

# 탁구 로봇을 위한 빠른 자세 분류 시스템 개발

## Development of Fast Posture Classification System for Table Tennis Robot

진성호<sup>1</sup>·권영우<sup>1</sup>·김윤정<sup>1</sup>·박미영<sup>1</sup>·안재훈<sup>1</sup>·강호선<sup>2</sup>·최지욱<sup>2</sup>·이인호<sup>†</sup>  
Seongho Jin<sup>1</sup>, Yongwoo Kwon<sup>1</sup>, Yoonjeong Kim<sup>1</sup>, Miyoung Park<sup>1</sup>, Jaehoon An<sup>1</sup>,  
Hosun Kang<sup>2</sup>, Jiwook Choi<sup>2</sup>, Inho Lee<sup>†</sup>

**Abstract:** In this paper, we propose a table tennis posture classification system using a cooperative robot to develop a table tennis robot that can be trained like a real game. The most ideal table tennis robot would be a robot with a high joint driving speed and a high degree of freedom. Therefore, in this paper, we intend to use a cooperative robot with sufficient degrees of freedom to develop a robot that can be trained like a real game. However, cooperative robots have the disadvantage of slow joint driving speed. These shortcomings are expected to be overcome through quick recognition. Therefore, in this paper, we try to quickly classify the opponent's posture to overcome the slow joint driving speed. To this end, learning about dynamic postures was conducted using image data as input, and finally, three classification models were created and comparative experiments and evaluations were performed on the designated dynamic postures. In conclusion, comparative experimental data demonstrate the highest classification accuracy and fastest classification speed in classification models using MLP (Multi-Layer Perceptron), and thus demonstrate the validity of the proposed algorithm.

**Keywords:** Motion Classification, Feature Extraction, Dimension Reduction, MLP

### 1. Introduction

탁구는 인체 관절의 복잡한 공간적 움직임이 수반되며 다양한 스트로크 자세가 존재하는 가장 인기 있는 스포츠 중 하나이다. 최근 탁구에 대한 관심과 인기가 높아짐에 따라 선수 기량 향상 방법론에 대한 연구가 활발하다. 선수 기량 향상을

위해서는 많은 훈련량과 훈련 시 실전과 같은 상황이 요구된다. 이를 위해 보통 전문가 또는 코치와 훈련을 진행하지만, 이 방법에는 전문 인력에 따른 비용과 한정된 시간이 존재하게 된다. 따라서 항상 훈련이 가능한 자동화된 지능형 훈련 시스템이 요구되는 상황이다.

흔히 탁구 경기 중에서는 상대방의 공간적 움직임에 대한 파악을 통해 미리 상대방의 스트로크를 예측하곤 한다. 이러한 공간적 특성을 로봇이 미리 학습한다면 사용자의 동작을 세밀하게 구분하여 대응할 수 있을 것이다. 최근에는 라켓에 센서를 부착하거나, 사람의 신체에 웨어러블 센서를 부착한 연구가 활발하다<sup>1,2</sup>. 깊이 카메라나 관성 센서 등의 감지 센서를 서로 융합하여 활용하기도 한다<sup>3-5</sup>. 하지만 이러한 기존의 대부분의 연구는 부착된 센서에만 초점을 맞추어져 있다. 신체나 라켓에 부착된 IMU 센서의 정보를 이용하여 초보 선수의 스트로크의 패턴을 분석하고, 이를 전문가의 패턴과 비교하여 기량 향상을 꾀하는 방식이다. 카메라로 획득한 신체의 생체역학에 대한 정보와 라켓에 부착된 IMU 센서를 융합한 연구도 진행 중이다<sup>6</sup>.

Received : Jun. 24. 2022; Revised : Jul. 27. 2022; Accepted : Aug. 16. 2022

\* This work was supported by Police-Lab 2.0 Program (www.kipot.or.kr) funded by the Ministry of Science and ICT (MSIT, Korea) & Korean National Police Agency (KNPA, Korea) [Project Name: Development and demonstration of unmanned patrol robot system for local police support / Project Number: 210121M05]

\*\* This work has supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. 2021R1C1C1009989)

1. MS Student, Department of Electrical and Electronic Engineering, Pusan National University, Busan, Korea (seongho1696, aqedfg2000, yunjeong1696, aldud452, dkswoons46@pusan.ac.kr)

2. Ph.D Student, Department of Electrical and Electronic Engineering, Pusan National University, Busan, Korea (hosun7379, jiwook7379@pusan.ac.kr)

† Assistant Professor, Corresponding author: Dept of Electronics Engineering, Pusan National University, Busan, Korea (inholee8@pusan.ac.kr)

이는 보통 본인의 동작 패턴을 더 잘 이해하고 올바른 스트로크를 숙달하기 위해 이용한다. 센서를 융합하여 사용하면 개별적으로 사용할 때 보다 더 좋은 결과를 얻을 수 있다. 하지만 개발 및 계산 측면에서는 확연한 복잡성을 띠게 된다. 또한 일반적인 환경에서 고가의 센서를 통한 기존의 인식 시스템을 사용하게 되면 이에 맞는 고사양의 환경을 구축해야 한다는 어려움이 존재한다<sup>8)</sup>. 때문에 시스템의 사용자가 일부 인원에만 국한되거나 대중적으로 사용되는데 제한이 생길 수가 있다. 그리고 인간의 행동을 인식하는데 대부분은 Segmentation 기술을 이용한다. 하지만 Segmentation 기술은 흔히 실시간으로 적용하기에는 기술적 한계가 존재한다고 알려져 있다. 게다가 행동 분류 시에 세밀한 동작보다는 큰 동작에 대한 분류에 용이하며, 비슷한 동작의 분류에는 어려움을 나타낸다. 결론적으로 널리 사용되는 Segmentation 기술은 빠른 분류에 부적합하다.

본 논문에서는 실전과 같은 훈련이 가능한 지능화된 훈련 시스템 개발을 목적으로 한다. 이를 위해 몇 가지 요구 조건을 갖춘 로봇을 필요로 한다. 가장 중요한 요건은 공이 오는 걸 잘 보고 잘 칠 수 있어야 하고, 이에 빠르게 인식이 가능해야 한다. 또한 빠른 관절 구동 속도를 요구하며 더 넓은 워크스페이스에서 동작하기 위해 높은 자유도가 요구된다. 현재 탁구 로봇 분야에서 가장 널리 사용되는 로봇은 델타 로봇이다. 델타 로봇의 가장 큰 특징은 아주 빠른 관절 구동 속도를 가지고 있다는 점이다. 하지만 4자유도의 낮은 자유도로 인해 본 논문에서 추구하는 실전과 같은 훈련 시스템을 기대하기는 어렵다. 사람과 같이 탁구공에 스핀을 주는 등의 행위에는 보다 높은 자유도를 필요로 하며 이상적으로는 휴머노이드 로봇을 사용해야 한다.

하지만 휴머노이드 로봇을 사용하게 되면 천문학적 비용은 물론, 장기간의 개발 시간이 요구되며 기계적, 소프트웨어적으로 해결해야 할 요소들이 다수 존재한다. 따라서 가장 현실적인 방안은 협동 로봇을 사용하는 방법일 것이다. 엔드 이펙터에 자유도가 존재하므로 스핀과 같은 인간과 유사한 타구가 가능하며 휴머노이드 로봇을 사용 시의 단점을 많이 상쇄시킬 수 있다. 그러나 협동 로봇은 아주 느린 관절 구동 속도를 가지고 있다. 따라서 본 논문에서는 이러한 느린 관절 구동 속도를 극복하기 위해 컴퓨터 비전 베이스로 상대방의 포즈를 빠르게 분류하고자 한다.

이에 따라 본 논문에서는 협동 로봇의 느린 제어 속도를 극복하기 위해, 탁구 라켓이나 사용자의 몸에 부착되는 어떠한 센서 없이 저가의 2D카메라 한 대만을 사용한 모션 분류 시스템을 제안한다<sup>9)</sup>. 이는 높은 하드웨어 사양을 요구하지도 않고 센서리스 시스템이기 때문에 접근성과 대중성에 용이하다는 장점이 있다<sup>10,11)</sup>. 본 연구에서는 훈련 시스템이 일부 사용자

에만 국한되는 Domain gap 문제 해결을 위하여 인체 관절에 대한 2D 관절 정보를 추출하고 다섯 가지 공간적 특징을 추출하였다<sup>12)</sup>. 또한, 추출된 고차원 데이터를 저차원의 잠재 공간으로 축소시켜 차원의 저주 문제로 인한 분류의 어려움을 최소화하였다<sup>13,14)</sup>. 그리고 하이퍼 파라미터가 수정된 MLP Classifier 모델을 사용하여 예측하고자 하는 자세에 대한 분류를 진행하였다<sup>15)</sup>.

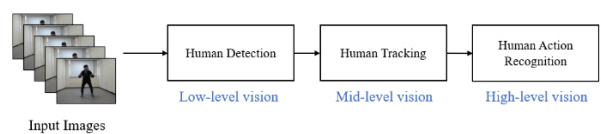
최종적으로 제안하는 알고리즘의 유효성 입증에 위해 동일한 실험 환경에서 보편적으로 사용되는 비교 모델과 비교 실험을 진행하였다<sup>16)</sup>. 비교 모델로 SVM Classifier와 KNN Classifier를 선정하였으며 MLP Classifier와 마찬가지로 하이퍼 파라미터를 수정하여 인식 및 분류에 최적의 모델을 생성하였다. 실험 데이터를 통해 Confusion Matrix를 사용하여 인식 정확도를 비교하고, 한 영상에 대한 자세 인식 속도 비교를 통해 제안하는 알고리즘의 유효성을 입증하였다.

2장에서는 System block diagram을 통해 제안하는 시스템의 전체적인 흐름과 시스템의 구조를 확인한다. 3장에서는 제안하는 시스템에 대한 세부적인 방법론에 대해 설명하며 4장에서는 알고리즘의 유효성 입증에 위한 평가 지표를 제시하고 이를 통해 시스템의 타당성을 확인한다. 마지막으로 5장에서는 본 연구의 결론 및 향후 연구 계획을 제시한다.

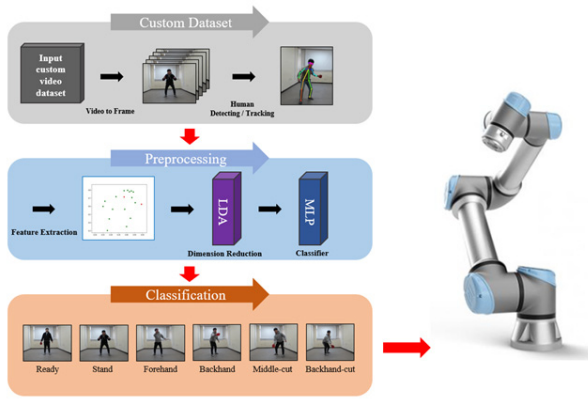
## 2. System Overview

일반적으로 인간의 행동 인식을 위해 [Fig. 1]과 같은 프레임 워크를 거치게 된다<sup>17)</sup>.

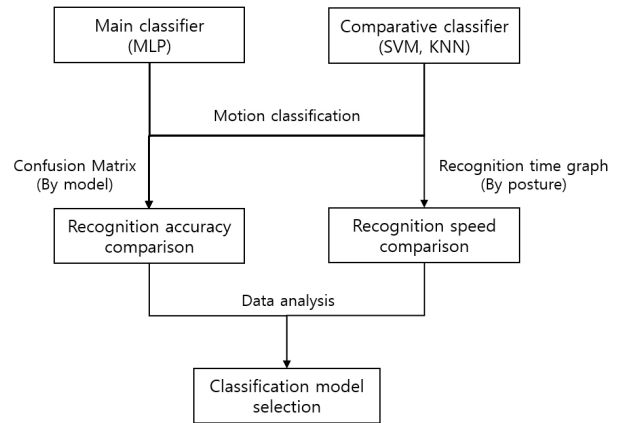
먼저 Input video를 Image data로 변환하고 이를 Low-level의 비전 작업으로 인체를 감지하게 된다. 다음으로 Mid-level의 비전 작업으로 인체를 추적한 뒤, High-level의 작업으로 인간의 행동을 인식한다<sup>18-22)</sup>. 본 논문에서는 OpenPose 알고리즘을 이용하여 인체 감지 및 추적을 수행하였다. 행동 인식을 위해서 자체적으로 생성한 Classifier 모델을 이용하여 각 Motion에 대한 분류를 수행하고 이를 통해 사람 행동 인식을 수행하였다<sup>23-26)</sup>. 시스템의 전체적인 알고리즘은 [Fig. 2]와 같다. 원하는 자세 분류를 위해 분류할 원하는 각각의 자세에 대한 영상을 촬영하고 이를 프레임별로 이미지 파일로 변환한다. 변환된 이미지 파일에 대하여 OpenPose 알고리즘을 사용해 인체 감지 및 추적을 수행하고<sup>27,28)</sup>, 수행 결과에 따른 인체



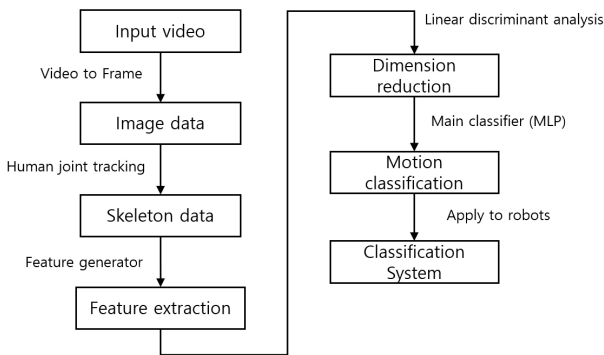
[Fig. 1] General framework for action recognition



[Fig. 2] Overall system algorithm



[Fig. 4] Evaluation block diagram



[Fig. 3] System block diagram

관절의 정보가 담긴 데이터를 획득한다<sup>[15]</sup>. 획득한 데이터는 각 관절에 대한 (x, y) 좌표와 클래스의 라벨 정보이다. OpenPose의 Skeleton 데이터를 획득함으로써 특정 인물에만 인식이 편향되는 문제를 해결함과 동시에, 여러 가지 센서를 융합한 기존의 인식 방법에 비해 높은 FPS 획득이 가능해진다.

추출한 정보를 이용하여 인체의 공간적 특징을 추출하였다. 추출한 특징은 원시 관절 위치 좌표, 관절 사이 길이, 관절 사이 각도, 상체의 방향 벡터, 손목의 방향 벡터로 총 다섯 가지이다. 이러한 특징을 통해 Domain gap 문제를 해결하고, 단순히 좌표만 추출했을 때보다 더 세밀한 자세 인식이 가능해진다. 추출한 다섯 가지의 특징은 High-Dimension의 데이터이다. 이를 그대로 Classifier를 통해 분류하고자 한다면 차원의 저주 문제로 인해 원하는 결과를 얻지 못할 가능성이 높다. 차원이 증가될수록 데이터 사이의 공간이 넓어짐에 따라 모델의 성능이 저하되기 때문이다. 이를 위해 지도 방식의 차원 축소 기법인 LDA (Linear Discriminant Analysis)를 이용하여 차원 축소를 수행한다<sup>[13]</sup>. 축소된 데이터는 기존의 데이터를 가장 잘 설명할 수 있는 잠재공간인 Low-Dimension의 데이터이며, 원하는 자세 분류를 위해 Classifier를 통하여 이 데이터에 대한 분류를 진행한다. Classifier로 MLP (Multi-Layer Perceptron)

을 사용하였는데<sup>[28]</sup>, 실험적으로 하이퍼 파라미터를 수정하여 가장 성능이 좋은 모델을 생성하였다. 비교 모델로 SVM (Support Vector Machine)과 KNN (K-Nearest Neighbors) Classifier를 선정하였고, 마찬가지로 각각의 모델에 대한 실험적으로 하이퍼 파라미터를 수정하여 가장 성능이 좋은 모델을 생성 후 MLP 생성 모델과 비교하였다<sup>[6]</sup>. 실험 및 평가는 각각의 생성된 Classifier 모델을 통해 분류하고자 하였던 자세에 대한 분류 정확도와 인식 속도를 측정하였다. 이를 통하여 제안하는 시스템에서 MLP Classifier 모델에 대한 가장 적합한 하이퍼 파라미터를 제시하고 평가 지표들 통해 제안하는 알고리즘에 대한 유효성을 입증한다. [Fig. 3]은 분류 시스템 개발을 위한 과정을 설명하는 시스템 블록 다이어그램이고, [Fig. 4]는 분류 시스템에 적용된 분류 모델의 정확도 및 분류 속도에 대한 평가 과정을 나타내는 블록 다이어그램이다.

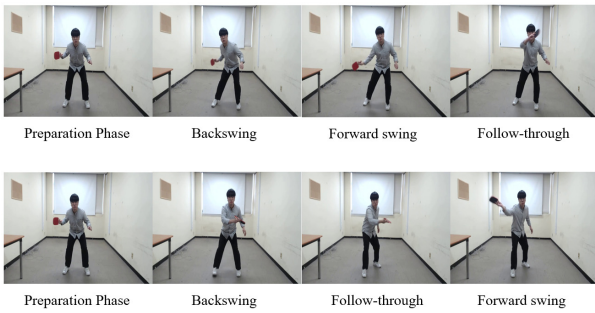
### 3. Methodology

#### 3.1 Custom Dataset

원하는 자세의 분류를 위해서 학습 데이터 수집은 가장 기초적인 과정이다. 학습을 위한 Train Set과 평가를 위한 Test Set 모두 직접 Customizing 한 데이터를 사용하였고, 적절한 비율로 나누어 학습에 사용하였다. 이번 장에서는 분류 자세에 대한 선정 기준과 선정된 자세에 대한 Training 과정을 설명한다.

##### 3.1.1 Posture for Classification

Stroke의 네 가지 기본 기술은 'Forehand Counter hit', 'Backhand Counter hit', 'Forehand Push', 'Backhand Push'이다. Forehand Counter hit는 가장 기본적인 Topspin Stroke 자세이고 backhand Counter hit는 Forehand Counter hit와 유사하지만 라켓의 뒷면을 이용한 stroke 자세이다. Forehand Push와 Backhand Push는



[Fig. 5] Biomechanical swing phase

Counter hit와 유사하지만 말 그대로 라켓을 이용하여 공을 미는 Stroke 자세이다. 본 논문에서는 유사하지만 다양한 자세에 대한 분류를 위해 Forehand Push와 Backhand Push를 대신하여 Middle Cut와 Backhand Cut 자세를 추가하였다. 또한 가장 기본적인 Standing 자세와 Ready 자세를 추가하여 총 여섯 가지 자세에 대한 분류를 수행하고자 한다.

탁구 자세 분류를 위해 자세에 맞는 이미지 데이터를 수집하여 여러 가지 이미지 데이터의 집합으로 군집화하고, 각 자세에 대한 클래스 라벨을 부여하였다. 군집화 된 이미지 데이터로부터 분류를 위한 정보 데이터가 정해지므로 어떠한 이미지들로 군집화 시킬 것인지 선정하는 작업은 중요하다. 본 논문에서는 [Fig. 5]와 같이 Preparation, Backswing, Forward swing, Follow through로 구분되는 동작을 통해 이미지를 군집화 하였다.

여섯 가지 자세 중 두 가지 자세는 별다른 Action을 취하지 않는 Stand 자세와 Ready 자세로 정적인 자세이고, 나머지 네 가지 자세는 Stroke action을 취한 동적인 자세들이다. 그 중 Forehand 자세와 Middle cut 자세는 [Fig. 5]의 상단과 같이 Preparation 단계, Backswing 단계, Forward swing 단계, Follow-through 단계 순으로 이미지 데이터를 군집화 하였고, Backhand 자세와 Backhand cut 자세는 [Fig. 5]의 하단과 같이 Preparation 단계, Backswing 단계, Follow-through 단계, Forward swing 단계 순으로 이미지 데이터를 군집화 하였다.

### 3.1.2 Custom training

지도학습(supervised learning)에서 모델을 효과적으로 검증하기 위해서는 Train Set와 Test Set을 적절한 비율로 나누는 것이 중요하다. Train Set를 사용하여 모델을 학습시키고 나면 이후에 Test Set을 이용하여 모델의 예측 및 분류 정확도를 계산할 수 있다. 보통 학습 데이터 중 80%를 Train Set, 20%를 Validation Set으로 사용하여 모델을 학습시키고, 전체 데이터 중 80%를 Train Set, 20%를 Test Set으로 설정하여 모델의 평가를 진행한다. 본 논문에서도 마찬가지로 Train Set의 비율과 Train Set에 전혀 관여되지 않은 인원에 대한 Test Set을 수집하여 모델의 학습 및 평가를 진행하였다. Test Set에는 키, 체형,

[Table 1] Composition of train set and test set

Action	Number	
	Train	Test
Stand	1857	469
Ready	3308	927
Forehand	1354	335
Backhand	1214	308
Middle-cut	1276	311
Backhand-cut	1315	330
Total	10324	2680

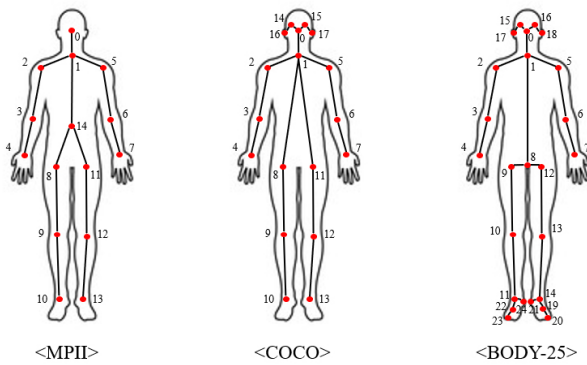
옷, 환경 등이 다른 인원에 대해 영상을 수집하였으며 이를 평가 지표로 활용하였다. 이로 인하여 특정 인물에 대해서만 인식이 제한되는 문제를 해결하였다. 각각의 Train Set 및 Test Set의 비율은 [Table 1]과 같다.

### 3.2 Human Tracking

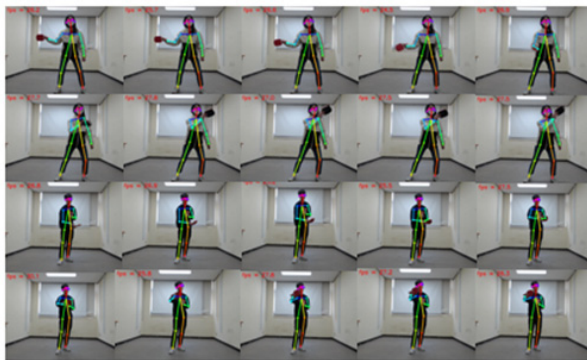
본 논문에서는 인체 감지 및 추적을 위해서 ‘OpenPose’ 알고리즘을 사용하였다. 인체를 감지하는 방법으로 크게 두 가지 방법이 존재한다. 첫 번째는 Top-down approach 방식이다. 이는 먼저 사람을 감지하고, Bounding Box 내부에서 Pose를 예측하는 방식이다. 각 사람마다 별도로 Pose를 예측하기 때문에 계산 비용이 크고 속도가 느리지만, 정확도 측면에서는 다른 방식에 비해 높은 결과를 보인다. 두 번째는 Bottom-up approach 방식으로, 영상에 포함된 사람의 Keypoint를 먼저 예측하고, Keypoint 간의 관계를 분석하여 Pose를 예측하는 방식이다. 이는 Top-down 방식에 비해 정확도는 떨어지지만, 사람을 감지하는 과정이 없어서 계산 비용이 확연하게 줄어들어 실시간 적용이 가능하다. 따라서 Bottom-up approach 방식의 OpenPose를 사용하여 높은 FPS 획득에 따른 실시간 적용이 가능하고, 원하는 인체의 특징 데이터를 획득하여 Domain gap 문제를 해결하도록 하였다. OpenPose에는 크게 ‘MPII’, ‘COCO’, ‘BODY-25’로 세 가지 모델이 존재한다. ‘MPII’는 상, 하체의 14가지 Key Point와 얼굴의 한 가지 Key Point가 포함되어 있고, ‘COCO’는 상, 하체의 13가지 Key Point와 얼굴의 5가지 Key Point가 포함되어 있다. ‘BODY-25’에는 상, 하체 14가지에 더하여 얼굴 5가지, 발 6가지 Key Point가 더해진 모델이다. 이후 과정인 특징 추출에 사용할 관절은 상체 부분이므로 최적화를 위해 가장 필요한 관절만을 인식할 필요가 있다. 따라서 총 18개의 출력 관절로 이루어져 있고 필요한 목과 손목의 좌표가 존재하는 COCO 모델을 사용하여 주요 관절에 대한 (x, y) 좌표를 수집하였다. 각 모델에 대한 예시는 [Fig. 6]과 같다.

COCO 모델을 사용하여 주요 관절의 Skeleton 데이터를 획





[Fig. 6] OpenPose model



[Fig. 7] Human body tracking example using openpose

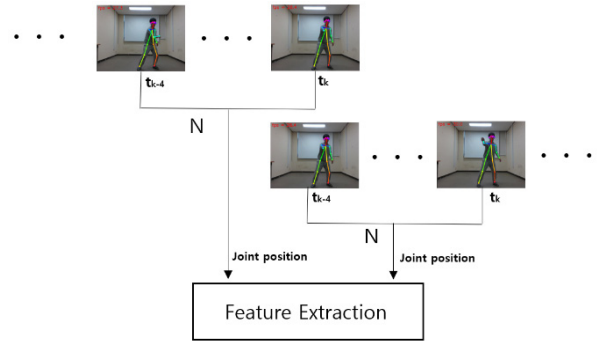
득한다. Skeleton 데이터를 이용하여 학습에 활용하고 실시간 동작 시 인체 추적에 사용한다. [Fig. 7]은 COCO모델을 사용한 실시간 인체 추적의 예시이다<sup>[27,28]</sup>.

### 3.3 Feature Extraction

COCO 모델을 사용하여 인체의 감지 및 추적을 수행하고, 각 관절의 (x, y)좌표 및 군집화 된 이미지 파일에 대한 클래스의 라벨 정보를 획득하였다. 이러한 Skeleton 데이터를 통해 Domain gap 문제를 해결하고, 여러 가지 센서를 융합한 기존의 방식에 비해 높은 FPS 획득이 가능하므로 실시간에 더 용이하다. 각 관절의 좌표만 추출하여 분류를 수행하면 어느 정도의 분류는 가능하겠지만, 비슷한 자세에서 분류를 실패할 확률도 있고 올바른 분류에 시간이 오래 걸릴 수도 있다. 따라서 추출한 Skeleton 데이터를 이용하여 인체의 공간적 특징을 몇 가지 추출하였다. 추출한 특징은 Raw Joint Position, Joint Length, Joint Angle, Upper Body Direction Vector, Wrist Direction Vector로 총 다섯 가지이다. 이를 구하기 위해 먼저 OpenPose로부터 감지된 Skeleton에서 N\_Frame에 대한 관절 위치 정보를 수집한다. 수집한 관절 위치 정보를 통해 다섯 가지의 특징을 추출하게 되며, 추출한 인체의 공간적 특징은 아래 [Table 2]와 같

[Table 2] Five spatial features of the human body

Skeleton Features	
Joint Position	Direction Vector
Raw Joint Position (RP)	Upper Body Direction Vector (BV)
Joint Length (JL)	Wrist Direction Vector (WV)
Joint Angle (JA)	

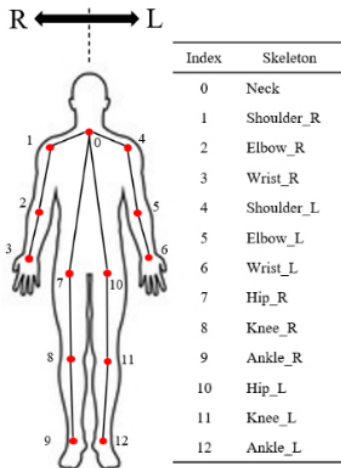


[Fig. 8] Method of extracting spatial features of the human body

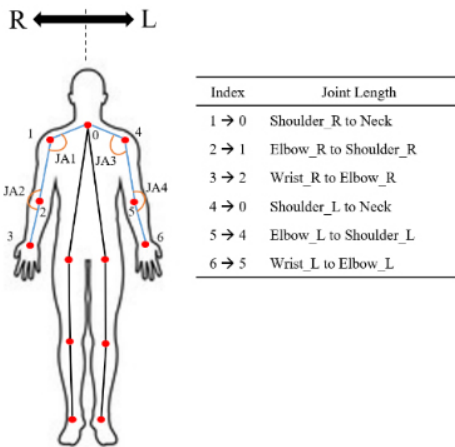
이 Joint Position 측면과 Direction Vector 측면으로 구분하였다. [Fig. 8]은 인체의 공간적 특징 추출 방식을 나타내는 그림이다. 군집화 된 이미지 데이터에서 슬라이딩 윈도우 기법을 적용하여 Window size N을 5로 지정한 뒤 5프레임마다 관절 위치 정보를 획득한다. 5프레임마다 모인 위치 정보를 이용하여 인간의 행동 분류를 위한 특징 추출이 가능해진다. 그림은 10 FPS에서 획득한 이미지 데이터로, 각 이미지 세트는 0.5초에 해당한다. 다음으로 특징 추출을 위한 전처리 과정으로 필요 없는 부위에 대한 관절 정보를 제거하였다. 최적화를 위해 상체의 필요한 부위만 사용하기 위해 COCO 모델을 사용했지만, 실제 모델에는 코, 눈, 귀 등의 신체 부분이 포함된다. 따라서 이를 제거하고 각 관절의 Index를 새롭게 정렬하여 특징 추출을 진행하였다.

#### 3.3.1 Joint Position

Joint Position 측면에는 Raw Joint Position (RP), Joint Length (JL), Joint Angle (JA)가 포함된다. [Fig. 9]는 앞서 설명한 분류 및 특징 추출에 필요 없는 목 위의 신체 부분을 모두 제거하여 Index를 새롭게 정렬한 그림이다. 새롭게 정렬한 Index의 정보를 이용하여 다섯 가지의 특징 추출을 진행하는데 우선 관절 위치 정보를 이용한 특징점들부터 추출하였다. Raw Joint Position은 [Fig. 9]의 Index들 중에서 목에 해당하는 Index 0, 팔에 해당하는 Index 1~6, 그리고 골반에 해당하는 Index 7과 10으로부터 추출하였다. 말그대로 먼저 원시 좌표를 추출하고 이후 이 정보를 이용하여 Joint Length와 Joint Angle을 획득하였다. 골반의 Index는 실제 분류를 위한 특징점으로서 중요한 역할을



[Fig. 9] Joint index after preprocessing



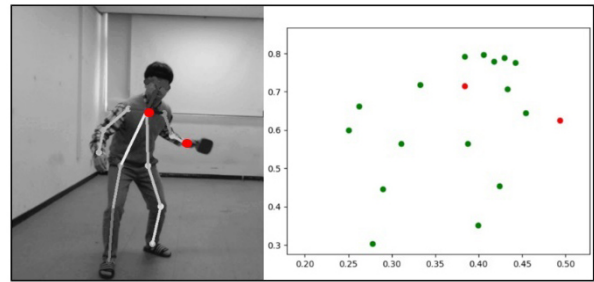
[Fig. 10] Joint length & angle

하지는 않지만, 골반의 Index를 중심으로 인체의 Skeleton이 감지되었음을 인지하기 위한 Trigger로서 사용하였다. 원시 좌표를 통해 추출한 길이 및 각도에 대한 예시는 [Fig. 10]과 같다.

[Fig. 10]은 앞서 추출한 원시 좌표로부터 추출한 관절의 길이와 각도에 대한 Index 표이다. 탁구세 팔의 움직임에 따라 자세가 결정되므로 상체 중 팔에 대한 길이 및 각도를 추출하였다. [Fig. 10]에 포함된 표와 같이 총 여섯 가지의 관절 사이 길이를 구하고, [Fig. 10]의 Skeleton에 표기된 JA1부터 JA4까지의 네 가지 관절 사이의 각도를 추출하였다.

### 3.3.2 Direction Vector

앞서 추출한 세 가지의 특징 이외에 두 가지의 방향 벡터를 특징으로 추출하였다. 탁구에서 Stroke 동작 수행 시에 자세 별 가장 확연한 차이를 보이는 것이 손목의 위치이다. 또한 상체의 무게 중심에 따라 세부적인 자세가 분류되므로, 손목과 목의 위치 변량을 이용하여 방향 벡터를 특징으로 추출하였다.



[Fig. 11] Direction vector extraction

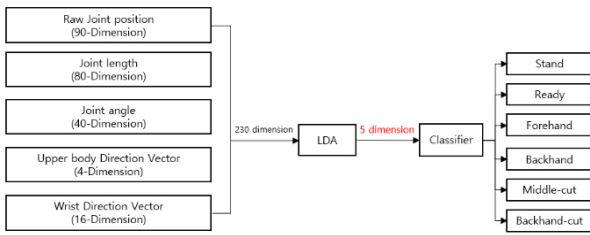
[Fig. 11]은 Forehand 자세에서의 프레임 별 진행 방향(Direction vector)를 나타내며 이를 통해 정적인 상태와 동적인 상태, 비슷한 Stroke 동작 등의 세밀한 분류가 가능해진다.

### 3.4 Dimension Reduction

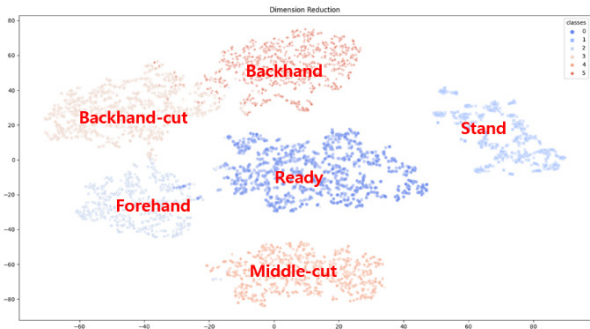
차원의 저주(Curse of dimensionality)라는 말이 있다. 이는 모델을 학습할 때 독립적 샘플이 많을수록 학습이 잘되는 반면, 차원이 커질수록 학습에 제한이 생기고 더 많은 데이터를 필요로 하는 현상을 의미한다. 때문에 차원이 증가할수록 학습 데이터 수가 차원 수보다 적어지면서 모델의 성능이 저하되게 된다. 이를 해결하기 위해 이론적인 해결책으로 학습 샘플의 밀도가 충분히 높아질 때까지 데이터를 모아 Train Set의 크기를 늘리는 방법이 있지만, 현실적으로 일정 밀도에 도달하기 위해 필요한 샘플의 수는 차원의 수가 증가함에 따라 기하급수적으로 증가한다. 이에 학습 샘플을 늘리기 보다는 역으로 차원을 축소시켜 차원의 저주 문제를 해결하였다. 본 논문에서는 차원 축소를 위해 LDA (Linear Discriminant Analysis)를 사용하였다<sup>13)</sup>. 흔히 알려진 대표적인 차원 축소 기법은 PCA (Principal Component Analysis)이다. PCA는 비지도 방식으로써 입력 데이터의 분산이 가장 큰 축을 찾으며 차원을 축소하는 방식이다<sup>14)</sup>. 반면 LDA는 지도학습 방식으로써 개별 클래스를 분류할 수 있는 기준을 최대한 유지하며 차원을 축소하기 때문에 분류 문제에 보다 더 적합하고 보다 뛰어난 성능을 나타낸다. 그렇게 축소된 데이터는 기존의 데이터를 가장 잘 설명할 수 있는 잠재공간인 저차원 공간의 데이터로 변환되며, 이로 인해 차원의 저주 문제를 해결하고 모델의 성능을 향상시킬 수 있다. 마찬가지로 본 연구에서 앞서 추출한 다섯 가지의 Feature들은 각각의 차원을 가지며 이를 합하면 고차원의 성질을 띠게 된다.

본 연구에서는 LDA 기법을 사용하여 [Fig. 12]과 같이 총 230차원의 고차원 데이터를 5차원의 저차원 잠재공간으로 축소시켰다. 이를 통해 모델 경량화 및 모델의 성능을 향상시켰다.

[Fig. 13]는 기존의 고차원 데이터에서 시각화를 위해 2차원



[Fig. 12] Dimension reduction using LDA



[Fig. 13] LDA application result

으로 나타낸 그림이다. 같은 클래스들의 데이터의 분산은 최소화시키고, 다른 클래스들 간의 분산은 최대화 시키고자 하는 LDA의 차원 축소 목적에 맞게 잘 수행됐음을 확인할 수 있다<sup>[13]</sup>. 이처럼 차원 축소는 단순히 데이터의 압축뿐만 아니라 차원 축소를 통한 원 데이터를 가장 잘 설명할 수 있는 잠재적인 요소를 추출하는 것에 목표를 둔다. 이와 같이 수많은 특징 점들이 일차적으로 분류된 과정에서 Classifier를 통해 실제 분류 과정을 거치게 된다.

### 3.5 Classifier Model

고차원의 원본 데이터들을 추출한 고유벡터를 이용해 입력 데이터를 선형 변환함으로써 원본 데이터와 유사한 저차원으로 투영된 데이터로 변환하였다. 이를 Classifier를 이용하여 본격적으로 분류 과정을 거치게 된다. 본 논문에서는 MLP Classifier를 사용한 분류 시스템을 제안한다. 따라서 MLP Classifier의 하이퍼 파라미터 수정을 통해 데이터에 최적화된 모델을 생성하고, 보편적으로 분류를 위해 사용되는 모델인 SVM과 KNN 모델을 비교 모델로 선정하여 생성한 MLP Classifier 모델과 비교하도록 한다<sup>[16]</sup>.

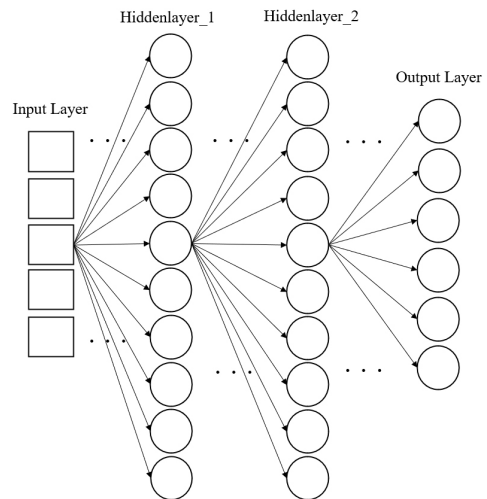
#### 3.5.1 MLP Classifier

MLP (Multi-Layer Perceptron)는 입력층과 출력층 사이에 각각 전체 결합하는 은닉층을 넣은 Neural Network 중 하나이다. 가장 기본적인 형태의 인공신경망 구조이며, 하나의 입력

층(Input layer), 하나 이상의 은닉층(Hidden layers), 하나의 출력층(Output layer)으로 구성되어 있다<sup>[15]</sup>.

본 논문에서는 [Fig. 14]와 같이 은닉층의 수를 2개로 고정하고 각 계층별 10개씩 노드를 할당하였다. Activation function이란, 노드에 입력된 값들을 비선형 함수에 통과시킨 후 다음 레이어로 전달하는데 사용되는 함수를 말한다. 따라서 입력 데이터를 다음 레이어로 어떻게 출력할지 결정하기 때문에 Neural Network에서 중요한 역할을 한다. 본 논문에서는 ReLU (Rectified Linear Unit) function과 Sigmoid function을 은닉층의 활성화 함수로 사용하고, 출력층에 Softmax function을 사용하여 각 모델의 Multi-Class Classification 성능을 비교하였다. [Table 3]은 수정 가능한 Hyperparameter와 각각의 임의 지정 범위를 나타내는 표이다.

본 논문에서는 Step size는 0.01로 고정하였으며 Threshold score는 0.8로 설정하였고 Validation 비율은 4:1로 맞추었다. 그리고 Activation function, Optimizer, Alpha(L2규제), Batch size,



[Fig. 14] Proposed MLP structure

[Table 3] MLP Hyperparameter

Hyperparameter (MLP)	
Name	Value
Output layer Act func	Softmax
Hidden layer Act func	Sigmoid / ReLU
Optimizer	SGD / Adam
Alpha	0.001 / 0.0001
Batch_size	10, 50, 100, 250, 500, 1000, 2000
Step_size	0.01
Epoch	100, 250, 500
Threshold score	0.8
Validation	4:1

Epoch 등의 변수들을 일정량만큼 수정하여 제안하는 시스템에서 가장 높은 정확도를 나타내는 모델을 생성하였다. 실험은 지정한 파라미터에 대해 열 번의 반복 실험을 통하여 모델 정확도의 평균 값을 산출하였고, 각 표에 그 수치들을 표기하였다.

[Table 4]에서는 은닉층의 Activation function으로 Sigmoid를 사용하여 Optimizer로 SGD와 Adam을 각각 비교하였다. 실험 결과 은닉층의 활성화 함수가 Sigmoid function일 경우, Optimizer로 SGD를 사용했을 때 Adam을 사용했을 때보다 전반적으로 성능이 뛰어났다. 하지만 Activation function=Sigmoid, Optimizer=Adam, Epoch=500, Alpha=0.0001, Batch size=1000 일 때 94.28로 가장 높은 성능을 보였다.

[Table 5]에서는 은닉층의 Activation function으로 ReLU를 사용하여 Optimizer로 SGD와 Adam을 각각 비교하였다. 실험 결과 은닉층의 활성화 함수가 ReLU function일 경우, 마찬가지로 Optimizer로 SGD를 사용했을 때 Adam을 사용했을 때보다 전반적으로 성능이 뛰어났다. 또한 Activation=Relu, Optimizer=SGD, Epoch=100, Alpha=0.0001, Batch size=2000 일 때 94.01로 가장 높은 성능을 보였다.

하이퍼 파라미터를 수정한 MLP 모델 중에서는 [Table 4]에 나와있는 Activation function=Sigmoid, Optimizer=Adam, Epoch=500,

Alpha=0.0001, Batch size=1000 일 때 모델의 성능이 가장 뛰어났다. 그리고 하이퍼 파라미터 수정에 따른 전체적인 정확도 비교를 하였을 때, 은닉층에 ReLU function을 사용했을 때보다 성능이 전반적으로 뛰어났다. 본 논문에서는 [Table 4]에서의 성능이 가장 뛰어났던 MLP Classifier 모델을 사용하여 SVM Classifier와 KNN Classifier와의 비교 실험 및 평가를 진행하였다.

### 3.5.2 SVM Classifier

SVM (Support Vector Machine)은 결정 경계(Decision Boundary)라고 불리는 분류를 위한 기준 선을 정의하는 모델이다. 앞서 생성한 MLP Classifier 모델과 비교를 위해 같은 방식으로 하이퍼 파라미터 수정을 통해 데이터에 가장 최적의 성능을 나타내는 모델을 생성하였다.

[Table 6]에서는 RBF Kernel을 사용하여  $\gamma$ 와 c값을 수정 후 정확도를 비교하였다.  $\gamma$ 는 하나의 데이터 샘플이 영향력을 행사하는 거리를 결정하는 역할을 하고,  $\gamma$ 가 클수록 작은 표준 편차를 가지며 데이터가 영향을 미치는 거리가 짧아지게 된다. 실험 결과  $\gamma$ 의 값이 0.001부터 0.1까지는 c값과 상관없이 대체적으로 높은 정확도를 나타냈고 그 외의  $\gamma$ 값에는 c값에 큰 영향을 받았다. 결과적으로  $\gamma$ 값이 0.0001이고 c 값이 10000

[Table 4] Classification result of the MLP-Sigmoid under parameters tuning

		Sigmoid											
		SGD						Adam					
Batch size	Epoch=100	Epoch=250		Epoch=500		Epoch=100	Epoch=250		Epoch=500				
	$\alpha=0.001$	$\alpha=0.0001$	$\alpha=0.001$	$\alpha=0.0001$	$\alpha=0.001$	$\alpha=0.0001$	$\alpha=0.001$	$\alpha=0.0001$	$\alpha=0.001$	$\alpha=0.0001$	$\alpha=0.001$	$\alpha=0.0001$	
10	93.21	93.51	94.09	93.70	93.99	93.51	91.76	90.99	90.41	92.93	90.99	93.02	
50	93.90	93.80	93.99	93.80	93.81	93.99	92.93	94.09	92.83	91.18	92.93	91.76	
100	93.80	93.99	93.51	93.90	94.09	94.08	93.80	92.93	93.02	92.05	94.09	91.96	
250	93.90	94.09	93.80	93.02	93.41	94.09	93.79	93.02	93.22	93.12	92.64	93.70	
500	94.09	93.99	93.70	94.09	93.99	94.19	93.31	93.21	93.60	93.41	92.50	93.70	
1000	93.60	93.70	93.90	93.90	93.70	93.90	93.51	93.90	93.60	92.34	93.31	<b>94.28</b>	
2000	93.60	93.51	93.79	93.60	93.51	93.31	92.83	93.31	93.12	93.70	93.60	93.41	

[Table 5] Classification result of the MLP-ReLU under parameters tuning

		ReLU											
		SGD						Adam					
Batch size	Epoch=100	Epoch=250		Epoch=500		Epoch=100	Epoch=250		Epoch=500				
	$\alpha=0.001$	$\alpha=0.0001$	$\alpha=0.001$	$\alpha=0.0001$	$\alpha=0.001$	$\alpha=0.0001$	$\alpha=0.001$	$\alpha=0.0001$	$\alpha=0.001$	$\alpha=0.0001$	$\alpha=0.001$	$\alpha=0.0001$	
10	88.28	87.02	87.21	86.05	86.72	85.47	88.57	90.02	89.92	89.53	88.27	89.15	
50	91.28	92.25	92.15	92.04	92.44	92.93	89.63	90.50	91.76	92.15	90.79	90.60	
100	92.33	91.47	91.18	91.96	91.76	91.57	90.70	92.04	91.00	91.96	90.89	92.83	
250	92.24	91.76	91.28	92.25	92.73	91.66	90.41	92.05	92.44	92.73	91.28	91.67	
500	93.12	93.02	92.83	93.22	93.02	91.57	92.05	91.67	91.38	92.25	91.47	91.18	
1000	93.80	93.60	92.34	93.02	93.12	93.31	90.02	91.28	90.60	90.12	91.76	91.57	
2000	93.70	<b>94.01</b>	93.21	93.51	92.83	93.31	91.38	91.08	92.64	91.57	93.22	92.83	



일 때 93.12로 가장 높은 정확도를 나타냈다.

[Table 7]에서는 Polynomial Kernel을 사용하여 Degree를 지정 후 c 값에 따른 정확도를 비교하였다. Polynomial Kernel에서는 모든 데이터 포인트들을 새로운 차원으로 변환하여 한 차원 위의 공간으로 투영시킨다. 이렇게 투영함으로써 군집화된 데이터군을 선형적으로 분리할 수 있는 평면을 생성 가능하다. 실험 결과 Degree가 1도일 때, c 값이 10일 때 가장 높은 정확도인 93.08로 나타났다. 결과적으로 SVM Classifier에서는 RBF Kernel을 사용했을 때 가장 높은 정확도를 나타냈으며, [Table 6]에서 가장 높은 정확도를 나타낸  $\gamma=0.0001$ ,  $c=10000$ 을 적용한 모델을 이용하여 비교 모델들과 비교 실험을 진행하였다.

### 3.5.3 KNN Classifier

KNN (K-Nearest Neighbor)은 새로운 입력 데이터에 대해 가장 가까운 데이터를 중심으로 새로운 데이터의 종류를 정해주는 가장 직관적인 모델이다. 최적의 성능은 [Table 8]과 같다.

[Table 8]에서는 맨해튼 거리와 유클리드 거리를 각각 적용하여 K값에 따른 정확도를 비교하였다. KNN에는 크게 세 가지 탐색 방식이 존재한다. Brute Force 방식은 말 그대로 데이터셋 내 지점의 모든 쌍의 거리를 억지로 계산하는 것이며, 작은 데이터셋에서는 경쟁력 있는 방식이다. 하지만 샘플 수의 증가에 따라 증가 속도보다 성능 하락이 커지면서 효율성이 떨어지게 된다. K-D Tree 방식은 이러한 Brute Force의 비효율적인 방식을 해결하기 위해 만들어진 방식으로, 데이터 샘플

[Table 6] Classification result of the SVM-RBF under parameters tuning

RBF								
c	$\gamma$							
	$10^{-7}$	$10^{-6}$	$10^{-5}$	$10^{-4}$	$10^{-3}$	$10^{-2}$	$10^{-1}$	1
0.1	21.51	21.51	21.51	55.97	91.97	92.34	90.89	53.19
1	21.51	21.51	50.10	91.18	92.44	92.54	88.66	59.59
$10^1$	21.51	50.10	91.18	92.44	92.83	92.34	87.02	59.40
$10^2$	50.10	91.18	92.44	92.83	92.83	91.96	86.05	59.40
$10^3$	91.18	92.44	92.83	92.93	92.93	89.73	87.40	59.40
$10^4$	92.44	92.83	92.93	<b>93.12</b>	92.25	87.89	86.63	59.40
$10^5$	92.83	92.93	92.83	92.83	91.28	88.57	86.72	59.40

[Table 7] Classification result of the SVM-Poly under parameters tuning

Poly										
c	Degree									
	1°	2°	3°	4°	5°	6°	7°	8°	9°	10°
0.1	92.83	84.01	89.34	80.81	86.72	78.29	85.85	78.20	85.37	78.01
1	92.93	83.33	87.60	79.46	85.66	78.59	86.53	78.29	85.47	78.49
$10^1$	<b>93.08</b>	83.24	87.01	79.44	86.43	78.49	85.76	78.20	85.17	78.44
$10^2$	93.04	83.19	87.15	79.40	86.41	78.43	85.77	78.17	85.11	78.45
$10^3$	93.01	82.66	87.13	79.41	86.43	78.49	85.72	78.18	85.13	78.42
$10^4$	92.93	82.50	87.18	79.36	86.62	78.51	85.70	78.17	85.09	78.41
$10^5$	92.83	82.36	87.20	79.34	86.52	78.50	85.70	78.17	85.05	78.36

[Table 8] Classification result of the KNN-Distance under parameters tuning

Distance						
k	Manhattan distance			Euclidean distance		
	Ball Tree	K-D Tree	Brute Force	Ball Tree	K-D Tree	Brute Force
5	92.24	92.24	92.24	92.15	92.15	92.15
15	92.73	92.73	92.73	<b>93.02</b>	93.02	93.02
25	92.63	92.63	92.63	92.82	92.82	92.82
35	92.63	92.63	92.63	92.15	92.15	92.15
45	92.53	92.53	92.53	92.24	90.69	90.69
55	92.53	92.53	92.53	92.24	92.24	92.24

에 대한 거리 집계 정보를 효율적으로 계산하여 탐색 계산 비용을 줄인 방식이다. Ball Tree 방식은 K-D Tree 방식을 개선한 방식으로 매우 고차원에서 그 효율성이 드러나는 방식이다. 실험 결과로는 세 방식의 정확도에서는 차이가 없었지만 각 자세에 대한 F1 score에서 미미한 차이가 존재하였다. 결과적으로는 유클리드 거리를 적용하고 K값이 15일 때 가장 높은 정확도인 93.02를 나타냈으며, 이 모델을 사용하여 비교 모델과의 비교 실험을 진행하였다.

### 4. Evaluation

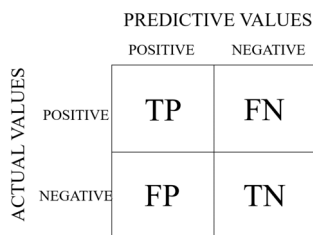
#### 4.1 Classification Performance of The Model

제안하는 모델의 성능 평가를 위해 Confusion Matrix를 사용하였다. Confusion Matrix란, 학습을 통한 모델의 예측 성능을 측정하기 위해 예측 값과 실제 값을 비교하기 위한 표이다. Confusion Matrix에는 Accuracy (A), Precision (P), Recall (R), F1-score 등의 항목이 포함되며, 본 논문에서는 Precision, Recall, F1-score를 이용하여 모델의 성능 비교를 진행한다. 탁구 자세 분류에서 Precision은 제대로 분류된 자세(True Positive, True Negative)의 비율을 나타낸다. True Positive와 True Negative는 실제 값을 맞게 예측한 부분이고, False Positive와 False Negative는 실제 값과 다르게 예측한 부분을 의미한다.

[Fig. 15]는 Confusion Matrix의 구조이다. 앞서 설명한 TP, TN, FP, FN을 수식적으로 계산하여 Accuracy, Precision, Recall, F1-score 등을 구한다.

[Table 9]는 Accuracy가 94.28인 MLP Classifier 모델에 대한 Confusion Matrix이다. middle-cut 자세에 대한 분류 성능만 조금 떨어질 뿐, 평균 93%의 F1-score를 획득하며 자세 분류에 대한 높은 성능을 나타냈다. 이외 Precision이나 Recall에서도 각각 94%, 93%의 높은 성능을 나타냈다. 이에 대하여 각 자세 별 실제 자세에 대하여 모델이 예측한 비율을 Visualize한 Matrix는 [Fig. 16]과 같다.

가운데 대각선의 굵은 글씨체로 표시된 비율은 실제 자세에 대하여 모델이 옳게 예측한 비율을 의미한다. (Recall) 이외의 비율은 실제 자세에 대하여 모델이 잘못 예측한 비율을 나



[Fig. 15] Confusion matrix structure

타낸다. MLP Classifier에서는 Middle-cut 자세를 Stand 자세로 약 19%의 비율로 오탐하였다. 이외에는 대체적으로 10% 이하의 오탐 비율로 높은 예측 성능을 보여주었다.

[Table 10]은 Accuracy가 93.12인 SVM Classifier 모델에 대한 Confusion Matrix 표이다. SVM 모델에서는 forehand 자세 및 middle-cut 자세에서 MLP 모델에 비해 상대적으로 낮은 F1-score를 나타냈다. 하지만 이외의 자세에 대해서는 전반적으로 높은 성능을 나타내며 Precision에서 93%, Recall에서 91%, 평균 F1-Score로 91%의 점수를 나타냈다. 이에 대한 각 자세 별 실제 자세에 대하여 모델이 예측한 비율을 Visualize한 Matrix는 [Fig. 17]과 같다.

SVM Classifier에서도 마찬가지로 middle-cut 자세를 오탐한 비율이 가장 컸는데, MLP Classifier와는 다르게 forehand 자세로 오탐한 비율 약 20%로 가장 높았다. 이외에도 10% 미만의 오탐 비율을 나타내며 대체적으로 양호한 성능을 나타냈다.

[Table 11]은 Accuracy가 93.02인 KNN Classifier 모델에 대한 Confusion Matrix 표이다. KNN Classifier에서는 forehand, backhand, middle-cut 자세에 대한 각 평가 지표 점수가 다른 모델에 비해 확연히 떨어졌다. 각 자세에 대한 평균 점수로 비교했을 때 타 모델들에 비해 가장 낮은 점수를 나타냈다. 이에 대한 각 자세 별 실제 자세에 대하여 모델이 예측한 비율을 Visualize한 Matrix는 [Fig. 18]과 같다.

[Table 9] Confusion matrix by MLP classifier

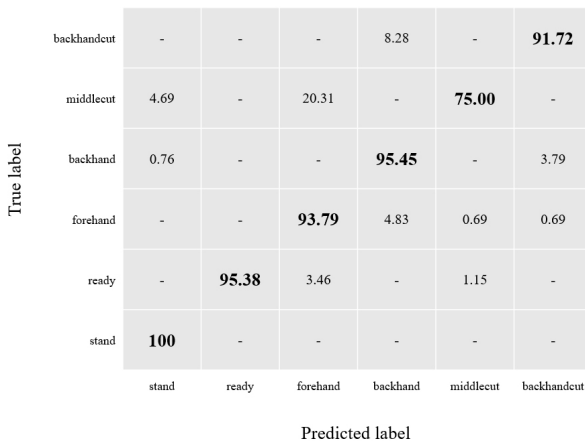
	Precision	Recall	F1-score
stand	0.90	1.00	<b>0.95</b>
ready	1.00	1.00	<b>1.00</b>
forehand	0.95	0.93	<b>0.94</b>
backhand	0.90	0.92	<b>0.91</b>
middle-cut	0.99	0.76	<b>0.86</b>
backhand-cut	0.92	0.95	<b>0.94</b>
Average	0.94	0.93	<b>0.93</b>

True label	Predicted label					
	stand	ready	forehand	backhand	middlecut	backhandcut
backhandcut	-	-	-	4.83	-	<b>95.17</b>
middlecut	18.75	0.78	4.69	-	<b>75.78</b>	-
backhand	-	-	-	<b>92.42</b>	-	7.58
forehand	0.69	-	<b>93.10</b>	4.14	0.69	1.38
ready	-	<b>99.62</b>	0.38	-	-	-
stand	<b>100</b>	-	-	-	-	-

[Fig. 16] MLP confusion matrix visualization

[Table 10] Confusion matrix by SVM classifier

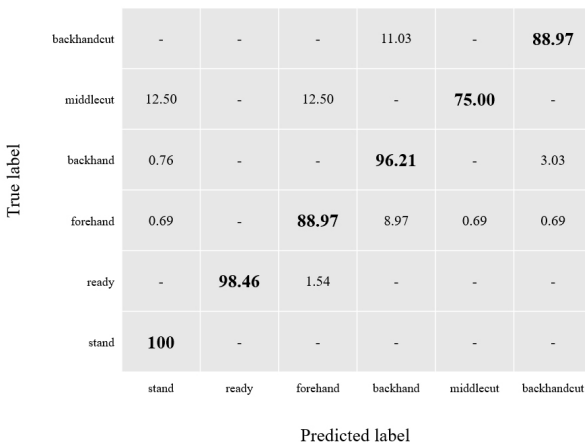
	Precision	Recall	F1-score
stand	0.97	1.00	<b>0.98</b>
ready	1.00	0.95	<b>0.98</b>
forehand	0.80	0.94	<b>0.86</b>
backhand	0.87	0.95	<b>0.91</b>
middle-cut	0.96	0.75	<b>0.84</b>
backhand-cut	0.96	0.92	<b>0.94</b>
Average	0.93	0.91	<b>0.91</b>



[Fig. 17] SVM confusion matrix visualization

[Table 11] Confusion matrix by KNN classifier

	Precision	Recall	F1-score
stand	0.93	1.00	<b>0.96</b>
ready	1.00	0.98	<b>0.99</b>
forehand	0.87	0.89	<b>0.88</b>
backhand	0.81	0.96	<b>0.88</b>
middle-cut	0.99	0.75	<b>0.85</b>
backhand-cut	0.96	0.89	<b>0.92</b>
Average	0.92	0.91	<b>0.91</b>



[Fig. 18] KNN confusion matrix visualization

KNN Classifier에서는 forehand를 backhand로 예측한 경우, middle-cut를 stand로 예측한 경우, middle-cut를 forehand로 예측한 경우, backhand-cut를 backhand로 예측한 오답 비율이 다른 모델에 비해 확연히 높았다.

세 모델에 대해 자세 별 Visualize된 recall 비율과 F1-score를 비교해봤을 때, MLP Classifier가 가장 낮은 오답 비율을 나타냈다. 하지만 비교한 세 모델 모두에서 ‘middle cut’ 자세에 대한 오답 비율이 가장 높게 나타났다. 이는 Visualize된 Confusion Matrix를 통해 확인 가능하듯이 ‘Stand’ 자세 및 ‘forehand’ 자세와 ‘middle cut’ 자세가 유사하기 때문에 나타난 결과이다. 표를 통한 최종 결과 비교 시에 이 부분에 대해서도 MLP Classifier의 오답 비율이 가장 낮았음을 확인할 수 있다.

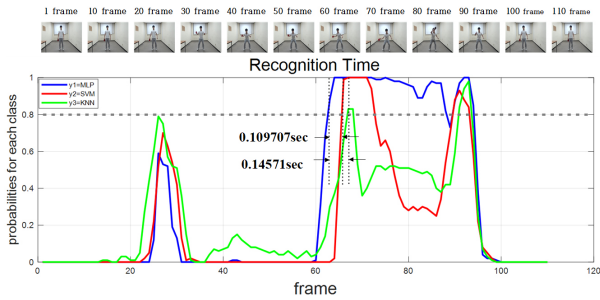
결과적으로 F1-Score에 대한 평균 값을 비교했을 때, MLP 모델에서 93%, SVM 모델에서 91%, KNN 모델에서 91%를 나타내며 제안하는 모델이 비교 모델에 비해 2%가량 높은 성능을 나타냈다<sup>[16]</sup>.

#### 4.2 Classification Speed of The Model

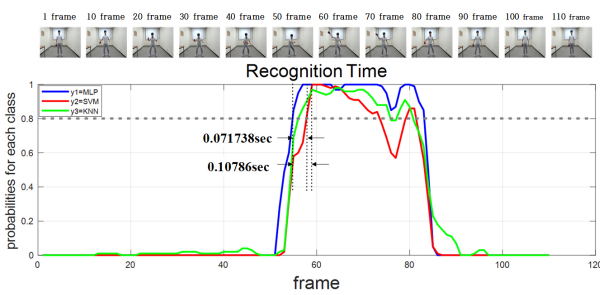
앞선 장에서 Confusion Matrix를 통한 각 비교 모델들의 분류 성능확인을 통해 제안하는 시스템의 유효성을 입증하였다. 실제 탁구 로봇이 얼마나 정확하게 예측하는지도 중요한 문제지만 얼마나 빠르게 예측하는지도 중요한 문제이다. 예측을 정확하게 하더라도 분류하는데 오랜 시간이 걸리게 되면 그 시간만큼 로봇이 반응하는데 필요한 시간이 줄어들기 때문이다. 즉, 분류 시간이 빠르면 빠를수록 전체적인 부담이 덜어지는 것이다. 또한 본 연구의 취지인 협동 로봇의 느린 관절 구동 속도를 극복하기 위한 빠른 분류 시스템 개발에 대한 검증을 위해서는 생성한 모델이 각 자세에 대한 분류 속도가 어느 정도 나타나는지 확인하는 과정이 필수적이다. 따라서 본 논문에서는 각 비교 모델들을 네 가지 자세에 대해 얼마나 빠르게 인식하는지 그래프를 통해 비교하였다. 느린 구동 속도도도 상대방의 타구에 대응할 여유 시간을 갖기 위해, 빠른 분류에 대한 기준 시간을 0.1초로 설정하였다. 분류를 위한 Threshold는 0.8로 지정하였고 총 여섯 가지의 자세에 대한 확률의 합은 1이 된다. 이에 따라 0.8 이상의 확률로 분류된 자세가 모델이 예측한 자세가 되고, 비교하고자 하는 자세를 최초 인식한 시간을 비교하여 제안하는 시스템의 타당성을 증명하였다. 네 자세에 대한 영상 모두 110 frame의 동일한 영상 길이를 가지고 있으며 30 fps로 촬영되었다. [Fig. 19] 부터 [Fig. 22]는 Forehand Stroke, Backhand Stroke, Middle-cut Stroke, Backhand-cut Stroke에 대하여 앞서 생성한 세 모델들의 분류 속도를 비교한 그래프이다.

각 Stroke에 대한 모델 별 분류 속도를 비교 결과, 평균적으로 모든 Stroke에 대하여 제안하는 MLP 모델에서 비교 모델에

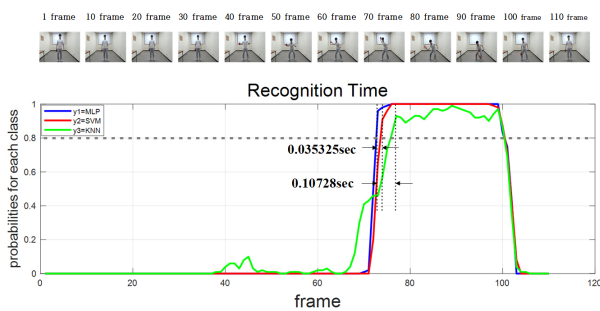
비해 빠른 분류 속도를 보였다. 동작을 취했다고 판단하는 순간부터 지정한 Threshold를 넘는 순간까지 모든 자세에서 0.1초 이내의 시간이 소요됐으며, 타 비교 모델들과 0.1초 이상의 시간 차가 발생하였다. 결론적으로 모든 분류 자세에 대하여 제안하는 모델이 빠른 분류의 기준 시간인 0.1초 이내로 판단을 수행하며 시스템의 유효성을 입증하였다.



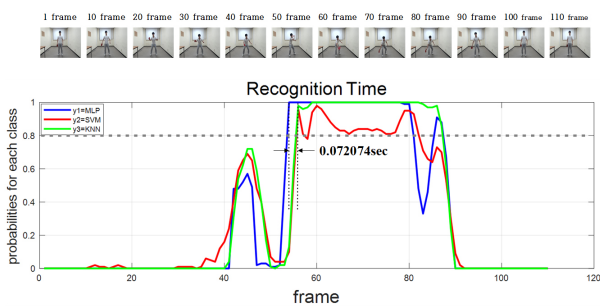
[Fig. 19] Recognition time of forehand stroke



[Fig. 20] Recognition time of backhand stroke



[Fig. 21] Recognition time of middle-cut stroke



[Fig. 22] Recognition time of backhand-cut stroke

## 5. Conclusion

본 논문에서는 자동화 훈련 시스템 개발을 목적으로, 협동 로봇의 느린 관절 구동 속도를 극복하기 위한 빠른 인식 시스템을 제안하였다. 이를 위해 탁구 라켓이나 사용자의 몸에 어떠한 센서 부착 없이 2D 카메라 한 대만을 사용함에 따라 저가형 시스템을 구축하였다.

또한 한 사람의 Training Set 구성만으로 신체가 다른 사람에 대한 분류도 성공적으로 수행하며 Domain Gap 문제를 해소하였다. 클래스 별로 군집화 된 데이터셋으로부터 Skeleton 데이터를 획득하고, 이에 대한 다섯 가지 신체의 공간적 특징을 추출하여 비슷한 자세도 구별 가능하도록 하였다. 또한 학습 데이터의 차원을 축소함으로써 필요 없는 데이터는 삭제하여 모델의 경량화를 이루었다. 이렇게 차원의 저주 문제를 해결함에 따라 모델의 성능을 향상시켰다.

최종 분류를 위한 Classifier로는 MLP classifier를 사용하였다. 이에 대한 타당성을 증명하기 위하여 보편적으로 사용되는 SVM classifier 및 KNN classifier와 Confusion Matrix를 통하여 모델의 분류 성능을 평가하였다. 또한 각 모델의 네 가지 자세별 인식 시간을 비교하였다. 이에 따라 제안하는 알고리즘에서 가장 높은 분류 능력과 가장 빠른 분류 시간이 나타났음을 데이터를 통하여 증명하고 제안하는 알고리즘의 유효성을 입증하였다.

본 연구에서는 탁구 선수의 기량 향상을 목적으로 연구를 진행하였지만, 타 스포츠 분야에서도 훈련 목적으로 활용 가능할 것이다. 또한 자세 교정 목적, 위급 상황 조기 발견, 범죄 상황 인식 등에도 활용이 가능할 것이다. 추후 새로운 하이퍼 파라미터를 적용하여 비슷한 자세에 대한 오탐 비율을 줄이고, 데이터셋 세분화, 인체의 공간적 특징 확장 등으로 성능이 보다 향상된 모델을 생성할 예정이다. 최종적으로 낙구 지점에 대한 자세 별 데이터셋을 생성하여 조기 인식 및 낙구 지점 예측 시스템 개발을 위한 연구를 진행할 것이다.

## Reference

- [1] M. Iosa, P. Picerno, S. Paolucci, and G. Morone, "Wearable inertial sensors for human movement analysis," *Expert Rev. Med. Devices*, vol. 13, no. 7, pp. 641-659, July, 2016, DOI: 10.1080/17434440.2016.1198694.
- [2] F. Li, K. Shirahama, M. A. Nisar, L. Köping, and M. Grzegorzec, "Comparison of feature learning methods for human activity recognition using wearable sensors," *IEEE Sensors*, vol. 18, no. 2, pp. 679, 2018, DOI: 10.3390/s18020679.
- [3] C. Chen, R. Jafari, and N. Kehtarnavaz, "A Real-Time Human Action Recognition System Using Depth and Inertial Sensor Fusion," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 773-781, February, 2016, DOI: 10.1109/jsen.2015.2487358.



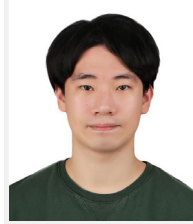
- [4] C. Chen R. Jafari, and N. Kehtarnavaz, "Improving Human Action Recognition Using Fusion of Depth Camera and Inertial Sensors," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 51-61, 2015, DOI: 10.1109/thms.2014.2362520.
- [5] N. Dawar and N. Kehtarnavaz, "Action Detection and Recognition in Continuous Action Streams by Deep Learning-Based Sensing Fusion," *IEEE Sensors Journal*, vol. 18, no. 23, pp. 9660-9668, December, 2018, DOI: 10.1109/jsen.2018.2872862.
- [6] A. I. Cuesta-Vargas, A. Galán-Mercant, and J. M. Williams, "The use of inertial sensors system for human motion analysis," *Phys. Ther. Rev.*, vol. 15, no. 6, pp. 462-473, 2010, DOI: 10.1179/1743288x11y.0000000006.
- [7] Z. Zhiqing, "Biomechanical Analysis and Model Development Applied to Table Tennis Forehand Strokes," *School of Mechanical and Aerospace Engineering*, 2016, DOI: 10.32657/10356/70334.
- [8] R. A. Minhas, A. Javed, A. Irtaza, M. T. Mahmood, and Y. B. Joo, "Shot Classification of Field Sports Videos Using AlexNet Convolutional Neural Network," *Computing and Artificial Intelligence*, vol. 9, January, 2019, DOI: 10.3390/app9030483.
- [9] J. Kim and Y. Do, "Human Detection in the Images of a Single Camera for a Corridor Navigation Robot," *The Journal of Korea Robotics Society*, vol. 8, no. 4, pp. 236-246, Dec., 2013, DOI: 110.7746/jkros.2013.8.4.238.
- [10] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: automatic estimation of 3D human pose and shape from a single image," *European Conference on Computer Vision*, vol. 9909, pp. 561-578, 2016, DOI: 10.1007/978-3-319-46454-1\_34.
- [11] Q. Dang, J. Yin, B. Wang, and W. Zheng, "Deep learning based 2d human pose estimation: A survey," *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 663-676, 2019, DOI: 10.26599/tst.2018.9010100.
- [12] Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, and B. Zhao, "A federated learning system with enhanced feature extraction for human activity recognition," *Knowledge-Based Systems*, vol. 229, October, 2021, DOI: 10.1016/j.knosys.2021.107338.
- [13] D. K. Choubey, M. Kumar, V. Shukla, S. Tripathi, and V. K. Dhandhanian, "Comparative Analysis of Classification Methods with PCA and LDA for Diabetes," *Current diabetes reviews*, vol. 16, no. 8, pp. 833-850, September, 2020, DOI: 10.2174/1573399816666200123124008.
- [14] F. Anowar, S. Sadaoui, and B. Selim, "Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE)," *Computer Science Review*, vol. 40, May, 2021, DOI: 10.1016/j.cosrev.2021.100378.
- [15] A. Iqbal and S. Aftab, "A Classification Framework for Software Defect Prediction Using Multi-filter Feature Selection Technique and MLP," *I.J. Modern Education and Computer Science*, vol. 1, pp. 18-25, February, 2020, DOI: 10.5815/ijmecs.2020.01.03.
- [16] H. I. Dino and M. B. Abdulrazzaq, "Facial Expression Classification Based on SVM, KNN and MLP Classifiers," *International Conference on Advanced Science and Engineering*, May, 2019, DOI: 10.1109/icoase.2019.8723728.
- [17] E. Vats and C. S. Chan, "Early detection of human actions—A hybrid approach," *Applied Soft Computing*, vol. 46, pp. 953-966, September, 2016, DOI: 10.1016/j.asoc.2015.11.007.
- [18] J. E. Lee, J. H. Kim, H. G. Lee, S. J. Kim, D. H. Kim, and G. T. Park, "Emergency Alarm Service for the old and the weak by Human Behavior Recognition in Intelligent Space," *The Journal of Korea Robotics Society*, vol. 2, no. 4, pp. 297-303, 2007, Accessed: January 27, 2022, [Online], <https://koreascience.kr/article/JAKO200723736027917.page>.
- [19] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," *Computer Vision and Pattern Recognition Conference*, June, 2018, DOI: 10.1109/cvpr.2018.00742.
- [20] M. Seo, S. Lee, and D.-G. Choi, "Spatial-temporal Ensemble Method for Action Recognition," *The Journal of Korea Robotics Society*, vol. 15, no. 4, pp. 385-391, 2020, DOI: 10.7746/jkros.2020.15.4.385.
- [21] A. Elhayek, O. Kovalenko, P. Murthy, J. Malik, and D. Stricker, "Fully automatic multi-person human motion capture for VR applications," *EuroVR 2018*, vol. 11162, pp. 28-47, 2018, DOI: 10.1007/978-3-030-01790-3\_3.
- [22] F. Sajjad, A. F. Ahmed, and M. A. Ahmed, "A study on the learning based human pose recognition," *2017 9th IEEE-GCC Conference and Exhibition (GCCCE)*, pp. 1-8, Manama, Bahrain, 2017, DOI: 10.1109/IEEEGCC.2017.8448200.
- [23] X. Cheng, L. Zhang, Y. Tang, Y. Liu, H. Wu, and J. He, "Real-Time Human Activity Recognition Using Conditionally Parametrized Convolutions on Mobile and Wearable Devices," *IEEE Sensors*, vol. 22, no. 6, March, 2022, DOI: 10.1109/jsen.2022.3149337.
- [24] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," *IEEE 13th Int. Conf. Wearable Implant. Body Sensor Netw (BSN)*, pp. 71-76, Jun, 2016, DOI: 10.1109/bsn.2016.7516235.
- [25] Z. Yang, O. I. Raymond, C. Zhang, Y. Wan, and J. Long, "DFTerNet: Towards 2-bit dynamic fusion networks for accurate human activity recognition," *IEEE Access*, vol. 6, pp. 56750-56764, 2018, DOI: 10.1109/access.2018.2873315.
- [26] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Netw. Appl.*, vol. 25, pp. 743-755, December, 2019, DOI: 10.1007/s11036-019-01445-x.
- [27] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 172-186, January, 2021, DOI: 10.1109/tpami.2019.2929257.
- [28] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," *Computer Vision and Pattern Recognition Conference*, pp. 7291-7299, 2017, DOI: 10.1109/cvpr.2017.143.



**진 성 호**

2019 경남과학기술대학교 메카트로닉스 공학과 학사  
2020~현재 부산대학교 전기전자공학과 석사과정

관심분야: 영상 인식, 매니플레이터 및 그리퍼 제어



**안 재 훈**

2021 부산대학교 전자공학과 학사  
2022~현재 부산대학교 전기전자공학과 석사과정

관심분야: 보행로봇 제어, 기계 학습



**권 영 우**

2021 부산대학교 전자공학과 학사  
2022~현재 부산대학교 전기전자공학과 석사과정

관심분야: 컴퓨터 비전, 자세 추정, Quadruped Robot



**강 호 선**

2017 부산대학교 항공우주공학과 학사  
2019 부산대학교 전기전자컴퓨터 공학과 석사  
2019~현재 부산대학교 전자공학과 박사과정

관심분야: 보행로봇 제어, 위치 인식, 영상 처리, 강화 학습



**김 윤 정**

2020 동의대학교 메카트로닉스공학과 학사  
2021~현재 부산대학교 전기전자공학과 석사과정

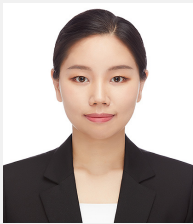
관심분야: 경로계획, 로봇 시뮬레이션, 매니플레이터 제어



**최 지 욱**

2017 경남과학기술대학교 전자공학과 학사  
2019 부산대학교 전기전자컴퓨터 공학과 석사  
2019~현재 부산대학교 전자공학과 박사과정

관심분야: 지능제어, 강인제어, 마이크로프로세서 응용



**박 미 영**

2020 경남과학기술대학교 메카트로닉스 공학과 학사  
2021~현재 부산대학교 전기전자공학과 석사과정

관심분야: 매니플레이터 제어, 시스템 설계



**이 인 호**

2009 한국과학기술원 기계공학과 학사  
2011 한국과학기술원 기계공학과 석사  
2016 한국과학기술원 기계공학과 박사  
2020~현재 부산대학교 전자공학과 조교수

관심분야: Robotics, 기계 시스템 자동화, 시스템 모델링