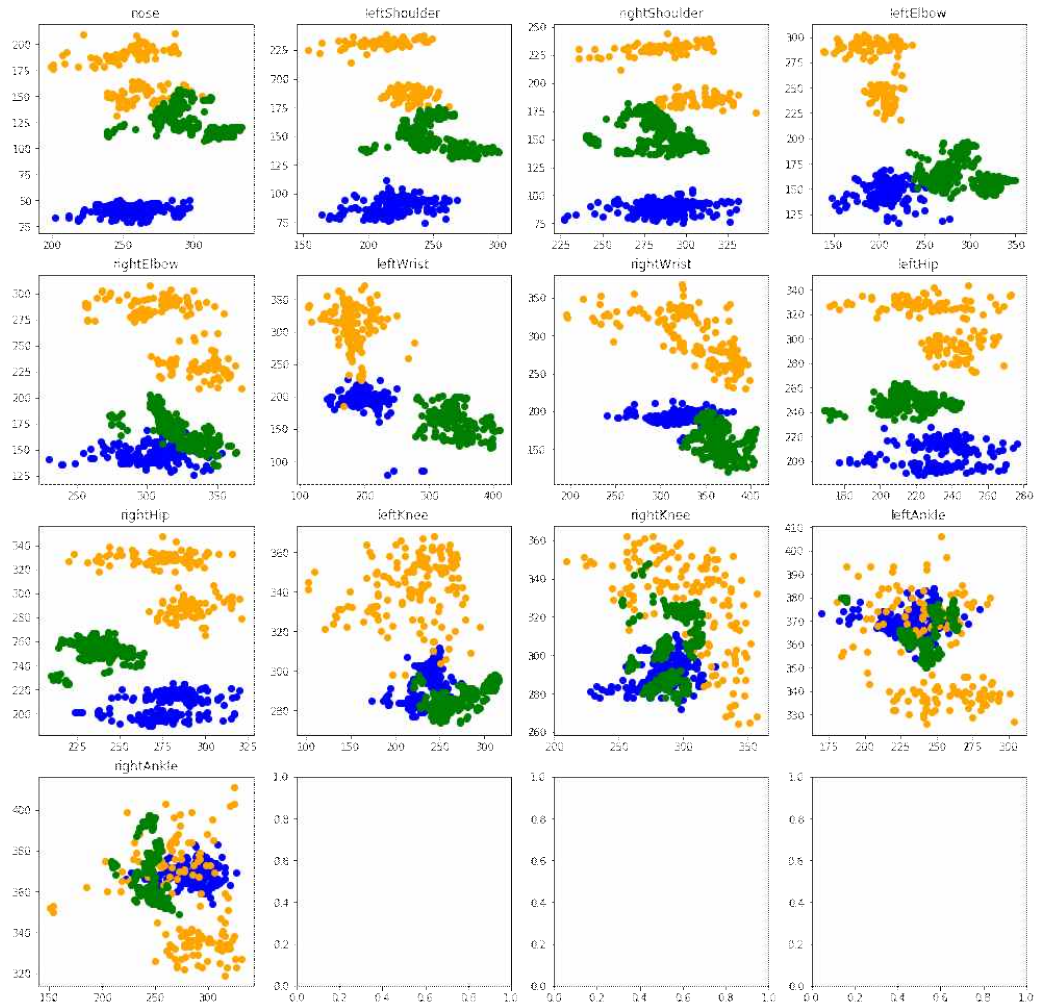
Python Matplotlib 라이브러리를 사용하여 서 있는 자세, 앉아 있는 자세, 스쿼트 자세 총 3가지의 자세의 관절 좌표를 [그림 4-27]에서 분석하였다. 파란색은 서 있는 자세, 오렌지색은 앉아있는 자세, 초록색은 스쿼트 자세이다.

첫 번째로, 상반신(Nose, Left Shoulder, Right Shoulder)의 Y 축 좌표를 보면 앉아 있는 자세, 스쿼트 자세, 서 있는 자세 순서로 Y 좌표 값이 커지는 것을 볼 수 있다. 이는 실제 자세와 명확하게 같은 결과를 보여주며, 이러한 상반신의 좌표 값이 자세를 분류하는 데 중요한 역할을 하는 것을 알 수 있다.

두 번째로, 스쿼트 자세인 초록색을 분석하였다. 이것은, 대각선으로 서서 팔을 앞으로 뻗는 자세로 스쿼트 자세 데이터 셋을 구성하였는데 이에 따라서, 왼쪽 어깨, 팔꿈치, 손목(Left Elbow, Left Shoulder, Left Wrist)의 X 좌표가 다른 자세보다 오른쪽에 위치하는 것을 알 수 있다. 이와는 반대로 오른

쪽 엉덩이(Right Hip)는 스쿼트 자세에서 상대적으로 왼쪽으로 빠지기 때문에 그래프에서도 X 좌표는 다른 자세보다 왼쪽에 위치하게 된다.

세 번째로, 하반신(무릎, 발목)에 대한 좌표 데이터는 세 자세를 분류하는 특징을 찾지 못해 이번 자세를 분류하는데 적합하지 않았다.

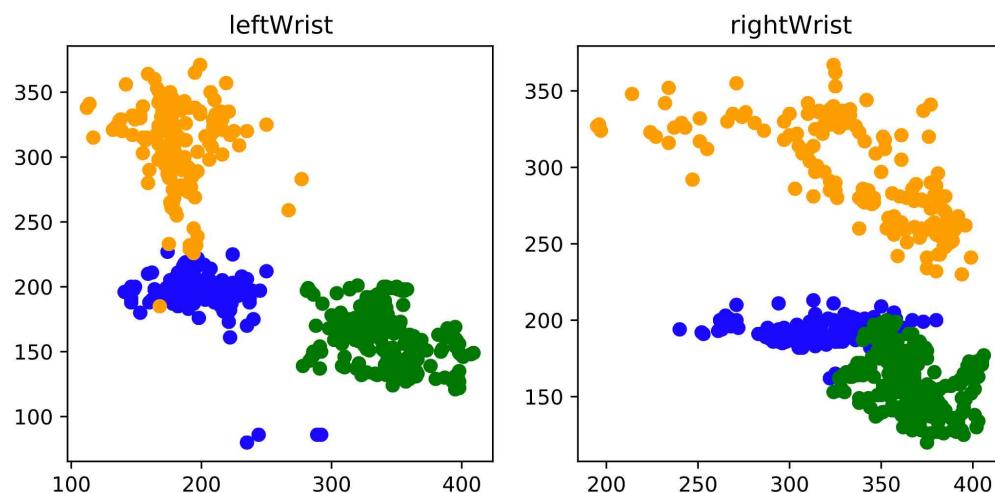


[그림 4-27] DataSet 분석

나) 손목 관점으로 DataSet분석

[그림 4-28]에서 두 그래프는 각각 왼쪽 손목, 오른쪽 손목 좌표를 나타

낸다. 공통으로 앉아 있는 자세인 오렌지색의 Y 좌표가 가장 큰 값을 가지고, 나머지 서 있는 자세와 스쿼트 자세의 Y 좌표는 비슷한 값을 가진다. 즉, 실제 앉은 자세에서는 손목의 위치가 바닥과 가깝게 되며, 서 있는 자세와 스쿼트 자세는 중간쯤에 위치된다. X 좌표를 보면, 서 있는 자세와 앉아 있는 자세는 비슷하게 위치한다. 그러나 스쿼트 자세는 대각선으로 선 자세에서 팔을 오른쪽으로 들기 때문에, 왼쪽 손목의 Y 좌표가 다른 자세보다 오른쪽으로 치우쳐져 있다. 최종적으로, 자세별 좌표의 특징이 명확해서 이를 바탕으로 구성한 딥러닝 모델이 100%에 가까운 정확도로 분류를 잘하는 것을 확인하였다. 그렇지만, 모델 훈련 같은 경우는 구글 PoseNet을 사용하였기 때문에 고려해야 할 사항으로 사용자 입장에서 카메라의 각도나 위치 그리고 배경이 영향 미친다는 것을 알 수 있었다.



[그림 4-28] 손목 관점으로 DataSet 분석(하태용 외, 2020)

다) 구현 방식

모든 코드는 PoseNet의 데모 코드 파일 camera.js²⁴⁾에 추가된다. 위치는 424번째 줄 코드 이후, 곧 인식한 자세의 KeyPoint와 그에 따른 골격을 화면

24) <https://github.com/tensorflow/tfjs-models/blob/master/posenet/demos/camera.js>

에 그려주는 바로 그 부분이다. 필요한 변수는 미리 선언해 주었음을 전제한다. 그리고, [그림 4-29]에서 손뽕(박수) 치는 동작을 인식하는 방법을 요약하는 세 가지 키워드는 손목 사이의 거리, 스위치 변수, 시간차 계산이다. 여기서 고려할 사항으로는 손뽕(박수) 친 직후에 손바닥이 일정 거리 이상 떨어지지 않은 경우로서 박수 횟수 증가하면 안 된다는 것이다.

손목 간 거리 정보만 가지고 손뽕(박수) 친 횟수를 세려고 할 때 손바닥을 맞댄 채 계속 가만히 있어도 손뽕(박수) 친 횟수가 증가하는 문제가 발생했다. 그래서 손바닥이 한번 맞대진 후에 일정 거리 이상 떨어졌는지 확인하는 스위치 변수를 두는 동시에, 손바닥이 포개지는 순간을 포착한 후 손뽕(박수) 횟수를 증가시키는 그 사이의 시간차를 계산하도록 하여 이중으로 이런 문제가 발생하지 않도록 하였다.

```
if (keypoints[7].score >= minPartConfidence && keypoints[8].score >= minPartConfidence &&
    keypoints[9].score >= minPartConfidence && keypoints[10].score >= minPartConfidence) {

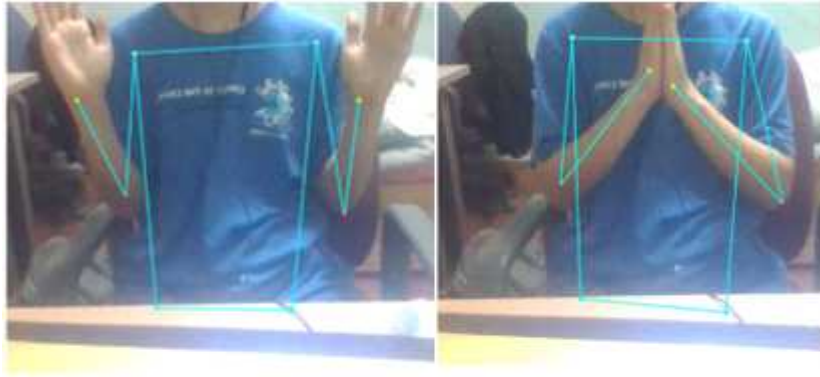
    if(count >= 1) {
        if (_switch != 0 && dist < _switch + 40)
            return;
        if (_switch != 0 && dist >= _switch + 40)
            _switch = 0;
    }

    if(dist <= 80 && _switch == 0) {
        start = new Date();
        var audio = document.getElementById("myAudio");
        if (count == 0 || (start - end >= 1000)) {
            _switch = dist;
            count = count + 1;
            document.getElementById("clap").innerHTML = '박수 ' + count;
            myAudio.play()
            myAudio.currentTime = 0;
            audio.play();
            end = start;
        }
    }
}
```

[그림 4-29] 손뽕(박수) 친 횟수를 세는 코드(하태용 외, 2020)

라) 동작 인식: 손뽕(박수) 치기

[그림 4-30]에서 박수 동작 인식 화면으로 스마트 폰에서 구현한 것으로 화면 하단에 손뽕(박수)치기 한 동작에 대한 카운트가 표시되고 있다.



박수 15

박수 16

[그림 4-30] 손뽕(박수) 동작 인식 화면(하태용 외, 2020)

3) 신체활동 목록과 MET 연계 구현

PoseNet 자세 추정의 손뽕(박수) 치기 운동량에 따른 결과 값과 신체활동 목록에 따라, MET 모델 설계에 따른 구현 결과로써 스마트 헬스케어로 연계된 내용을 [그림 4-31]과 같이 전체 흐름도를 보여 준다.



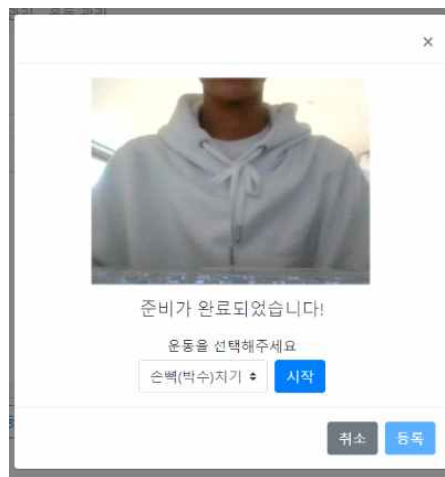
[그림 4-31] MET 모델 설계에 따른 구현 결과 (하태용 외, 2020)

첫 번째로, 사용자가 버튼 상단의 캘린더에서 날짜를 선택한 뒤 등록 버튼을 누르면, [그림 4-32]에서 '촬영하여 등록'은 카메라를 통해 사용자의 운동 활동을 인식하여 해당 내역을 등록하는 버튼이다.



[그림 4-32] 운동 등록
화면(하태용 외, 2020)

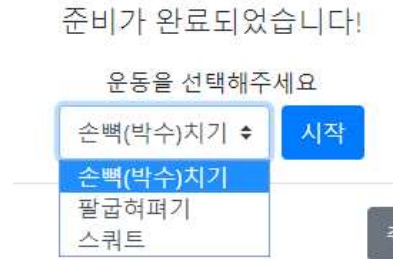
두 번째로, [그림 4-33]에서 ‘촬영하여 등록’시 Html Media API를 통해 디바이스 기기의 카메라 기능을 사용한다.



[그림 4-33] 운동 항목 선택
화면(하태용 외, 2020)

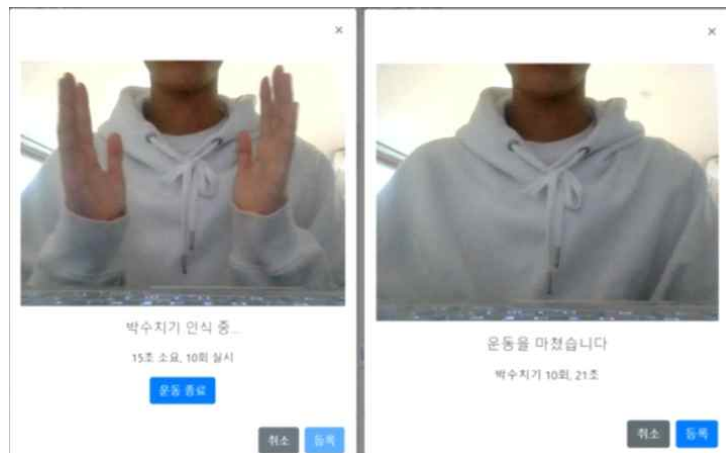
세 번째로, [그림 4-34]에서 사용자가 운동 항목을 선택하는 화면이다. 운동 종목을 선택하도록 드롭 다운 형식으로 구현하였다. Backend API를 통해

DB에 등록된 운동 목록을 불러와 출력시키는 방식으로 작동한다.



[그림 4-34] 데이터베이스에 저장된 운동 종목(하태용 외, 2020)

네 번째로, 시작 버튼을 누르면 [그림 4-35]에서 사용자의 자세를 추정하여 선택된 운동을 인식한다. 운동 진행 중에는 경과 시간과 운동 횟수를 실시간으로 출력한다. 운동 완료 시 등록 버튼을 통해 운동 등록 화면으로 넘어가며, 인식되었던 운동 결과가 해당 화면에 자동으로 입력된다.



[그림 4-35] 손뼉(박수)치기 인식 화면(하태용 외, 2020)

다섯 번째로, [그림 4-36]에서 운동한 시간을 바탕으로 손뼉(박수) 치기에 맞는 MET 값과 계산하여 소모한 열량을 출력한다. 자동으로 입력된 항목들 외에도 사용자가 직접 작성할 수 있으며, 최종적으로 등록 버튼을 누르면 Backend API를 통해 사용자 데이터베이스에 운동 기록이 저장된다.

[그림 4-36] 손뼉(박수)치기
소모 열량 (하태용 외, 2020)

[그림 4-36] 화면상의 소모 열량을 계산하면 다음과 같다. 먼저, 손뼉(박수) 치기 시작 시간과 종료 시간의 운동량이 31초이고 운동 종목에 대한 MET가 2이다. 거기에, 기초 체력인 몸무게 60 Kg 이므로 이것을 식에 대입하면 $60\text{ Kg} \times 2\text{ MET} \times 1,050\text{ cal/hr/kg} = 126\text{ Kcal/hr/kg}$ 가 된다. 여기서, 분모 환산 계수 60분으로 나누면 2.1 Kcal/hr/kg 가 된다. 결론적으로, 21초에 대한 실제적인 운동 칼로리를 계산하면 분모 환산 계수 60초로 나누면 0.73 Kcal/hr/kg 으로 화면에 표시된 결과값이 된다.

4) 스마트 헬스케어 시스템

스마트 헬스케어 시스템으로 다음과 같은 메뉴로 이루어진다.

[그림 4-37] 홈 화면은 사용자에게 서비스에 관한 프로모션을 제공하는 역할을 수행한다.

Smart Healthcare

스마트 헬스케어 대사 관리 서비스

회원 가입

로그인

대사 관리

대사 관리 안내 텍스트입니다.

정보 입력하기

식사 관리

식사 관리 안내 텍스트입니다.

식사 기록하기

운동 관리

운동 관리 안내 텍스트입니다.

운동 기록하기

[그림 4-37] 홈 화면

[그림 4-38] 대사 관리 화면에서는 사용자가 자신의 건강 정보를 입력하고 관리할 수 있으며, 입력된 정보를 바탕으로 기초대사량과 비만도가 계산된다.

대사 관리

기초대사량	건강 정보
1999 kcal	신장 170cm
비만도	체중 50kg
24.69	저밀도 콜레스테롤 100mg/DI
현재 체중 과체중	허리둘레 70cm
정상 체중 16.5-23	혈압 75-120
	중성지방 160mg/DI
	고밀도 콜레스테롤 30mg/DI
	공복혈당 110mg/DI

[정보 수정](#)

[그림 4-38] 대사 관리 화면

[그림 4-39] 식사 관리 화면에서는 사용자가 날짜별로 식사 기록을 확인하고 입력할 수 있으며, 이때 입력에는 음식 인식 기술을 활용할 수 있다.

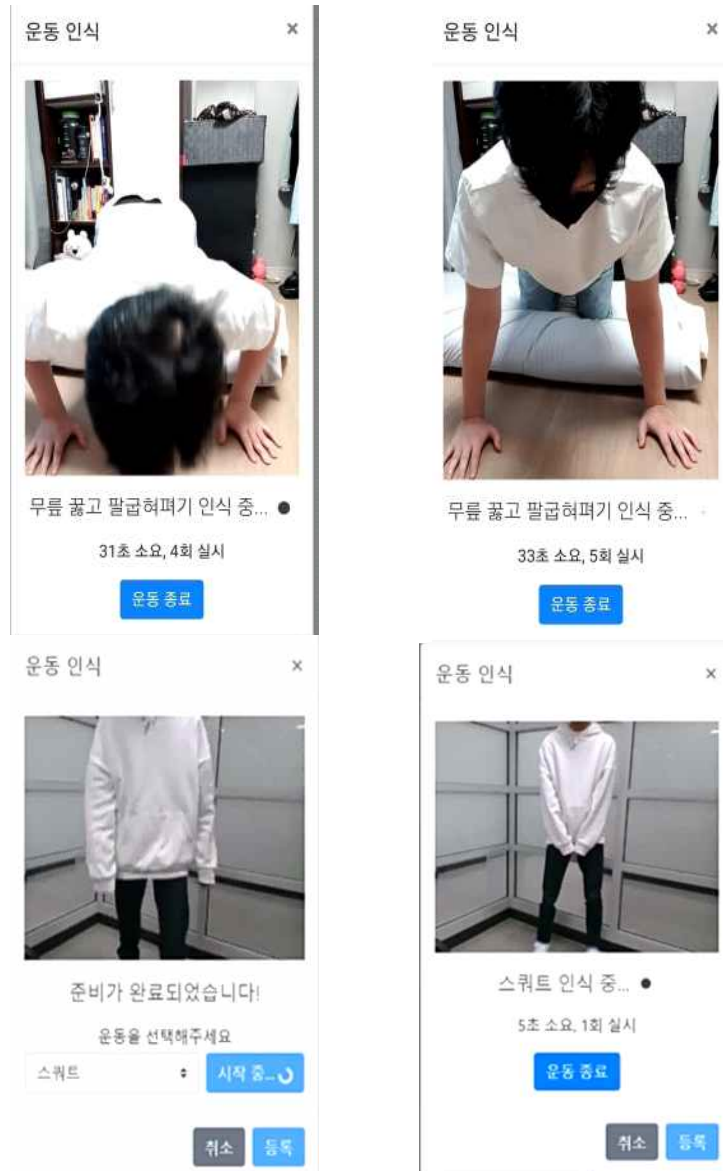
일시	음식	열량
오후 4시 20분	김치찌개, 밥, 삼겹살, 상추	600 kcal
오후 4시 20분	김치찌개, 밥	600 kcal
오후 4시 20분	김치찌개, 밥	600 kcal
오후 4시 20분	김치찌개, 밥	600 kcal

일시	종목	소모 열량
오후 4시 20분	말굽여머기 10회, 1분	230kcal
오후 4시 20분	스쿼트 19회, 1분 38초	300kcal

- 99 -

5) 스마트 헬스케어 추가 운동

[그림 4-34]의 운동 선택 종목에서 보이듯이 [그림 4-41]에서 손뼉(박수) 치기와는 별도의 추가적인 운동 종목으로 팔굽혀펴기와 스쿼트 동작 구현을 확인할 수 있다.



[그림 4-41] 추가 운동 동작 관리

제 2 절 스마트 헬스케어 도입에 따른 기대효과

1) 일반적 기대 효과

최근 IT 정보통신기술과 BT 생명공학기술이 융합한 건강관리 의료 서비스 기술의 발전과 환경 변화를 직시하고, 미래 전략적 대응 방안을 마련할 필요가 있다고 본다. 이에 본 연구에서 헬스케어 산업의 패러다임 변화를 인지하고 서비스는 예측을 통한 개인 맞춤형 보건의료로 탈바꿈 되고 있다. 이 거대한 변화는 ‘스마트 헬스케어’로 귀결이 될 것 같다. 이제는 기존 건강관리 의료 서비스 공급자와 소프트 웨어 및 인프라 공급자 그리고 스마트 헬스케어 기기 사업자 간 협업하면서 기존 건강관리 의료 서비스를 스마트화하게 하고 있다. 그만큼 스마트 헬스케어 시장은 거듭 성장할 것으로 예상하며, 스마트 디바이스는 PC와 달라서 사용자들에게 이동성을 제공하고, 더불어 GPS, 가속도 센서 등 다양한 센서가 기본적으로 장착되어 있어 이를 통한 다양한 스마트 헬스케어 서비스에 부가적으로 활용이 가능하다고 본다.

현재 의료진을 중심으로 개발하던 의료기기는 이제는 소비자의 관점에서 재조명되어 보다 유용한 기기로 탈바꿈하고 있다. 인공지능 기술의 확산으로 기존 사물인터넷과 같은 센서나 네트워크를 통해 동작하는 의료기와 스마트 기기의 융합으로 이제 모바일 의료기기라는 하나의 트렌드를 만들고 있다. 이제 단순한 건강관리에서부터 중증질환의 치료에 이르기까지 가능하게 하는 인공지능 기반의 스마트 헬스케어 시스템의 서비스가 중요한 역할을 할 것으로 기대된다.

2) 기술적 측면

스마트폰 등의 모바일 디바이스를 활용하여 개인의 건강 상태를 실시간으로 확인할 수 있는 콘텐츠에 대한 관심 증가로 이어지고 있으나, 관련 기술의 발전과 시장에서의 수용성 측면에서는 초기 시장 진입 단계로 지속적인 기술 개발과 콘텐츠의 개발 및 확산이 요구되고 있는 상황이다. 따라서 이러한 요

구 사항을 보다 효율적으로 해결하고자 하는 수단으로 인공지능 딥러닝 기술을 적용한 객체 인식 및 검출 알고리즘을 활용함으로써 가시적인 성과를 보이고 있다.

스마트 헬스케어는 다양한 전문 기술 및 서비스 영역 간의 융합을 통해 구현되는 영역으로 파트너십이라는 협력관계가 절대적이라 할 수 있다. 과거에 경쟁했던 기업과도 이제는 협업 체제를 구축해야 하며, 관련성이 없는 기업과도 협업을 요구하는 시기이다. 기존의 제약회사와 의료기관 그리고 의료기기 기업과 같은 전통 사업자 뿐만 아니라 이제는 웨어러블 디바이스 관련 사업과 모바일 소프트웨어 그리고 통신사들과 같은 사업자와의 협업이 요구되는 산업이다.

4차 산업혁명 기반 기술들이 다양한 산업에 적용되고 있는 가운데, 건강 의료 분야에도 상당한 속도로 확산되고 있다. 가장 주목되고 있는 기술은 인공지능 이미지 인식과 빅데이터 기술로써, 정밀 의료나 맞춤형 건강관리 의료 서비스뿐만 아니라 예측 및 예방 등 건강관리 의료 서비스의 변화를 만들어 가는 기술로 인식되고 있다.

3) 연구의 성과

인공지능 기반의 스마트 헬스케어는 건강관리 의료 서비스 제공 공간이 가정, 피트니스센터, 길거리 등 실생활 전 영역으로 확장되어 개인의 건강 정보가 평생 동안 축적됨에 따라 기존 의료 시스템과의 호환을 통해 스마트 헬스케어가 가능하리라 본다. 또한, 건강관리 의료 서비스가 단발성 질병 치료에서 이제는 평생 치료의 개념으로 확장하는 연구가 필요하리라 판단된다.

사용자의 편의성이 효과적으로 반영된 스마트 헬스케어 애플리케이션 개발을 위한 동작 연구로서 향후 스마트 헬스케어 시스템 구성에 도움을 주고자 한다. 이를 위한 구체적 개발 방법으로 인공지능의 딥러닝 뼈대 인식 기술 기반으로 실시간 자세 추정 기법을 적용하여 운동자에게 카운트 및 자세 교정 정보를 전달하고 실제 트레이너와 함께 운동하는 효과를 얻게 하여, 궁극적으로는 개인 맞춤형 운동 트레이너 서비스의 구현이 인공지능 피트니스 애플리케이션

플리케이션과 스마트 헬스케어 시장에서 우수한 경쟁력을 갖춘 응용 분야가 될 것으로 예상된다.

첫 번째, 데이터베이스에 저장된 개인별 운동에 대한 종목과 운동량 등 이력 정보를 분석하여 운동 소모 칼로리를 계산하고 운동 종목에 맞게 칼로리를 소비했는지를 비교 점검하여 사용자에게 분석한 결과를 전달 및 운동에 대한 건강 정보를 제공하여 최종적으로 의사 결정하게 하는 서비스를 제공한다.

두 번째, 활동량 정보인 운동 이력정보를 분석하여 사용자의 건강에 대한 사후 관리 및 예방을 목적으로 모니터링 서비스를 제공한다.

인공지능 기반 스마트 헬스케어 시스템을 활용한 산업의 발전과 융합을 선도하여 스마트 헬스케어 산업의 융복합 新시장을 개척하고 새로운 건강관리 서비스 시장에서 양질의 일자리를 창출할 수 있는 스마트 헬스케어가 주도하는 성과가 있길 바란다.

제 5 장 결 론

기존 헬스케어 관련 운동 Application은 단순히 따라하기형이거나 제품 기능만 활용한 단순하게 데이터를 보여주는 형태를 취하다 보니 실제 사용자의 동작을 분석하기 어려우며, 실제 사용자의 흥미 및 몰입을 유발하는데 도움이 되지 않는다는 단점을 가지고 있다.

또한, 기존 콘텐츠는 운동량에 대한 기초적인 정보를 제공하고는 있으나, 운동량에 대한 칼로리를 분석하여 제공하는 콘텐츠나 디바이스는 아직 출시되지 않고 있다. 이를 개선하기 위해 인공지능 자세 추정 기반의 스마트 헬스케어 시스템 개발을 통해 “언제나, 어디서나” 건강관리가 가능하게 하여 사용자의 운동 형태와 운동 시간을 감지하고 운동량에 비례하여 소모 열량을 계산하여 보여주는 기능이 있다면, 사용자 측면에서 실제적인 모니터링을 통한 동기 유발과 즐거움을 줄 수 있어 적절한 운동 효과를 볼 수 있을 것이다.

본 연구에서는 인공지능의 딥러닝 뼈대 인식 기술을 기반으로 하는 실시간 자세 추정 기법을 적용하여 운동자에게 카운트 및 자세 교정 정보를 전달하고 실제 트레이너와 함께 운동하는 효과를 얻게 하여, 궁극적으로는 개인 맞춤형 운동 트레이너 서비스의 구현이 인공지능 피트니스 애플리케이션과 스마트 헬스케어 시장에서 우수한 경쟁력을 갖춘 응용 분야가 될 것으로 예상된다.

다음과 같이 연구에 대한 결과를 요약하고자 한다.

첫 번째, LSP DataSet을 가지고 PoseNet으로 자세 추정 학습을 해 보았으며, 그 결과 눈, 코, 귀 KeyPoint에 대한 결과가 타 모델과 해석의 차이로 다소 차이가 나지만, 나머지 부위는 성능이 떨어지지 않음을 확인하였다.

두 번째, 손뼉(박수) 치기 동작의 운동량 분석에 인공지능 딥러닝 뼈대 인식 기술 기반의 실시간 자세 추정 기법을 적용하여 손뼉(박수) 치기라는 하나의 동작에 따른 운동량을 분석하는 프로그램을 구현하였다.

세 번째, 동작 인식과 운동 관리를 동시에 수행할 수 있는 시스템 구성을 위해 신체 활동 목록의 동작 유형별 신진대사 해당치(MET)를 활용한 운동관

리 애플리케이션을 구현하였고, 이를 활용하여 운동 종목과 시간 그리고 운동량의 분석 제시로 사용자의 실시간 운동 칼로리 소모량에 대한 계산 정보를 효과적으로 제공할 수 있을 것으로 기대된다.

특히, PoseNet은 헬스케어 플랫폼의 많은 기능 중 스마트 헬스케어 서비스를 개발하고자 하는 목적에 상당히 부합하는 모델이라 그 활용성이 매우 클 것이라 판단된다. 또한, 앞으로 건강에 대한 관심이 높아짐에 따라 비접촉 비즈니스와 관련된 서비스로 헬스케어 산업에 호재로 작용할 것으로 보이며, 추후 연구 과제로 음식 이미지 인식을 통한 식사관리를 추가한다면 스마트 헬스케어 대사 관리 시스템 발전을 가속화하는 촉매제가 될 것으로 보인다.

향후 연구에서는 본 스마트 헬스케어 시스템의 식사관리에 인공지능 음식 인식을 추가하여 스마트 헬스케어 시스템을 완성하고자 하다. Input으로 음식 관리 Output으로 운동 관리를 하여 지속적인 모니터링을 통해 인공지능 운동량 분석과 음식 인식 분야에 대한 보다 구체적인 정보를 제공하여 건강관리 서비스가 될 수 있도록 하는 지속적인 연구를 추진할 예정이다.

이번 연구를 통해 도출된 스마트 헬스케어 결과물들이 다른 연구에 밑거름이 되어 보다 더 발전된 연구를 할 수 있는 방향 제시를 하고, 해당 연구들이 스마트 헬스케어에 적용될 수 있는 밑거름이 되기를 기대한다.

참 고 문 헌

1. 국내문헌

- 고은지. (2018). 헬스케어는 IT 기업들의 새로운 성장 동력이 될 수 있을까. 서울: LG경제연구원.
- 구형일. (2018). 전기의 세계 ; 인공지능 및 딥러닝 동향. 『대한전기학회』, Vol. 67, No. 7, pp. 7-12.
- 국가기술표준원. (2018). 『2018 스마트헬스 10대 표준화 전략트렌드』, 서울: International Standards Ensure Positive Change.
- 국가기술표준원. (2016). 『표준기반 R&D로드맵』.
- 국립암센터. (2010). 『신체활동(Physical Activity)』.
- 김문구, 박종현, 오지선. (2018). 『인공지능 헬스케어 국내외 동향 및 활성화 방향』, 서울: IITP 정보통신기술진흥센터.
- 김문구. (2016). 『인공지능 헬스케어의 산업생태계와 발전방향』, 서울: 전자통신연구원.
- 김수범. (2018). 글로벌 헬스케어 주요 이슈, 『보건산업진흥원』, KHIDI Brief Vol. 261.
- 김영수, 정재진. (2019). e헬스케어 비즈니스모델에 관한 연구, 『벤처창업연구』, Vol. 14, No. 1, pp. 167-185.
- 김옥남. (2019). 『u-Healthcare가 다가온다』, LG ERI리포트.
- 김정곤, 이서진. (2016). 『주요국의 ICT 융합 의료산업 전략 및 시사점』, 서울: 대외경제정책연구원.
- 김주원. (2011). 스마트 공공보건의료 서비스 도입 방안, 『한국정보화진흥원』, IT 정책연구시리즈, 7호.
- 문세연, 윤영미, 한태화, 이상은, 장혁재, 송시영, 김형창. (2017). 디지털 헬스케어 서비스에 대한 일반인의 인식. 『디지털콘텐츠학회논문지』, Vol. 18, No. 4, pp. 621-629.

- 박선주, 유희숙, 안정은. (2014). 지속가능한 국가발전을 위한 창조비타민 해외전략 분석 -보건 의료 분야의 新, 『ICT융합전략 제2호』.
- 박재갑. (2010). 『한국인을 위한 신체활동 가이드라인 2010 심포지엄』, 서울: 국립중앙의료원.
- 배장은, 김승인. (2015). 여가만족도 향상을 위한 가상현실 게임과 u-Healthcare 기반의 피트니스 서비스 제안. 『디지털디자인학연구』, Vol. 15, No. 1, pp. 133-144.
- 보건복지부. (2014). 『2014년 지역사회 통합 건강증진사업 안내-신체활동 분야』. 서울: 보건복지부.
- 산업통상자원부. (2013). 『헬스케어 신시장 창출을 위한 정책연구』. 서울: 산업기반실 창의산업정책관실 바이오나노과.
- 산업통상자원부. (2015). 『스마트 헬스케어산업 활성화 방안』. 서울: 산업통상자원부.
- 삼성경제연구소. (2011). 『헬스케어 3.0 시대 요약』. 서울: 유진투자증권.
- 삼정 KPMG 경제연구원. (2018). 스마트 헬스케어의 현재와 미래 『Issue Monitor』, 제79호.
- 식품의약품안전처. (2018). 『스마트헬스케어 의료기기 기술표준 전략보고서』. 서울: 식품의약품안전처 식품의약품안전평가원.
- 안무업, 최기훈. (2009). 『원격진료, 이헬스 및 유헬스로의 발전과정』. 서울: 대한의사협회.
- 양윤준. (2016). 성인대상 국가건강검진 서식 및 상담 매뉴얼 개정 “운동 평가 및 상담”. 『대한임상건강증진학회』, 춘계학술대회.
- 유선길. (2018). “디지털 헬스케어 주요 기술 개발 및 발전 방향에 대한 연구”. 순천향대학교 대학원 박사학위논문.
- 유진투자증권. (2018). 『헬스케어와 4차 산업혁명』. 서울: 산업자료.
- 이난경, 이종옥, 황경태. (2014). 헬스케어 정보기술 분야의 연구 프레임워크 및 연구동향. 『정보화정책』, Vol. 21, No. 3, 가을호, pp. 3-32.
- 이미라, 김완수. (2009). 성인의 건강증진을 위한 신체활동 실천 모니터링 도구. 『대한임상건강증진학회지』, Vol. 9, No. 2.

- 이미영. (2011). 한국인 신체활동 가이드 라인개발을 위한 측량평가 측면에서의 준비. 『한국체육측정평가학회지』, Vol. 13, No. 3, pp. 17-31.
- 이민경. (2018). 『4차 산업혁명 시대의 헬스케어 동향과 시사점』. 서울: KDB 산업기술리서치센터.
- 이승철, 정해동, 박승태, 김수현. (2017). 딥러닝. 『소음진동』, Vol. 27, No. 3, pp. 19-25.
- 이진수. (2014). 디지털 헬스케어 플랫폼과 주요기업 동향. 『보건산업브리프』, Vol. 140, pp. 1-12.
- 전성표. (2018). “스마트융합 헬스케어 개인별 맞춤형 건강관리 케어시스템 개발에 관한 연구 : 근골격계 부상예방 및 개인 맞춤형 활동 모니터링 케어시스템을 탑재한 스마트 신발 개발을 중심으로”. 한성대학교 대학원 박사학위논문.
- 정보통신산업진흥원. (2017). 『스마트 헬스케어 서비스 분야 도입사례 분석』. 서울: 정보통신산업진흥원.
- 정성희. (2015). 고령자·만성질환자의 건강관리를 위한 헬스케어서비스 활용과 과제. 『고령화리뷰』, Vol. 3, No. 2, pp. 25-40.
- 정영호, 고숙자, 정지영. (2012). 『건강증진이 경제에 미치는 효과에 관한 연구』. 서울: 한국보건사회연구원.
- 질병관리본부. (2013). 『2012국민건강통계』. 서울: 질병관리본부.
- 질병관리본부. (2016). 『정책연구용역사업 최종결과보고서, 국민건강영양조사 신체활동측정도구 결과 활용 방안 연구』. 서울: 질병관리본부.
- 최동운, 송행숙, 김수용. (2009). USN 기반 u-헬스케어 에이전트 시스템의 활동량 계산 알고리즘. 『한국엔터테인먼트산업학회』, 학술대회 논문집, Vol. 3, No. 1, pp. 107-110.
- 최윤희, 정혜린. (2015). 모바일 세계가 주목하는 미래 스마트 헬스케어 산업. 『e-KIET 산업경제정보』, 제609호, pp. 1-12.
- 하나금융경영연구소. (2016). 『국내외 헬스케어 산업 현황과 전망』. 서울: 하나금융경영연구소.
- 하태용, 이후진. (2019). 인공지능 기반의 행동인식을 통한 개인 운동 트레이

- 너 구현의 방향성 제시. 『Journal of the Korea Convergence Society』, Vol. 10, No. 6, pp. 235-242.
- 하태용, 이후진. (2020). 인공지능 기반의 자세 추정에 따른 모바일 헬스케어 동작 연구 . 『대한전자공학회』, Vol. 57, No. 1, pp. 63-69.
- 하태용, 이후진. (2020). 인공지능 기반의 스마트 헬스케어 운동관리를 위한 애플리케이션 구현. 『대한전자공학회』, Vol. 57, No. 6, pp. 44-51.
- 한국보건사회연구원. (2016). 『정보통신기술(ICT)과 보건의료서비스 융합 활성화를 위한 정책과제』. 서울: 한국보건사회연구원.
- 한국보건사회연구원. (2018). 『미래형 보건의료기술의 포괄적 성과 제고를 위한 정책과제』. 서울: 한국보건사회연구원.
- 한국보건산업진흥원. (2018). 『인공지능(AI) 기반 의료기기 현황 및 이슈(1)』. 서울: Khidi.
- 한국정보화진흥원. (2011). 『헬스케어 新시장 창출을 위한 정책 연구』. 서울: 한국정보화진흥원.
- 한석민. (2018). Deep Learning 비전기술동향(전기의세계). 『대한전기학회』, Vol. 67, No. 7, pp. 13-17.
- 황원식, 최윤희. (2017). 『4차 산업혁명 시대의 신성장동력, 스마트 헬스케어 산업』. 서울: 산업연구원.
- <https://ko.wikipedia.org>. (2018).
- IT동아. (2009). <https://it.donga.com/28617/>.
- IT동아. (2019). 『4차 산업혁명과 직업의 미래 10. 헬스케어의 현재와 미래사회』. 서울: IT동아.
- KISTI. (2016). 『Market Reort』. 서울: KISTI.
- NIPA. (2017). 『스마트 헬스케어 서비스 분야도입사례 분석집』. 서울: NIPA.
- TTA. (2018). 『표준화 전략맵-스마트 헬스』. 서울: TTA.

2. 국외문헌

- A. K. Triantafyllidis, V. G. Koutkias, and I. Chouvarda. (2013). A Pervasive Health System Integrating Patient Monitoring, Status Logging and Social Sharing. *Biomedical and Health Informatics, IEEE Journal*, Vol. 17, No. 1, pp. 30–37.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton. (2012). Imagenet Classification with Deep Con-volutional Neural Networks. *In: Advances in Neural Information Processing Systems*. pp. 1097–1105.
- A. M. McDaniel, D. L. Schutte, and L. O. Keller. (2008). Consumer Health Informatics: From Genomics to Population Health. *Nursing Outlook*, Vol. 56, No. 5, pp. 216–223.
- A. Newell, K. Yang, and J. Deng. (2016). Stacked Hourglass Networks for Human Pose Estimation. *Computer Vision – ECCV 2016*, pp. 483–499.
- A. Toshev, and C. Szegedy. (2013). Human Pose Estimation via Deep Neural Networks. *In CVPR*.
- ACSM. (2000). www.acsm.org.
- Allied market research. (2018). *Forecast, 2018–2025 Artificial Intelligence*
- B. C. Corbin. (1996). *Concept of Physical Fitness*.
- B. Ives, S. Hamilton, and G. B. Davis. (1980). A Framework for Research in Computerbased Management Information Systems. *Management Science*, Vol. 26, No. 9, pp. 910–934.
- B. M. Choi and T. R. Kim. (2015). Global Health and Humman Development with Growth of Digital Health Care Industry, *Global Social Welfare Review*, Vol. 5, No. 2, pp. 95–120.
- C. C. Poon and Y. T. Zhang. (2008). Perspectives on High Technologies for Low-Cost Healthcare. *Engineering in Medicine and Biology*

- Magazine, IEEE*, Vol. 27, No. 5, pp. 42–47.
- C. Tudor-Locke, J. E. Williams, and J. P. Reis. (2004). Utility of Pedometers for Assessing Physical Activity. *Construct Validity. Sports Medicine*, Vol. 34, pp. 281–291.
- C. Tudor-Locke, Y. Hatano, and R. P. Pangrazi. (2008). Revisiting “How many Steps are Enough?”. *Medicine & Science in Sports & Exercise*, Vol. 40, No. 7 Suppl, pp. S537–S543.
- C. Szegedy, W. Liu, and P. Sermanet. (2015). Going Deeper with Convolutions. *In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9.
- Computerphile. (2016). Resizing Images. [Video]. Available at: https://youtu.be/AqscP7rc8_M [Accessed 3 Jul. 2018].
- E. Insafutdinov, L. Pishchulin, and B. Andres. (2016). DeepCut: *Joint Subset Partition and Labeling for Multi Person Pose Estimation In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- E. Nageba, P. Rubel, and J. Fayn. (2012). Context-aware mobile services adaptation to dynamic resources. Application to mHealth. *In Mobile and Wireless Networking (iCOST), 2012 International Conference on Selected Topics*.
- Frost & Sullivan. (2015). *Cognitive Computing and Artificial Intelligence Systems in Healthcare*.
- Frost & Sullivan. (2018). *Global Healthcare Market Predictions Revealed : Growth Opportunities, Technology, and Trends*, 2017.12.
- G. Thomase and K. Gunningf. (2013). Introducing a nationally shared electronic patient record: Case study comparison of Scotland, England, Wales and Northern Ireland. *International Journal of Medical Informatics*, pp. e125–e138.
- H. Jeong, S. Park, S. Woo, and S. Lee. (2016). Rotating Machinery

- Diagnostics Using Deep Learning on Orbit Plot Images, *Procedia Manufacturing*, Vol. 5, pp. 1107–1118.
- <https://blog.openai.com/faulty-reward-functions/>.
- https://github.com/rwightman/posenet-python/blob/master/posenet/convert_er/wget.py.
- <https://medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow-js-7dd0bc881cd5>
- <https://sites.google.com/site/compendiumofphysicalactivities>.
- <http://www.splex.co.kr/wp-content/uploads/2015/05/smarthealth>.
- I. Cubić, I. Markota, and I. Benc. (2010). Application of session initiation protocol in mobile health systems., *In MIPRO, Proceedings of the 33rd International Convention*, pp. 367–371.
- IDC “*IDC Worldwide Black Book: 3rd Platform Edition*”.
- J. Carreira, P. Agrawal, and K. Fragkiadaki. (2015). *Human Pose Estimation with Iterative Error Feedback*.
- J. G. Park. (2010). Physical Activity Guideline for Korean. *Physical Activity Guideline Symposium, National medical center*.
- J. K. Ko. (2000). MD, Exercise Testing, *Korean Circulation*, J; Vol. 30, No. 5, pp. 653–674.
- J. Long, E. Shelhamer, and T. Darrell. (2015). Fully Convolutional Networks for Semantic Segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440.
- J. Tompson, A. Jain, and Y. LeCun. (2014). Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In: *Advances in Neural Information Processing Systems*, pp. 1799–1807.
- J. Tompson, R. Goroshin, and A. Jain. (2015). Efficient Object Localization Using Convolutional Networks. *In CVPR*.

- J. Clark, and D. Amodei. (2016). *Faulty Reward Functions in the Wild*.
- K. He, X. Zhang, and S. Ren. (2015). Deep Residual Learning for Image Recognition. *IEEE Conference on 2015*.
- K. He, X. Zhang, and S. Ren. (2016). Deep Residual Learning for Image Recognition. *Computer Vision and Pattern Recognition, CVPR* 2016.
- L. Pishchulin, E. Insafutdinov, and S. Tang. (2016). Deepcut: *Joint Subset Partition and Labeling for Multi Person Pose Estimation*. In *CVPR*.
- L. V. Dyk. (2014). A Review of Telehealth Service Implementation Frameworks. *International Journal of Environmental Research and Public Health*, pp. 1279–1298.
- M. Andriluka, L. Pishchulin, and P. Gehler. (2014). 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 3686–3693.
- M. Eichner and V. Ferrari. (2010). We are family: Joint Pose Estimation of Multiple Persons. In *ECCV*.
- M. ElHelw, J. Pansiot, and D. McIlwraith. (2009). An Integrated Multi-Sensing Framework for Pervasive Healthcare Monitoring. In *Pervasive Computing Technologies for Healthcare, PervasiveHealth 2009, 3rd International Conference on , IEEE*, pp. 1–7.
- M. J. Lamonte and B. E. Ainsworth. (2001). Quantifying Energy Expenditure and Physical Activity in the Context of Dose Response. *Med. Sci. Sports Exerc.*, Vol. 33, No. 6, pp. S370–S378.
- M. Jette, K. Sidney, and G. Blumchen. (1993). Metabolic Equivalents (METs) in Exercise Testing, Exercise Prescription, and Evaluation of Functional Capacity, *Clinical Cardiology*, Vol. 16, Issue 3.

- M. Kochen. (1985). Are MIS Frameworks Premature?, *Journal of Management Information Systems*, pp. 92–100.
- M. Volk, J. Sterle, and U. Sedlar. (2015). Safety and Privacy Considerations for Mobile Application Design in Digital Healthcare, *International Journal of Distributed Sensor Network*, Vol. 11, 2015.
- Markets and Markets. (2016). *Artificial Intelligence(AI) Market by Technology, Application and Geography–Global Forecast to 2020*.
- Michigan Vision and Learning Lab. (2016). *Stacked Hourglass Networks for Human Pose Estimation (Training Code)*. Available at: <https://github.com/umich-vl/pose-hg-train>.
- N. Mohammadzadeh and R. Safdari. (2014). Patient Monitoring in Mobile Health : Opportunities and Challenges. *Med Arh*, Vol. 68, No. 1, pp. 57–60.
- Notice to Readers Publication of Surgeon General's Report On Physical Activity and Health. (1996). Surgeon General Report on Physical Activity, *Morbidity and Mortality Weekly Report*, Vol. 45, No 27, pp. 591–592.
- O. Postolache, P. S. Girão, and M. Ribeiro. (2011). Enabling telecare Assessment with Pervasive Sensing and Android OS Smartphone, *IEEE International Symposium on Medical Measurements and Applications*, pp. 288–293.
- S. Adibi. (2013). Biomedical Sensing Analyzer (BSA) for Mobile–Health (mHealth)–LTE. *IEEE Journal of Biomedical and Health Informatics* , Vol. 18, No. 1, pp. 345–351.
- S. E. Wei, V. Ramakrishna, and T. Kanade. (2016). Convolutional pose machines. *Computer Vision and Pattern Recognition (CVPR)*, IEEE Conference on 2016.
- S. Ioffe and C. Szegedy. (2015). Batch Normalization: Accelerating Deep

Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning*.

- S. Johnson and M. Everingham. (2010). Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. *In BMVC'10*.
- S. N. Yun, T. W. Jeon, and H. J. Lee. (2002). Survey of Physical Activities and Exercise Programs of the Public Health Centers, *J Korea Community Health Nursing Academic Society*, Vol. 16, No. 1, pp. 148–164.
- S. Ren, K. He, and R. Girshick. (2009). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *In NIPS'2015*.
- S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach. NJ, USA: Prentice Hall Press, 2009.
- T. Mettler and D. A. Raptis. (2012). What Constitutes the Field of Health Information Systems? Fostering a Systematic Framework and Research Agenda. *Health informatics journal*, Vol. 18, No. 2, pp. 147–156.
- T. Mettler and M. Eurich. (2012). A “Design Pattern”-based Approach for Analyzing Ehealth Business Models. *Health Policy and Technology*, Vol. 1, No. 2, pp. 77–85.
- T. M. Wood and W. Zhu. (2006). Measurement Theory and Practice in Kinesiology. Champaign, IL: Human Kinetics.
- T. Pfister, J. Charles, and A. Zisserman. (2015). Flowing Convnets for Human Pose Estimation in Videos. *In ICCV*.
- U.S. Department of Health & Human Services. (2008). *Physical Activity Guidelines for American*. Retrieved November 1, 2011, from <http://www.health.gov/paguidelines/>.
- U.S. Surgeon General. (2005). *The Surgeon General's call to Action to*

Improve the Health and Wellness of Persons with Disabilities.
<http://www.surgeongeneral.gov/library/disabilities/>.

- US Department of Health and Human Services. (1996). Physical Activity and Health: a Report of the Surgeon General. Atlanta, Georgia: US Department of Health and Human Services. *Public Health Service, CDC, National Center for Chronic Disease Prevention and Health Promotion.*
- W. L. Haskell, I . M. Lee, and R. R. Pate. (2007). Physical Activity and Public Health: *updated Recommendation for Adults from the American College of Sports Medicine and the American Heart Association, Circulation*, Vol .116, No. 9, pp. 1081–1093.
- W. Ouyang, X. Chu, and X. Wang. (2014). Multi-source Deep Learning for Human Pose Estimation, *In CVPR*.
- X. Chen and A. Yuille. (2014). Articulated Pose Estimation by a Graphical Model with image Dependent Pairwise Relations. *In NIPS*.
- Y. LeCun and Y. Bengio. (1995). Convolutional Networks for Images, Speech, and Time Series, *The Handbook of Brain Theory and Neural Networks*, Vol. 3361, No. 10.
- Y. LeCun, Y. Bengio, and G. Hinton. (2015). Deep Learning, *Nature*, Vol. 521, pp. 436–444.
- Y. LeCun, L. Bottou, and Y. Bengio. (1998). Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324.

ABSTRACT

A Study on the Smart Healthcare through Human Pose Estimation Based on Artificial Intelligence

Ha, Tae-Yong

Major in Smart Convergence Product

Dept. of Smart Convergence Consulting

The Graduate School

Hansung University

Smart healthcare, which can receive health care services anytime, anywhere by using various advanced information and communication technologies, is emerging. There have been many attempts to converge with IT technology in the healthcare field for a long time, and recent changes are not new. With the advent of smart phones, it is becoming possible to provide health care services that are free from temporal and spatial constraints, and the quality of new health care services is converged by cutting-edge analysis technologies such as Artificial Intelligence(AI) and Big Data. The level is improving.

Although the development of artificial intelligence technology, the future healthcare service can analyze and learn a lot of health information related to the body by itself to predict when the disease will develop. In addition, it is expected to be able to help prevent disease manifestation through personalized diagnosis and lifestyle information.

Artificial intelligence-based smart healthcare can expand the space for

providing health care services to all areas of real life such as homes, fitness centers, and streets. As the personal health information accumulates throughout life, smart healthcare is possible through compatibility with the existing medical systems. Moreover, it is considered that medical services are needed to be expanded from the treatment of single disease to the concept of lifelong treatment.

In order to enable health management “anytime, anywhere” through the development of a smart healthcare system based on AI pose estimation in this study, Since it takes the form of simply following the existing healthcare-related exercise application or simply showing the data of the existing product, it does not help the interest and immersion of real users. To improve this, if there is a function that calculates and shows the calories burned in proportion to the amount of exercise by detecting the type of exercise and the movement of the user, the user can see the exercise effect through more realistic monitoring.

In this study, we learned the pose estimation with PoseNet using LSP DataSet as a deep learning(DL) based learning through pose estimation. Compared with other pose estimation main models, the results for the eyes, nose, and ear KeyPoints are somewhat different due to the differences in interpretation from other models. It was confirmed that the performance did not significantly drop in all other parts. Thus, PoseNet was selected as the optimal model, and it was verified through the comparison analysis with the existing pose estimation algorithms. We set the optimal exercise model called hand clapping(applause), which is the basic exercise motion for recognition, and implemented the motion algorithm in JavaScript. A smart healthcare exercise management system was implemented by grasping the user's motion and exercise amount and using the corresponding calories in the physical activity list for the exercise amount. In addition, through the implementation of the smart

healthcare system, it was also confirmed that the user's calories burned and that existing devices could be used to save time and money.

Based on the results of this study, we can expect that it will be possible to utilize various health management services and systems of smart healthcare using artificial intelligence motion recognition.

【Key word】 AI, Health Care, CNN, MET, Deep Learning