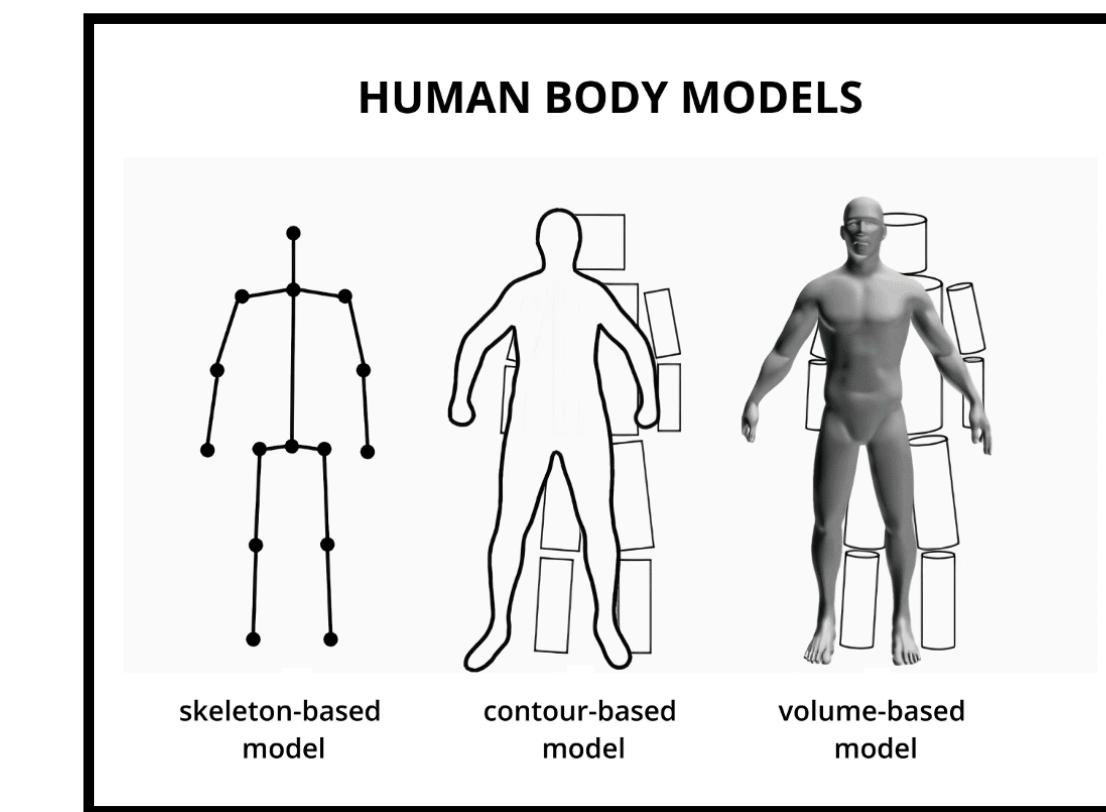
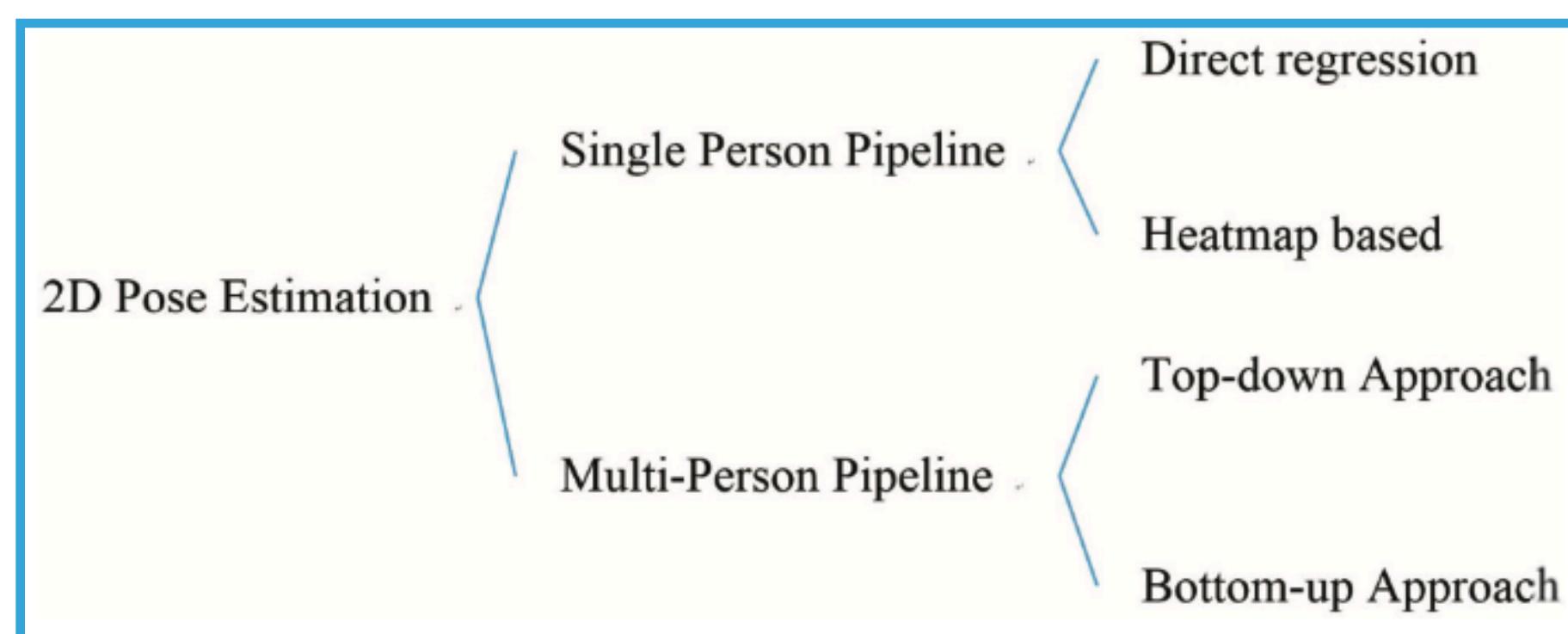


HUMAN POSE ESTIMATION

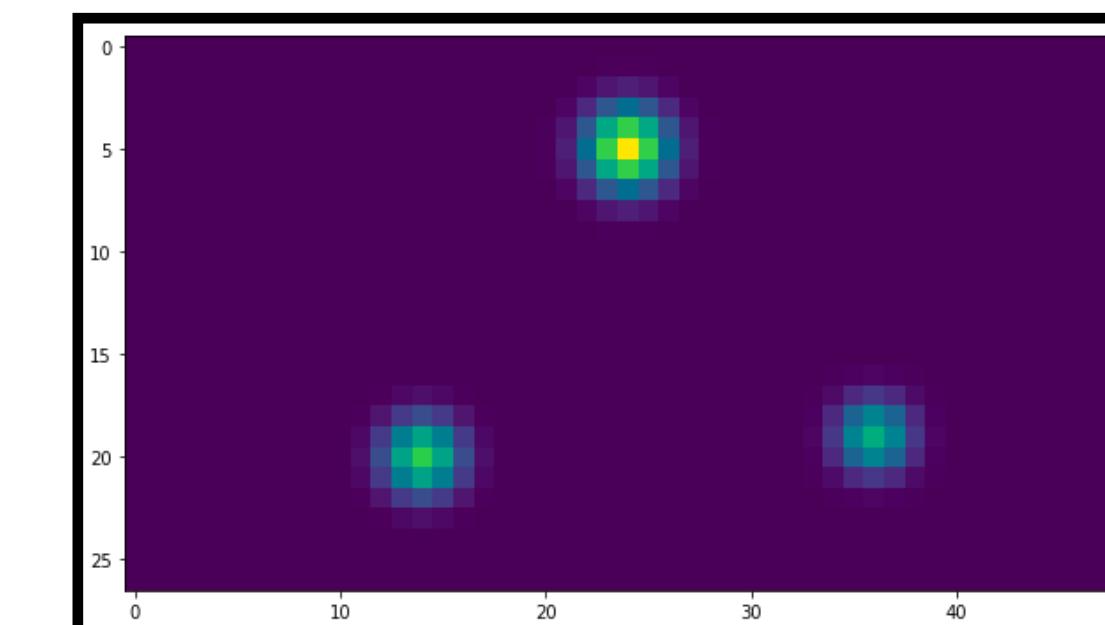
- ▶ 이미지나 영상에서 사람의 관절 위치를 정확하게 추정하는 문제
- ▶ 실제 적용을 위해선 높은 정확도와 "실시간" 추론이 필요
- ▶ **skeleton-based model**
 - part(joint, key point): 관절
 - limb(part pair, part connection): 두 관절 사이의 연결
- ▶ **2D Pose Estimation** (2D or 3D Pose Estimation)



contour: 여러개의 직사각형으로 표현 (기하학적)
volume: 3D mesh 데이터를 활용

ABSTRACT

- ▶ **Keyword: multi-person, realtime**
- ▶ 2개의 논문: 이전 버전 논문(CVPR, 2017) 발표 후 개선하여 업데이트 버전(arXiv, 2019) Release
 - 오늘 리뷰할 논문은 2019년 버전: 이전 논문과 비교
- ▶ **Bottom-up** 방식
 - Top-down 방식의 한계
 - Bottom-up with **PAFs** 제시
- ▶ **Heatmap**을 통해 이미지 내 사람의 관절 위치 추정
 - 관절 별 좌표 (X)
 - 픽셀 당 각 관절이 나타날 확률 (O)

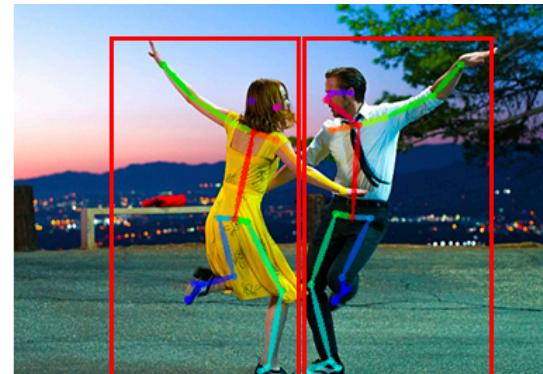


Heatmap

BOTTOM-UP APPROACH

Top-down approach

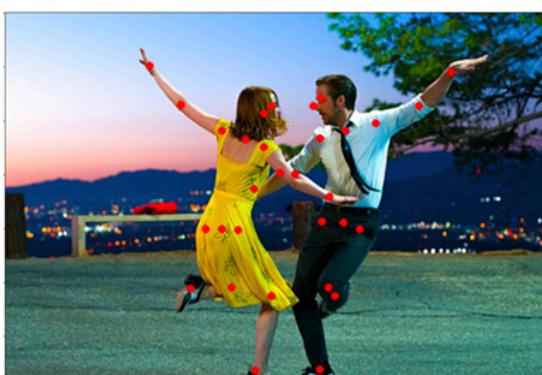
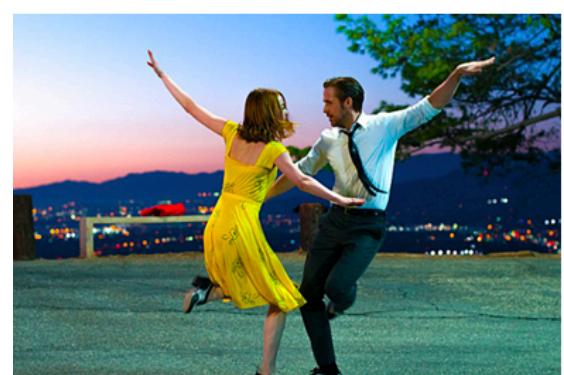
1) 사람을 Detection 하고, 2) Bounding Box 내부에서 포즈를 추정



사람 감지 (object detection) => pose 추정 * 사람수

Bottom-up approach

1) 사람의 keypoint를 모두 추정하고, 2) keypoint간의 상관관계를 분석하여 포즈를 추정



관절 부위 감지 => 관절 부위 이음

Challenge of human estimation: multiple people의 pose estimation

- 1) unknown number of people (despite of any position or scale)
- 2) difficult parts association (예) contact, occlusion, or limb articulations)
- 3) runtime complexity: due to the number of people in image

Top-down 방식이 아니라 Bottom-up 방식 이용

- Top-down 방식의 한계

- ▶ person detector를 실패하면, 보완할 방법 없음.
- ▶ runtime이 사람 수에 비례함

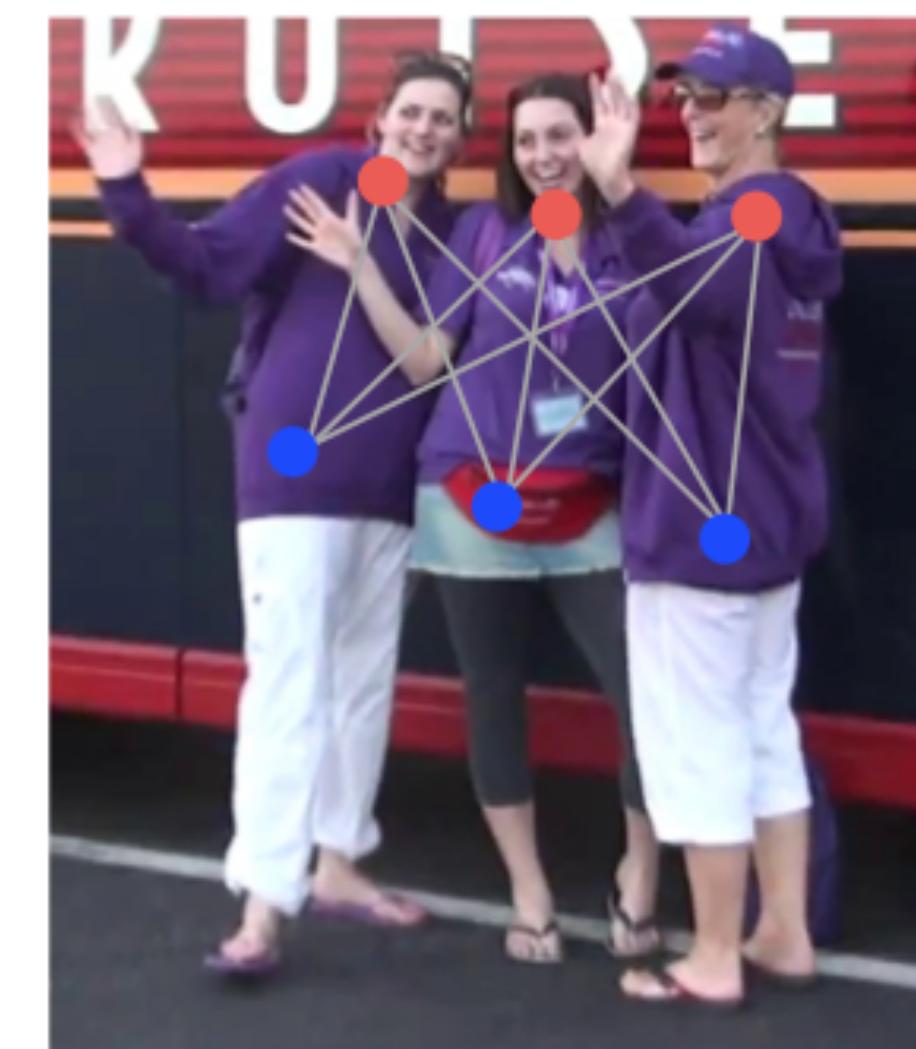
(각 person detector 마다 a single-person pose estimator이 run함)

- Bottom-up 방식 이용

- ▶ detect의 실패에 robust함 (object detection 과정 x)
- ▶ runtime이 사람 수와 크게 영향 없음
- 초기 bottom-up 모델은 한 이미지 당 수 분이 걸려 cost적 이점이 없었으나 이후 개선

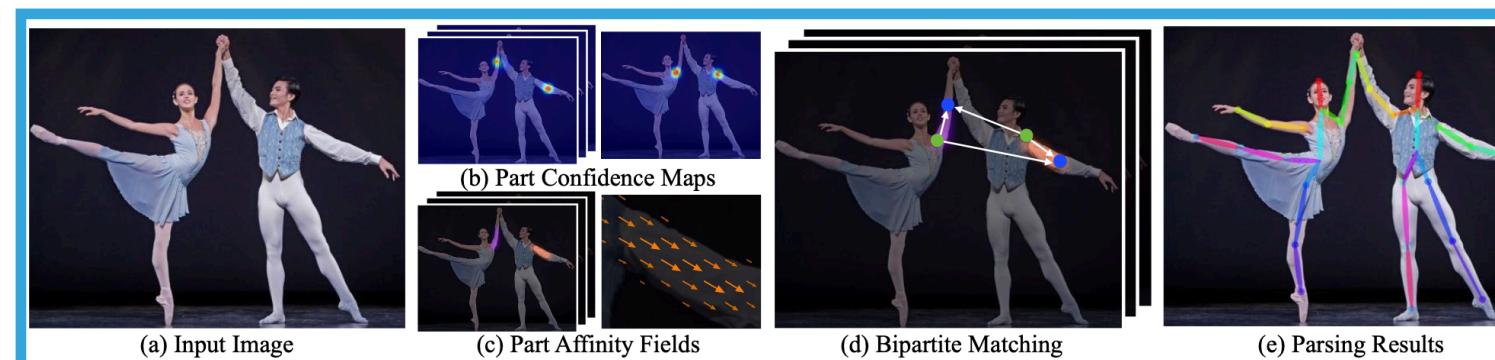
PAFs 제시 (논문에서 제안된 방식)

- ▶ 기존 Bottom-up 방식의 Challenge: accuracy & runtime 성능 개선 노력
 - 찾은 관절을 매칭할 수 있는 조합이 매우 많음. 매칭하는데 정확도가 떨어지고, 조합 증가에 따른 계산량의 증가 등의 문제가 존재. (= fully connected graph에서 integer linear programming를 해결하는 것은 NP-hard problem 야기)
 - **PAFs 제시 (+ 그리디 매칭)**
 - 신체 부위 사이의 중간점을 따내는 등 위치 정보를 추가하는 기법들이 제안되어 있었지만, 방향 정보 없이는 표현에 한계가 존재
- ▶ PAFs: 여러 사람의 관절들 사이에서 unstructured pairwise relationship를 인코딩하는 일련의 flow fields로 구성된 표현
 - PAFs = a set of 2D vector fields that encode the location and orientation of limbs over the image domain
 - PAF refinement is crucial for maximizing accuracy, while body part prediction refinement is not that important.



OPENPOSE: REALTIME MULTI-PERSON 2D POSE ESTIMATION USING PART AFFINITY FIELDS

METHOD

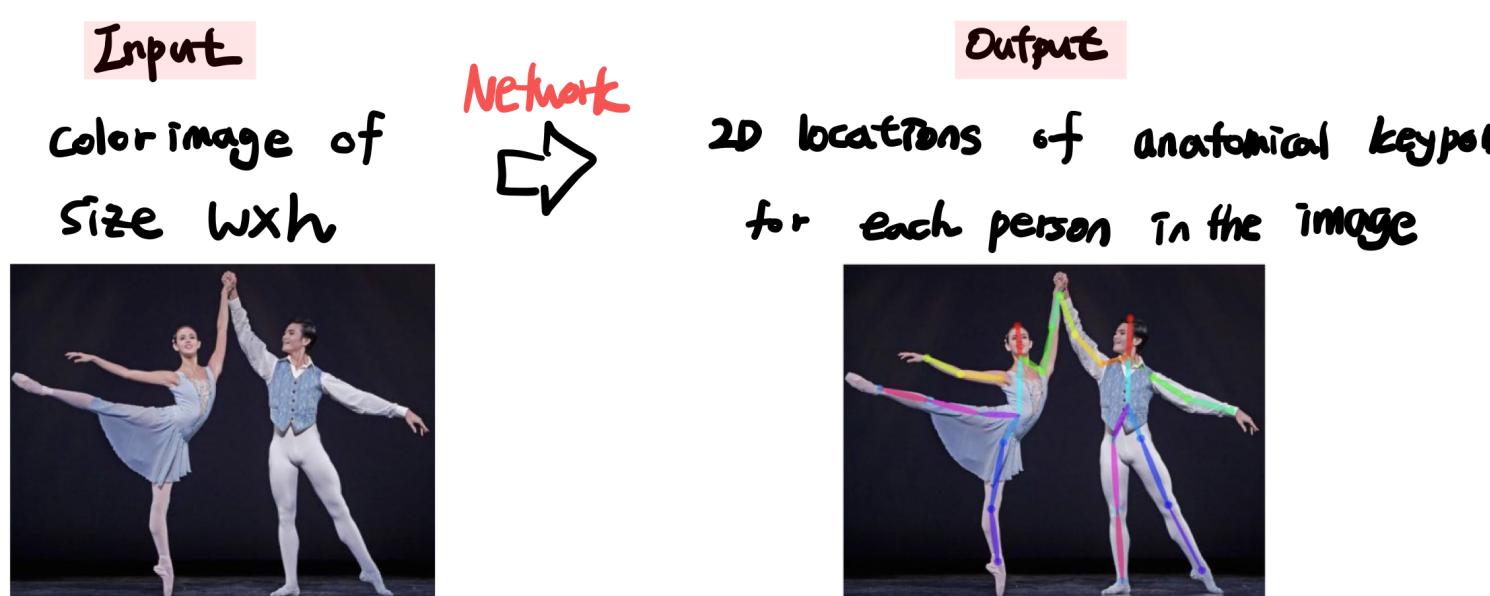


overall pipeline

- input: a color image of size $w \times h$ (a)
- 1) 관절 찾기: 이미지에서 Part Affinity Fields (b) & Part Confidence Maps (c)를 거쳐 관절 예측
- 2) 찾은 관절간의 관계 matching - Bipartite Matching (d)
- output: 이미지의 모든 사람(each)에 대한 keypoints의 2D locations (e)

Detail description (필기)

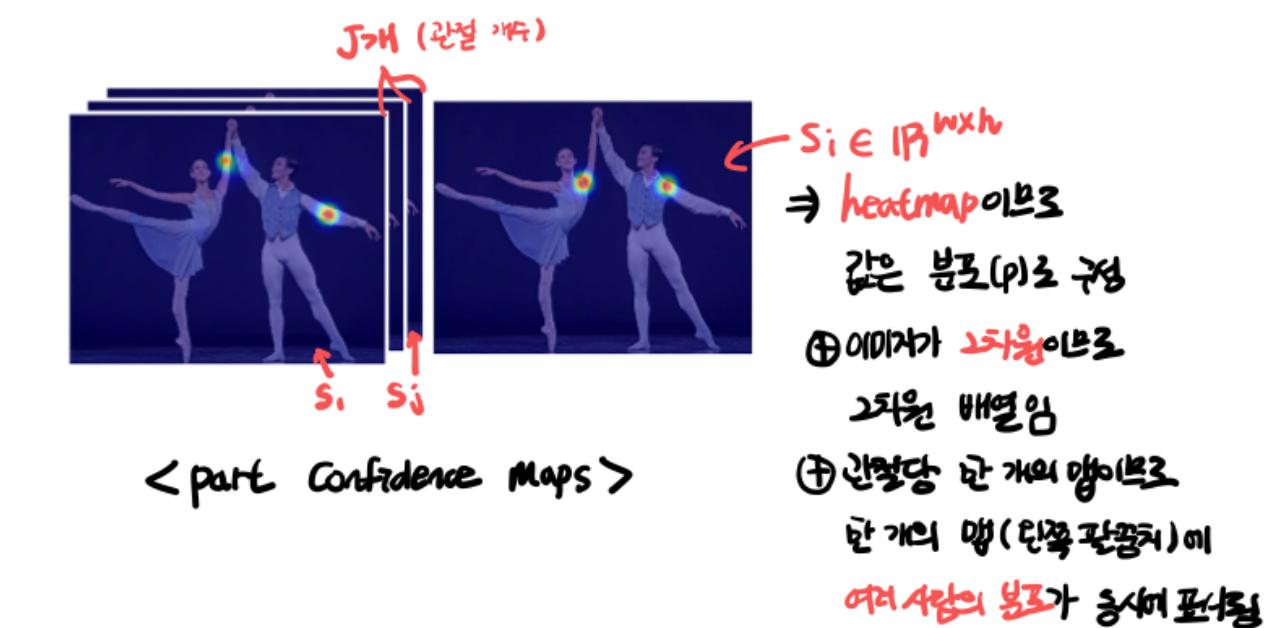
1 Input & output



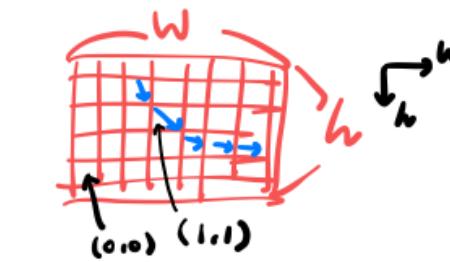
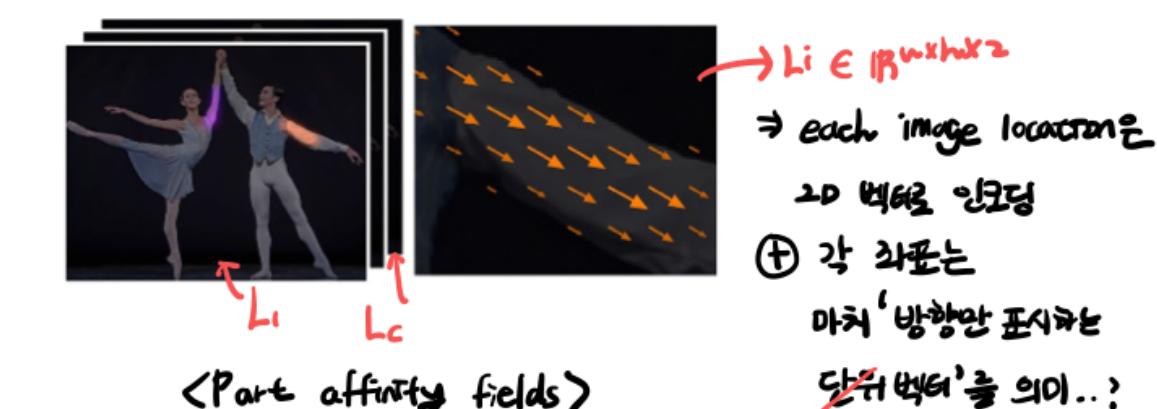
2 feed forward

예측 대상: 2D Confidence maps of body part locations S
predict 2D
⊕
2D vector fields of part affinity fields L

① $S_{\text{합}} = (S_1, S_2 \dots S_J), S_i \in \mathbb{R}^{w \times h}, i \in \{1, \dots, J\}$
 ↳ confidence maps - 관절 당 한개 (J 개)



② $L_{\text{합}} = (L_1, L_2, \dots L_c), L_c \in \mathbb{R}^{w \times h \times 2}, c \in \{1, \dots, C\}$
 ↳ Vector fields - limb 당 한개



2 greedy inference

Confidence maps & PAFs → **파싱** → 2D keypoints for all people in the image



NETWORK ARCHITECTURE (NETWORK DEPTH)

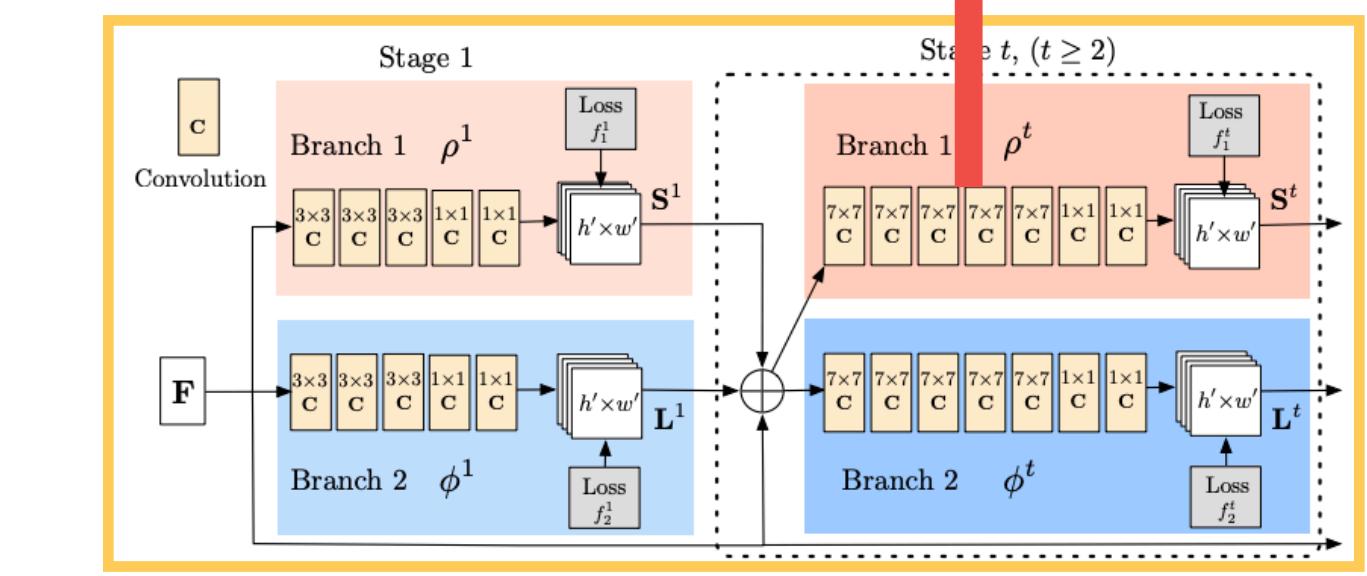
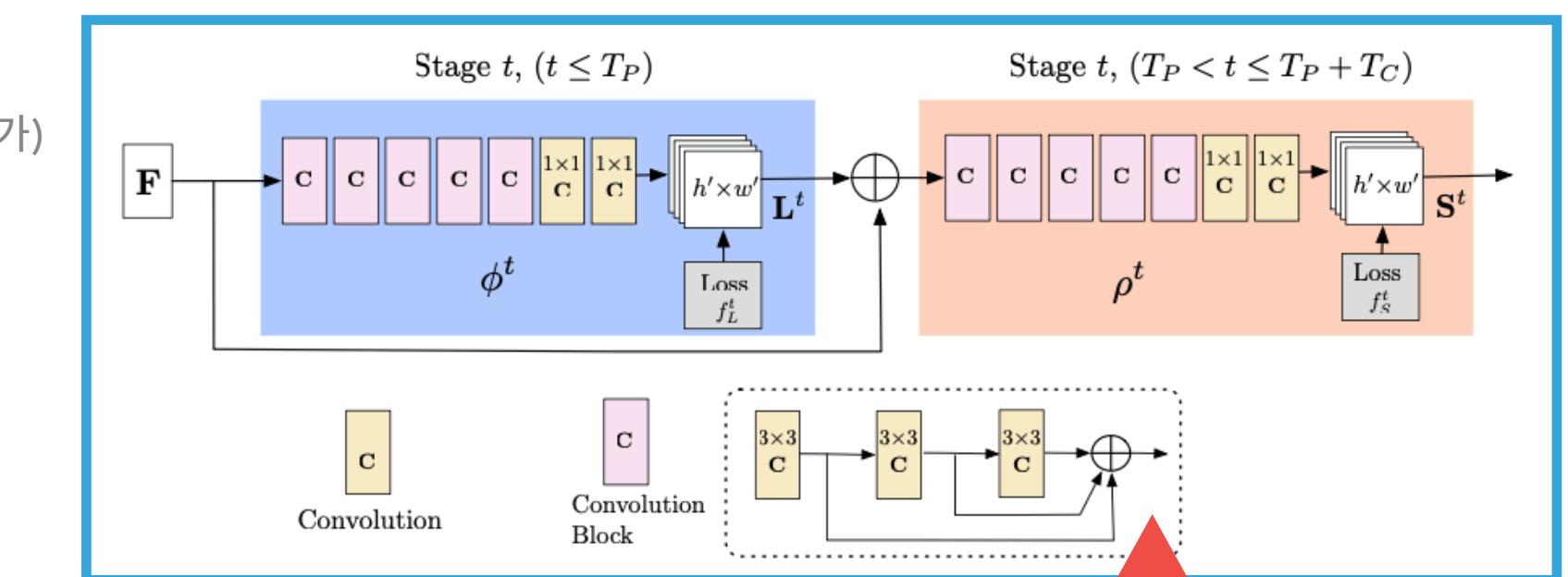
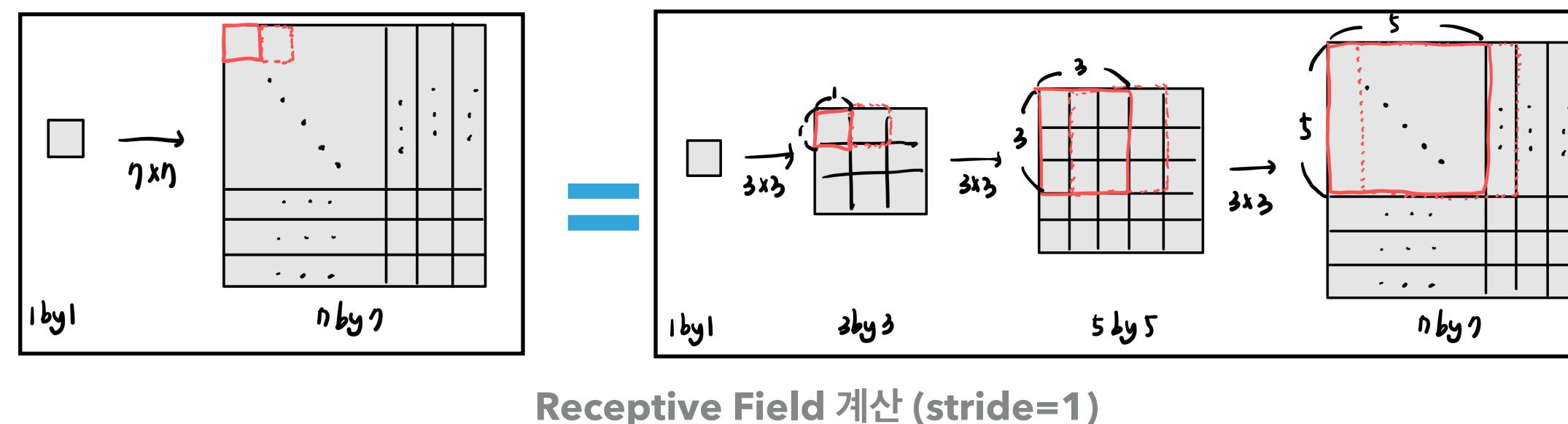
- ▶ feed forward = 순방향 네트워크

History ▶ Iterative prediction architecture with intermediate supervision at each stage

- 각 end of stage마다 intermediate supervision 시행
 - ▶ 최종 prediction에서만 loss를 구하는 것이 아니라 중간 과정에서도 loss를 구함

업데이트 ▶ (원래 버전) 7x7 convolutional kernels -> 3개의 연속된 3x3 convolutional kernels

- receptive field 보존 (그림 참고)
 - ▶ 각 stage의 입력 이미지에 대해 하나의 필터가 커버할 수 있는 이미지 영역의 일부 (레이어가 깊어질수록 field는 선형적으로 증가)
- more nonlinearity의 효과: 파라미터 수는 줄어들고 depth는 깊어짐
 - ▶ 계산 감소 (원래버전: $97 = 2 \cdot 7^2 \cdot 1$, 현버전: $51 = (2 \cdot 3^2 \cdot 1) \cdot 3$)
 - ▶ non-linearity layers이 3배로 증가 => network can keep both lower level and higher level features
- 3개의 컨볼루션 연산에 대한 출력을 concat

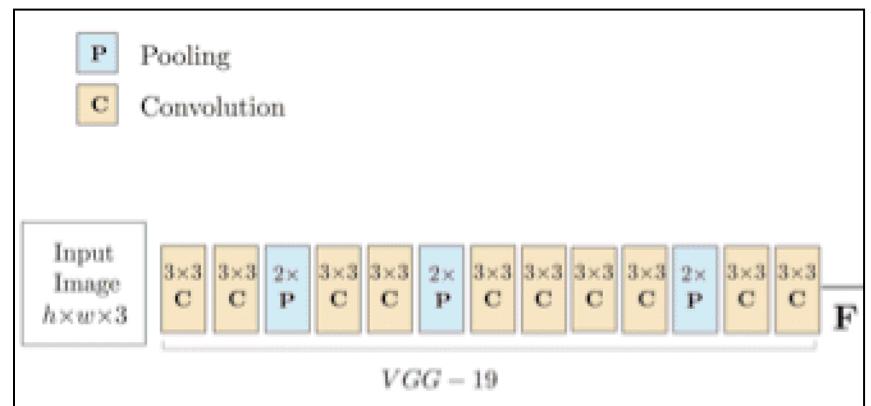


NETWORK ARCHITECTURE (MULTI-STAGE CNN)

History ▶ multi-stage CNN

- first stage - PAFs (L) 예측
- last stage - Confidence maps (S) 예측
- PAF stage와 Confidence Maps stage를 병렬에서 직렬로 수정
 - ▶ stage 당 연산량 절반 감소
 - ▶ PAF 예측을 개선할수록 CM 예측도 개선되지만 그 반대는 성립하지 않음 (그림1)

Step ▶ Feature map 생성: color image를 VGG-19에 통과시킴



▶ L^t 예측 (T_P 반복)

Stage 1

$$L^1 = \phi^1(F)$$

Stage 2~ T_P

$$L^t = \phi^t(F, L^{t-1}), \forall 2 \leq t \leq T_P$$

▶ S^t 예측 (T_C 반복)

Stage T_P

$$S^{T_P} = \rho^t(F, L^{T_P}), \forall t = T_P$$

Stage ($T_P + 1$)~ $T_P + T_C$

$$S^t = \rho^t(F, L^{T_P}, S^{t-1}), \forall T_P < t \leq T_P + T_C$$

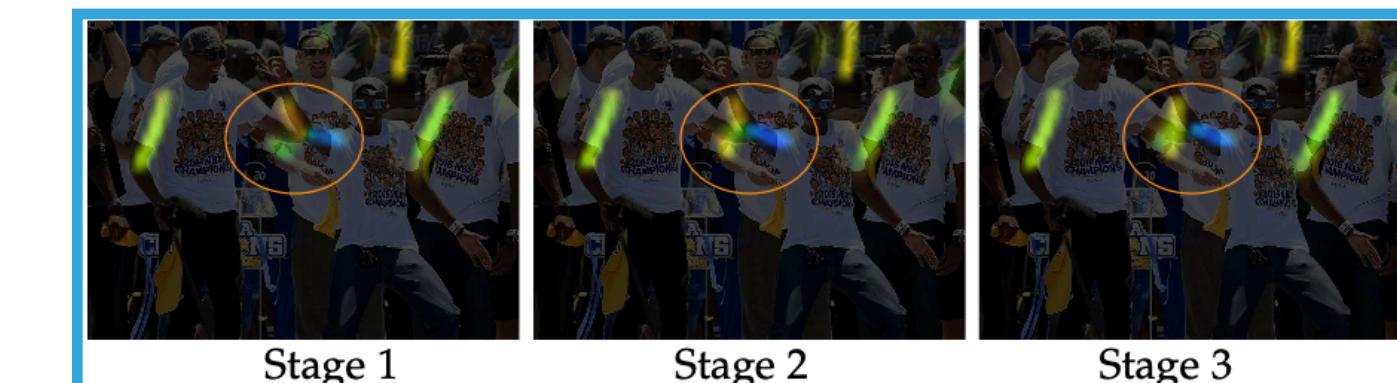
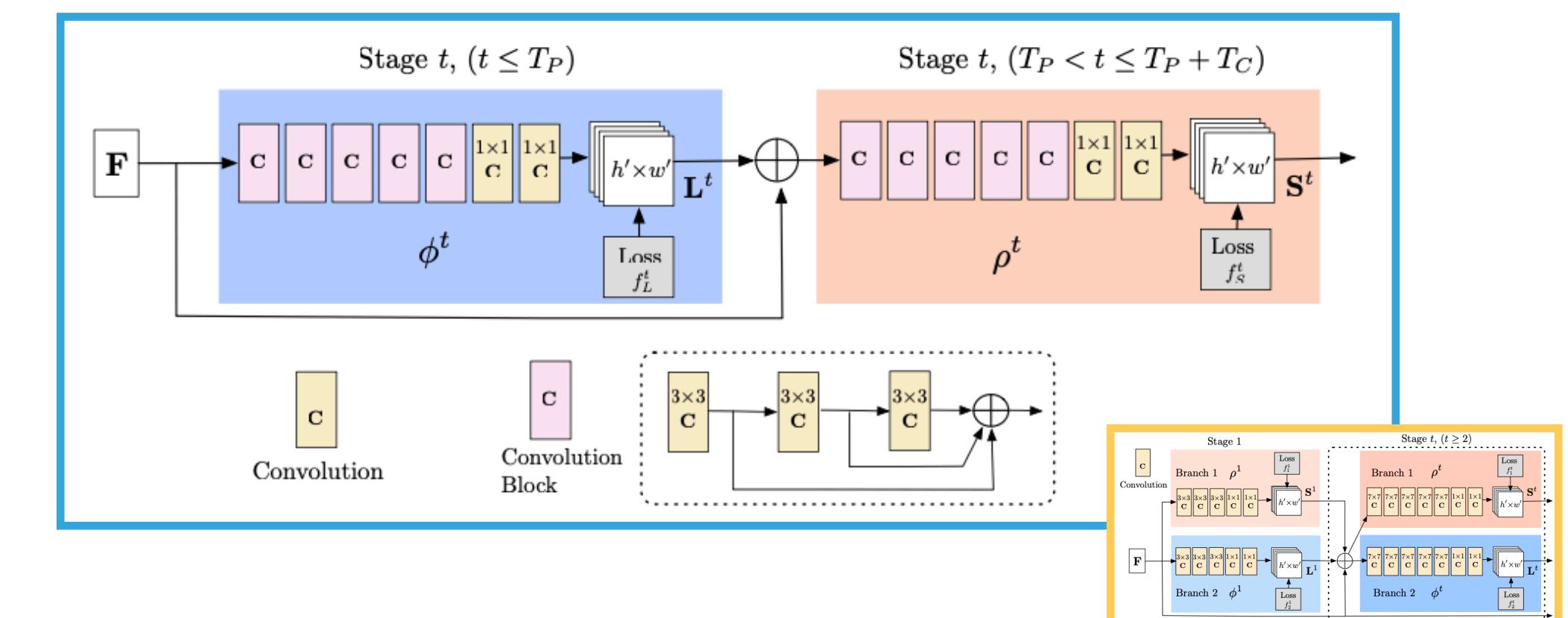


그림1



NETWORK ARCHITECTURE (MULTI-STAGE CNN)

업데이트 ▶ intermediate supervision

- end of stage마다 loss 구함

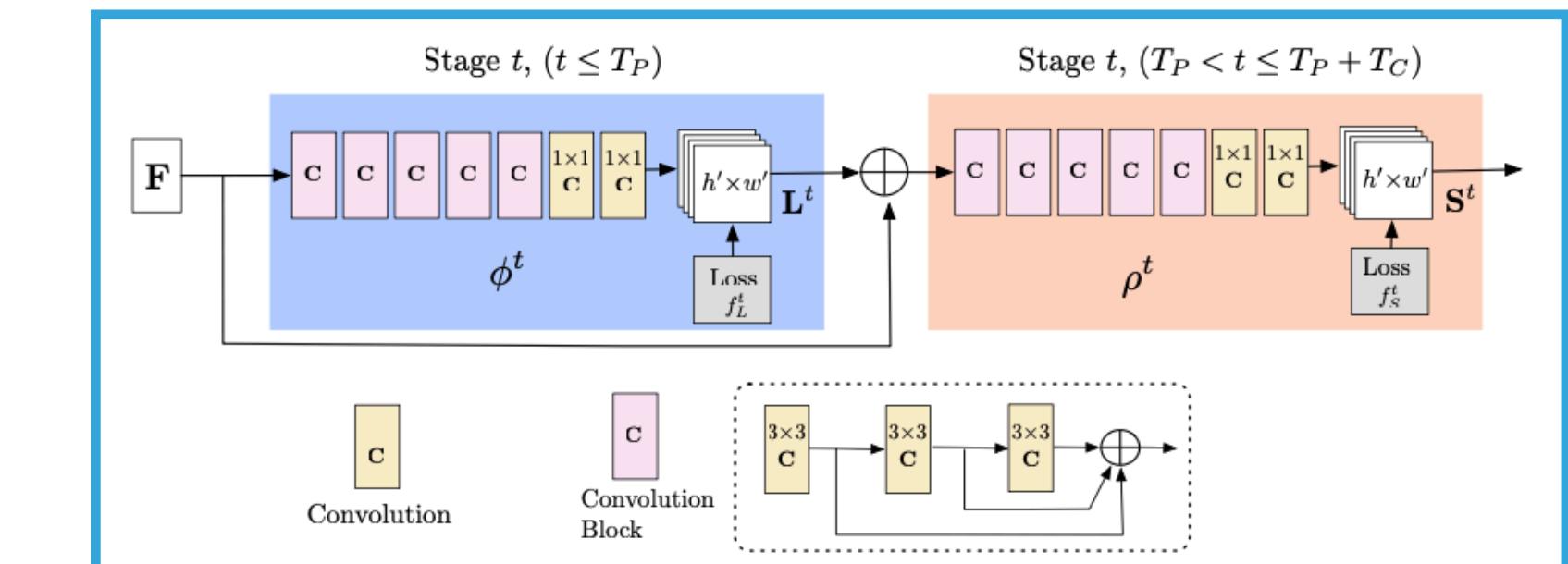
▶ 최종 loss function: 각 stage 별 loss function을 모두 합함

$$\begin{aligned} f_{\mathbf{L}}^{t_i} &= \sum_{c=1}^C \sum_{\mathbf{p}} \mathbf{W}(\mathbf{p}) \cdot \|\mathbf{L}_c^{t_i}(\mathbf{p}) - \mathbf{L}_c^*(\mathbf{p})\|_2^2, \\ f_{\mathbf{S}}^{t_k} &= \sum_{j=1}^J \sum_{\mathbf{p}} \mathbf{W}(\mathbf{p}) \cdot \|\mathbf{S}_j^{t_k}(\mathbf{p}) - \mathbf{S}_j^*(\mathbf{p})\|_2^2, \end{aligned} \quad > \quad f = \sum_{t=1}^{T_P} f_{\mathbf{L}}^t + \sum_{t=T_P+1}^{T_P+T_C} f_{\mathbf{S}}^t.$$

- L_2 loss
- $\mathbf{W}(\mathbf{p})$: binary mask
 - ▶ $\mathbf{W}(\mathbf{p}) = 0$ (annotation = 0일 때 (가려져서 라벨링 x))
 - ▶ true-positive의 penalty를 피함

▶ $\mathbf{L}(\mathbf{p})$ 와 $\mathbf{S}(\mathbf{p})$ 의 Ground Truth 생성해야함

- 가지고 있는 dataset은 annotated color image임



▶ 정의

- J: keypoint 개수, C: Limb 개수
- 집합 L: limb마다 PAFs(vecotor fields)를 가짐

$$L = (L_1, L_2, \dots, L_C)$$

$$L_c \in R^{w \times h \times 2}, c \in \{1 \dots C\}$$

(L 결과는 c개의 관절관계만큼 있음)

($L_c = (w \cdot h)^2$ 크기의 실수 집합, 각 image location은 2차원 벡터를 인코딩)

- 집합 S: keypoint마다 confidence maps를 가짐

$$S = (S_1, S_2, \dots, S_J)$$

$$S_j \in R^{w \times h}, j \in \{1 \dots J\}$$

(S 결과는 j개의 관절만큼 있음)

($S_j = (w \cdot h)$ 크기의 실수 집합, 이미지에 대한 Heatmap)

지도학습 - GROUND TRUTH 생성

- 지도학습에서 loss를 측정하기 위해 annotated dataset에서 GT 생성

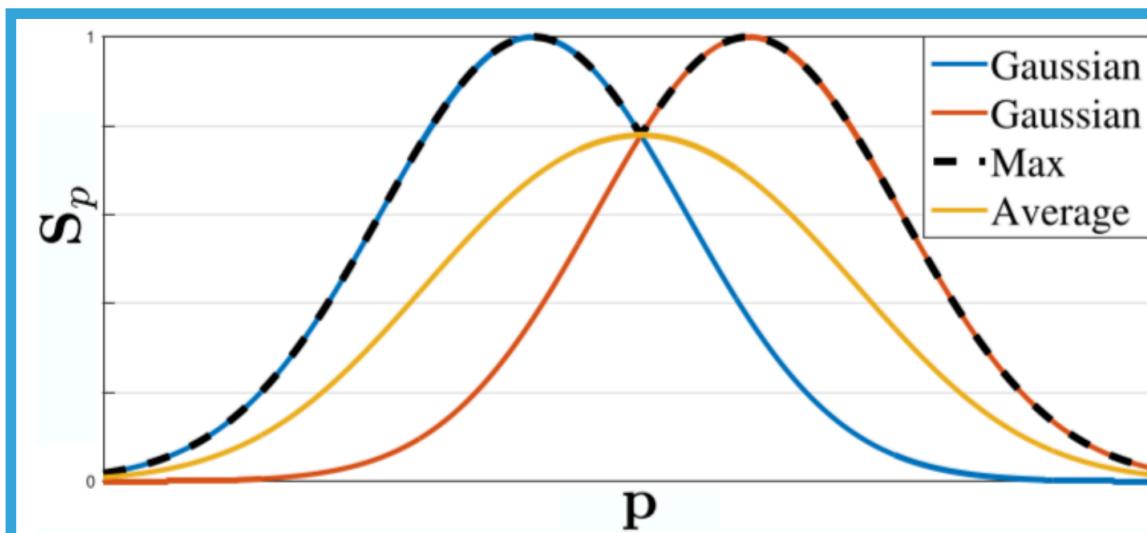
- Confidence Maps에 대한 GT 생성

$$S_{j,k}^*(\mathbf{p}) = \exp\left(-\frac{\|\mathbf{p} - \mathbf{x}_{j,k}\|_2^2}{\sigma^2}\right)$$

- ▶ 가우시안 분포 이용
- ▶ k번째 사람, j번째 관절에 대한 픽셀 p에서의 confidence map, x: k번째 사람의 j 번째 관절에 대한 GT

$$S_j^*(\mathbf{p}) = \max_k S_{j,k}^*(\mathbf{p})$$

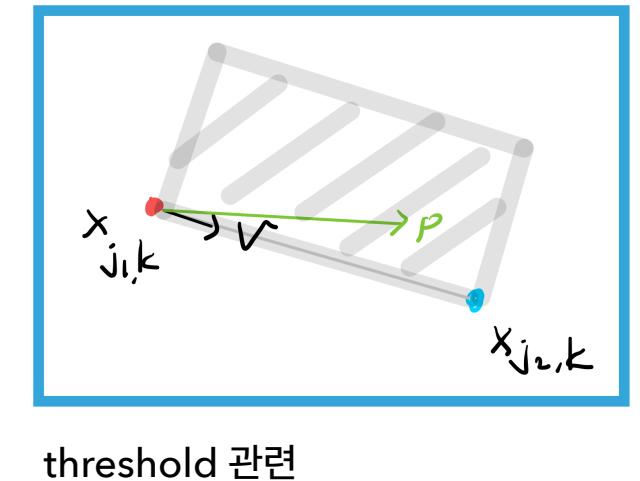
- ▶ k번째 사람의 heat map, 각 사람의 관절 cm을 max aggregation
- ▶ peak가 중요한 정보이므로
- ▶ test 시, 각 사람의 관절을 non-maximum suppression



- PAFs에 대한 GT 생성

- ▶ original: x 좌표

$$\mathbf{L}_{c,k}^*(\mathbf{p}) = \begin{cases} \mathbf{v} & \text{if } \mathbf{p} \text{ on limb } c, k \\ \mathbf{0} & \text{otherwise.} \end{cases}$$



threshold 관련

- ▶ unit vector: $\mathbf{v} = (\mathbf{x}_{j2,k} - \mathbf{x}_{j1,k}) / \|\mathbf{x}_{j2,k} - \mathbf{x}_{j1,k}\|_2$

- ▶ p의 threshold: $0 \leq \mathbf{v} \cdot (\mathbf{p} - \mathbf{x}_{j1,k}) \leq l_{c,k}$ and $|\mathbf{v}_\perp \cdot (\mathbf{p} - \mathbf{x}_{j1,k})| \leq \sigma_l$
($l_{c,k} = \|\mathbf{x}_{j2,k} - \mathbf{x}_{j1,k}\|_2$, σ_l : limb width를 정하는 하이퍼파라미터)

- ▶ 픽셀 당 최종 PAF: p에서 겹치는 모든 사람의 벡터의 평균

$$\mathbf{L}_c^*(\mathbf{p}) = \frac{1}{n_c(\mathbf{p})} \sum_k \mathbf{L}_{c,k}^*(\mathbf{p})$$

($n_c(\mathbf{p})$: the number of non-zero vectors at point \mathbf{p} across all k people)

