The OpenPose algorithm

Marco Sutti

May 24, 2021

Overview

- Realtime multi-person 2D pose estimation.
- Human 2D pose estimation: problem of localizing anatomical keypoints or parts.
  - From an image captured with RGB smartphone/tablet camera, use Deep Learning to estimate positions of body joints.
  - Convolutional Neural Network (CNN) that enables to predict location of joints of interest!
- Open source algorithm.

OpenPose: [Cao et al. '17, Cao et al. '21]

OpenPose – Main ideas

- Input: color image of size $w \times h$.
- Output: array of matrices, including:
  - A set of 2D Confidence Maps (CMs) $\mathbf{S} = (\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_J)$, where J is the number of parts, and $\mathbf{S}_j \in \mathbb{R}^{w \times h}$, $j \in \{1 \ldots J\}$. $\rightsquigarrow$ They show the location of parts (e.g., wrist, elbow, knee, etc.).

  - A set of Part Affinity Fields (PAFs) $\mathbf{L} = (\mathbf{L}_1, \mathbf{L}_2, \ldots, \mathbf{L}_C)$, where C is the number of limbs (or part pairs), and $\mathbf{L}_c \in \mathbb{R}^{w \times h \times 2}$, $c \in \{1 \ldots C\}$. $\rightsquigarrow$ Set of 2D vector fields that encode the degree of association between parts.

Pipeline

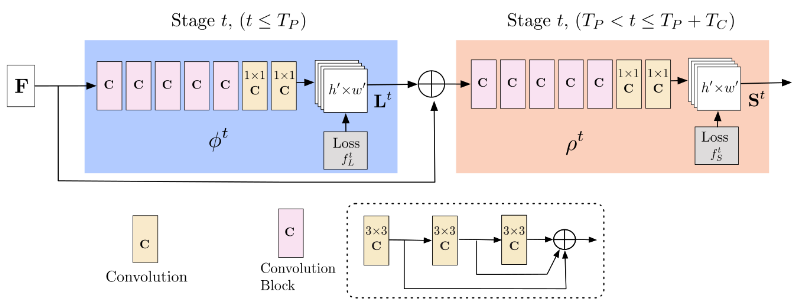
(a) Input Image (b) Part Confidence Maps (c) Part Affinity Fields (d) Bipartite Matching (e) Parsing Results

- Input: (a) color image of size $w \times h$.
- Convolutional Neural Network: jointly predicts (b) 2D CMs $\mathbf{S}$ for part detection and (c) 2D vector fields $\mathbf{L}$ of PAFs for part association.
- Parsing step: (d) performs a set of bipartite matchings to associate body part candidates in order to form limbs.
- Output: (e) assemble the 2D keypoints into full body poses for all people in the image.

Network Architecture

Multi-stage convolutional architecture

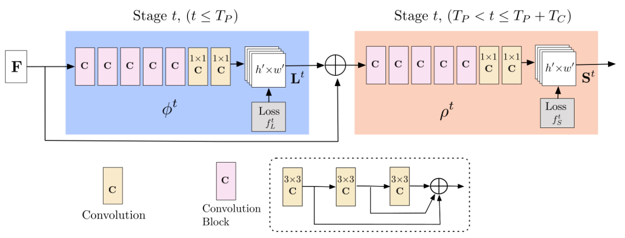
- Iteratively predicts PAFs $\mathbf{L}^t$ (left branch) and CMs $\mathbf{S}^t$ (right).



- Iterative prediction architecture: refines the prediction over successive stages, $t \in \{1, \ldots, T\}$, with intermediate supervision at each stage.

Iterative prediction architecture: [Wei et al. '16]

Joint Detection and Association/1

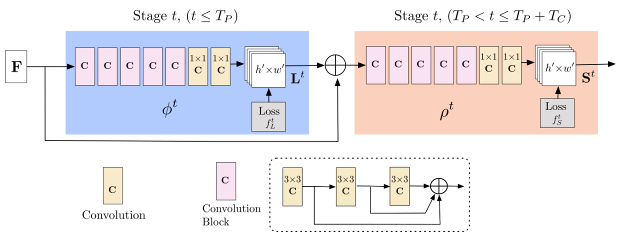


- The image is analyzed by a CNN, giving a set of feature maps **F**.
- First stage: $\mathbf{L}^1 = \phi^1(\mathbf{F})$, where ϕ^1 is the CNN for inference at Stage 1.
- Subsequent stages: concatenation of the PAF predictions $\mathbf{L}^{t-1}$ and the original **F** to produce refined predictions:

$$\mathbf{L}^t = \phi^t(\mathbf{F}, \mathbf{L}^{t-1}), \quad 2 \leqslant t \leqslant T_P,$$

where ϕ^t refers to CNNs for inference at Stage t, and T_P is the total number of PAFs stages.

Joint Detection and Association/2



- After T_P iterations, the process is repeated for the CMs detection, starting with the most updated PAF prediction, $\mathbf{L}^{T_P}$:

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

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

  where ρ^t refers to CNNs for inference at Stage t, and T_C is the total number of CM stages.

Remark: Refined PAF predictions improve the CM results (the opposite does not hold).

Loss Functions

- To guide the network to predict PAFs in the first branch and CMs in the second branch, we apply a loss function at the end of each stage.
- L_2 loss between estimated predictions and groundtruth fields and maps.
- Loss function of the PAF branch at stage t_i:

$$f_{\mathbf{L}}^{t_i} = \underbrace{\sum_{c=1}^{C}}_{\substack{\text{sum over}\\ \text{all limbs}}} \sum_{\mathbf{p}} \mathbf{W}(\mathbf{p}) \cdot \|\mathbf{L}_c^{t_i}(\mathbf{p}) - \mathbf{L}_c^{*}(\mathbf{p})\|_2^2.$$

- Loss function of the CM branch at stage t_k:

$$f_{\mathbf{S}}^{t_k} = \underbrace{\sum_{j=1}^{J}}_{\substack{\text{sum over}\\ \text{all parts}}} \sum_{\mathbf{p}} \mathbf{W}(\mathbf{p}) \cdot \|\mathbf{S}_j^{t_k}(\mathbf{p}) - \mathbf{S}_j^{*}(\mathbf{p})\|_2^2.$$

- $\mathbf{W}$ is a binary mask with $\mathbf{W}(\mathbf{p}) = 0$ when the annotation is missing at the pixel $\mathbf{p}$.
- Overall objective:

$$f = \sum_{t=1}^{T_P} f_{\mathbf{L}}^{t} + \sum_{t=T_P+1}^{T_P+T_C} f_{\mathbf{S}}^{t}.$$

Confidence Maps for Part Detection/1

- Training: to evaluate $f_{\mathbf{S}}$, generate groundtruth CMs $\mathbf{S}^*$ from the annotated 2D keypoints.
- Confidence map (CM): 2D representation of the belief that a particular body part can be located in any given pixel.
  - If single person in image: single peak should exist in each CM if the corresponding part j is visible.
  - If multiple people: there should be a peak corresponding to each visible part j for each person k:

Confidence Maps for Part Detection/2

- For each person k, we generate individual groundtruth CMs $\mathbf{S}^*_{j,k}$

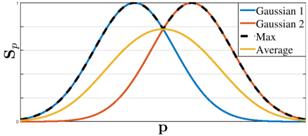
$$\mathbf{S}^*_{j,k}(\mathbf{p}) = \exp\left(-\frac{\|\mathbf{p} - \mathbf{x}_{j,k}\|_2^2}{\sigma^2}\right),$$

  with $\mathbf{x}_{j,k}$ groundtruth position of body part j for person k, σ^2 variance.

- Groundtruth CM for part j: to evaluate $f_{\mathbf{S}}$, aggregation of the individual groundtruth CMs via a max operator:

$$\mathbf{S}^*_j(\mathbf{p}) = \max_k \mathbf{S}^*_{j,k}(\mathbf{p}).$$

Max vs average: taking the max of the CMs instead of their average allows to keep distinct the nearby peaks:

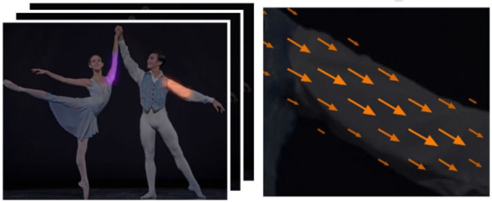


Part Affinity Fields for Part Association/1

Given a set of detected body parts, how do we assemble them to form the limbs of an unknown number of people?

$\rightsquigarrow$ Part Affinity Fields (PAFs).

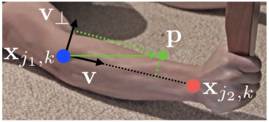
- They encode both location and orientation over the support of the limb.



- Each PAF is a 2D vector field for each limb.
  - For each $\mathbf{p}$ in the area belonging to a given limb, a 2D vector encodes the direction that points from one part of the limb to the other.
  - Each limb has a corresponding PAF joining its two associated body parts.

Part Affinity Fields for Part Association/2

Consider a single limb c:



- $\mathbf{x}_{j_1,k}$ and $\mathbf{x}_{j_2,k}$ groundtruth positions of body part j_1 (right elbow ●) and j_2 (right wrist ●) from the limb c for person k.
- Training: to evaluate f_{L}, define the groundtruth PAF $\mathbf{L}^*_{c,k}$ at an image point $\mathbf{p}$ as

$$\mathbf{L}^*_{c,k}(\mathbf{p}) = \begin{cases} \mathbf{v} & \text{if } \mathbf{p} \text{ is on limb } c, \text{ for person } k, \\ \mathbf{0} & \text{otherwise,} \end{cases} \qquad \mathbf{v} = \frac{\mathbf{x}_{j_2,k} - \mathbf{x}_{j_1,k}}{\|\mathbf{x}_{j_2,k} - \mathbf{x}_{j_1,k}\|_2}.$$

- Groundtruth PAF for limb c: average of the groundtruth PAFs of all people in the image, i.e.,

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

where $n_c(\mathbf{p})$ is the number of nonzero vectors at $\mathbf{p}$ across all k people.

Part Affinity Fields for Part Association/3

- Testing: we measure the association between two candidate part locations $\mathbf{d}_{j_1}$ and $\mathbf{d}_{j_2}$ by computing the line integral of the corresponding PAF along the line segment connecting $\mathbf{d}_{j_1}$ and $\mathbf{d}_{j_2}$,

$$E = \int_{u=0}^{u=1} \mathbf{L}_c(\mathbf{p}(u)) \cdot \frac{\mathbf{d}_{j_2} - \mathbf{d}_{j_1}}{\|\mathbf{d}_{j_2} - \mathbf{d}_{j_1}\|_2} \, \mathrm{d}u,$$

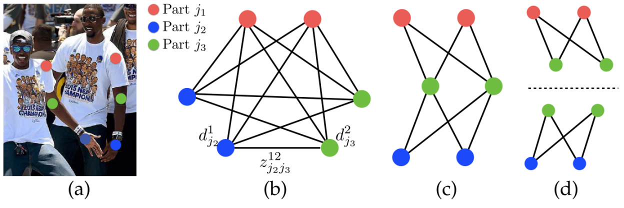
where $\mathbf{p}(u)$ is the parametrized segment connecting $\mathbf{d}_{j_1}$ and $\mathbf{d}_{j_2}$

$$\mathbf{p}(u) = (1-u)\mathbf{d}_{j_1} + u\mathbf{d}_{j_2}.$$

$\rightsquigarrow$ This gives a score for each candidate limb.

Multi-Person Parsing Using PAFs/1

How do we handle the case of multiple people in the same image?



- Due to multiple people in the image, we may have several candidates for each part (Fig. (a)). Example: we have two candidates for both j_1 (left shoulder ●), j_2 (left hand ●), and j_3 (left elbow ●).
- Each candidate is scored using the line integral computation on the PAF, i.e.,

$$E = \int_{u=0}^{u=1} \mathbf{L}_c(\mathbf{p}(u)) \cdot \frac{\mathbf{d}_{j_2} - \mathbf{d}_{j_1}}{\|\mathbf{d}_{j_2} - \mathbf{d}_{j_1}\|_2} \, du.$$

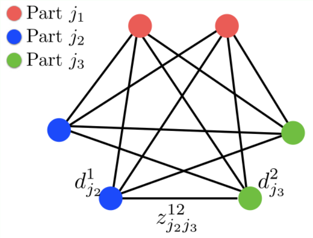
- Finding the optimal parse is a K-dimensional matching problem that is known to be NP-Hard (Fig. (c)).
- OpenPose uses a greedy relaxation that produces high-quality matches.

Multi-Person Parsing Using PAFs/2

- Set of body part detection candidates for multiple people:

$$\mathcal{D}_{\mathcal{J}} = \{\mathbf{d}_j^m : \text{for } j \in \{1 \dots J\}, m \in \{1 \dots N_j\}\}.$$

- Find the pairs of part detection candidates that are connected limbs.

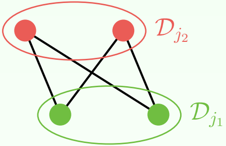


- $z_{j_1 j_2}^{mn} \in \{0, 1\}$ indicates whether two detection candidates $\mathbf{d}_{j_1}^m$ and $\mathbf{d}_{j_2}^n$ are connected. Example: $z_{j_2 j_3}^{12}$ for $\mathbf{d}_{j_2}^1$ and $\mathbf{d}_{j_3}^2$.
- Goal: find the optimal assignment for the set of all possible connections

$$\mathcal{Z} = \{z_{j_1 j_2}^{mn} : \text{for } j_1, j_2 \in \{1 \dots J\}, m \in \{1 \dots N_{j_1}\}, n \in \{1 \dots N_{j_2}\}\}.$$

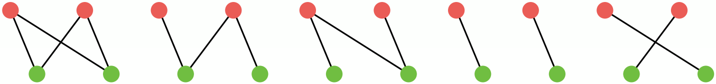
Multi-Person Parsing Using PAFs/3

- Consider a single pair of parts j_1 and j_2 for the cth limb.



  - Nodes of the graph: sets of body part detection candidates $\mathcal{D}_{j_1}$ and $\mathcal{D}_{j_2}$.
  - Edges: all possible connections between pairs of detection candidates. Plus: each edge is weighted by the affinity score E_{mn}.

- Finding the optimal association reduces to a maximum weight bipartite graph matching problem.
  $\rightsquigarrow$ A bipartite graph is a graph whose vertices can be divided into two disjoint and independent sets U and V such that every edge connects a vertex in U to one in V.
  $\rightsquigarrow$ Matching: subset of the edges chosen s.t. no two edges share a node.



NO **NO** **NO** **OK** **OK**

Multi-Person Parsing Using PAFs/4

- Goal: find a matching with maximum weight for the chosen edges, i.e.,

$$\max_{\mathcal{Z}_c} E_c = \max_{\mathcal{Z}_c} \sum_{m \in \mathcal{D}_{j_1}} \sum_{n \in \mathcal{D}_{j_2}} E_{mn} \cdot z_{j_1 j_2}^{mn},$$

$$\text{s.t.} \quad \forall m \in \mathcal{D}_{j_1}, \sum_{n \in \mathcal{D}_{j_2}} z_{j_1 j_2}^{mn} \leqslant 1, \qquad \forall n \in \mathcal{D}_{j_2}, \sum_{m \in \mathcal{D}_{j_1}} z_{j_1 j_2}^{mn} \leqslant 1,$$

  where E_c: overall weight of the matching for limb type c, $\mathcal{Z}_c$: subset of $\mathcal{Z}$ for limb type c, E_{mn}: part affinity score between parts $\mathbf{d}_{j_1}$ and $\mathbf{d}_{j_2}$.

- The two inequalities enforce that no two edges share a node, i.e., no two limbs of the same type share a body part.
- Determining $\mathcal{Z}$ is a NP-Hard problem.

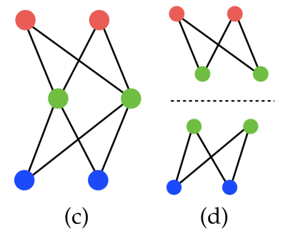
$\rightsquigarrow$ Add two relaxations to the optimization.

Multi-Person Parsing Using PAFs/5

⇝ Add two relaxations to the optimization problem:

(1) Choose a minimal number of edges to obtain a spanning tree skeleton (c) rather than using the complete graph.

(2) Decompose the matching problem into a set of bipartite matching subproblems (d).
⇝ We obtain the limb connection candidates for each limb type independently.



⇝ With all limb connection candidates, we can assemble the connections that share the same part detection candidates into full-body poses.

Examples

⇝ Latest portable version of OpenPose for Windows from `https://github.com/CMU-Perceptual-Computing-Lab/openpose`.

Conclusions

- Open source.
- Efficient while preserving accuracy.
- Uses 2D videos/images instead of 3D.

References:

[1] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using part affinity fields", IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 1, pp. 172–186, Jan. 2021.

[2] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields", in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1302–1310.

謝謝！