Dynamic Hand Gesture Recognition

송 재 훈

Vision & Display Systems Lab.

Dept. of Electronic Engineering, Sogang University

Outline

- Hand Gesture Recognition ?
- Applications
- Input data : RGB, Depth
- Flow of Gesture Recognition
- DG-STA : Construct Dynamic Graphs for Hand Gesture Recognition via Spatial-temporal Attention
- Conclusion

Hand Gesture Recognition (HGR)

- Goal : Classifying a set of discrete hand poses

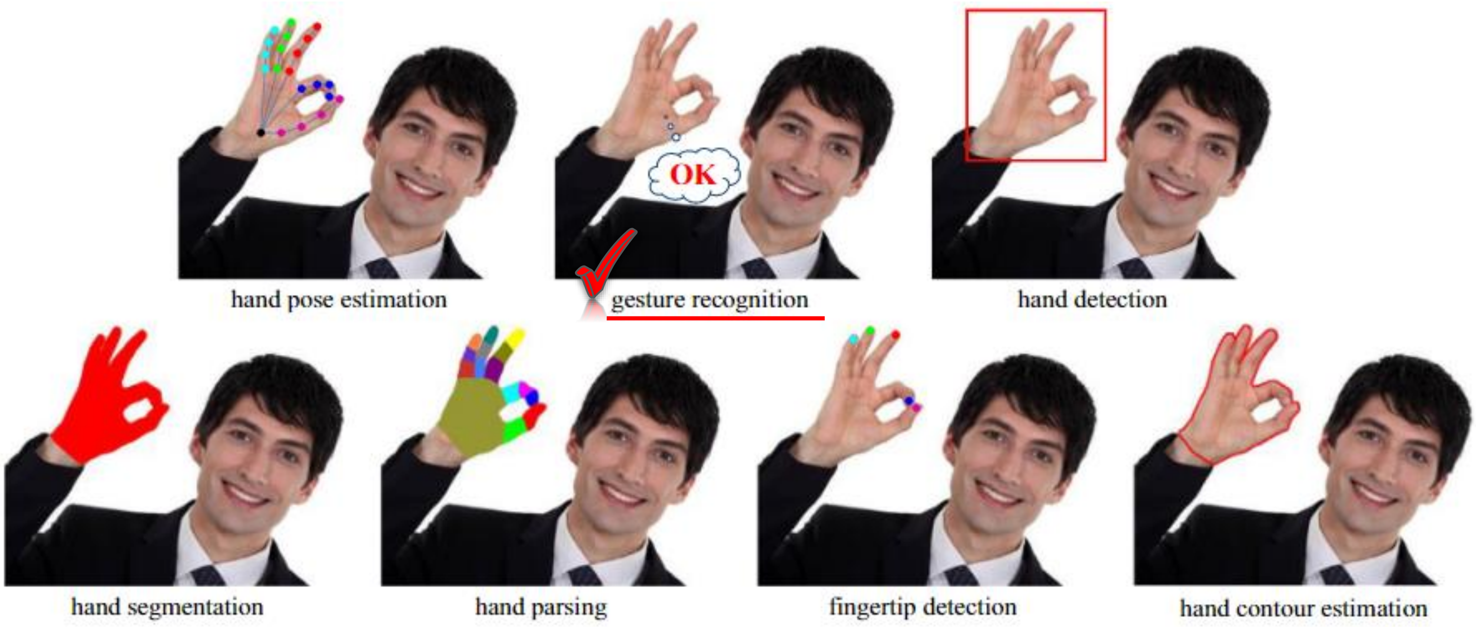
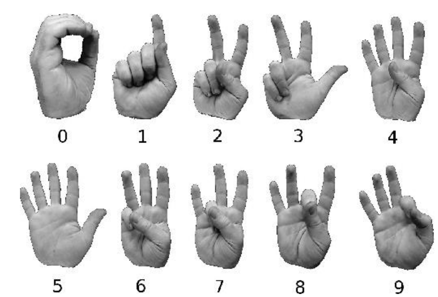


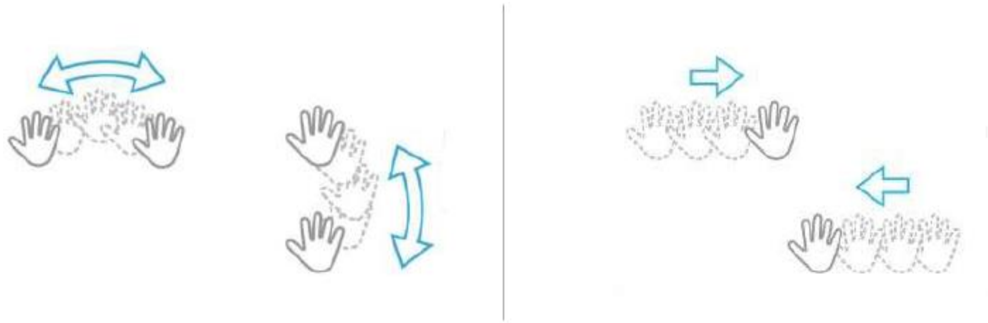
Fig. Hand Gesture Recognition and its similar 6 fields

Hand Gesture Recognition (HGR)

- Two categories
  - ✓ Static : identify hand gestures from a single image
    - comparison with reference images
      ex) ASL (American Sign Language)



  - ✓ Dynamic : identify animated hand gestures



Applications

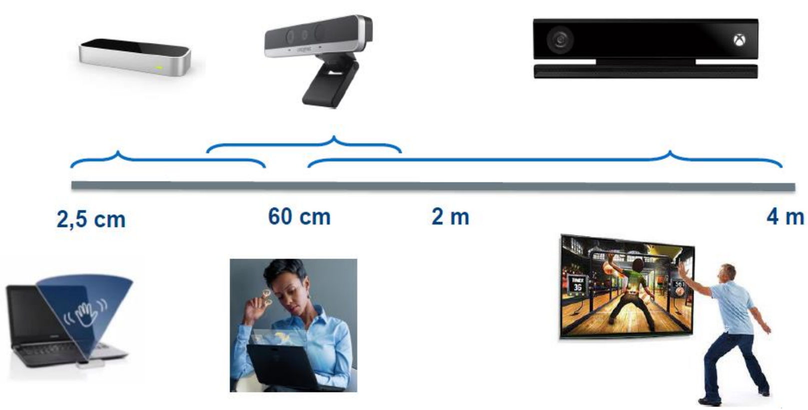
- Important skill for HCI (Human Computer Interaction)

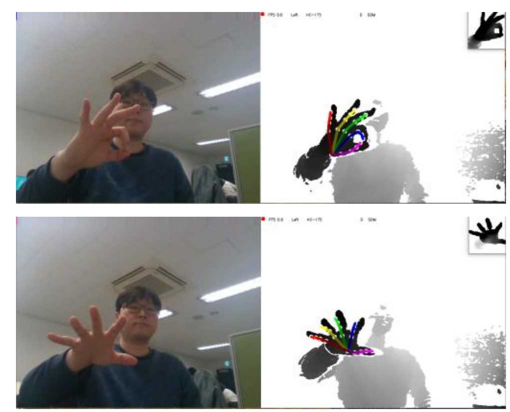
- Future

Input data

- RGB image
  - Hand segmentation 후 optical flow나 skeleton 좌표 등을 이용하여 분류

- Depth image (depth / pose)
  - Depth camera의 등장으로 simple segmentation 및 depth 좌표 활용 가능해짐
  - Hand pose estimation 기술발전으로 실시간으로 hand skeleton sequence 생성 가능



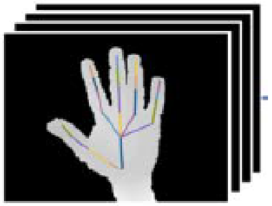


Flow of Gesture Recognition

- Recognition of the dynamic gestures based on the hand joint coordinate

Depth camera

→ **Depth Stream** → **Hand Segmentation** → **Skeletonization (pose estimation)** → **Gesture Recog.**



✓ Assume that a hand is nearest object from the camera

✓ Segmentation : depth thresholding → center of mass (COM) of hand → crop

✓ Hand pose estimation : predict the 3D (x, y, z) coordinate of joints

✓ Skeleton sequences : have high semantic information and small data size

Paper Information

- Construct Dynamic Graphs for Hand Gesture Recognition via Spatial-Temporal Attention

- Authors : Chen, Y.[1], Zhao, L., Peng, X., Yuan, J., Metaxas, D.N.

[1] Department of Computer Science, Rutgers University, New Jersey, USA

- BMVC 2019

Abstract

- Key Idea

  1) Construct a fully-connected graph from a hand skeleton

  2) Node features and edges are automatically learned via a self-attention mechanism

  3) Self-attention performs in both spatial and temporal domains

  4) leverage the spatial-temporal cues of joint positions

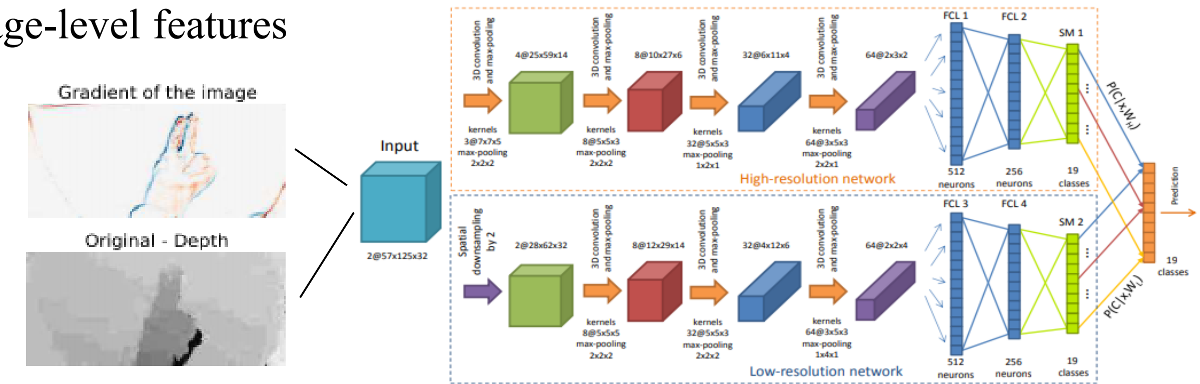
  5) spatial-temporal mask : significantly cut down the computational cost by 99%

Previous work

- Categories based on Input data

  1) Image-based : rely on image-level features

  ex. HGR with 3D CNN [1]

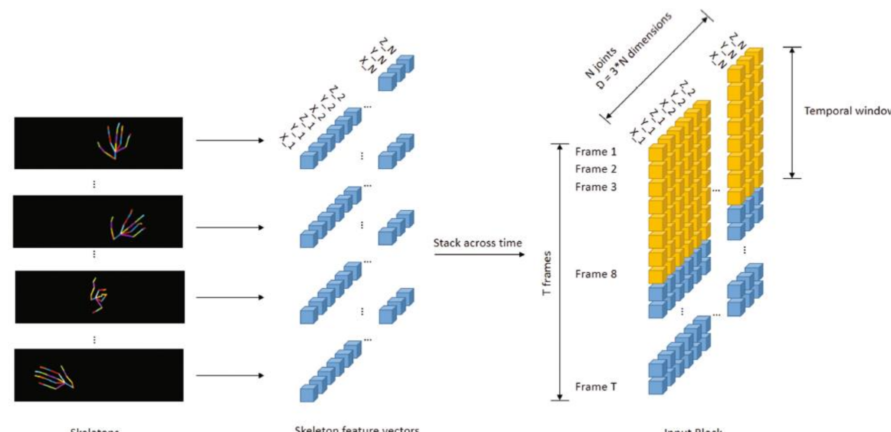


  2) Skeleton-based : sequence of hand joints with 2D or 3D coordinates

  ex. STA-Res-TCN [2]

  - concatenate the joint coordinates

  → spatial structures and temporal dynamics of hand skeletons are not explicitly exploited



DG-STA (Dynamic Graph-Based Spatial Temporal Attention)

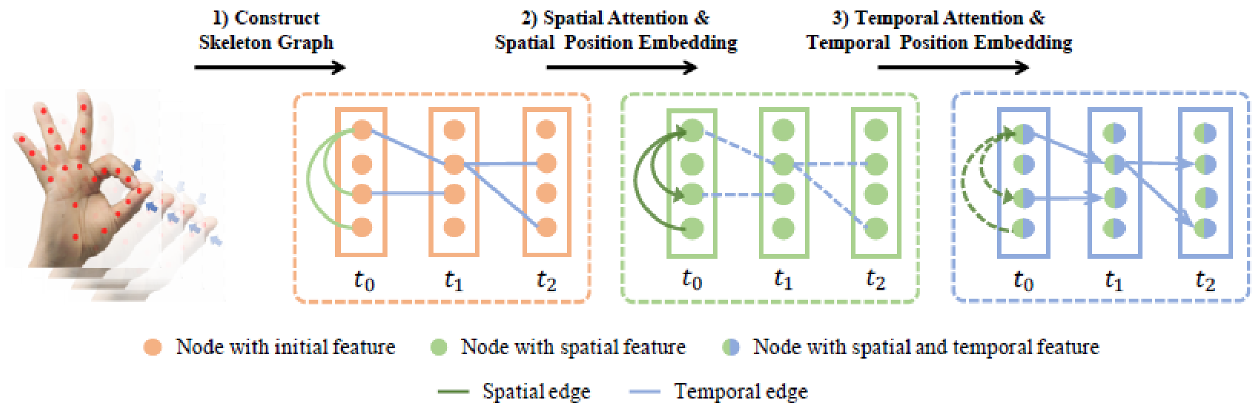


Fig 1. node는 hand joint, 점선은 끊어진 edge 의미 → edge weights와 node features 학습

- Contributions
  - ✓ 서로 다른 동작을 모델링하도록 graph 학습 (pre-defined graph 미사용)
    → 표현력이 향상된 action-specific graphs
  - ✓ spatial-temporal position embedding : 기존 temporal position embedding 유사
    → encodes the identity and temporal order information of each node
  - ✓ spatial-temporal mask operation : applied to the matrix of scaled dot-products among all nodes
    → improves the computational efficiency

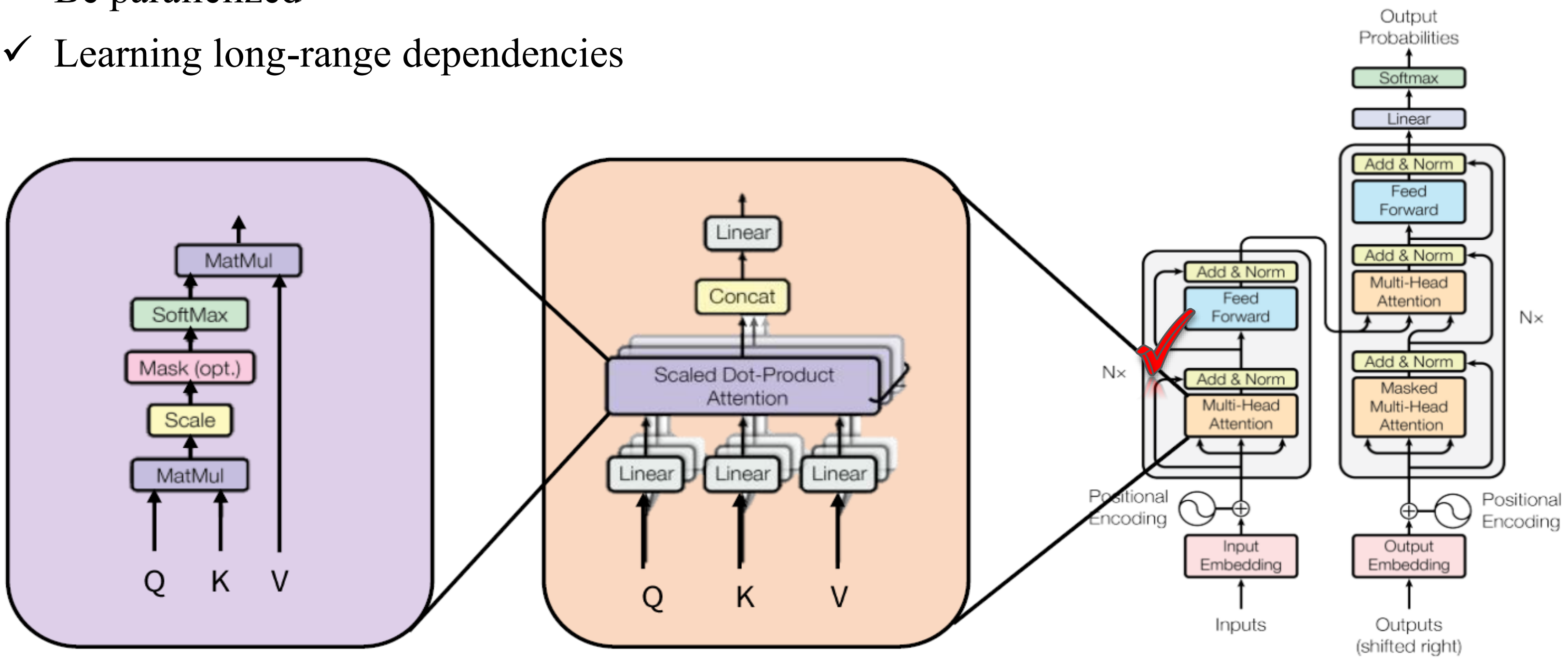
Related work

- Self-Attention
  - "Attention is all you need [1]" 논문의 transformer 에서 사용
    → 기계 번역에서 문장 내 단어들 간 temporal / semantic 관계 모델링
  - widely used in computer vision and natural language processing tasks
  - 본 논문에서는, graph로 표현된 hand skeletons를 포함하는 spatial-temporal information을 학습하기 위해 사용

- Sequential data processing networks
  1) RNN-based : 입력 시퀀스를 순차적으로 처리하여 병렬처리 어려움
     연산 시간, 계산 복잡도↑
  2) CNN-based : local neighbor만 처리, global 연산 수행 시 반복 처리로 연산량↑
  3) Self-attention : RNN, CNN 구조 사용 X
     Key = Query = Value , dot product 사용

Related work

- Self-Attention
  - ✓ Computational complexity ↓
  - ✓ Be parallelized
  - ✓ Learning long-range dependencies



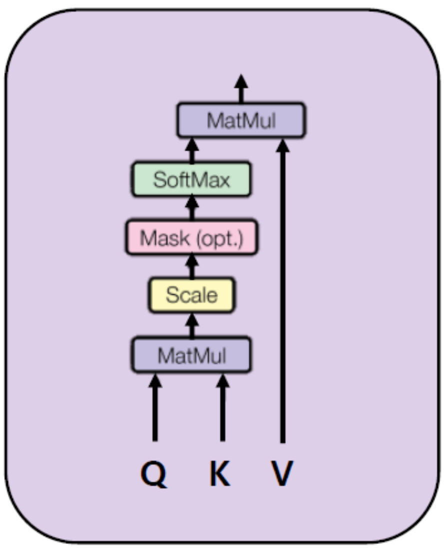
Scaled Dot-Product Attention

Multi-Head Attention

Transformer

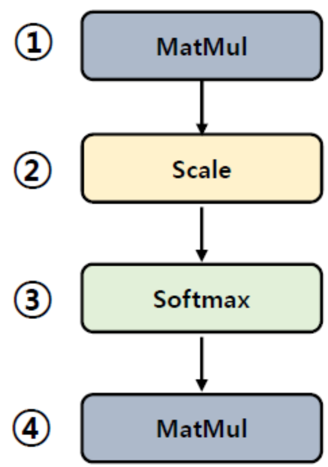
Related work _ Self-Attention

- Scale dot-product attention
  - ✓ Query = Key = Value
  - ✓ Similarity function = Dot product
  - ✓ Normalize by Softmax
  - ✓ Weight sum of Value vectors



Scaled Dot-Product Attention

Generalized Attention Form



: Weight sum of value vectors

$$A(q,K,V) = \sum_{i} softmax(f(K,q))\,V$$

① $f(K,Q) = QK^{T} \quad (K = KW^{K}, Q = QW^{Q}, V = QW^{V})$

② $\frac{QK^{T}}{\sqrt{d_k}}$: Scaled-dot product

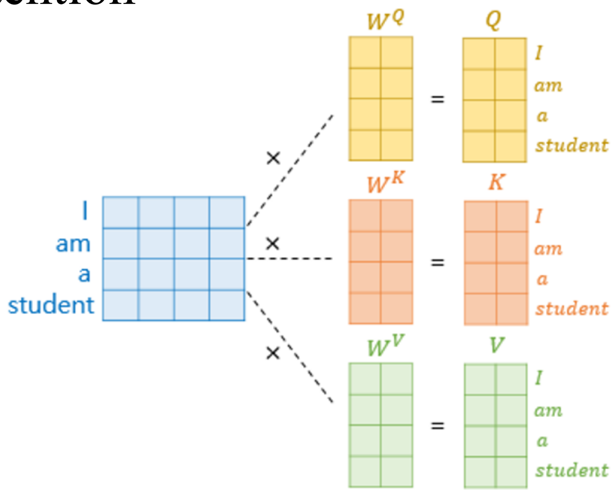
③ $softmax(\frac{QK^{T}}{\sqrt{d_k}})$

④ $softmax(\frac{QK^{T}}{\sqrt{d_k}})V$

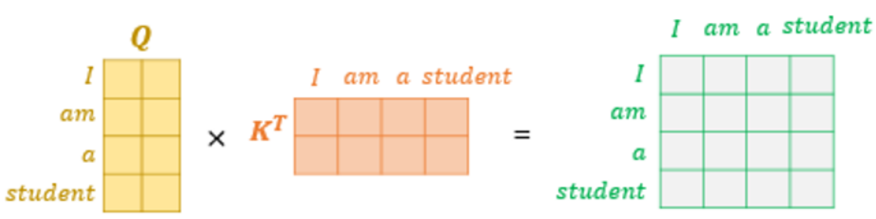
Related work _ Self-Attention

- Scale dot-product attention

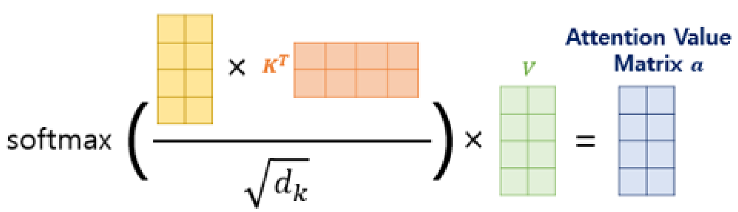
  ✓ MatMul



✓ Scaled dot-product

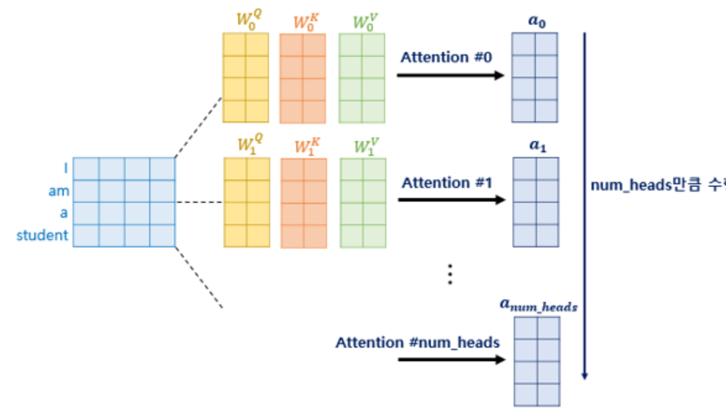


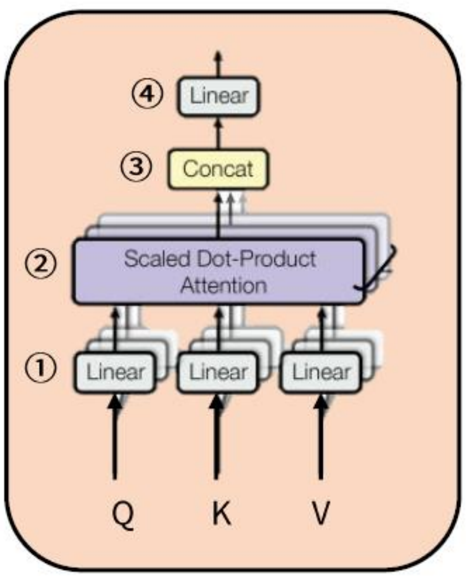
✓ MatMul (softmax)



Related work _ Self-Attention

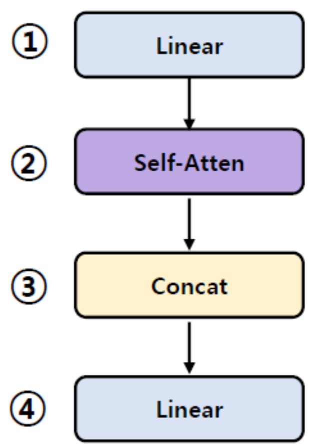
- Multi-head attention
  - ✓ Learning diverse input features (독립적으로 w 학습)





Multi-Head Attention

Self-Attention $SA(q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$



① $Q` = QW_i^Q \quad K` = KW_i^K \quad V` = VW_i^V \quad (i = 1 \dots h)$

② $\text{head}_i = SA(Q`, K`, V`)$

③ $[\text{head}_1, \text{head}_2, \dots, \text{head}_h]$

④ $[\text{head}_1, \text{head}_2, \dots, \text{head}_h]W^O$

$= \boldsymbol{MultiHead(Q, K, V)}$: concat mat. 동일 크기

Methodology

- Skeleton graph initialization
  - ✓ *T* frames video에서 hand skeleton 표현하기 위해 각 frame의 *N*개 hand joints 추출
  - ✓ skeleton graph $G = (V, E)$ 구성, V = node, E = edge
  - ✓ $\mathbf{f}_{(t,i)}$: the <u>feature vector</u> of the <u>node $v_{(t,i)}$</u> → node의 3D 좌표로부터 extracted (3 → 128)
  - ✓ node의 feature vector : $F = \{\mathbf{f}_{(t,i)} | t = 1, \ldots, T, i = 1, \ldots, N\}$
    - A spatial edge $v_{(t,i)} \rightarrow v_{(t,j)} (i \neq j)$ connects two different nodes at the same time step.
    - A temporal edge $v_{(t,i)} \rightarrow v_{(k,j)} (t \neq k)$ connects two nodes at different time steps.
    - A self-connected edge $v_{(t,i)} \rightarrow v_{(t,i)}$ connects the node with itself.
- Dynamic graph construction via Spatial-Temporal attention
  - ✓ Spatial attention model A_S : initial node feat. *F*를 입력 받아 spatial information 업데이트
  - ✓ Temporal attention model A_T : 위 features에서 temporal information 추가 업데이트
  - ✓ Average pooled 된 후 classification을 위한 feature representation으로 사용
  - ✓ Multi-head attention 적용

Methodology

- Spatial-Temporal attention
  - ✓ Transformer의 Self-attention과 동작이 거의 같음
  - ✓ 3개의 FC layers로 Key , Query, Value vectors 생성 (h는 head 의미)

$$\mathbf{K}^h_{(t,i)} = W^h_K \mathbf{f}_{(t,i)}, \quad \mathbf{Q}^h_{(t,i)} = W^h_Q \mathbf{f}_{(t,i)}, \quad \mathbf{V}^h_{(t,i)} = W^h_V \mathbf{f}_{(t,i)}, \tag{1}$$

  - ✓ scaled dot-product (query 와 key vector) → normalize by Softmax function

$$u^h_{(t,i)\to(t,j)} = \frac{\langle \mathbf{Q}^h_{(t,i)}, \mathbf{K}^h_{(t,j)} \rangle}{\sqrt{d}}, \quad \alpha^h_{(t,i)\to(t,j)} = \frac{\exp\left(u^h_{(t,i)\to(t,j)}\right)}{\sum_{n=1}^{N} \exp\left(u^h_{(t,i)\to(t,n)}\right)}, \tag{2}$$

d : key, query, value vectors의 dimension

  - ✓ weighted sum of the value vectors within the same time step

$$\bar{\mathbf{f}}^h_{(t,i)} = \sum_{j=1}^{N} \left(\alpha^h_{(t,i)\to(t,j)} \cdot \mathbf{V}^h_{(t,j)} \right), \tag{3}$$

  - ✓ concatenates the spatial attention features learned by all heads

$$\tilde{\mathbf{f}}_{(t,i)} = \text{Concate}\left[\bar{\mathbf{f}}^1_{(t,i)}, \bar{\mathbf{f}}^2_{(t,i)}, \ldots, \bar{\mathbf{f}}^H_{(t,i)}\right], \tag{4}$$

H : number of spatial attention heads

Methodology

- Spatial-Temporal attention
  - ✓ temporal attention model A_T takes the output node features from the spatial attention
  - ✓ spatial과 동일한 multi-head attention mechanism in the temporal domain
    → temporal attention model output : encodes both spatial & temporal information

- Spatial-Temporal Position Embedding
  - ✓ Transformer 와 동일한 방법으로 진행
  - ✓ RNN, CNN 처럼 순서나 위치 정보가 없음
    → position을 알 수 있는 position embedding vector를 더해줌

$$\hat{\mathbf{f}}_{(t,i)} = \mathbf{A}_T\left(\underline{\mathbf{p}^T_{(t,i)}} + \mathbf{A}_S\left(\mathbf{f}_{(t,i)} + \underline{\mathbf{p}^S_{(i)}}\right)\right), \tag{5}$$

  - ✓ Values are set using the sine and cosine functions of different frequencies
    (Transformer 논문 동일 방법)

Methodology

- Efficient Implementation
  - ✓ Transformer에서 사용한 mask 기능 유사 (사용하지 않는 key 값은 0으로 masking)
  - ✓ propose a novel scheme to facilitate the implementation of DG-STA
    1) compute the matrix of the scaled dot-products among all nodes (Softmax 전단계)
    2) apply spatial-temporal mask operation → focus on the spatial or temporal domain
  - ✓ Matrix of the scaled dot-products W (before normalization)

$$\mathbf{W} = \mathbf{Q} \otimes \mathbf{K}^{\top}$$

  - ✓ Mask operation W의 element (temporal edge 의미)는 η(매우 큰 음수)로 변경
    → Softmax (exponential) 함수에서 0이 됨
  - ✓ Spatial mask 적용 후,

$$\bar{\mathbf{W}}_S = \underbrace{\phi}_{\text{softmax}}(\mathbf{W} \odot \mathbf{M}_S + (1 - \mathbf{M}_S) \times \eta),$$

$\odot$: element-wise dot operation

  - ✓ Temporal mask 도 유사 : temporal or self-connect edge 를 제외하고 0으로 masking

Methodology

- Efficient Implementation
  - ✓ Spatial-temporal mask operation을 적용하여 computation time을 99% 줄임.

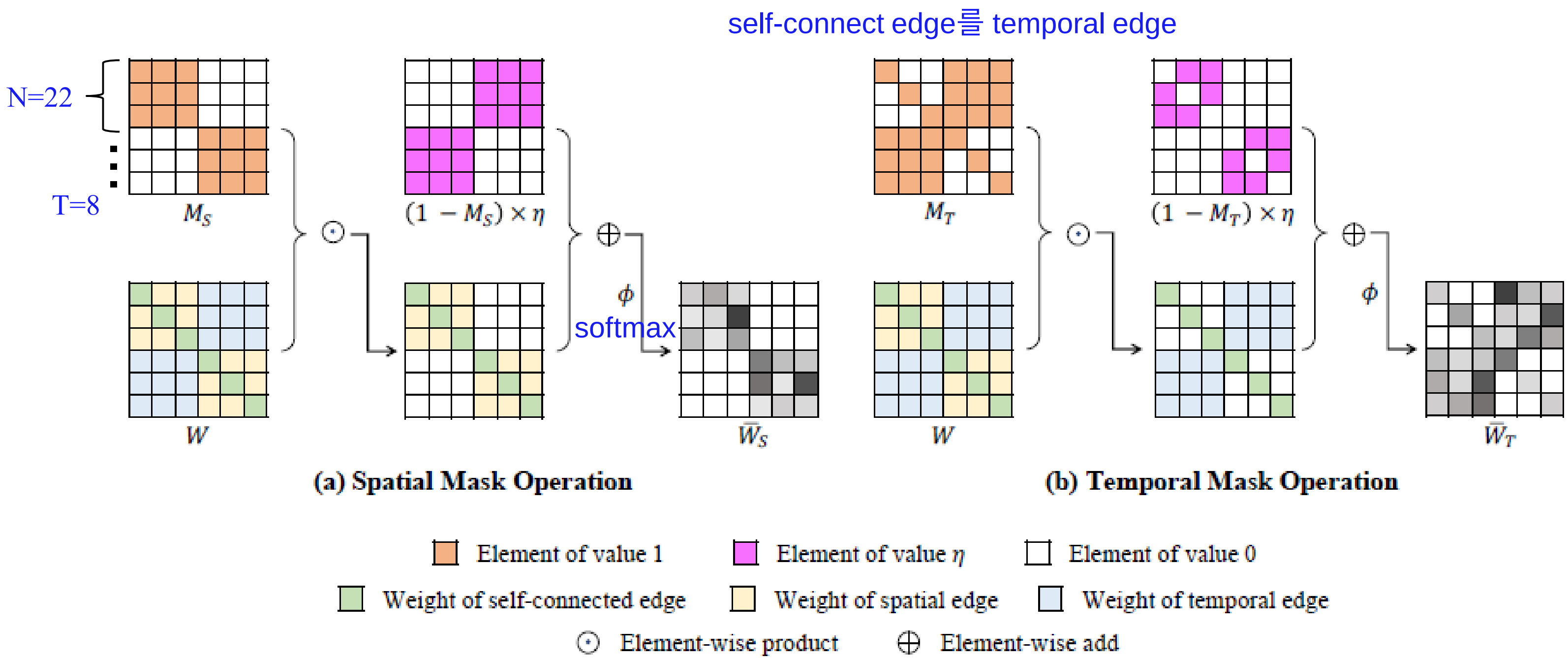


Fig 2. Illustration of the proposed spatial and temporal mask operations.

Experiments

- Implementation details
  - ✓ head number of the spatial & temporal attention = 8
  - ✓ dimension of Query, Key, Value vectors = 32
  - ✓ hand joint의 input 3D 좌표는 128 dim.의 initial node feature로 project됨

    : (NxT , 3) → (NxT, 128)
  - ✓ Add the Spatial position embedding
  - ✓ Spatial Attention 출력이 다시 temporal pos. embedding을 더한 후 Temporal Att. 수행
  - ✓ 모든 node features를 average pooled 하여 vector로 만든 후 FC를 거쳐 classify 됨

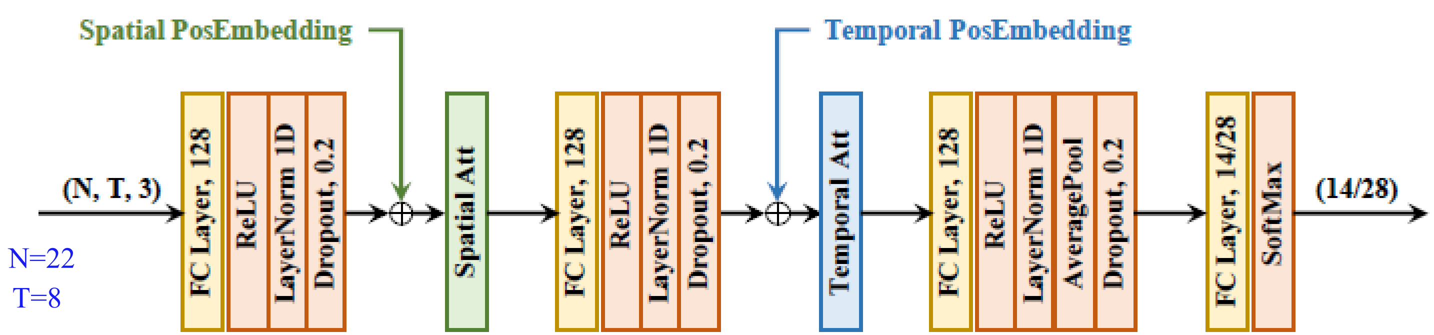
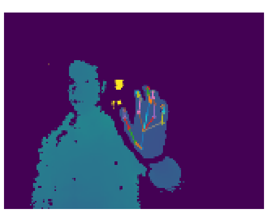


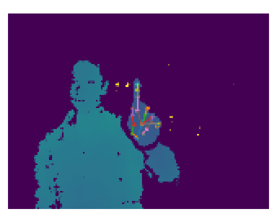
Fig 3. The network architecture of the proposed DG-STA

Datasets

- DHG-14/28 & SHREC'17 Track (HPE의 MSRA dataset (21joints) 과 유사)
  - ✓ Intel Realsense camera / 640 x 480 해상도 / 30 fps / gesture 길이 20~ 50 frames
  - ✓ 14 개 gesture sequence / 28명 참가자 1~10회 수행 / 2800 sequences
  - ✓ 2D depth image & 3D world space 22개 joints 포함
  - ✓ Two configurations : one single finger & whole hand
  - ✓ DHG-14/28 dataset은 test dataset이 없음
    - → leave-one-subject-out cross-validation strategy 사용 (20개 subjects)
  - ✓ 14 gestures (w/o single finger configuration) / 28 gestures (both configurations)

Tap





Swipe Left

Experiments

- Ablation study
  ✓ 3 major components :
  1) Fully-Connected skeleton graph structure (FSG) : Sparse skeleton graph structure (SSG)와 비교
     - ST-GCN [1] 처럼 spatial edge를 natural hand joints connection로 정의
     - temporal edges도 연속 frame들의 같은 joints 간 연결
  2) spatial-temporal attention model (STA) : GAT [2] 로 downgrade
     - spatial-temporal 구분 없이 one attention module로 전체 graph에 적용
  3) spatial-temporal position embedding (STE) : STE 미 적용 실험
     - STE로 encod되는 identity 와 temporal order informatio의 중요성을 보여줌

  ✓ proposed method (FSG+STA+STE) achieves the best performance

Setting	FSG+STA	FSG+GAT+STE	SSG+STA+STE	DG-STA
14 Gestures (D)	84.3	90.8	89.8	**91.9**
28 Gestures (D)	77.3	87.8	86.6	**88.0**
14 Gestures (S)	88.9	92.7	91.5	**94.4**
28 Gestures (S)	80.1	86.2	87.7	**90.7**

Table 1. Ablation study of accuracy (%) on the DHG-14/28 (D) and SHREC’17 Dataset (S)

서강대학교 SOGANG UNIVERSITY

VDS LAB

Experiments

- Comparison with previous methods
  - ✓ hand-crafted feature / deep learning based approach / a graph-based method 등과 비교
  - ✓ hand의 dynamics와 structures 를 활용할 수 있는 제안 방법과 ST-GCN이 outperform함

  - ✓ Proposed method achieves the state-of-the-arts performance

Method	14 Gestures	28 Gestures
SoCJ+HoHD+HoWR [8]	83.1	80.0
Chen *et al.* [5]	84.7	80.3
CNN+LSTM [21]	85.6	81.1
Res-TCN [13]	86.9	83.6
STA-Res-TCN [13]	89.2	85.0
ST-GCN [39]	91.2	87.1
✓ **DG-STA (Ours)**	**91.9**	**88.0**

Table 2: Comparisons of accuracy (%) on DHG-14/28 Dataset.

Method	14 Gestures	28 Gestures
Oreifej *et al.* [26]	78.5	74.0
Devanne *et al.* [10]	79.4	62.0
Classify Sequence by Key Frames [9]	82.9	71.9
Ohn-Bar *et al.* [25]	83.9	76.5
SoCJ+Direction+Rotation [7]	86.9	84.2
SoCJ+HoHD+HoWR [8]	88.2	81.9
Caputo *et al.* [2]	89.5	-
Boulahia *et al.* [1]	90.5	80.5
Res-TCN [13]	91.1	87.3
STA-Res-TCN [13]	93.6	**90.7**
ST-GCN [39]	92.7	87.7
✓ **DG-STA (Ours)**	**94.4**	**90.7**

Table 3: Comparisons of accuracy (%) on SHREC'17 Track Dataset.

Conclusion

- Skeleton-based hand-gesture recognition 방법
- Graph-based spatial-temporal attention method를 사용
- Fully-connected skeleton graph를 활용하여 edge weight 학습과 시공간 정보 추출
- 가장 좋은 성능을 보이며 skeleton-based human action recognition 사용 가능

- HPE(좌표 추정) + DHGR(분류): 카메라 입력으로부터 제스처 인식까지 통합 진행 중

Thank You