(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0081950
(43) 공개일자 2024년06월10일

(51) 국제특허분류(Int. Cl.)
G06V 40/20 (2022.01) *G06V 10/40* (2022.01)
G06V 10/762 (2022.01) *G06V 10/764* (2022.01)
G06V 20/40 (2022.01)
(52) CPC특허분류
G06V 40/20 (2022.01)
G06V 10/40 (2023.08)
(21) 출원번호 10-2022-0165732
(22) 출원일자 2022년12월01일
심사청구일자 2022년12월01일

(71) 출원인
한국과학기술연구원
서울특별시 성북구 화랑로14길 5 (하월곡동)
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
조정현
서울특별시 성북구 화랑로14길 5
김익재
서울특별시 성북구 화랑로14길 5
(뒷면에 계속)
(74) 대리인
특허법인로얄

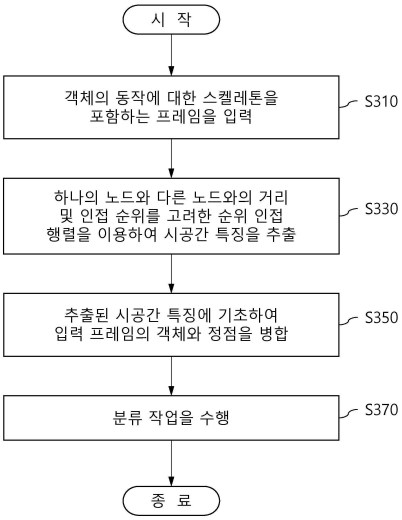
전체 청구항 수 : 총 16 항

(54) 발명의 명칭 **순위 그래프 컨벌루션 네트워크를 이용한 동작 처리 방법 및 장치**

(57) 요 약

본 명세서는 그래프 컨벌루션 네트워크(graph convolutional network)를 이용하는 스켈레톤 기반의 동작 인식 기술에 관한 것으로, 동작 처리 장치가 객체의 동작에 대한 스켈레톤(skeleton)을 포함하는 프레임을 입력받고, 스켈레톤에 대해 하나의 노드와 다른 노드와의 거리 및 인접 순위를 고려한 순위 인접 행렬(rank adjacency matrix)을 이용하여 시공간 특징을 추출하고, 추출된 시공간 특징에 기초하여 입력 프레임의 객체와 정점(vertex)을 병합하며, 분류 작업을 수행한다.

대 표 도 - 도3



(52) CPC특허분류
G06V 10/7625 (2023.08)
G06V 10/764 (2023.08)
G06V 20/46 (2022.01)
G06T 2207/20044 (2013.01)
(72) 발명자
박운상
서울특별시 마포구 삼개로 33, 도화3지구우성아파트 6동 1502호

이햇살
서울특별시 성동구 왕십리로 410, 센트라스 116동 1004호

명 세 서

청구범위

청구항 1

그래프 컨벌루션 네트워크(graph convolutional network, GCN)를 이용한 동작 처리 방법에 있어서,

동작 처리 장치가 객체의 동작에 대한 스켈레톤(skeleton)을 포함하는 프레임을 입력받는 단계;

상기 동작 처리 장치가 상기 스켈레톤에 대해 하나의 노드와 다른 노드와의 거리 및 인접 순위를 고려한 순위 인접 행렬(rank adjacency matrix)을 이용하여 시공간 특징을 추출하는 단계;

상기 동작 처리 장치가 추출된 상기 시공간 특징에 기초하여 입력 프레임의 객체와 정점(vertex)을 병합하는 단계; 및

상기 동작 처리 장치가 분류 작업을 수행하는 단계;를 포함하는, 동작 처리 방법.

청구항 2

제 1 항에 있어서,

상기 시공간 특징을 추출하는 단계는,

공간 그래프 컨벌루션 레이어 및 시간 컨벌루션 레이어를 각각 적어도 하나 이상 포함하는 모듈을 통해 수행되며,

상기 공간 그래프 컨벌루션 레이어는,

특징 채널 축을 따라 노드 인덱싱(node indexing)을 수행하고,

완전 연결 레이어(fully-connected layer)를 적용하며,

순위 인접 행렬을 사용하여 정점(vertex)을 취합하되 프레임 간에 공유되는 어텐션 마스크(attention mask)를 적용하는, 동작 처리 방법.

청구항 3

제 2 항에 있어서,

상기 순위 인접 행렬은,

입력된 스켈레톤 데이터에 포함된 관절(joint)을 나타내는 노드 각각에 대해 다른 노드와의 거리를 산출하고, 산출된 거리가 작은 순서대로 정렬하되 순위 메트릭(metric)을 통해 상기 하나의 노드와 가까운 소정 개수의 다른 노드에 대해 인접 순위를 나타내는 행렬을 생성하는, 동작 처리 방법.

청구항 4

제 3 항에 있어서,

상기 순위 인접 행렬은,

입력 프레임의 모든 노드에 대해 하나의 노드와 다른 노드간의 유클리디언 거리(Euclidean distance)를 산출하여 거리 행렬을 생성하고, 생성된 상기 거리 행렬을 기반으로 순위 범위가 설정된 순위 메트릭을 통해 순위 부여 및 필터링을 통해 인접 순위를 나타내는 행렬을 생성하는, 동작 처리 방법.

청구항 5

제 2 항에 있어서,

상기 노드 인덱싱은,

특징 채널 축을 따라 원-핫(one-hot) 정점 인덱스의 임베딩을 수행함으로써 관절 특징을 동적으로 취합하는, 동

작 처리 방법.

청구항 6

제 2 항에 있어서,

상기 어텐션 마스크는,

특징을 취합하기 전에 어텐션을 적용하여 각각의 순위 하위 집합(subset)에 대한 마스크를 학습하되, 프레임 간에 공유되는 어텐션 마스크로서 정적 곱셈 마스크(static multiplication mask)를 적용하는, 동작 처리 방법.

청구항 7

제 2 항에 있어서,

상기 시공간 특징을 추출하는 단계는,

3개의 채널 스테이지로 구성되고, 각각의 채널 스테이지는 공간 그래프 컨벌루션 레이어 및 시간 컨벌루션 레이어의 블록이 4개, 3개 및 3개로 구성되어 총 10개의 블록을 형성하는 모듈을 통해 수행되는, 동작 처리 방법.

청구항 8

제 1 항에 있어서,

상기 입력 프레임의 객체와 정점을 병합하는 단계는, 전역 평균 풀링(global averaging pooling, GAP)을 통해 수행되고,

상기 분류 작업을 수행하는 단계는, 소프트맥스(softmax)를 적용함으로써 수행되는, 동작 처리 방법.

청구항 9

하나 이상의 명령어(instructions)을 저장하는 하나 이상의 비-일시적인(non-transitory) 컴퓨터 판독 가능 매체(computer-readable medium)에 있어서,

하나 이상의 프로세서에 의해 실행 가능한(executable) 상기 하나 이상의 명령어는, 그래프 컨벌루션 네트워크(graph convolutional network, GCN)를 이용한 동작을 처리하되,

객체의 동작에 대한 스켈레톤(skeleton)을 포함하는 프레임을 입력받고,

상기 스켈레톤에 대해 하나의 노드와 다른 노드와의 거리 및 인접 순위를 고려한 순위 인접 행렬(rank adjacency matrix)을 이용하여 시공간 특징을 추출하고,

추출된 상기 시공간 특징에 기초하여 입력 프레임의 객체와 정점(vertex)을 병합하며,

분류 작업을 수행하는, 컴퓨터 판독 가능 매체.

청구항 10

제 9 항에 있어서,

상기 순위 인접 행렬은,

입력된 스켈레톤 데이터에 포함된 관절(joint)을 나타내는 노드 각각에 대해 다른 노드와의 거리를 산출하고, 산출된 거리가 작은 순서대로 정렬하되 순위 메트릭(metric)을 통해 상기 하나의 노드와 가까운 소정 개수의 다른 노드에 대해 인접 순위를 나타내는 행렬을 생성하는, 컴퓨터 판독 가능 매체.

청구항 11

객체의 동작에 대한 스켈레톤(skeleton)을 포함하는 프레임을 입력받는 입력부; 및

그래프 컨벌루션 네트워크(graph convolutional network, GCN)를 이용하여 상기 프레임에 대한 동작을 처리하는 처리부;를 포함하고,

상기 처리부는,

상기 스켈레톤에 대해 하나의 노드와 다른 노드와의 거리 및 인접 순위를 고려한 순위 인접 행렬(rank adjacency matrix)을 이용하여 시공간 특징을 추출하고, 추출된 상기 시공간 특징에 기초하여 입력 프레임의 객체와 정점(vertex)을 병합하며, 분류 작업을 수행하는, 동작 처리 장치.

청구항 12

제 11 항에 있어서,

상기 처리부는,

공간 그래프 컨벌루션 레이어 및 시간 컨벌루션 레이어를 각각 적어도 하나 이상 포함하는 모듈을 통해 상기 시공간 특징을 추출하며,

상기 공간 그래프 컨벌루션 레이어는,

특징 채널 축을 따라 노드 인덱싱(node indexing)을 수행하고,

완전 연결 레이어(fully-connected layer)를 적용하며,

순위 인접 행렬을 사용하여 정점(vertex)을 취합하되 프레임 간에 공유되는 어텐션 마스크(attention mask)를 적용하는, 동작 처리 장치.

청구항 13

제 12 항에 있어서,

상기 처리부는,

입력된 스켈레톤 데이터에 포함된 관절(joint)을 나타내는 노드 각각에 대해 다른 노드와의 거리를 산출하고, 산출된 거리가 작은 순서대로 정렬하되 순위 메트릭(metric)을 통해 상기 하나의 노드와 가까운 소정 개수의 다른 노드에 대해 인접 순위를 나타내는 상기 순위 인접 행렬을 생성하는, 동작 처리 장치.

청구항 14

제 13 항에 있어서,

상기 처리부는,

입력 프레임의 모든 노드에 대해 하나의 노드와 다른 노드간의 유클리디언 거리(Euclidean distance)를 산출하여 거리 행렬을 생성하고, 생성된 상기 거리 행렬을 기반으로 순위 범위가 설정된 순위 메트릭을 통해 순위 부여 및 필터링을 통해 인접 순위를 나타내는 상기 순위 인접 행렬을 생성하는, 동작 처리 장치.

청구항 15

제 12 항에 있어서,

상기 처리부는,

특징 채널 축을 따라 원-핫(one-hot) 정점 인덱스의 임베딩을 수행함으로써 관절 특징을 동적으로 취합하는, 동작 처리 장치.

청구항 16

제 12 항에 있어서,

상기 처리부는,

특징을 취합하기 전에 어텐션을 적용하여 각각의 순위 하위 집합(subset)에 대한 마스크를 학습하되, 프레임 간에 공유되는 어텐션 마스크로서 정적 곱셈 마스크(static multiplication mask)를 적용하는, 동작 처리 장치.

발명의 설명

기 술 분 야

[0001] 본 명세서는 동작을 인식하는 기술에 관한 것으로, 특히 그래프 컨벌루션 네트워크(graph convolutional network)를 이용하는 스켈레톤 기반의 동작 인식 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 동작 인식은 컴퓨터 비전과 인공 지능에서 매우 중요한 작업이 되었다. 인간과 컴퓨터의 상호 작용, 게임, 비디오 감시, 비디오 이해 등 다양한 응용 분야에서 동작 인식이 널리 사용되기 때문이다. COVID-19 등 감염병 확산으로 집에서 보내는 시간이 늘어나면서 행동인식 재택 트레이닝 시스템이 더욱 요구되고 있다. 또한 행동인식의 적용 범위가 반려동물의 행동인식까지 확대되고 있다.

[0003] 활용되는 입력 데이터의 종류에 따라 동작 인식 방식은 크게 이미지 기반, 스켈레톤 기반, 하이브리드 방식으로 분류된다. 이미지 기반 접근 방식에서 이미지 쌍 간의 점 대응을 나타내는 광학 흐름(optical flow)은 일반적으로 관심 대상의 겉보기 움직임을 나타내는 데 사용되었다. 그러나 이러한 방법에는 종종 시간이 많이 걸리고 스토리지가 많이 필요한 하위 프로세스가 필요하다. 또한 이미지 기반 방법의 성능은 조명과 같은 광학 노이즈의 영향을 받을 수 있다. 이러한 문제가 완화되더라도 이미지 기반 접근 방식은 개인 식별 정보(personally identifiable information, PII) 문제에서 자유롭지 않다. 노인 환자를 위한 병원 서비스와 같은 실제 상황에서는 이러한 접근 방식이 제한적이다.

[0004] 이러한 맥락에서 스켈레톤 기반 접근법의 장점은 분명한다. 이미지 기반 접근 방식에서 광학 흐름을 추출하는 것처럼 관심 대상의 포즈를 설명하기 위해 연결된 좌표 집합을 의미하는 스켈레톤을 추출하는 과정은 비디오에서 수행된다. 그러나 이러한 유형의 방법은 표현이 간결하고 프라이버시가 없기 때문에 상대적으로 가볍다. Microsoft Kinect와 같은 비용 효율적인 깊이 센서와 Openpose와 같은 적절한 포즈 추정기의 보급으로 인해 이 백서에서 동작 인식 방법에 대한 스켈레톤 데이터를 쉽게 얻을 수 있다.

[0005] 초기 단계의 스켈레톤 기반 동작 인식 연구에서는 스켈레톤 시퀀스에서 의사(pseudo) 이미지를 생성하거나 CNN(convolutional neural networks)과 같은 포즈 예측 모델에서 히트맵(heatmap)을 얻었다. 이러한 접근 방식은 이미지 기반 방법과 유사하다. 그러나 의사 이미지와 같은 중간 형태의 데이터를 생성하는 것은 스켈레톤 데이터를 압축적으로 사용하는 것과 충돌하고 저사양 컴퓨터에서 더 깊은 신경망 학습을 방해한다. 따라서 스켈레톤 기반 동작 인식을 위해 CNN을 보다 일반적인 그래프 구조로 일반화하는 GCN(graph convolutional network)이 제시되었다.

선행기술문헌

비특허문헌

[0006] (비특허문헌 0001) S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in Proc. AAAI, 2018, pp. 7444-7452.

발명의 내용

해결하려는 과제

[0007] 본 명세서의 실시예들이 해결하고자 하는 기술적 과제는, 종래의 그래프 기반의 행동인식 기술에서 하나의 노드에서 근접 행렬(Adjacent Matrix)을 산출함에 있어서, 임의로 정해진 규칙을 따르거나 학습 파라미터를 사용하여 계산함으로 인해 나타나는 행동 인식 정확도가 낮거나 그래프 노이즈에 취약한 단점을 극복하고자 한다.

과제의 해결 수단

[0008] 상기 기술적 과제를 해결하기 위하여, 본 명세서의 일 실시예에 따른 그래프 컨벌루션 네트워크(graph convolutional network, GCN)를 이용한 동작 처리 방법은, 동작 처리 장치가 객체의 동작에 대한 스켈레톤(skeleton)을 포함하는 프레임을 입력받는 단계; 상기 동작 처리 장치가 상기 스켈레톤에 대해 하나의 노드와 다른 노드와의 거리 및 인접 순위를 고려한 순위 인접 행렬(rank adjacency matrix)을 이용하여 시공간 특징을 추출하는 단계; 상기 동작 처리 장치가 추출된 상기 시공간 특징에 기초하여 입력 프레임의 객체와 정점(vertex)을 병합하는 단계; 및 상기 동작 처리 장치가 분류 작업을 수행하는 단계;를 포함한다.

[0009] 일 실시예에 따른 동작 처리 방법에서, 상기 시공간 특징을 추출하는 단계는, 공간 그래프 컨벌루션 레이어 및 시간 컨벌루션 레이어를 각각 적어도 하나 이상 포함하는 모듈을 통해 수행되며, 상기 공간 그래프 컨벌루션 레이어는, 특징 채널 축을 따라 노드 인덱싱(node indexing)을 수행하고, 완전 연결 레이어(fully-connected layer)를 적용하며, 순위 인접 행렬을 사용하여 정점(vertex)을 취합하되 프레임 간에 공유되는 어텐션 마스크(attention mask)를 적용할 수 있다.

[0010] 일 실시예에 따른 동작 처리 방법에서, 상기 순위 인접 행렬은, 입력된 스켈레톤 데이터에 포함된 관절(joint)을 나타내는 노드 각각에 대해 다른 노드와의 거리를 산출하고, 산출된 거리가 작은 순서대로 정렬하되 순위 메트릭(metric)을 통해 상기 하나의 노드와 가까운 소정 개수의 다른 노드에 대해 인접 순위를 나타내는 행렬을 생성할 수 있다.

[0011] 나아가, 이하에서는 상기 기재된 동작 처리 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.

[0012] 상기 기술적 과제를 해결하기 위하여, 본 명세서의 일 실시예에 따른 동작 처리 장치는, 객체의 동작에 대한 스켈레톤(skeleton)을 포함하는 프레임을 입력받는 입력부; 및 그래프 컨벌루션 네트워크(graph convolutional network, GCN)를 이용하여 상기 프레임에 대한 동작을 처리하는 처리부;를 포함하고, 상기 처리부는, 상기 스켈레톤에 대해 하나의 노드와 다른 노드와의 거리 및 인접 순위를 고려한 순위 인접 행렬(rank adjacency matrix)을 이용하여 시공간 특징을 추출하고, 추출된 상기 시공간 특징에 기초하여 입력 프레임의 객체와 정점(vertex)을 병합하며, 분류 작업을 수행한다.

[0013] 일 실시예에 따른 동작 처리 장치에서, 상기 처리부는, 공간 그래프 컨벌루션 레이어 및 시간 컨벌루션 레이어를 각각 적어도 하나 이상 포함하는 모듈을 통해 상기 시공간 특징을 추출하며, 상기 공간 그래프 컨벌루션 레이어는, 특징 채널 축을 따라 노드 인덱싱(node indexing)을 수행하고, 완전 연결 레이어(fully-connected layer)를 적용하며, 순위 인접 행렬을 사용하여 정점(vertex)을 취합하되 프레임 간에 공유되는 어텐션 마스크(attention mask)를 적용할 수 있다.

[0014] 일 실시예에 따른 동작 처리 장치에서, 상기 처리부는, 입력된 스켈레톤 데이터에 포함된 관절(joint)을 나타내는 노드 각각에 대해 다른 노드와의 거리를 산출하고, 산출된 거리가 작은 순서대로 정렬하되 순위 메트릭(metric)을 통해 상기 하나의 노드와 가까운 소정 개수의 다른 노드에 대해 인접 순위를 나타내는 상기 순위 인접 행렬을 생성할 수 있다.

발명의 효과

[0015] 상기된 본 명세서의 실시예들에 따르면, 하나의 노드와 다른 노드들 간의 거리 및 인접 순위를 기준으로 순위 인접 행렬을 도출하여 GCN 아키텍처를 행동 인식에 적용하는 순위-GCN(Rank-GCN)을 제안하였다. 순위-GCN에서는 정점 사이의 쌍별(pairwise) 거리를 기반으로 순위 인접 그래프를 정의하고 가장 가까운 순위와 가장 먼 순위에 따라 정점 특징을 누적하는 방식을 도출함으로써, 행동 인식 정확도에 대한 성능 향상뿐만 아니라 보다 실용적인 시나리오에서 특정 노드의 교환, 위치 이동 및 드롭에 대한 강인성을 얻을 수 있다.

도면의 간단한 설명

[0016] 도 1은 부정확한 스켈레톤 정보가 동작 인식 성능에 영향을 줄 수 있는 두 가지 상황을 예시하였다.

도 2는 본 명세서의 실시예들이 제안하는 순위 그래프 컨벌루션 네트워크를 이용한 동작 인식을 위한 전체 모델 구조를 도시하였다.

도 3은 일 실시예에 따른 그래프 컨벌루션 네트워크(graph convolutional network, GCN)를 이용한 동작 처리 방법을 도시한 흐름도이다.

도 4는 본 명세서의 실시예들이 제안하는 '순위'의 개념을 설명하기 위해 인접 이웃을 정의하는 다양한 방법을 비교하여 도시한 도면이다.

도 5는 도 2의 동작 인식 모델에서 순위 그래프 컨벌루션 레이어를 보다 구체적으로 도시한 도면이다.

도 6은 순위 인접 행렬 생성 과정을 설명하기 위한 도면이다.

도 7은 도 5의 순위 그래프 컨벌루션 레이어에서 프레임별 동작 과정을 보다 구체적으로 도시한 도면이다.

도 8 및 도 9는 각각 순위 그래프 컨벌루션 레이어의 수행 과정을 설명하기 위한 도면 및 의사 코드(pseudocode)를 예시한 도면이다.

도 10은 일 실시예에 따른 그래프 컨벌루션 네트워크를 이용한 동작 처리 장치를 도시한 블록도이다.

도 11은 강인성(robustness) 테스트를 위한 세 가지 실험 설정을 예시한 도면이다.

도 12 내지 도 17은 도 11의 강인성 테스트의 결과를 예시한 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0017] 본 명세서의 실시예들을 구체적으로 설명하기에 앞서, 본 명세서의 실시예들이 구현되는 동작 인식 기술 분야에서 나타나는 목표와 이를 해결하기 위해 고려될 수 있는 기술적 수단을 순차적으로 소개하도록 한다.

[0018] GCN(graph convolutional network)은 2D 또는 3D 관절(joint) 좌표를 노드로, 연결(즉, 뼈)을 에지(edge)로 구성한 불규칙 그래프 위에 인접한 정점에 대한 특징을 취합하여 정점(vertex)(즉, 스켈레톤의 관절)에서 특징을 학습한다. 입력 데이터의 공간적 차원과 시간적 차원 모두에 적용된다. 이 취합 전략에 따라 다양한 방법을 구분할 수 있다. 단순화를 위해 신체 관절 사이의 물리적 연결이 사용되었지만 이상적인 특징 취합 전략은 로컬 이웃을 넘어 구조적으로 떨어져 있더라도 강한 상관 관계가 있는 노드 간의 장거리 종속성을 반영해야 한다. 따라서 이전에 개발된 방법은 경험적으로(heuristically) 이웃 정점을 미리 정의하고 데이터에서 인접 정보를 학습하는 것과 같은 암시적 방법이다.

[0019] 그러나 전역 이웃을 사용하더라도 인접 정보는 일반적으로 입력 비디오의 시간적 차원에 대해 고정되어 있으며 스켈레톤 기반 방법은 이미지 기반 방법이 광학 노이즈에 민감한 것처럼 관절 좌표에서 민감한 노이즈이다.

[0020] 도 1은 부정확한 스켈레톤 정보가 동작 인식 성능에 영향을 줄 수 있는 두 가지 상황을 예시하였다. 왼쪽 사진은 한 팔로 사탕을 나누는 사람을 보여주고 오른쪽 사진은 책상 앞에서 팔을 쭉 뻗고 있는 사람을 보여준다. 두 경우 모두 점선으로 표시된 원과 선이 불가피하게 이동되어 부정확한 스켈레톤 정보를 보여준다. 이러한 문제 인식에 기반하여 본 명세서의 실시예들은 역동적이고 물리적으로 의미 있는 방식으로 작업이 실제로 발생하는 위치(노란색 원과 선)에 초점을 맞춤으로써 강인성을 높일 수 있는 기술적 수단을 제안하고자 한다.

[0021] 이하에서 제시되는 본 명세서의 실시예들은 시간적 차원을 따라 인접 행렬을 동적으로 계산하는 효과적이고 강력한 프레임워크인 순위 그래프 컨벌루션 네트워크(Rank-GCN)을 제안한다. 제안하는 실시예들의 주요 목표는 다음과 같다.

[0022] 순위 그래프 컨벌루션 네트워크는 공간적 차원과 시간적 차원에서 글로벌 정보를 사용하는 새로운 방법을 제안한다. 동적 인접 행렬 생성을 위해 학습 가능한 매개변수를 사용하는 이전 방법과 비교할 때 순위 그래프 컨벌루션 네트워크는 매개변수가 적고 구현하기 쉽고 더 해석 가능한 결과를 생성한다. 인간이 만든 방법은 딥 러닝 기반 방법보다 취약하다는 인식이 있지만, 순위 그래프 컨벌루션 네트워크의 접근 방식은 이전 방법보다 더 나은 성능을 보여줄 뿐만 아니라 해석 가능한 전망도 있다.

[0023] 기하학적 거리 측정을 사용하고 순위 그래프 컨벌루션 알고리즘을 도입하여 인접 행렬을 계산하는 문제를 해결한다. 예를 들어, 거리 임계값을 직접 사용하는 대신 거리 순위를 사용한다. 인접한 관절 그룹을 결정하기 위해 순위를 사용함으로써 인접 노드가 더 잘 활용되고 최첨단 방법보다 활동 인식에서 성능과 강인성을 제공한다.

[0024] 이하에서는 도면을 참조하여 본 명세서의 실시예들을 구체적으로 설명하도록 한다. 다만, 하기의 설명 및 첨부된 도면에서 실시예들의 요지를 흐릴 수 있는 공지 기능 또는 구성에 대한 상세한 설명은 생략한다. 덧붙여, 명세서 전체에서, 어떤 구성 요소를 '포함'한다는 것은, 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라, 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0025] 본 명세서에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 명세서를 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "구비하다" 등의 용어는 설시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0026] 특별히 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 명세서가 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미이다. 일

반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미인 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0027] CNN 기반 스켈레톤 동작 인식

[0028] 초기 스켈레톤 기반 행동 인식 연구에서는 전통적인 CNN 모델이 대부분 채택되었다. CNN 모델을 활용하기 위해 일부 구현예들은 일련의 스켈레톤을 3채널 이미지로 전처리하여 의사 이미지를 생성했다. 예를 들어, 전면, 상단 및 측면의 세 가지 다른 보기에서 관절 궤적의 컬러 맵을 만든 다음 이 세 가지 다른 보기의 예측 점수를 융합했다. BPI(Body Pose Evolution Image) 및 BSI(Body Shape Evolution Image) 접근법은 각각 관절의 시간축을 따라 순위 풀링을 적용하고 3D 관절의 정규화된 좌표를 연결하여 사용되었다.

[0029] 스켈레톤을 이용한 의사 영상 기반 동작 인식의 단점은 컨벌루션 연산이 인접한 관절에만 적용되고 그리드(grid)에 이미지로 표현된다는 점이다. 즉, 여러 가지 그럴듯한 관절 조합을 공동으로 고려해야 하지만 커널 크기 3의 컨벌루션으로 세 개의 관절만 고려한다. 이를 해결하기 위해 BSI 방식은 인체를 따라 움직이는 중복 관절을 설정한다. 한편, HCN은 채널의 관절 축을 교체하는 추가 레이어를 구현하여(T×V×C에서 T×C×V로) VGG-19의 수정된 버전으로 구성되었다. 이러한 스와핑 레이어는 추가 비용 없이 상당한 성능 향상을 가져오며, 광범위한 인접 관절에서 수행되는 비로컬 작업이 동작 인식에 중요함을 보여준다.

[0030] 히트맵 기반의 3D CNN 동작 인식 모델인 PoseC3D도 소개되었다. PoseC3D는 3D 또는 (2+1)D 컨벌루션 레이어를 사용하여 시공간적 특징을 추출하는 3D CNN 모델의 변형이다. 2D 스켈레톤 입력에서 생성된 각 관절의 히트맵은 PoseC3D에서도 사용된다. PoseC3D는 3D 레이어의 깊은 블록을 쌓아 시공간적 특징을 추출함으로써 지역성 문제를 해결한다. 하지만 PoseC3D는 심도가 깊기 때문에 GCN 기반 모델보다 계산 비용이 많이 든다. 또한 PoseC3D 모델은 GCN 기반 모델보다 관절 감지 실패에 대해 견고하다는 것을 보여주었다.

[0031] 스켈레톤 동작 인식을 위한 GCN

[0032] CNN 기반 방법의 변형에 대한 위의 관찰에서 "이웃 관절의 인접성"이라는 개념을 사용하여 GCN을 사용한 동작 인식이 기존 CNN 기반 방법보다 더 잘 수행됨을 추론할 수 있다.

[0033] GCN은 ST-GCN(spatiotemporal GCN)으로 수정되어 처음으로 동작 인식에 사용되었다. ST-GCN이 제안된 후 많은 다른 유사한 방법이 탐색되었다. GCN의 확장인 ST-GCN은 인접한 관절을 공간 영역에서 그룹으로 나누는 부분 집합 분할 방법으로 개발되었으며, 시간 영역에서 각 관절의 역학을 캡처하기 위해 1D 컨벌루션 계층을 사용했다. AGCN(adaptive GCN)과 AAGCN(adjacency-aware GCN)에서는 중간 특징의 외적을 적용하고 결합하여 학습 가능한 인접 행렬을 만든다.

[0034] MS-G3D는 인접 행렬을 시간 방향의 추가 차원으로 확장하여 공간 관계에 비해 더 포괄적인 범위의 시공간 노드를 캡처할 수 있다. Shift-CNN에서 영감을 얻은 Shift-GCN은 인접 행렬을 사용하여 특징을 취합하는 대신 이동 메커니즘으로 구현되었다. Efficient-GCN은 계산에 더 적은 매개변수를 사용하기 위해 분리 가능한 컨벌루션 레이어를 포함하고 입력 데이터 스트림에 대해 초기 융합 방법을 채택했다. 특히 초기 융합을 채택하여 멀티스트림 앙상블에 대한 모델 매개변수의 수를 획기적으로 줄였다. 다른 GCN 모델과 달리 Distance-GCN은 관절 사이의 유클리드 거리를 기반으로 새로운 인접 행렬을 만들었으며, 관절 사이의 쌍별(pairwise) 거리를 활용하면 단순히 물리적으로 연결된 인접성을 사용하는 대신 동작 인식 성능이 향상된다는 것을 보여주었다.

[0035] 동작 인식을 위한 GCN에 대한 관련 연구에서 인접 행렬을 설계하는 것이 성능에 중요한 영향을 미친다고 말할 수 있다. 본 명세서의 실시예들은, Distance-GCN에서 나아가, 이웃 관절을 분할하기 위해 실제 메트릭을 채택하는데, 관심 관절로부터의 거리에 따라 이웃 관절을 순서대로 정렬하는 기술적 수단을 제안한다.

[0036] 도 2는 본 명세서의 실시예들이 제안하는 순위 그래프 컨벌루션 네트워크를 이용한 동작 인식을 위한 전체 모델 구조를 도시하였다. GCN 기반 동작 인식 모델과 유사하게, 본 명세서의 실시예들이 제안하는 Rank-GCN은 3개의 채널 스테이지가 있는 인터리빙 공간 그래프 컨벌루션 레이어와 시간 1D 컨벌루션 레이어의 10개 블록의 유사한 구성을 사용하여 구현될 수 있다. 여기서, 공간 컨벌루션을 위해 Rank GCN 레이어를 사용하였다.

[0037] 우선, 객체의 동작에 대한 스켈레톤(skeleton)(210)을 포함하는 프레임을 입력받는다. 입력 데이터 크기는 P×T×V×C이며, 여기서 P는 시퀀스(sequence)에 있는 사람 수, T는 프레임 수, V는 관절 수, C는 2D 또는 3D 좌표의 차원이다. 어텐션(attention) 메커니즘으로 수정되거나 실험적으로 구성될 수 있는 미리 정의된 고정 그래프일 수 있는 인접 행렬 A로 표현되는 그래프가 주어지면, 공간 그래프 컨벌루션 레이어와 1D 시간 컨벌루션 레이

어의 여러 블록을 입력 데이터에 적용하여 고차원의 시공간(spatiotemporal) 특징을 추출한다(220). 특히, 스켈레톤에 대해 하나의 노드와 다른 노드와의 거리 및 인접 순위를 고려한 순위 인접 행렬(rank adjacency matrix)을 이용하여 시공간 특징을 추출한다. 여기서, 시공간 특징을 추출하는 과정은, 도 2의 공간 그래프 컨벌루션 레이어 및 시간 컨벌루션 레이어를 각각 적어도 하나 이상 포함하는 모듈(220)을 통해 수행되며, 예를 들어, 3개의 채널 스테이지로 구성되고, 각각의 채널 스테이지는 공간 그래프 컨벌루션 레이어 및 시간 컨벌루션 레이어의 블록이 4개, 3개 및 3개로 구성되어 총 10개의 블록을 형성하는 모듈을 통해 수행될 수 있다.

[0038] 그런 다음, 추출된 상기 시공간 특징에 기초하여 입력 프레임의 객체(예를 들어, 사람)와 객체들의 정점(vertex)을 병합한다. 이를 위해, 전역 평균 풀링(global averaging pooling, GAP)(230)이 적용될 수 있다. 마지막으로, 분류 작업을 수행한다. 이를 위해, 소프트맥스(softmax)(240)가 적용될 수 있다.

[0039] 도 3은 일 실시예에 따른 그래프 컨벌루션 네트워크(graph convolutional network, GCN)를 이용한 동작 처리 방법을 도시한 흐름도이다.

[0040] S310 단계에서, 동작 처리 장치는, 객체의 동작에 대한 스켈레톤(skeleton)을 포함하는 프레임을 입력받는다.

[0041] S330 단계에서, 상기 동작 처리 장치는, 상기 스켈레톤에 대해 하나의 노드와 다른 노드와의 거리 및 인접 순위를 고려한 순위 인접 행렬(rank adjacency matrix)을 이용하여 시공간 특징을 추출한다. 이 과정에서는 스켈레톤을 구성하는 노드(또는 정점)에 대해 단지 1-홉(hop)의 연결 관계만을 고려하거나 단순 거리만을 고려하는 것이 아니라, 스켈레톤의 구조에만 의존하지 않는 순위 기반의 근접성을 고려하였다. 인체의 구조상 멀리 떨어진 부분 인접하는 경우(예를 들어, 손과 입이 인접하는 경우)에도 인접한 거리에 따라 노드들을 가까운 순서대로 정렬하고 일정한 개수의 노드를 인접 노드로 파악하는 접근 방식을 제안한다. 이 과정은 공간 그래프 컨벌루션 레이어 및 시간 컨벌루션 레이어를 각각 적어도 하나 이상 포함하는 모듈을 통해 수행될 수 있다.

[0042] S350 단계에서, 상기 동작 처리 장치는, S330 단계를 통해 추출된 상기 시공간 특징에 기초하여 입력 프레임의 객체와 정점(vertex)을 병합한다.

[0043] S370 단계에서, 상기 동작 처리 장치는, 분류 작업을 수행한다.

[0044] 앞서 기술한 S330 단계에서 시공간 특징을 추출하기 위한 공간 그래프 컨벌루션 연산은 다음 수학식으로 정리될 수 있다.

수학식 1

[0045]
$$v_{i,out} = \sum_{v_j \in \mathcal{N}(v_{i,old})} FC(v_{j,in}) \cdot A_{ij}$$

[0046] 여기서 v, A, N 및 FC()는 각각 정점, 인접 행렬, 인접 노드 집합 및 완전 연결 레이어(fully-connected layer)를 나타낸다.

[0047] 도 2를 통해 예시한 바와 같이, 본 명세서의 실시예들이 제안하는 순위 그래프 컨벌루션 네트워크(Rank-GCN) 모델은 또한 공간적 특징을 위한 10개의 인터리빙 순위 그래프 컨벌루션 레이어와 시간적 특징을 위한 1D 컨벌루션 레이어로 구성될 수 있다. 공간적 특징 외에도 입력 스트림 및 인접 행렬에 따라 순위 그래프 컨벌루션 레이어는 시공간적 특징을 추출하여 신체 제스처의 보다 복잡한 표현을 얻을 수 있다.

[0048] 도 4는 본 명세서의 실시예들이 제안하는 '순위'의 개념을 설명하기 위해 인접 이웃을 정의하는 다양한 방법을 비교하여 도시한 도면으로, 관심 정점을 검정색 노드로 표시하였다.

[0049] 도 4의 (a)는 ST-GCN, AGCN 및 AAGCN에서 사용될 수 있는 방식으로, ×는 가상 노드를 나타낸다. 도 4의 (a)에서는 관심 정점(검정색 노드)의 근접 노드로 점선 안의 노드를 사전에 정의하여 사용할 수 있는데, 이 경우 물리적으로 연결된 1-홉(hop) 이웃만 고려하기 때문에 로컬 이웃의 커버리지가 매우 제한적이다. 반면, 도 4의 (b)와 (c)에서는 광범위한 이웃을 다룬다.

[0050] 도 4의 (b)는 Distance-GCN 방법으로, D_1과 D_2는 (b)의 관심 노드를 중심으로 하는 동심원의 반지름을 나타낸다. 도 4의 (c)는 본 명세서의 실시예들이 제안하는 순위-GCN 방법이 적용되었다. 관심 정점(검정색 노드)에서 가까운 거리를 기준으로 근접 노드를 동적으로 정의하되, 가까운 순서(rank) 정보를 기준으로 인접도

를 계산하면 정확도 및 안정성 측면에서 행동 인식 성능이 향상됨을 확인할 수 있었다.

[0051] 도 4의 (b)와 (c)를 비교하면, (b)의 반지름이 D_1인 실선과 반지름이 D'_1인 점선 원은 일부 하위 집합에 대해 두 가지 가능한 범위를 보여준다. 3개의 관절 중 2개는 상기 일부 하위 집합이 점선 원과 같이 훈련 과정에서 학습된 '약간' 더 작은 범위를 갖는 경우 제외된다. 그리고, 이는 요소(관절)의 수가 변경되기 때문에 성능에 영향을 미칠 수 있다. 도 4의 (c)의 방법은 순위 전략을 채택함으로써 이웃 노드의 약간의 거리 변화에 영향을 받지 않고 각 하위 집합에 대해 안정적인 요소 수를 유지한다. D'_1의 거리에 있는 노드와 비교할 때 도 4의 (b)에서 불안정성 문제가 발생할 수 있는 반면 Rank-GCN 파티션 그룹은 (c)에서 안정적이다.

[0052] 이하에서는 도 4 내지 도 6을 참조하여 그래프 기반 동작 인식을 위한 Rank-GCN의 주요 모듈인 순위 그래프 컨벌루션 레이어 모듈을 소개한다.

[0053] 도 5는 도 2의 동작 인식 모델에서 순위 그래프 컨벌루션 레이어를 보다 구체적으로 도시한 도면이다.

[0054] 본 명세서의 실시예들이 제안하는 공간 그래프 컨벌루션 레이어는, 특징 채널 축을 따라 노드 인덱싱(node indexing)을 수행하고, 완전 연결 레이어(fully-connected layer)를 적용하며, 순위 인접 행렬을 사용하여 정점(vertex)을 취합하되 프레임 간에 공유되는 어텐션 마스크(attention mask)를 적용할 수 있다.

[0055] 노드 인덱싱(node indexing)

[0056] 동작 인식을 위한 기존의 GCN은 고정된 규칙으로 관절 특징을 취합(aggregate)하므로 주어진 지점에서 어떤 노드가 취합될지 예상할 수 있다. 그러나 동적으로 취합이 수행되는 경우 고정된 취합 관절 세트가 없을 수 있다. 이 문제를 해결하기 위해 특징 채널 축(axis)을 따라 원-핫(one-hot) 정점 인덱스의 임베딩을 추가함으로써, 관절 특징을 동적으로 취합할 수 있다

[0057] 순위 인접 행렬(rank adjacency matrix)

[0058] 도 6은 순위 인접 행렬 생성 과정을 설명하기 위한 도면이다.

[0059] 동작 역학을 보다 효과적으로 캡처하기 위해, 새로운 인접 행렬 생성 방법인 순위 인접 행렬을 제안한다. 시간 t의 프레임과 관심 있는 중심 노드 v_t^i가 주어지면 스칼라 값을 출력하는 메트릭 함수 $\mathcal{M}$을 사용하여 노드 사이의 거리를 계산한다. 거리 행렬 D_t^i는 입력 프레임의 모든 노드를 다음과 같이 반복하여 얻는다.

수학식 2

[0060] $$D_t^i = \{\mathcal{M}(v_t^i, v_t^j) | j = 1, \ldots, V\} \in \mathbb{R}^{V \times 1}$$

[0061] 여기서 v_t^i은 정점의 좌표, 속도 및 가속도일 수 있다.

[0062] 거리 행렬을 기반으로 순위 행렬 $A_{ti} \in \mathbb{Z}^{R \times V \times 1}$은 순위 범위 $\Gamma = \{\gamma_r = (s_r, e_r) | r = 1, \ldots, R\}$로 순위를 매기고 필터링하여 도출된다. 여기서 s_r과 e_r는 각각 범위의 시작과 끝이고 R은 순위 범위의 수이다. 따라서 정점 i 및 순위 r에 대한 프레임 t에서의 순위 메트릭은 다음과 같다.

수학식 3

[0063] $$A_{t,r}^i = filter(rank(D_t^i), \gamma_r) \in \{0, 1\}^{V \times 1}$$

[0064] 스켈레톤 입력이 주어지면 스켈레톤의 프레임을 $S = \{S_t \in \mathbb{R}^{V \times C} | t = 1, \ldots, T\}$로 표시한다. 순위 그래프 컨벌루션 레이어는 도 9를 통해 제시된 알고리즘에 따라 동작한다.

[0065] 도 7은 도 5의 순위 그래프 컨벌루션 레이어에서 프레임별 동작 과정을 보다 구체적으로 도시한 도면으로, 주어진 순위 인접 행렬을 사용하여 정점(vertex)을 취합하고 프레임 간에 공유되는 어텐션 마스크(attention mask)

를 적용하는 방법을 보여준다.

[0066] 이상의 수학식 2 및 3을 통해 살펴본 바와 같이, 순위 인접 행렬은, 입력된 스켈레톤 데이터에 포함된 관절(joint)을 나타내는 노드 각각에 대해 다른 노드와의 거리를 산출하고, 산출된 거리가 작은 순서대로 정렬하되 순위 메트릭(metric)을 통해 상기 하나의 노드와 가까운 소정 개수의 다른 노드에 대해 인접 순위를 나타내는 행렬을 생성할 수 있다. 구현의 관점에서, 순위 인접 행렬은, 입력 프레임의 모든 노드에 대해 하나의 노드와 다른 노드간의 유클리디언 거리(Euclidean distance)를 산출하여 거리 행렬을 생성하고, 생성된 상기 거리 행렬을 기반으로 순위 범위가 설정된 순위 메트릭을 통해 순위 부여 및 필터링을 통해 인접 순위를 나타내는 행렬을 생성할 수 있다.

[0067] 순위별 어텐션 마스크(rank-wise attention mask)

[0068] 어텐션 메커니즘으로 인한 많은 성능 향상으로 인해 어텐션 모듈도 새롭게 고안하였다. 예를 들어, 인접 행렬 또는 취합된 특징에 어텐션 메커니즘을 적용할 수 있으나, 본 명세서의 실시예들에서는 사전 취합 단계에 어텐션을 적용하여 각 순위 하위 집합(subset)에 대한 최적의 마스크를 학습할 수 있다. 어텐션 모듈을 가능한 한 가볍게 만들기 위해 알고리즘 1에서 M으로 표시되는 간단한 정적 곱셈 마스크(static multiplication mask)를 채택할 수 있다. 마스크 모듈은 순위별, 정점별 및 채널별 곱셈 마스크로, 계산 복잡성과 가중치 매개변수 수가 약간만 증가하여 일관된 성능 향상을 가져올 수 있다. 요약하건대, 어텐션 마스크는, 특징을 취합하기 전에 어텐션을 적용하여 각각의 순위 하위 집합에 대한 마스크를 학습하되, 프레임 간에 공유되는 어텐션 마스크로서 정적 곱셈 마스크를 적용할 수 있다.

[0069] 이상에서 설명한 모듈들을 함께 결합하여 입력된 스켈레톤 데이터에 따라 매 프레임마다 다르게 특징을 취합하는 순위-GCN(Rank-GCN) 레이어를 제안하였다. 순위-GCN 레이어의 전체 프로세스는 도 9의 알고리즘에 제시되었다. 여기에서 메트릭 함수 $\mathcal{M}$을 변경하여 다양한 종류의 순위 행렬 A를 활용할 수 있다.

[0070] 도 8 및 도 9는 각각 순위 그래프 컨벌루션 레이어의 수행 과정을 설명하기 위한 도면 및 의사 코드(pseudocode)를 예시한 도면이다. 여기서, T는 시퀀스(sequence)의 수이고, V는 관절(joint)의 수이며, C는 채널의 수이다. 도 9를 참조하면, 거리가 가장 가까운 순서(rank)를 기준으로 근접 행렬 A를 계산하고 이를 기반으로 행동 인식에 사용되는 특징을 추출하는 그래프 컨벌루션 알고리즘을 제시하고 있다.

[0071] 도 10은 일 실시예에 따른 그래프 컨벌루션 네트워크를 이용한 동작 처리 장치(20)를 도시한 블록도로서, 도 3의 동작 처리 방법을 하드웨어 구성의 관점에서 재구성한 것이다. 따라서, 여기서는 설명의 중복을 피하고자 각 구성의 동작 및 기능의 개요만을 약술하도록 한다.

[0072] 입력부(21)는, 객체의 동작에 대한 스켈레톤(skeleton)(10)을 포함하는 프레임을 입력받는 구성이다.

[0073] 처리부(23)는, 그래프 컨벌루션 네트워크(graph convolutional network, GCN)를 이용하여 상기 프레임에 대한 동작을 처리하는 구성으로, 상기 스켈레톤에 대해 하나의 노드와 다른 노드와의 거리 및 인접 순위를 고려한 순위 인접 행렬(rank adjacency matrix)을 이용하여 시공간 특징을 추출하고, 추출된 상기 시공간 특징에 기초하여 입력 프레임의 객체와 정점(vertex)을 병합하며, 분류 작업을 수행한다.

[0074] 상기 처리부(23)는, 공간 그래프 컨벌루션 레이어 및 시간 컨벌루션 레이어를 각각 적어도 하나 이상 포함하는 모듈을 통해 상기 시공간 특징을 추출하며, 상기 공간 그래프 컨벌루션 레이어는, 특징 채널 축을 따라 노드 인덱싱(node indexing)을 수행하고, 완전 연결 레이어(fully-connected layer)를 적용하며, 순위 인접 행렬을 사용하여 정점(vertex)을 취합하되 프레임 간에 공유되는 어텐션 마스크(attention mask)를 적용할 수 있다.

[0075] 상기 처리부(23)는, 입력된 스켈레톤 데이터에 포함된 관절(joint)을 나타내는 노드 각각에 대해 다른 노드와의 거리를 산출하고, 산출된 거리가 작은 순서대로 정렬하되 순위 메트릭(metric)을 통해 상기 하나의 노드와 가까운 소정 개수의 다른 노드에 대해 인접 순위를 나타내는 상기 순위 인접 행렬을 생성할 수 있다. 보다 구체적으로, 상기 처리부(23)는, 입력 프레임의 모든 노드에 대해 하나의 노드와 다른 노드간의 유클리디언 거리(Euclidean distance)를 산출하여 거리 행렬을 생성하고, 생성된 상기 거리 행렬을 기반으로 순위 범위가 설정된 순위 메트릭을 통해 순위 부여 및 필터링을 통해 인접 순위를 나타내는 상기 순위 인접 행렬을 생성할 수 있다.

[0076] 상기 처리부(23)는, 특징 채널 축을 따라 원-핫(one-hot) 정점 인덱스의 임베딩을 수행함으로써 관절 특징을 동적으로 취합할 수 있다. 또한, 상기 처리부(23)는, 특징을 취합하기 전에 어텐션을 적용하여 각각의 순위 하위 집합(subset)에 대한 마스크를 학습하되, 프레임 간에 공유되는 어텐션 마스크로서 정적 곱셈 마스크(static

multiplication mask)를 적용할 수 있다.

[0077] 상기된 본 명세서의 실시예들은 순위-GCN(Rank-GCN)이라고 명명한 새로운 인접 행렬을 사용하여 강력한 스켈레톤 기반 동작 인식 방법을 제시하였다. 순위-GCN는 '인접성(adjacency)'의 개념을 재정의하여 인접 노드에서 특징을 누적하기 위해 인접성 행렬을 생성하는 방식을 제공한다. 순위 인접 행렬(rank adjacency matrix)이라고 명명한 새로운 인접 행렬은 관심 노드로부터의 유클리드 거리를 포함한 메트릭(metric)에 따라 모든 노드의 순위를 지정하여 생성될 수 있다. 이러한 방식은 인접성을 구성하기 위해 1-홉(hop) 인접 노드만을 사용하는 GCN 방식과 차별화된다.

[0078] 이하에서는 서로 다른 메트릭 함수를 서로 다른 입력 스트림과 결합하여 결과 모델이 상호 보완적인지 확인하기 위한 실험 및 분석 결과를 제시한다. 실험에서는 다음 세 가지 데이터 세트를 사용하였다.

[0079] 첫째, NTU RGBCD 60은 RGB, 깊이 맵, 적외선 및 3D 스켈레톤 데이터의 네 가지 다른 양식이 포함된 데이터 세트이다. 여기에는 40개의 피사체, 3개의 카메라 뷰 및 60개의 동작이 포함된 56,880개의 샘플이 포함되어 있다. CS(cross-subject) 및 CV(cross-view)의 두 개의 공식 벤치마크 훈련-테스트 스플릿(split)이 활용된다. 데이터 세트에는 3D의 RGB, 깊이 맵 및 적외선 방식으로 투영되는 2D 스켈레톤도 포함된다. 3D 스켈레톤 데이터는 미터 단위로 표시된다. 모든 양식은 Kinects-V2 센서에 의해 캡처되고 3D 스켈레톤은 깊이 맵에서 추론된다. 깊이 맵 및 ToF 센서의 한계로 인해 일부 샘플은 스켈레톤 좌표에 많은 노이즈가 있다. 샘플의 길이는 최대 300프레임이고 보기의 사람 수는 최대 4명이다. 가장 활동적인 두 사람과 300프레임을 선택한다. 300개 미만의 프레임 또는 2명 미만의 샘플은 AAGCN에서 사용되는 방법으로 전처리된다.

[0080] 둘째, NTU RGBCD 120은 60개의 새로운 작업 클래스가 추가된 NTU RGBCD 60의 증분 확장이다. XSet(cross-set) 및 XSub(cross-subject)의 두 개의 공식 벤치마크 훈련-테스트 스플릿을 사용한다. NTU RGBCD 60에 사용된 것과 동일한 전처리 방법이 적용된다.

[0081] 셋째, Skeletics-152는 VIBE 포즈 추정기로 Kinetics700 데이터 세트에서 추출한 스켈레톤 동작 데이터 세트이다. Kinetics-700에는 사람이 없는 활동도 있고 인간이 상호작용하는 맥락 내에서 분류해야 하는 활동도 있기 때문에 총 700개 클래스 중 152개 클래스가 Skeletics-152를 만들기 위해 선택된다. VIBE 포즈 추정기의 정확한 포즈 추정 성능으로 인해 Skeletics-152는 NTU-60 스켈레톤보다 노이즈가 훨씬 적다. 샘플에 나타나는 사람의 수는 1에서 10까지이며 평균은 2.97이고 표준편차는 2.8이다. 샘플의 길이는 25프레임에서 300프레임 범위이며 평균은 237.8이고 표준편차는 74.72이다. 수행된 모든 실험에 대해 샘플에서 최대 2명을 선택한다. NTU-60에는 관절의 좌표가 미터 단위로 포함되어 있지만, Skeletics-152에는 [0,1] 범위의 정규화된 값이 있다. 프레임이 300개 미만이거나 사람이 3명 미만인 샘플은 0으로 채워지고 훈련 및 테스트를 위해 추가 전처리가 수행되지 않는다.

[0082] 본 명세서의 실시예들에 따른 순위-GCN 방법의 강인성(robustness)을 입증하기 위해 도 11과 같이 세 가지 다른 실험 설정을 구성하여 시각화하였다. 도 11의 (a)는 무작위 변환(translation)이고, (b)는 관절의 무작위 누락(drop)이며, (c)는 관절의 무작위 뒤바꿈(swap)으로, 모든 수정은 프레임 방식으로 이루어진다.

[0083] NTU RGBCD 60 데이터 세트의 CS 분할을 실험한다. 포즈 추정이 개선되었지만 추론된 관절 사이의 정렬 불량이 여전히 발생할 수 있다. 다음 실험의 캡처 장치인 Kinects V2는 자주 떨리는 문제가 있다. 이 실험에서는 스트림의 앙상블이 아닌 관절 스트림만 사용한다. 비교를 위해 제안 모델의 상위 버전으로 MS-G3D를 선택하고 하위 버전으로 AAGCN을 선택한다. 우수한 모델은 다양한 오류에 취약할 수 있다는 점을 우려하여 AAGCN을 비교 모델로 설정하였다.

[0084] 무작위 변환(translation) 실험을 위해, 길이는 같지만 방향이 다른 벡터로 모든 관절을 변환한다. 모든 프레임과 각 프레임의 모든 조인트에 대해 [0,1] 범위에서 균일하게 생성된 변환 벡터가 적용된다. 여기서 1은 변환 벡터 중 최대 길이이다. 도 12 및 도 13은 무작위 변환의 실험 결과를 보여준다.

[0085] 정점의 무작위 누락(drop) 실험은 관절 하위 집합의 추론이 폐색 또는 포즈 예측 시스템 오류로 인해 실패한다고 가정한다. 총 25개의 관절 중 d개의 관절을 선택하고 선택한 관절을 0.5의 확률로 (0,0,0)으로 설정한다. 도 14 및 도 15에서 볼 수 있듯이 d = 0, 1, 2, 3, 4, 5로 실험한다. 임의의 관절을 누락시켰을 때 본 명세서의 실시예들에 따른 순위-GCN 모델이 다른 모델을 능가한다는 것을 확인하였고, 실험 결과는 순위-GCN 모델이 관절 동작 인식 모델이 부분 관절 하위 집합에 액세스할 수 없는 열악한 환경에서 다른 모델보다 더 견고함을 시사한다.

[0086] 정점의 무작위 뒤바꿈(swap) 실험에서는 모든 관절을 무작위 순서로 치환한다. 각 테스트 인스턴스에 대해 임의의

의 시작점을 사용하여 순열 시퀀스 1의 길이를 0에서 300으로 변경한다. 도 16의 결과는 다른 두 가지 강인성 테스트와 달리 AAGCN의 성능이 급격히 떨어짐을 보여준다. 이는 AAGCN의 인스턴스별 인접 행렬 생성이 유해한 결과를 초래하는 반면, 순위-GCN 모델의 동적 순위 인접 행렬 접근 방식은 순열 관절을 매우 잘 처리한다는 것을 의미한다. 다만, 도 17에서 다중 스트림의 경우 Efficient-GCN이 Rank-GCN보다 우수함을 보여준다. Efficient-GCN의 전처리 전략으로 인해 Efficient-GCN의 아키텍처가 순위-GCN 방법보다 이 실험에 더 적합하다고 가정하였다.

[0087] 상기된 본 명세서의 실시예들에 따르면, 하나의 노드와 다른 노드들 간의 거리 및 인접 순위를 기준으로 순위 인접 행렬을 도출하여 GCN 아키텍처를 행동 인식에 적용하는 순위-GCN(Rank-GCN)을 제안하였다. 순위-GCN에서는 정점 사이의 쌍별(pairwise) 거리를 기반으로 순위 인접 그래프를 정의하고 가장 가까운 순위와 가장 먼 순위에 따라 정점 특징을 누적하는 방식을 도출함으로써, 행동 인식 정확도에 대한 성능 향상뿐만 아니라 보다 실용적인 시나리오에서 특정 노드의 교환, 위치 이동 및 드롭에 대한 강인성을 얻을 수 있다.

[0088] 본 명세서에 따른 실시예는 다양한 수단, 예를 들어, 하드웨어, 펌웨어(firmware), 소프트웨어 또는 그것들의 결합 등에 의해 구현될 수 있다. 하드웨어에 의한 구현의 경우, 본 명세서의 일 실시예는 하나 또는 그 이상의 ASICs(application specific integrated circuits), DSPs(digital signal processors), DSPDs(digital signal processing devices), PLDs(programmable logic devices), FPGAs(field programmable gate arrays), 프로세서, 콘트롤러, 마이크로 콘트롤러, 마이크로 프로세서 등에 의해 구현될 수 있다. 펌웨어나 소프트웨어에 의한 구현의 경우, 본 명세서의 일 실시예는 이상에서 설명된 능력 또는 동작들을 수행하는 모듈, 절차, 함수 등의 형태로 구현될 수 있다. 소프트웨어 코드는 메모리에 저장되어 프로세서에 의해 구동될 수 있다. 상기 메모리는 상기 프로세서 내부 또는 외부에 위치하여, 이미 공지된 다양한 수단에 의해 상기 프로세서와 데이터를 주고 받을 수 있다.

[0089] 한편, 본 명세서의 실시예들은 컴퓨터로 읽을 수 있는 기록 매체에 컴퓨터가 읽을 수 있는 코드로 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다. 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등을 포함한다. 또한, 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다. 그리고 실시예들을 구현하기 위한 기능적인(functional) 프로그램, 코드 및 코드 세그먼트들은 본 명세서가 속하는 기술 분야의 프로그래머들에 의하여 용이하게 추론될 수 있다.

[0090] 하나 이상의 명령어(instructions)을 저장하는 하나 이상의 비-일시적인(non-transitory) 컴퓨터 판독 가능 매체(computer-readable medium)에 있어서, 하나 이상의 프로세서에 의해 실행 가능한(executable) 상기 하나 이상의 명령어는, 그래프 컨벌루션 네트워크(graph convolutional network, GCN)를 이용한 동작을 처리하되, 객체의 동작에 대한 스켈레톤(skeleton)을 포함하는 프레임을 입력받고, 상기 스켈레톤에 대해 하나의 노드와 다른 노드와의 거리 및 인접 순위를 고려한 순위 인접 행렬(rank adjacency matrix)을 이용하여 시공간 특징을 추출하고, 추출된 상기 시공간 특징에 기초하여 입력 프레임의 객체와 정점(vertex)을 병합하며, 분류 작업을 수행한다. 여기서, 상기 순위 인접 행렬은, 입력된 스켈레톤 데이터에 포함된 관절(joint)을 나타내는 노드 각각에 대해 다른 노드와의 거리를 산출하고, 산출된 거리가 작은 순서대로 정렬하되 순위 메트릭(metric)을 통해 상기 하나의 노드와 가까운 소정 개수의 다른 노드에 대해 인접 순위를 나타내는 행렬을 생성할 수 있다.

[0091] 이상에서 본 명세서에 대하여 그 다양한 실시예들을 중심으로 살펴보았다. 본 명세서에 속하는 기술 분야에서 통상의 지식을 가진 자는 다양한 실시예들이 본 명세서의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 명세서의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 명세서에 포함된 것으로 해석되어야 할 것이다.

부호의 설명

[0092] 10: 입력 스켈레톤 데이터

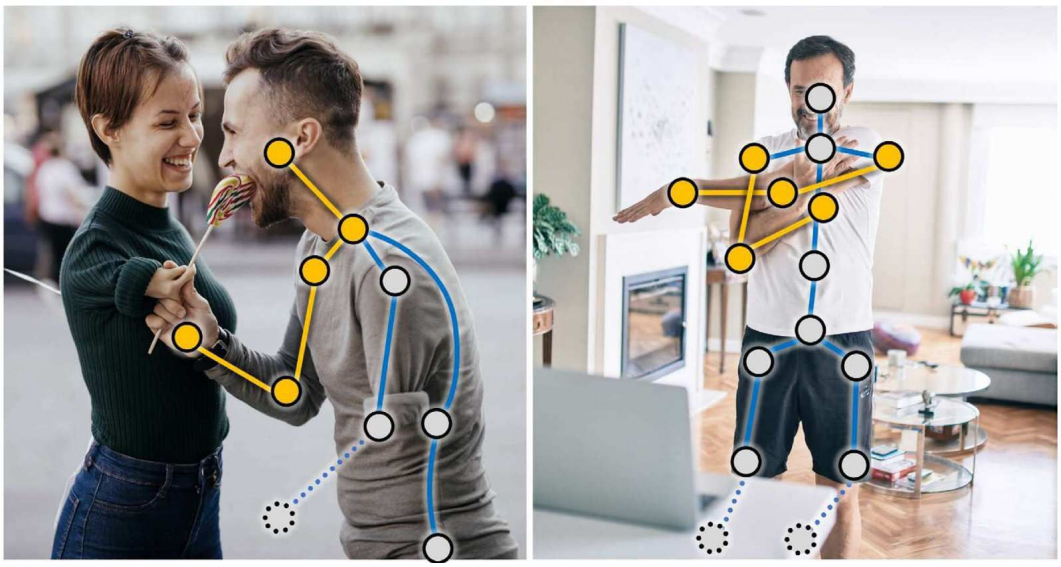
20: 동작 처리 장치

21: 입력부

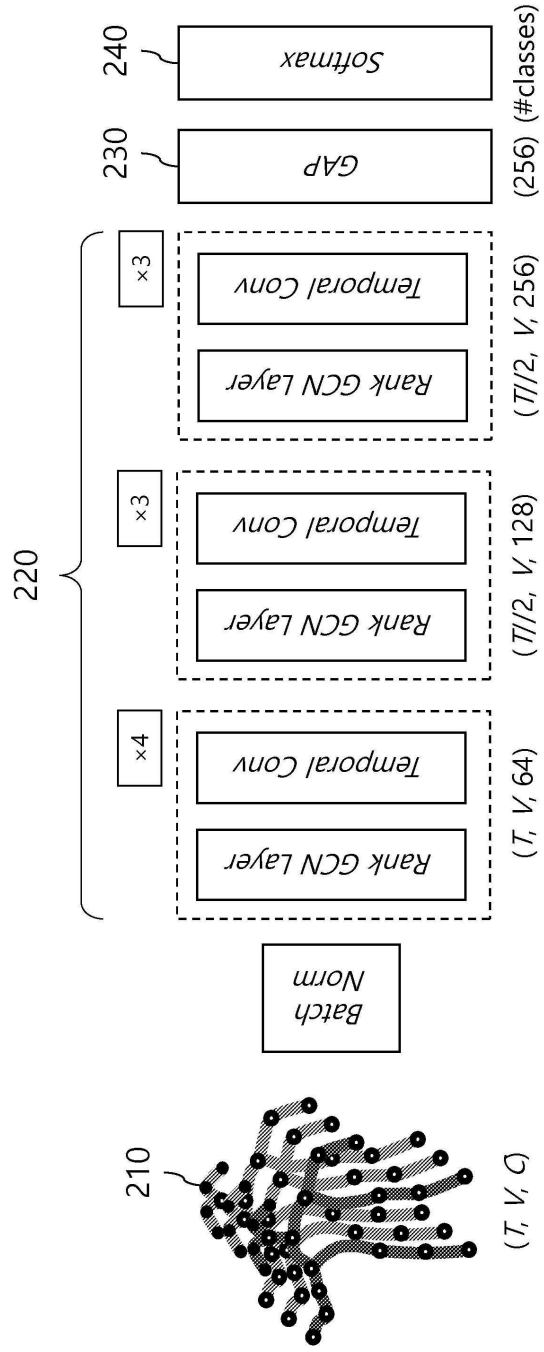
23: 처리부

도면

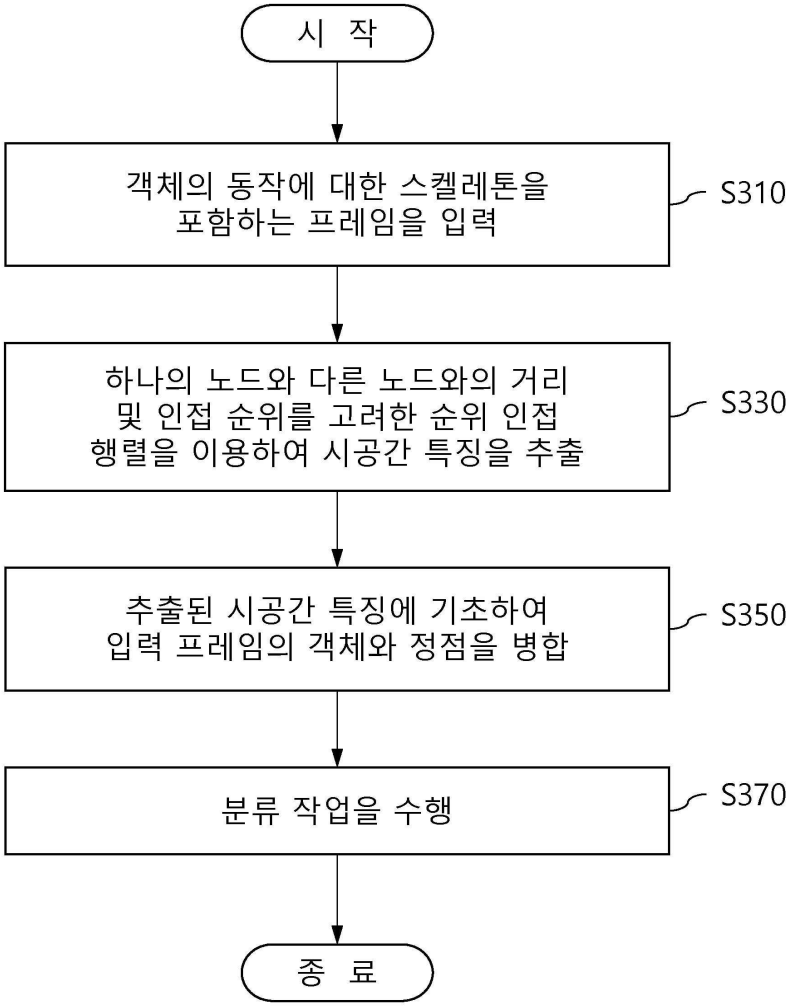
도면1



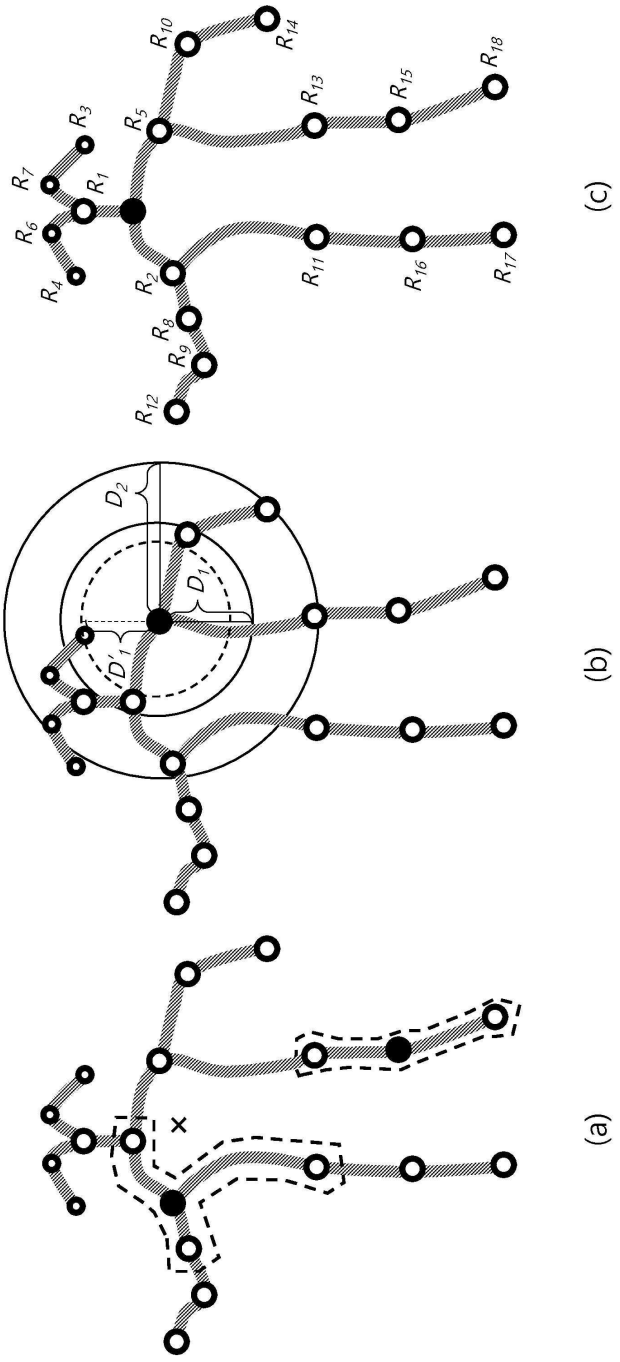
도면2


240
Softmax
(#classes)
230
GAP
(256)
220
×3
Temporal Conv
Rank GCN Layer
(T//2, V, 256)
×3
Temporal Conv
Rank GCN Layer
(T//2, V, 128)
×4
Temporal Conv
Rank GCN Layer
(T, V, 64)
Batch Norm
210
(T, V, C)

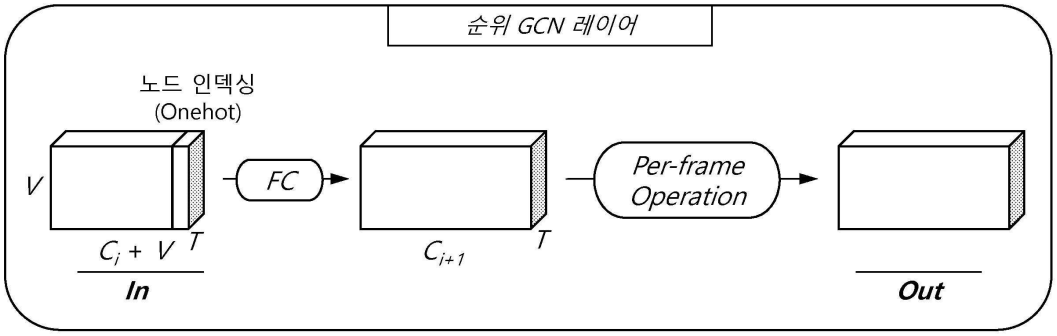
도면3


시 작
객체의 동작에 대한 스켈레톤을 포함하는 프레임을 입력
S310
하나의 노드와 다른 노드와의 거리 및 인접 순위를 고려한 순위 인접 행렬을 이용하여 시공간 특징을 추출
S330
추출된 시공간 특징에 기초하여 입력 프레임의 객체와 정점을 병합
S350
분류 작업을 수행
S370
종 료

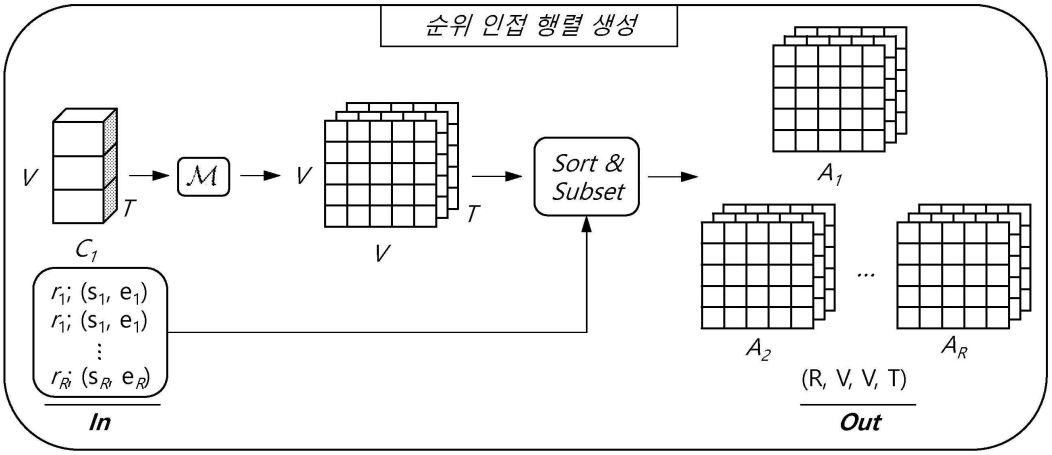
도면4


R10
R14
R18
R13
R15
R3
R5
R7
R1
R6
R4
R2
R11
R16
R17
R8
R9
R12
(c)
D2
D1
D'1
(b)
(a)

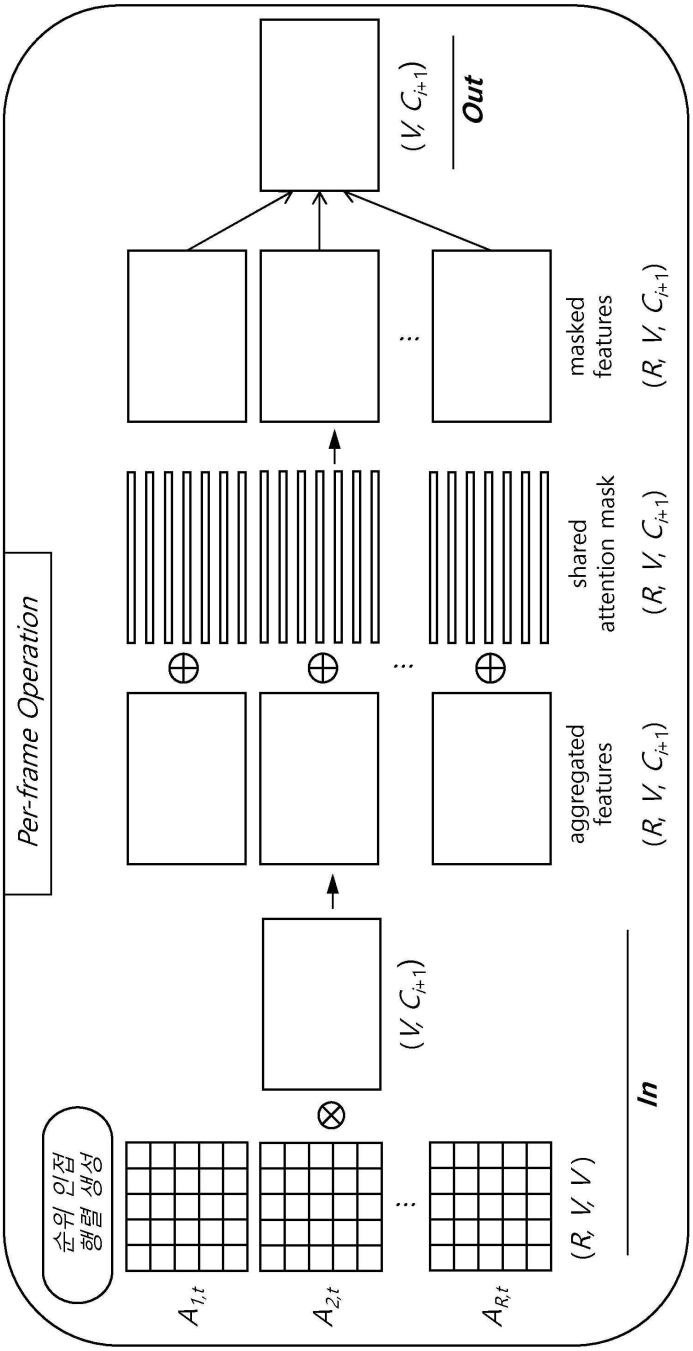
도면5


순위 GCN 레이어
노드 인덱싱
(Onehot)
V
FC
Per-frame
Operation
Ci + V
T
In
Ci+1
T
Out

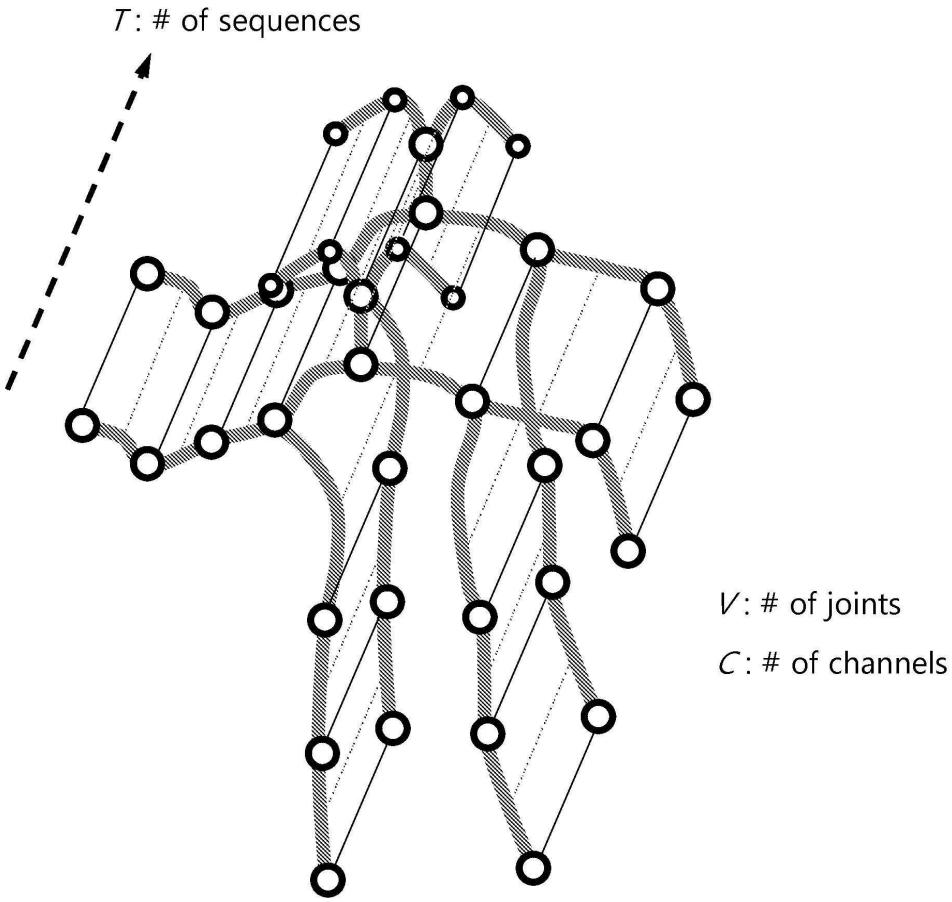
도면6


순위 인접 행렬 생성
V
T
C_1
$\mathcal{M}$
V
V
T
Sort &
Subset
A_1
A_2
...
A_R
r_1; (s_1, e_1)
r_1; (s_1, e_1)
⋮
r_R; (s_R, e_R)
In
(R, V, V, T)
Out

도면7


Per-frame Operation
순위 인접 행렬 생성
$A_{1,t}$
$A_{2,t}$
$A_{R,t}$
(R, V, V)
In
(V, C_{i+1})
aggregated features
(R, V, C_{i+1})
shared attention mask
(R, V, C_{i+1})
masked features
(R, V, C_{i+1})
(V, C_{i+1})
Out

도면8



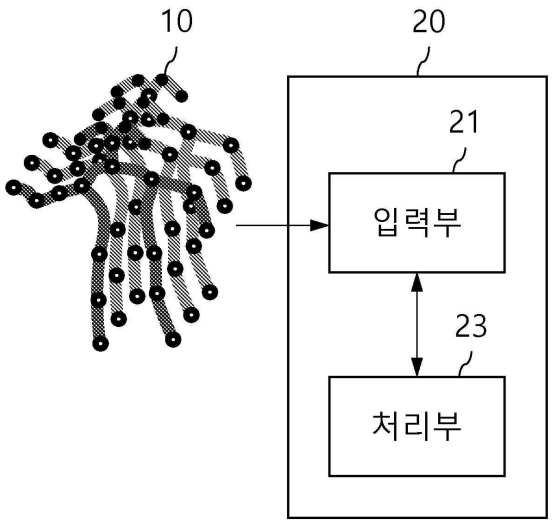
도면9

Algorithm 1 Algorithm for Rank-GCN layer

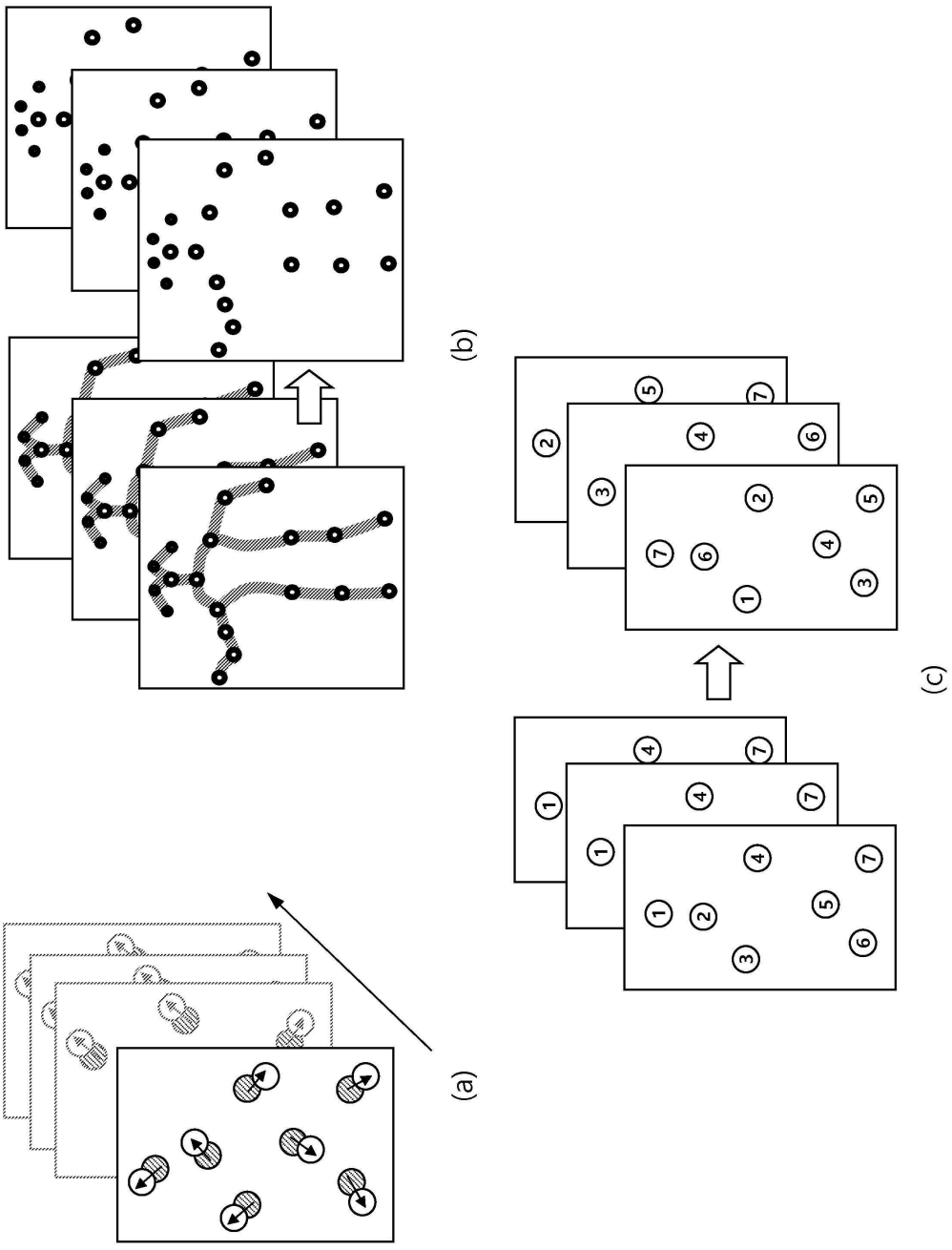
Input: Joints of a skeleton sequence $S \in \mathbb{R}^{T_1 \times V \times C_1}$,
Features from previous layer $F_i \in \mathbb{R}^{T_i \times V \times C_i}$,
Shared attention mask $M \in \mathbb{R}^{R \times V \times C_{i+1}}$
Rank ranges $\Gamma = \{\gamma_r = (s_r, e_r), 0 \leq s_r, e_r \leq V - 1\}$
Output: Output features $F_{i+1} \in \mathbb{R}^{T_{i+1} \times V \times C_{i+1}}$
Variable: Distance matrix $D_i \in \mathbb{R}^{T_i \times V \times V}$ and
Variable: Adjacency distance matrix $A \in \mathbb{R}^{|R| \times T_i \times V \times V}$

$D_1 \leftarrow \mathcal{M}(S)$ // *Pairwise metric function*
$D_i \leftarrow \text{Resize}(D_1, T_i)$ // *Resize* D_1 *to fit the length* T_i
$\mathcal{R} \leftarrow \text{Rank}(D_i)$
$F' = FC(F_i)$ // $C_i \rightarrow C_{i+1}$
$F_{i+1} \leftarrow 0$
for $r = 1$ to R **do**
  $A[r] = 1$ if $s_r \leq \mathcal{R} \leq e_r$ // *Element wise operation*
  $F_{i+1} = F_{i+1} + Dot(F', A[r]) \cdot M[r]$
end for
return F_{i+1}

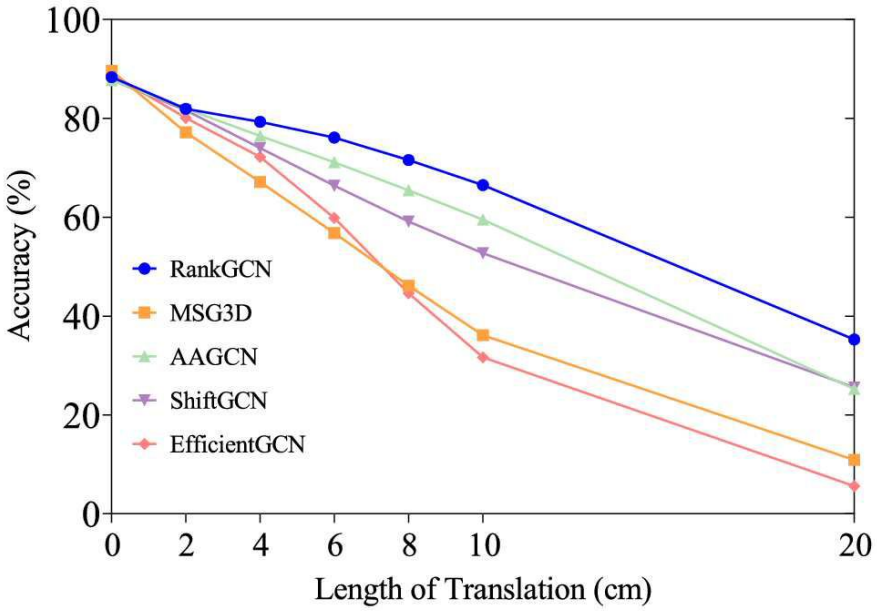
도면10



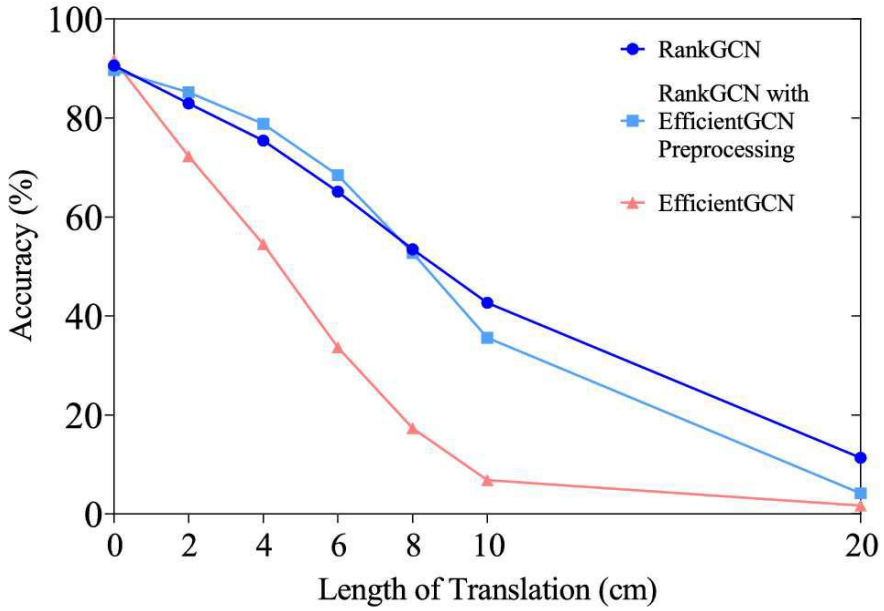
도면11



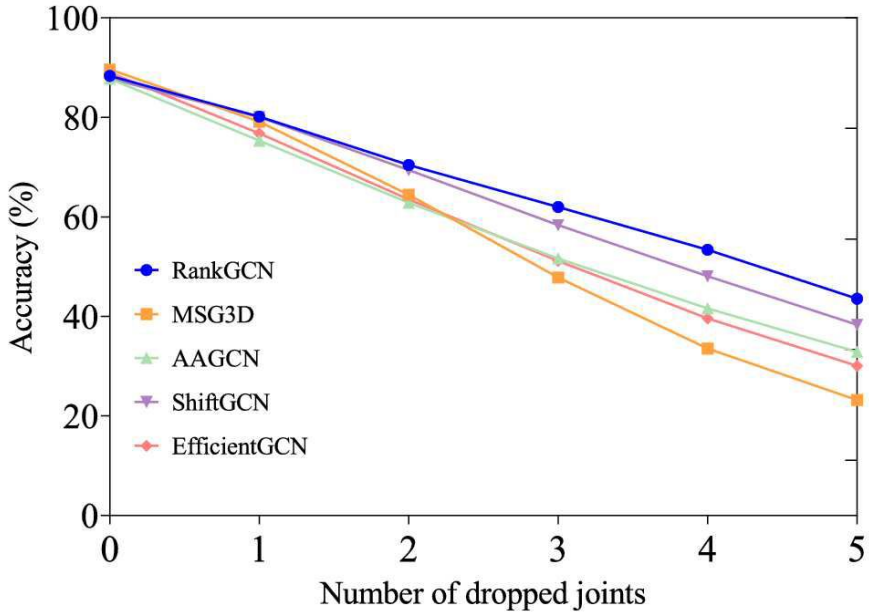
도면12


100
80
60
40
20
0
Accuracy (%)
RankGCN
MSG3D
AAGCN
ShiftGCN
EfficientGCN
0
2
4
6
8
10
20
Length of Translation (cm)

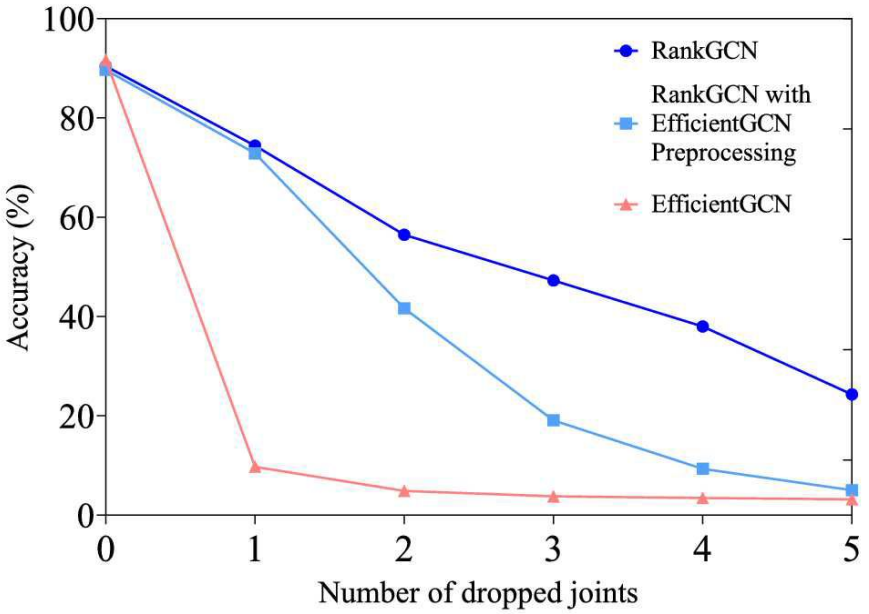
도면13


100
80
60
40
20
0
Accuracy (%)
RankGCN
RankGCN with EfficientGCN Preprocessing
EfficientGCN
0
2
4
6
8
10
20
Length of Translation (cm)

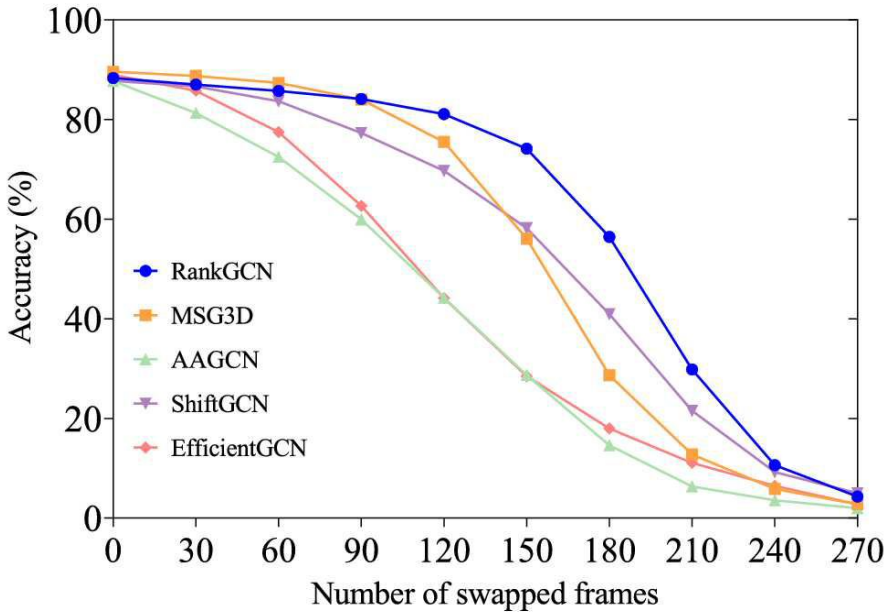
도면14


100
80
60
40
20
0
Accuracy (%)
RankGCN
MSG3D
AAGCN
ShiftGCN
EfficientGCN
0
1
2
3
4
5
Number of dropped joints

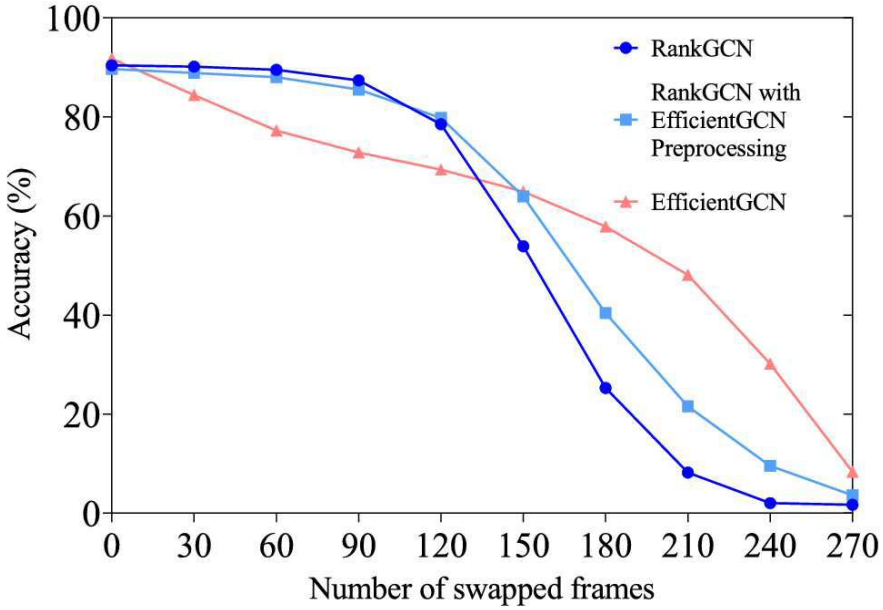
도면15


100
80
60
40
20
0
Accuracy (%)
RankGCN
RankGCN with
EfficientGCN
Preprocessing
EfficientGCN
0
1
2
3
4
5
Number of dropped joints

도면16


100
80
60
40
20
0
Accuracy (%)
RankGCN
MSG3D
AAGCN
ShiftGCN
EfficientGCN
0
30
60
90
120
150
180
210
240
270
Number of swapped frames

도면17


100
80
60
40
20
0
Accuracy (%)
RankGCN
RankGCN with EfficientGCN Preprocessing
EfficientGCN
0
30
60
90
120
150
180
210
240
270
Number of swapped frames