CS640: Introduction To Pose Estimation

Mahir Patel

Problem Definition

- Goal: Extract abstract structure of your subject from given input image.
- Why would we need this?

Image source: Her, 2013

Problem Definition

- Goal: Extract abstract structure of your subject from given input image.
- Why would we need this?
  - Fine grained localization
  - Activity/Motion Analysis
  - Applications: Robotics, AR/VR, Activity Feedback (sports, exercise, Surveillance.

Image source: Her, 2013

Problem Definition

- Goal: Extract abstract structure of your subject from given input image.
- Why would we need this?
  - Fine grained localization
  - Activity/Motion Analysis
  - Applications: Robotics, AR/VR, Activity Feedback (sports, exercise, Surveillance.
- How do we quantify this abstract structure?

Image source: Her, 2013

Problem Definition

- Estimate locations of certain “keypoints” or “joints”.
- Keypoints collectively define an abstract and instantaneous description of an activity.
- What should your model predict?

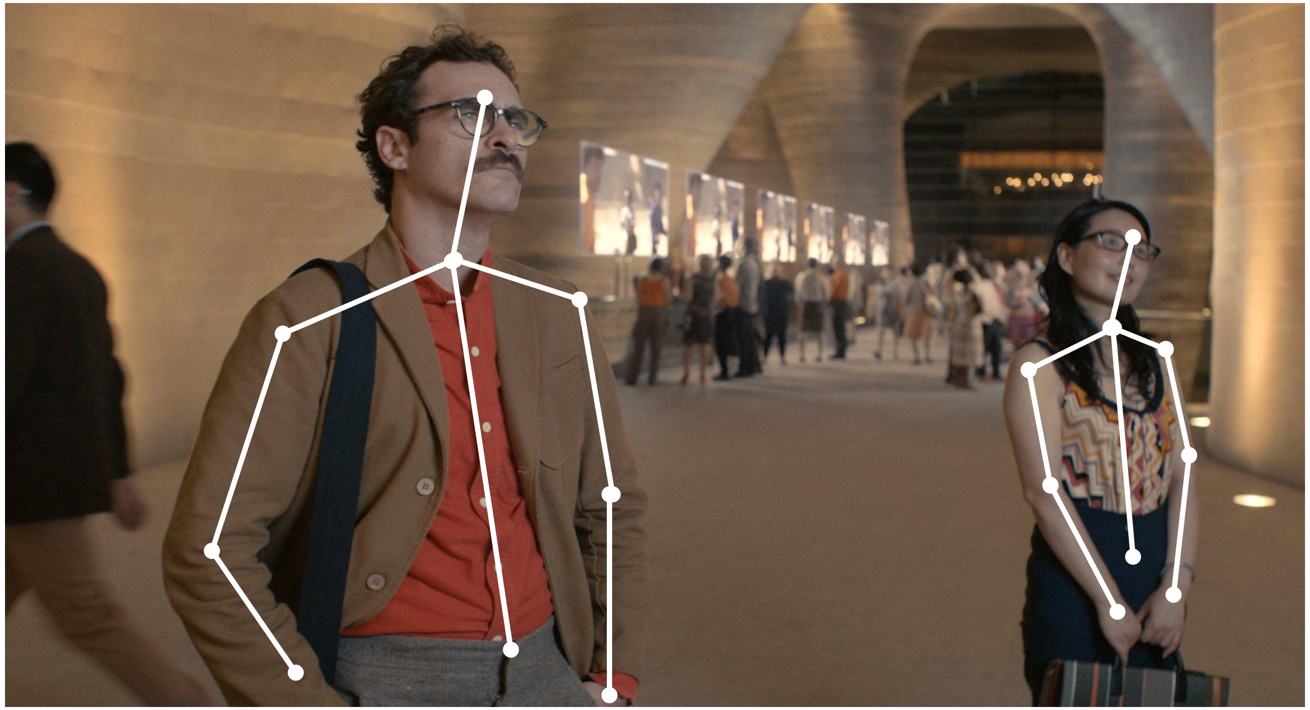


Image source: Her, 2013

Problem Definition

- Estimate locations of certain "keypoints" or "joints".
- Keypoints collectively define an abstract and instantaneous description of an activity.
- What should your model predict?
  - Image Coordinates of each keypoint

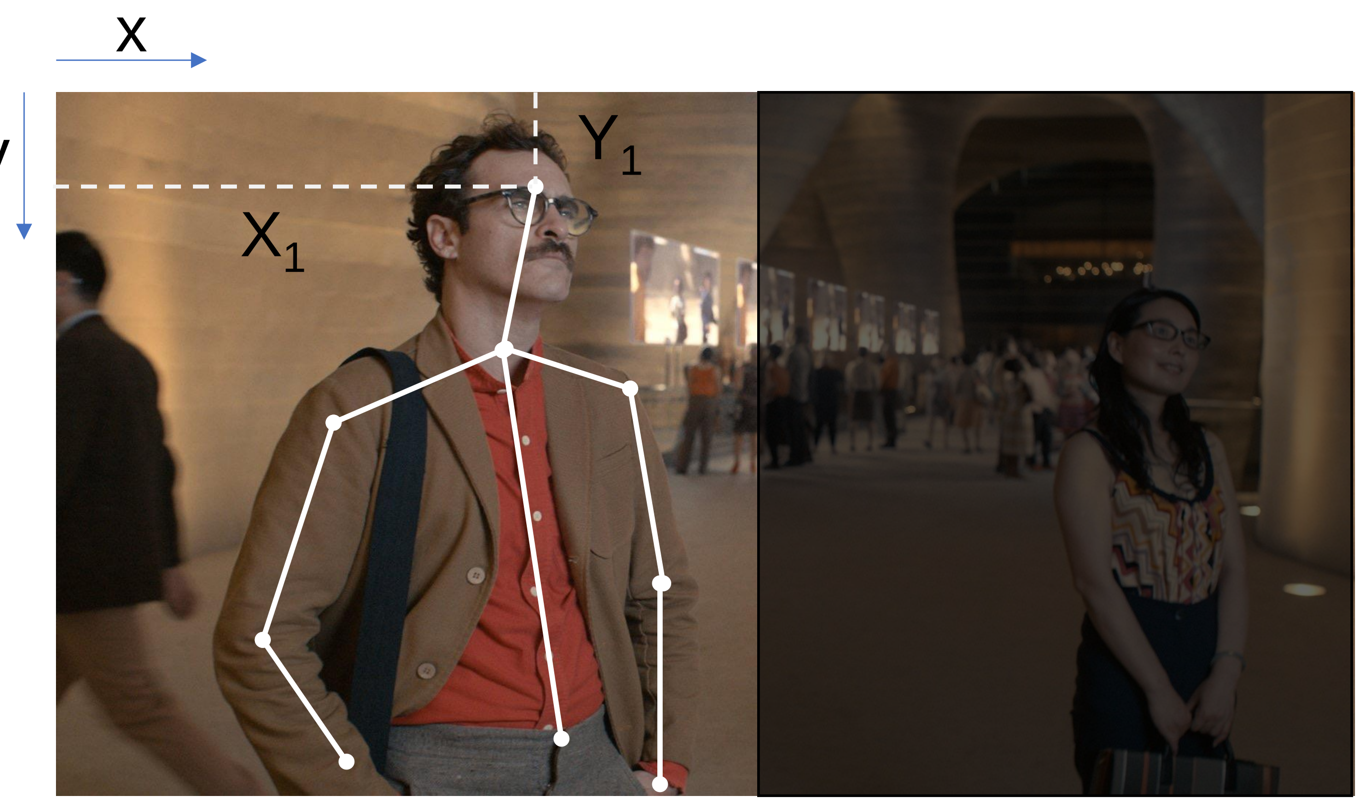


Image source: Her, 2013

2D Pose Estimation - Overview

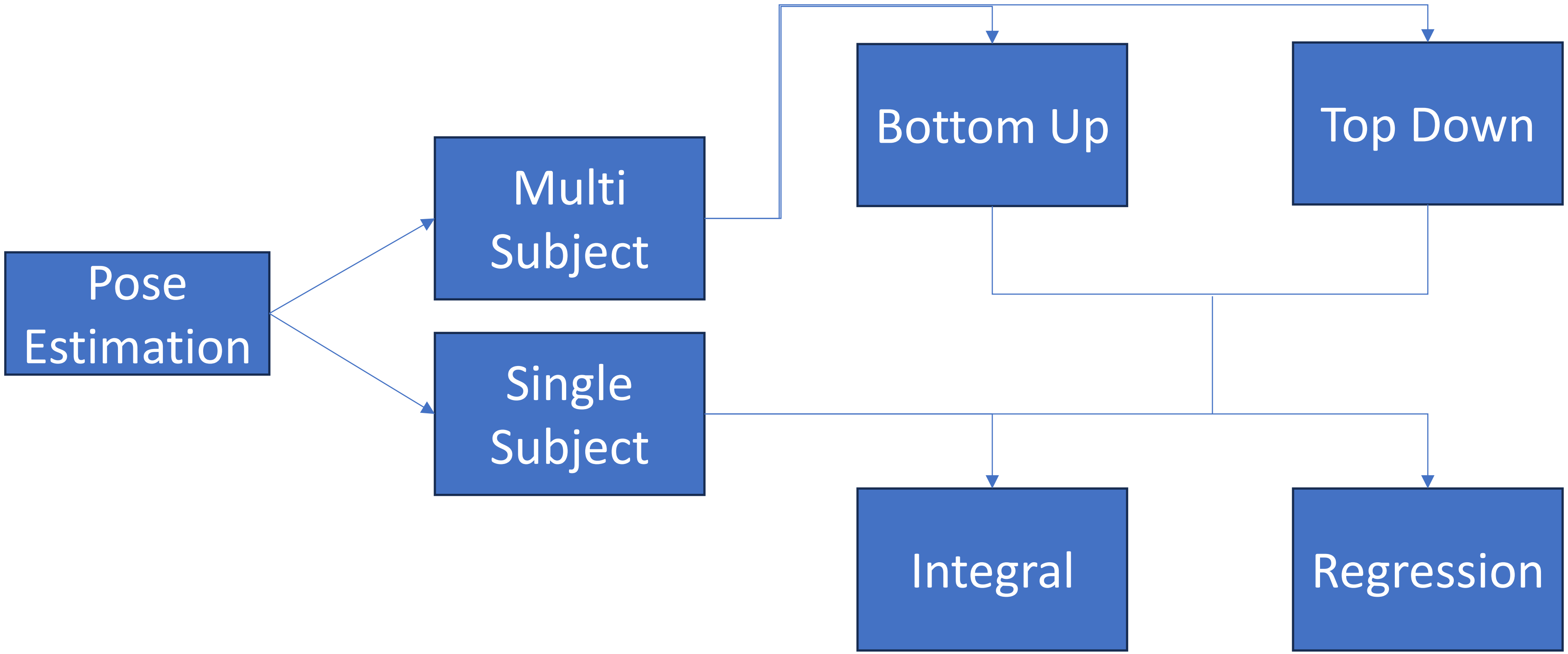


Image source: Her, 2013

2D Pose Estimation - Overview

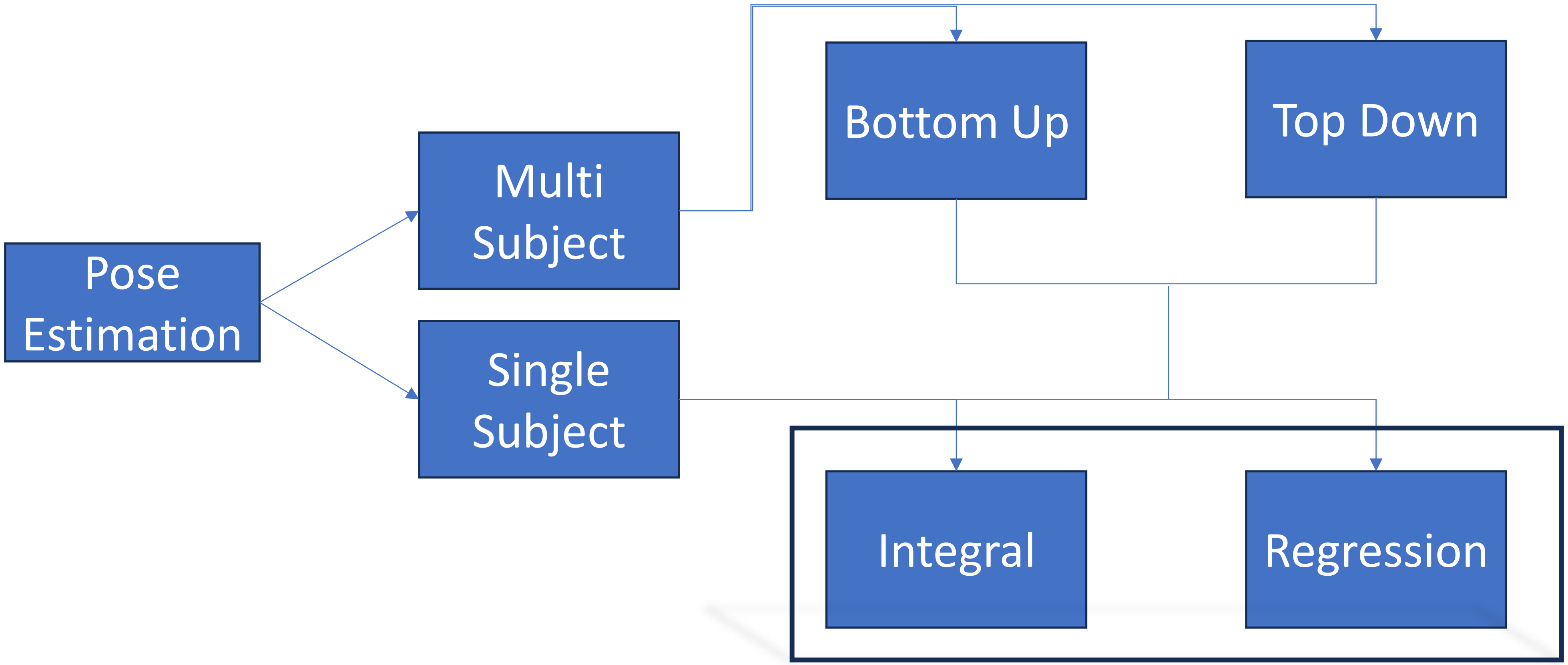
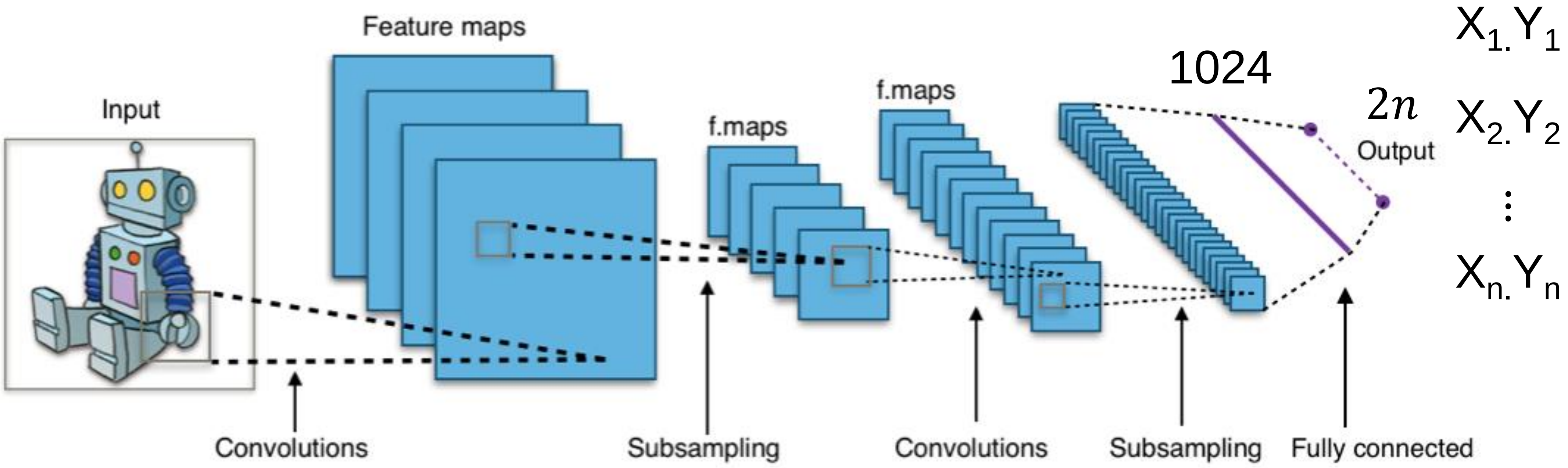


Image source: Her, 2013

Regression Based Approaches

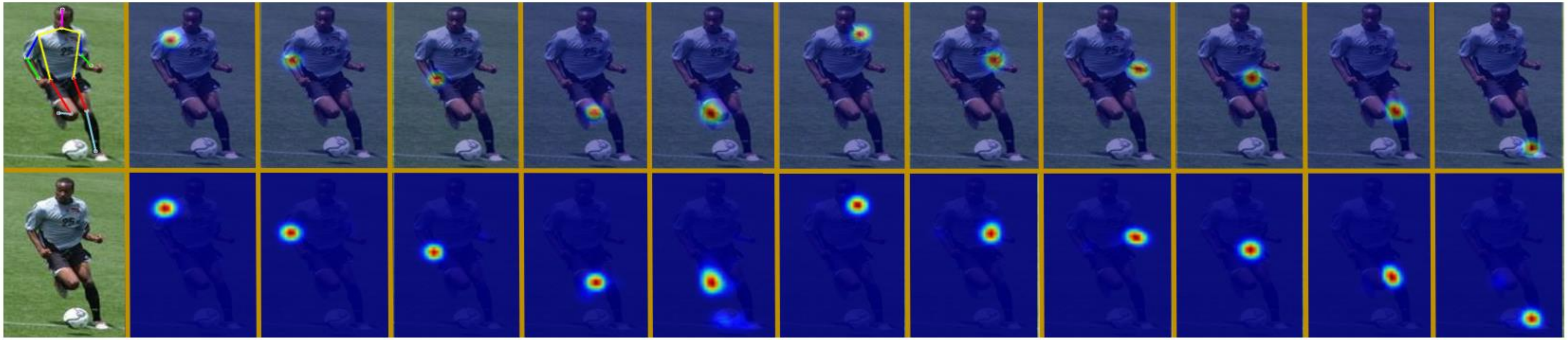
- Directly compute $n \times 2$ real number from input image.
- Labels are discrete 2D positions



source

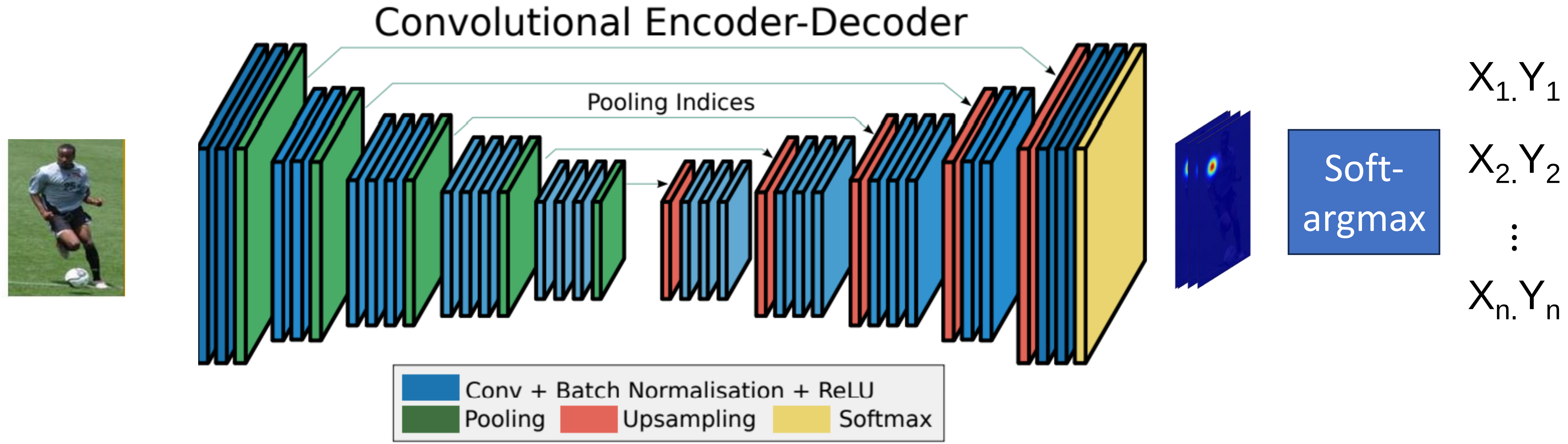
Integral Approaches

- Predict n $H \times W$ “heatmaps” where the hottest region in each heatmap corresponds to the location of the keypoint on the image.
- Uses feature map up-sampling techniques to reconstruct the higher resolution heatmaps



source

Integral Approaches



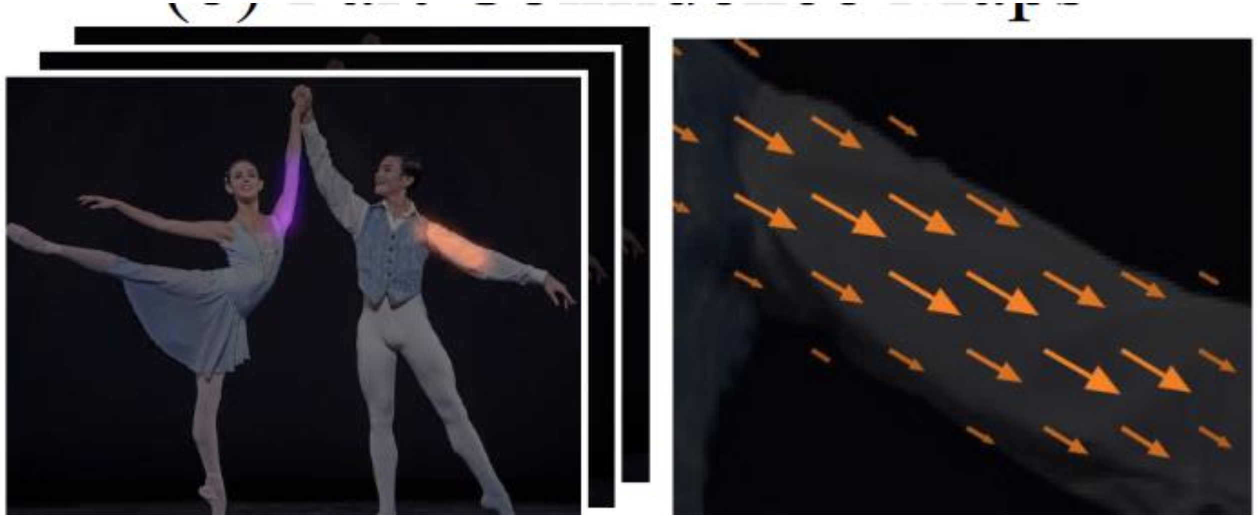
source

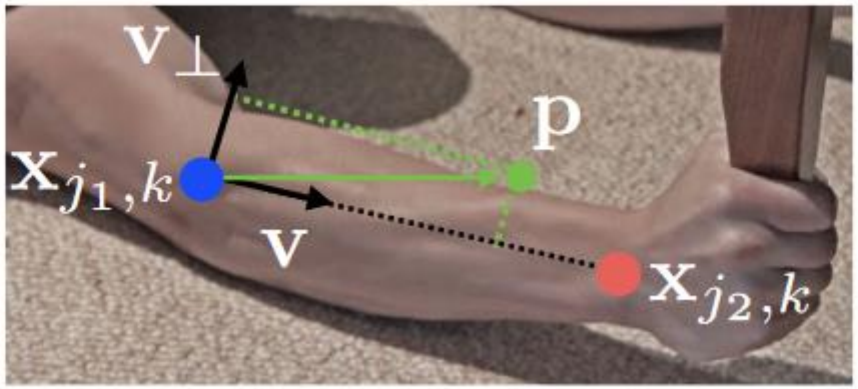
Integral Approaches

- Additional spatial bio-mechanics priors can be induced by designing additional prediction targets
- Each PAF is a HxWx2 heatmap where a pixel P is assigned v if they are within a certain threshold distance from the line-segment between the two joints.

$$v = \frac{x_{j_2,k} - x_{j_1,k}}{||x_{j_2,k} - x_{j_1,k}||_2}$$

Parts **A**ffinity **F**ields





OpenPose

Regression Vs Integral Approaches

- Most of the recent works opt for Integral approaches since it preserves spatial relations among pixels.
- The regression-based approaches have an infinite range of output. Which makes optimization difficult.
- Model cannot predict locations outside of the image in integral approaches. Range is limited to $H \times W$
- Using only convolutional layers reduces the number of parameters.

2D Pose Estimation - Overview

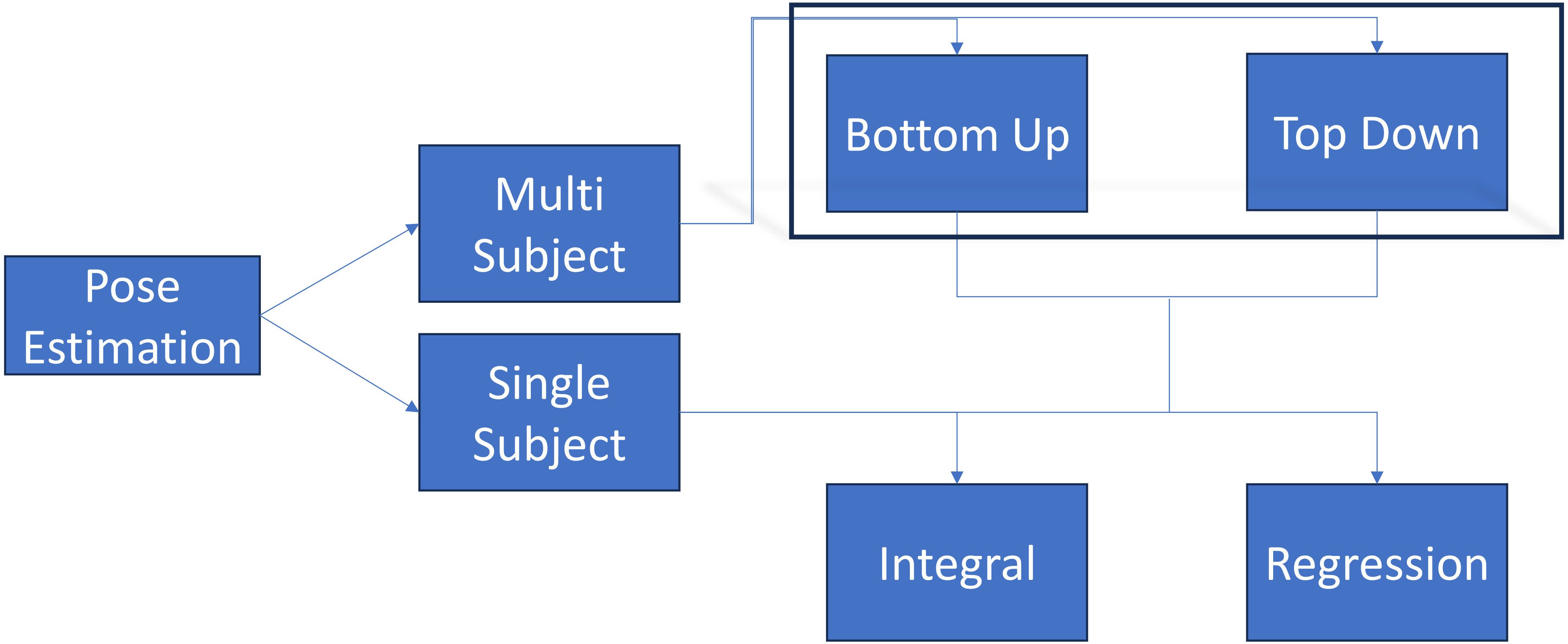
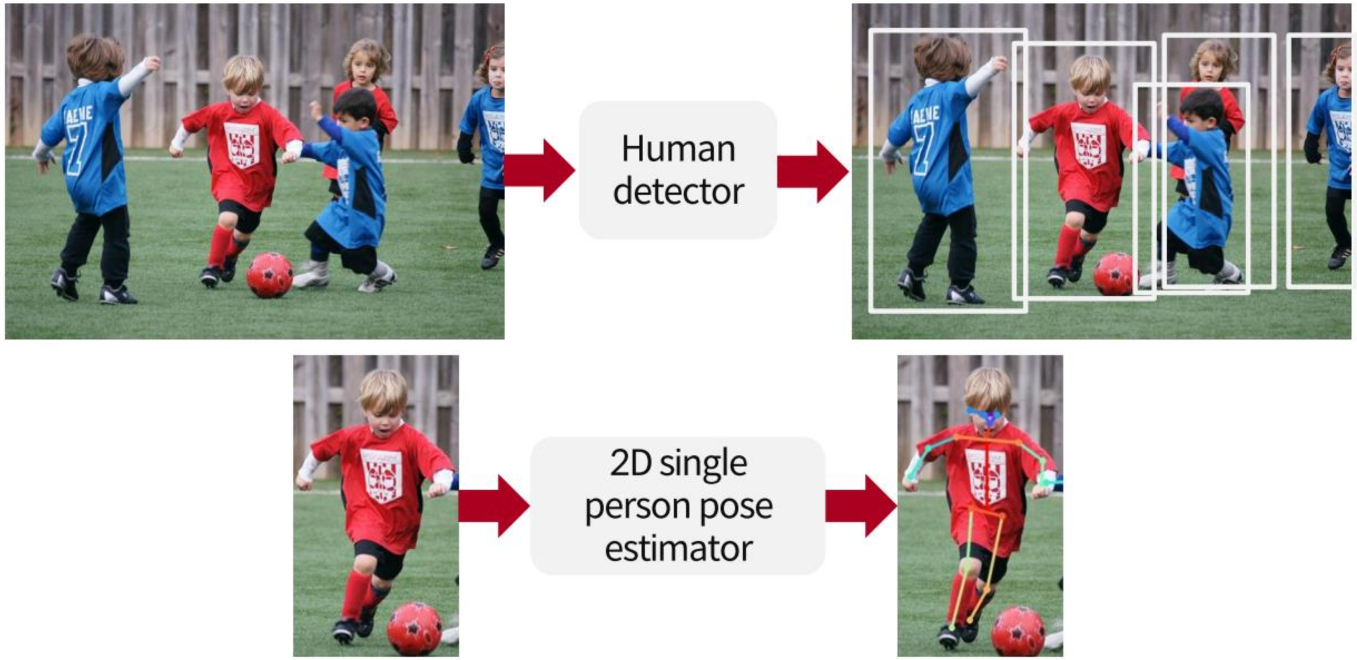


Image source: Her, 2013

Top-Down vs Bottom-Up

- Predict a bounding box for each subject.
- Perform single person pose estimation for each detection.



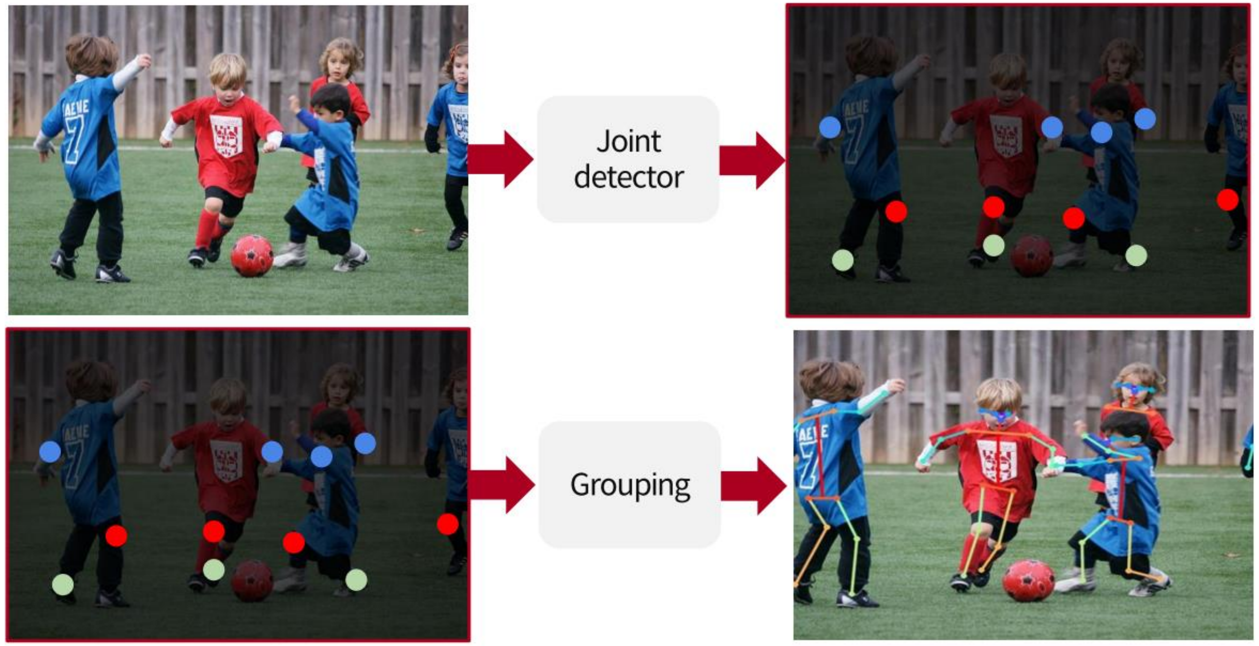
Top-Down

source

Top-Down vs Bottom-Up

- Predict all keypoints.
- Perform data association to group them into individual skeletons.

source



Bottom-Up

Top-Down vs Bottom-Up

What approach is better?

Top-Down vs Bottom-Up

What approach is better?

Top-Down

- Requires additional model for detecting bounding boxes.
- The computation requirement is directly proportional to the number of subjects in the image.

Bottom-Up

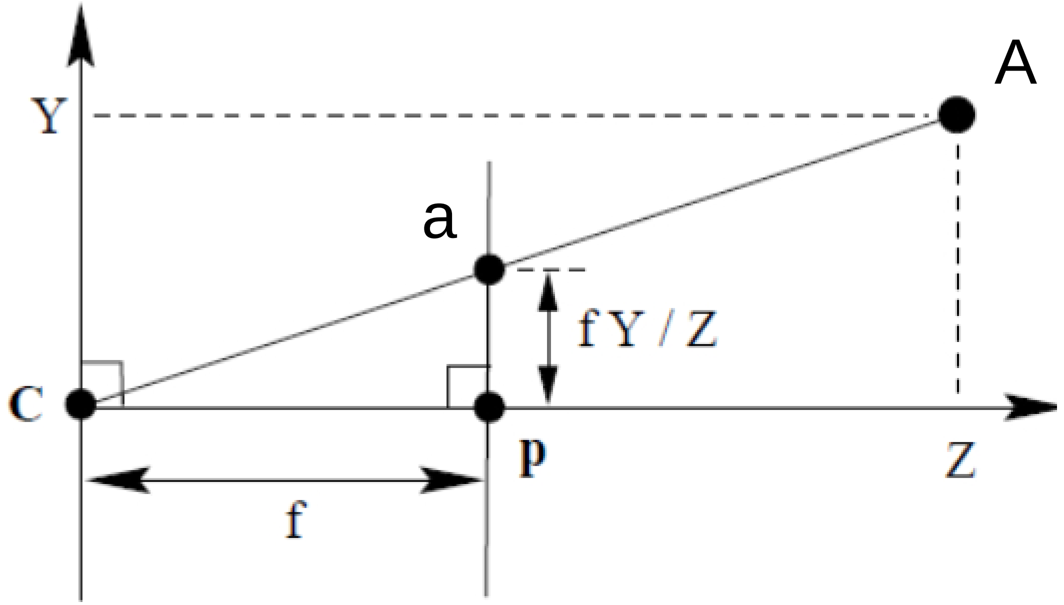
- Data association might not be straightforward, especially in situation where multiple subjects are in very close proximity.
- Normally requires expensive matching algorithms or bio-mechanics priors.

3D Pose Estimation

- Camera model mapping 3D points in world coordinate system to the image plane.
- 3D Pose Estimation architectures
- Multi-View Consistency for self-supervised 3DPE

Camera Model

- Naïve Pinhole Camera Model
- Assuming zero distortion



C = camera center
P = principal point (Image center)
f = focal length
A = point in 3D at (X,Y,Z)
a = point on image plane at (x,y)

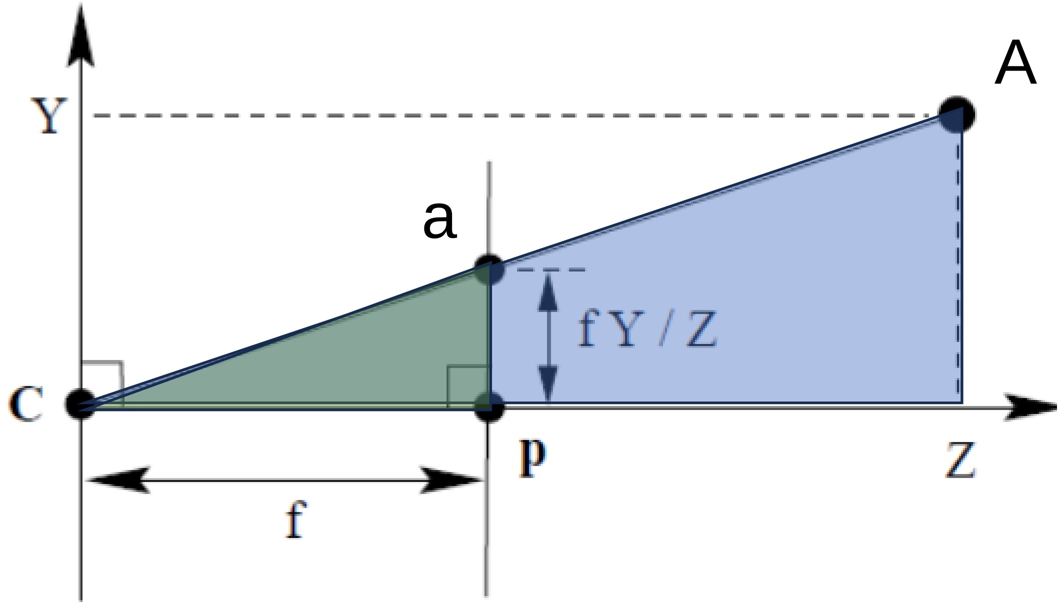
Camera Model

$$\frac{y}{f} = \frac{Y}{Z}$$

$$\therefore y = \frac{fY}{Z}$$

Similarly,

$$x = \frac{fX}{Z}$$



C = camera center
P = principal point (Image center)
f = focal length
A = point in 3D at (X,Y,Z)
a = point on image plane at (x,y)

BOSTON UNIVERSITY

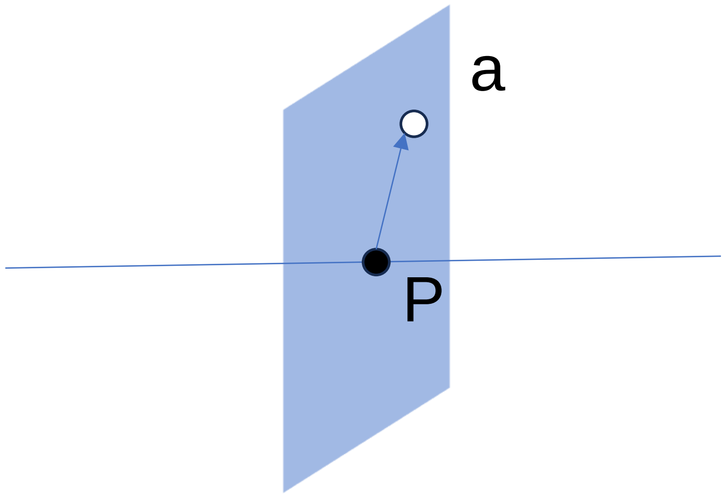
Camera Model

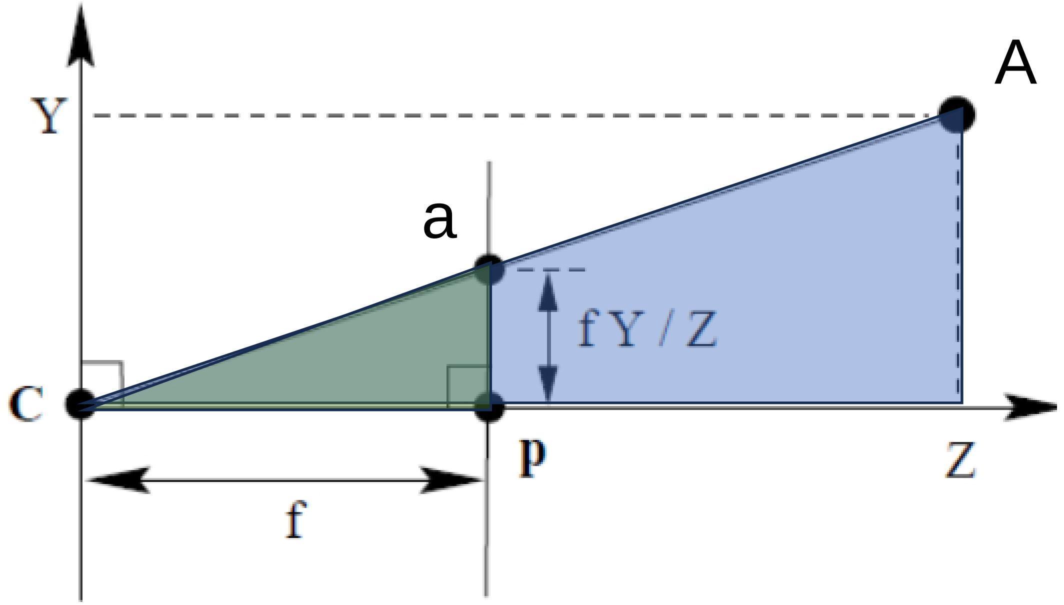
$$\frac{y}{f} = \frac{Y}{Z}$$

$$\therefore y = \frac{fY}{Z}$$

Similarly,

$$x = \frac{fX}{Z}$$





C = camera center
P = principal point (Image center)
f = focal length
A = point in 3D at (X,Y,Z)
a = point on image plane at (x,y)

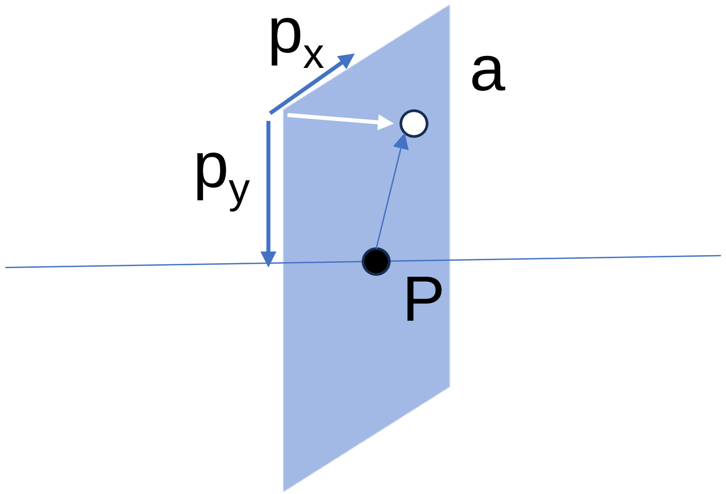
Camera Model

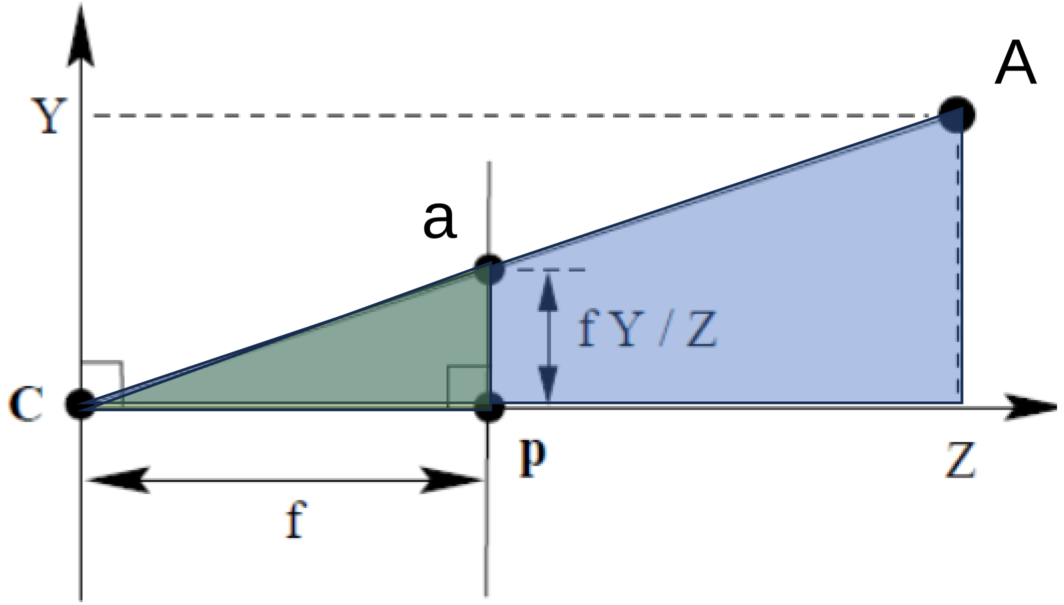
$$\frac{y}{f} = \frac{Y}{Z}$$

$$\therefore y = \frac{fY}{Z}$$

Similarly,

$$x = \frac{fX}{Z}$$





C = camera center
P = principal point (Image center)
f = focal length
A = point in 3D at (X,Y,Z)
a = point on image plane at (x,y)

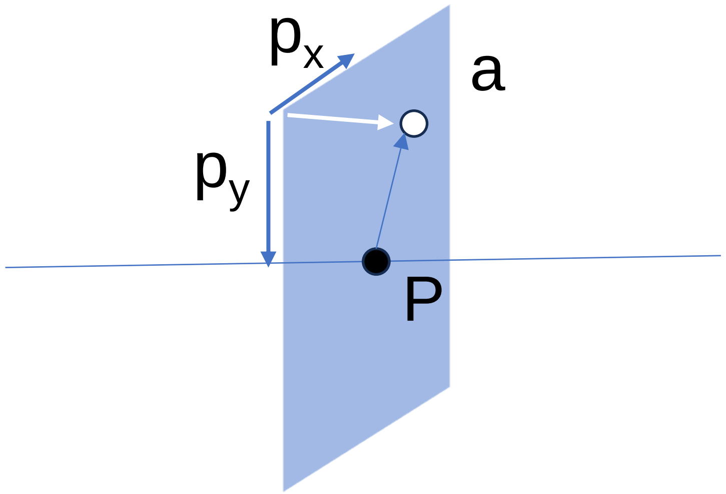
Camera Model

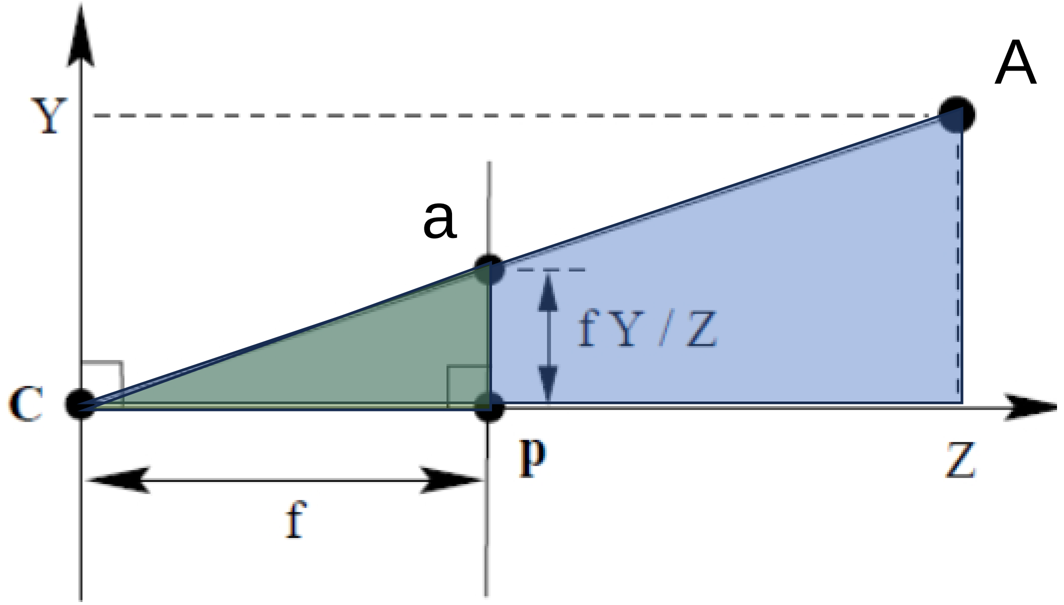
$$\frac{y}{f} = \frac{Y}{Z}$$

$$\therefore y = \frac{fY}{Z} + p_y$$

Similarly,

$$x = \frac{fX}{Z} + p_x$$



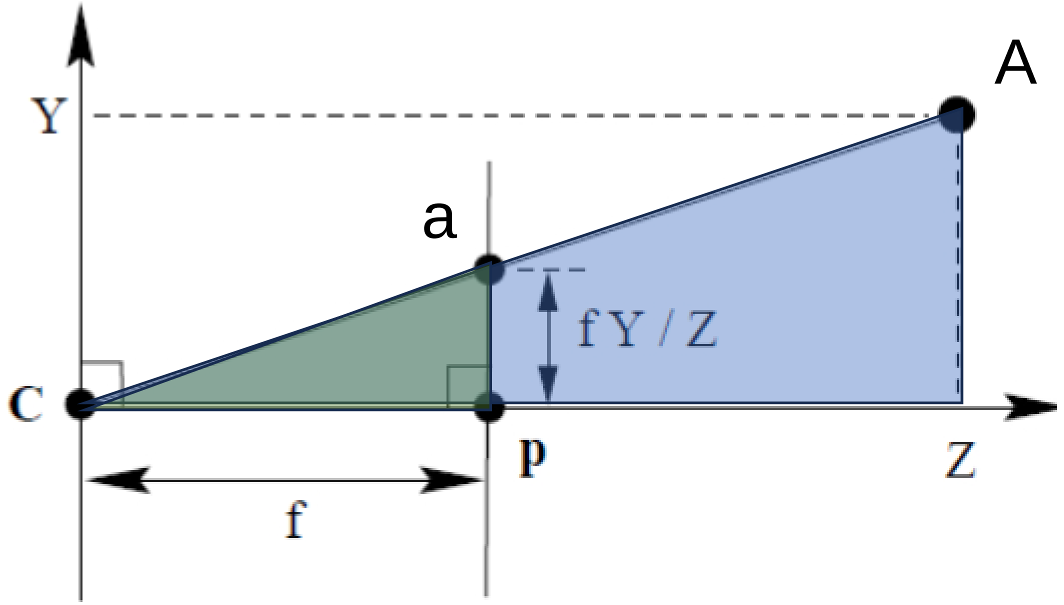


C = camera center
P = principal point (Image center)
f = focal length
A = point in 3D at (X,Y,Z)
a = point on image plane at (x,y)

Camera Model

$$\begin{matrix} \dfrac{fX}{Z} + p_x \\ \dfrac{fY}{Z} + p_y \\ 1 \end{matrix} = \begin{bmatrix} f_x & 0 & p_x & 0 \\ 0 & f_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Intrinsic Matrix: K



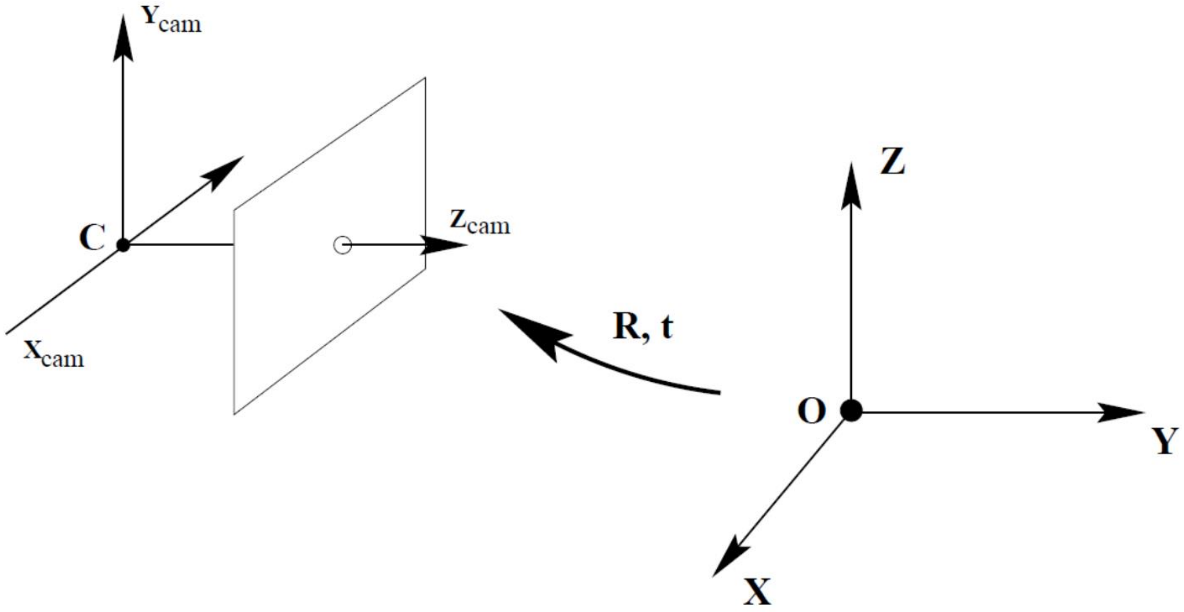
C = camera center
P = principal point (Image center)
f = focal length
A = point in 3D at (X,Y,Z)
a = point on image plane at (x,y)

World Coordinate to Camera Coordinate

$$\mathbf{X}_{cam} = \begin{bmatrix} R & t \end{bmatrix} \mathbf{X}$$

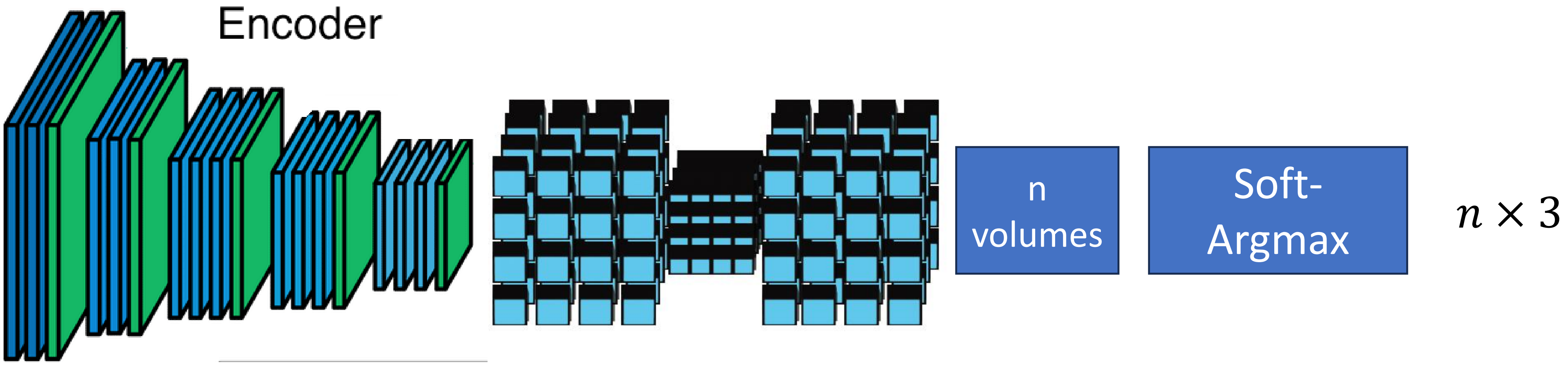
$$\mathbf{x} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \mathbf{X}$$

$$\begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_8 & t_3 \end{bmatrix}$$



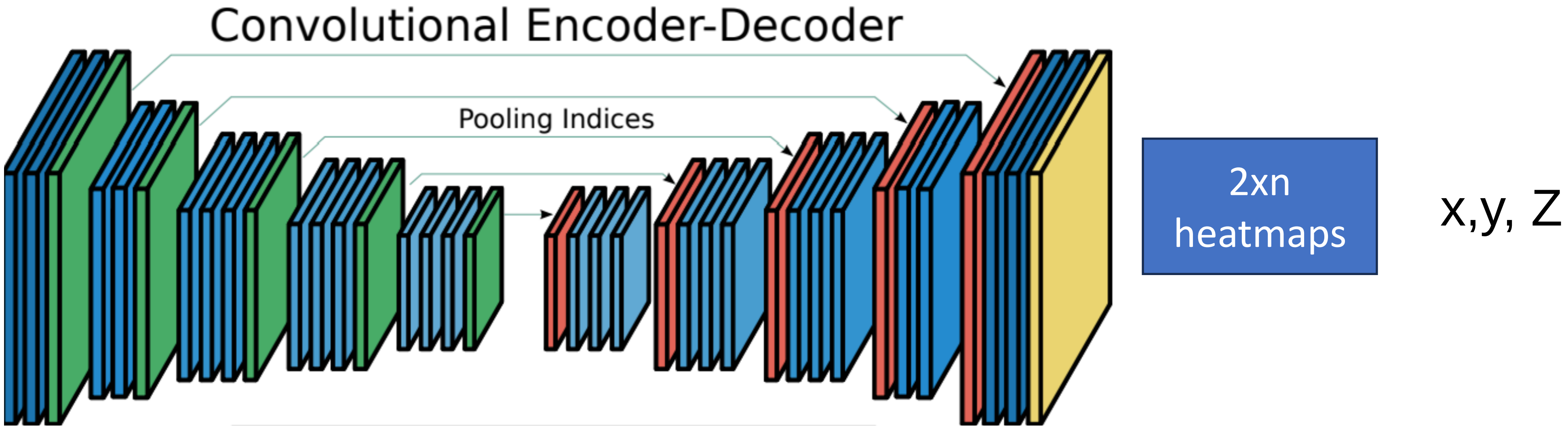
3D Pose Estimation Architectures

- 1. 2D CNN Encoder – 3D CNN Decoder
  - Predicts n volumetric heatmaps where the hottest region in the 3D volume gives the position (usually relative to some joint)
  - Different techniques for converting 2D features to 3D
    - Reshape
    - Project using inverse intrinsic matrix.



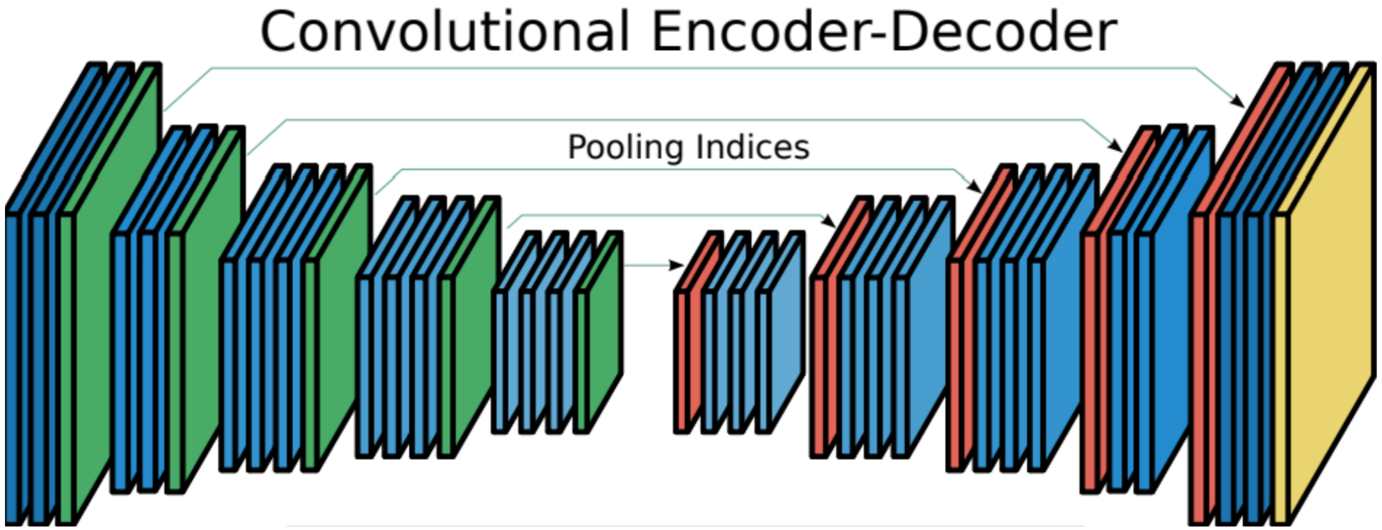
3D Pose Estimation Architectures

- 2. 2.5D Representation
  - Decouple X-Y plane from the Z plane.
  - Represent 3D pose using two heatmaps per keypoint, one for localizing the 2D joint and one for regressing the depth value.
  - The model outputs x,y image coordinates and Z coordinate.



3D Pose Estimation Architectures

- 2. 2.5D Representation
  - Decouple X-Y plane from the Z plane.
  - Represent 3D pose using two heatmaps per keypoint, one for localizing the 2D joint and one for regressing the depth value.
  - The model outputs x,y image coordinates and Z coordinate.



2xn heatmaps

x,y, Z

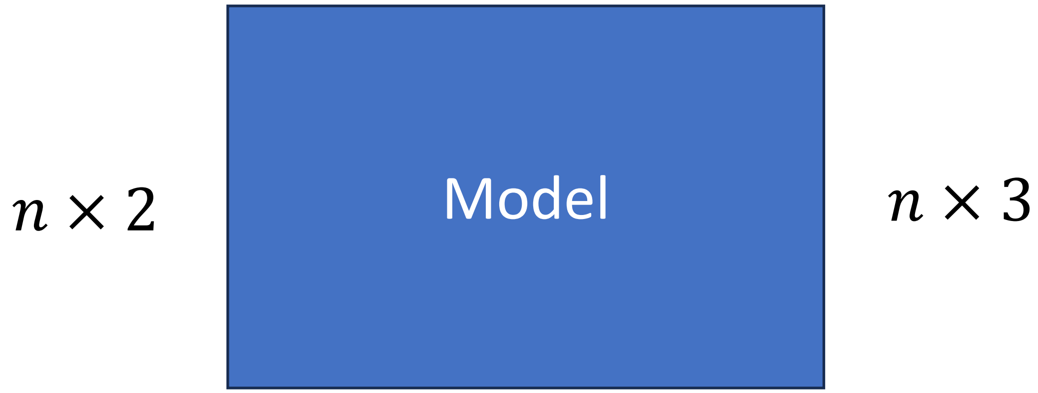
$$\frac{y}{f} = \frac{Y}{Z}$$

$$\therefore Y = \frac{yZ}{f}$$

Note: Ignoring camera center for simplifying notations

3D Pose Estimation Architectures

- 3. 2D to 3D Lifting
  - Regression based approach.
  - Uses off the shelf pose estimation model to get 2D poses and maps them to corresponding 3D poses using sequence of fully connected layers.
  - Recent works use transformer architecture instead.



Multi-View Consistency for Weak Supervision

- Acquiring 3D ground truth data is very expensive.
- We can use multi-view geometry and 2D predictions to add weak supervision.
- Idea: For given multi-view images taken at same time, the 3D pose generated for each image should be the same.
- Alternatively, if a 3D pose from one view is accurate, when projected to the other views, it should align with their predicted 2D poses.

Multi-View Consistency for Weak Supervision

$$P_{v_1 \to v_2} = M_{v_1 \to v_2} \times P_{v_1}$$

Rotate 3D Pose from view 1 to view 2

$$\hat{P}_{v_1 \to v_2} = \frac{P_{v_1 \to v_2}}{d(P^j_{v_1 \to v_2}, P^i_{v_1 \to v_2})} \times d(P^j_{v_2}, P^i_{v_2})$$

Normalize scale using the length of the limb from joint **i** to joint **j**.

$$u_{v_1 \to v_2} = K_{v_2} \times \hat{P}_{v_1 \to v_2}$$

Project poses to target image plane

$$loss = MSE(u_{v_1 \to v_2}, u_{v_2})$$

Compute Loss w.r.t the predicted 2D pose from the target view.

Repeat for Each View

Learning Outcomes

- Pose Estimation : Problem Formulation
- Different sub-research directions
- General architectures for 2D/3D pose estimation
- Naïve Camera Model
- Multi-View Consistency for Weak 3D Supervision.