Image and Video Applications

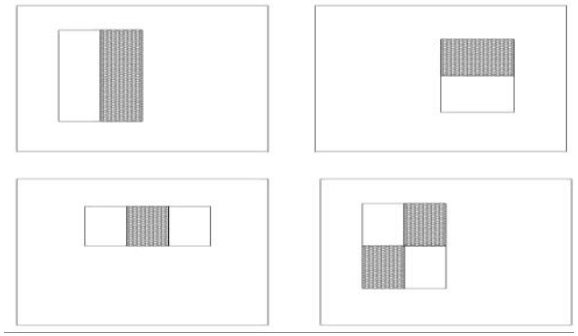
Jenq-Neng Hwang, Professor
Department of Electrical & Computer Engineering
University of Washington, Seattle WA
hwang@uw.edu

Face Detection and Recognition (Identification)

Detection

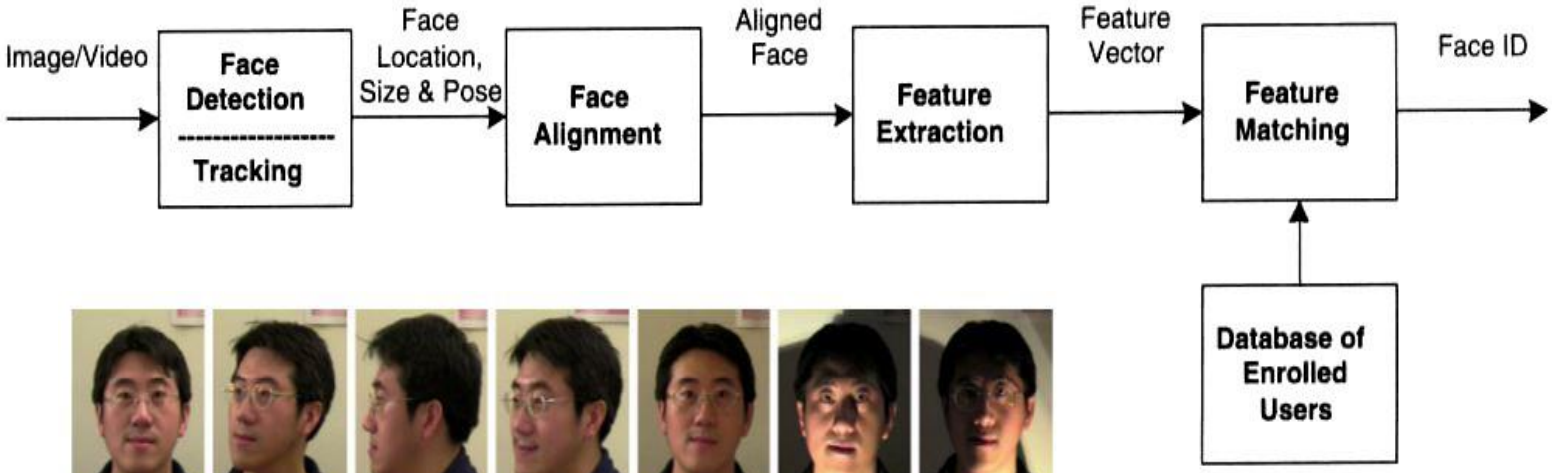
Recognition

"Sally"

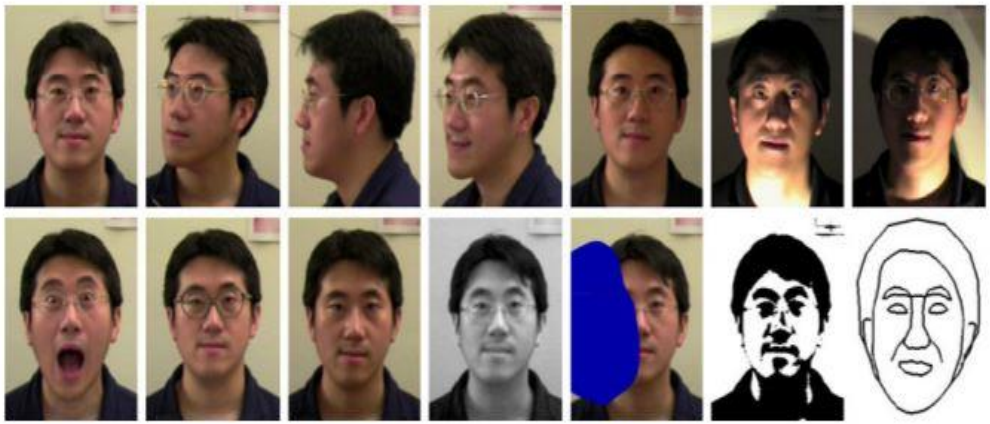


P. Viola and M. Jones, "*Rapid object detection using a boosted cascade of simple features,"* CVPR 2001.

Face Recognition (Identification) & Challenges



challenges



S. Z. Li and A. K. Jain. *Handbook of Face recognition*, 2005

CPSC 4600/5600 Course: Biometrics and Cryptography, The University of Tennessee at Chattanooga

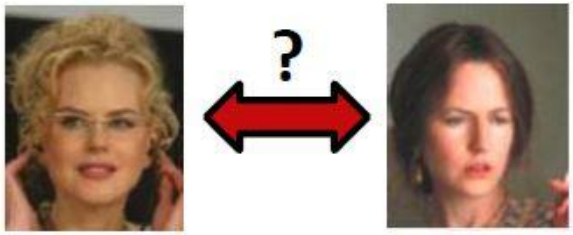
Face Identification (ID) vs. Face Verification

- Face identification: multi-class classification
  - classify an image into one of *N* identity classes

Inter-person variation

- Face verification: binary classification (Y/N)
  - Verify two images belonging to the same person or not

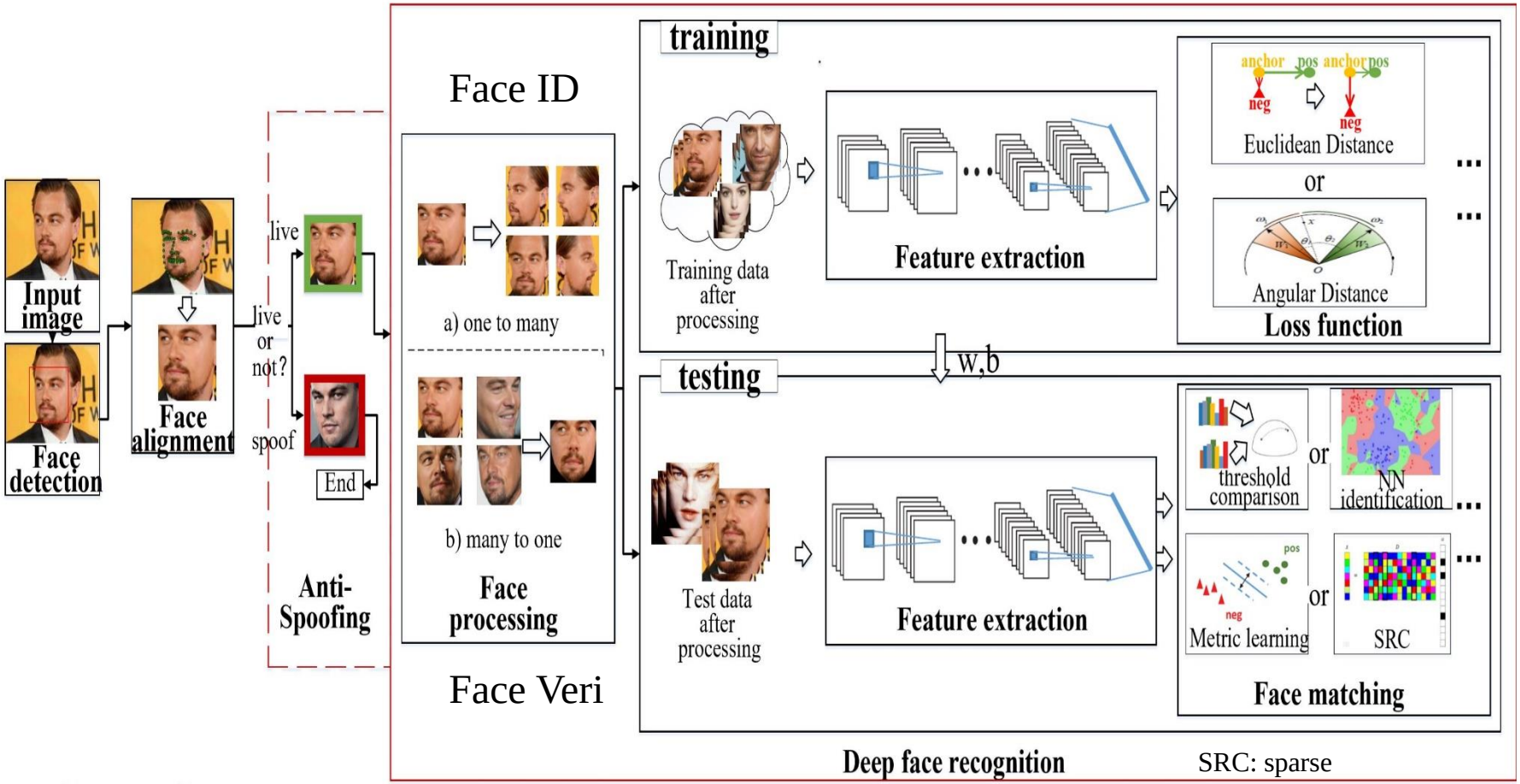
Intra-person variation



Yi Sun, Ding Liang, Xiaogang Wang, Xiaoou Tang, "DeepID3: Face Recognition with Very Deep Neural Networks," ArXiv 2015

http://www.face-rec.org/interesting-papers/

Face ID: Training & Testing



SRC: sparse representation classifier

Labelled Face Datasets

No.	Aim	# Persons	Total # images
1	Labeled Faces In the Wild (LFW), 2007	5,749	13,233
2	Oxford Visual Geometry Group (VGG)	2622	1,635,159
3	Facebook	4030	4.4M
4	Google	8M	200M

- LFW: Standard benchmark for face verification
- Aligned faces are provided, and 1680 people with two or more images.

Labeled Faces In the Wild (LFW) Dataset

- Face Verification: Given a pair of images specify whether they belong to the same person
- 13K images, 5.7K people
- Standard benchmark in the community
- Several test protocols depending upon availability of training data, within and outside the dataset.

Oxford VGG Celebrity

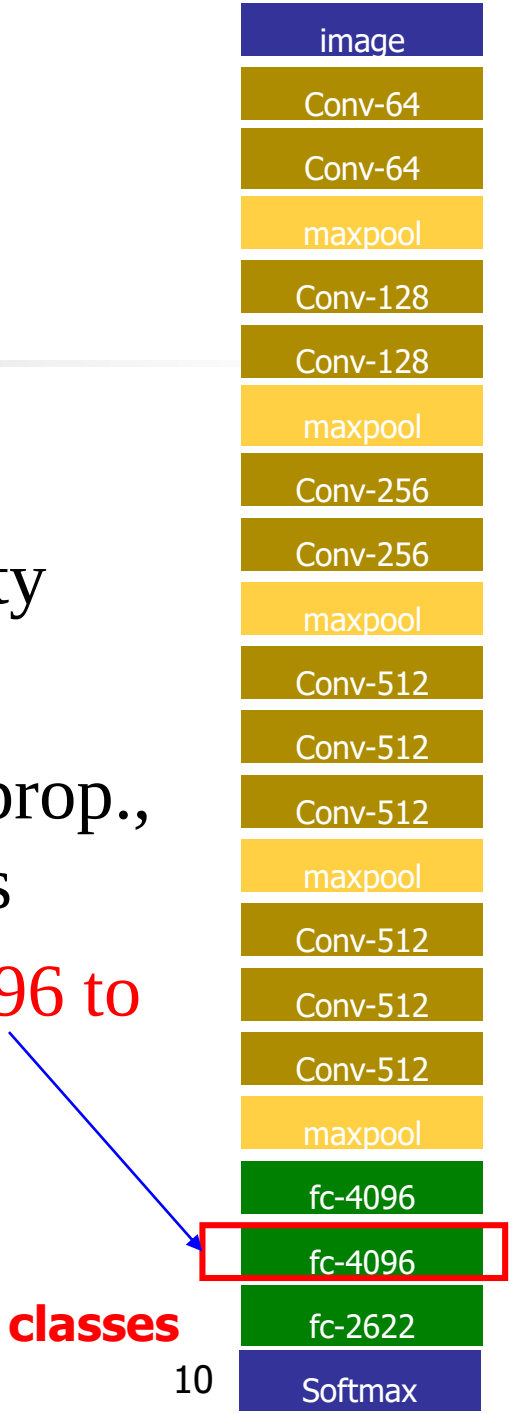
- Collect representative images for each celebrity ~200/identity
- Remove people with low representation on Google.
- Remove overlap with public benchmarks
- 2622 celebrities for the final dataset

NIST IJB-A Dataset

- The IARPA Janus Benchmark-A face challenge (IJB-A) is an open challenge dataset provided by NIST,
- The IJB-A contain 500 subjects with 5,397 images and 2,042 videos split into 20,412 frames,11.4 images and 4.2 videos per subject.

VGG Deep CNN

- 3 x 3 Convolution Kernels (very small)
- Conv. Stride 1 pixel, ReLU non-linearity
- Random Gaussian Initialization
- Stochastic Gradient Descent with backprop., accumulator descent for 256 batch sizes
- Learning a linear projection *W* from 4096 to 1024 dimensions (metric learning, 10 epochs, with the backbone fixed)



2622 classes

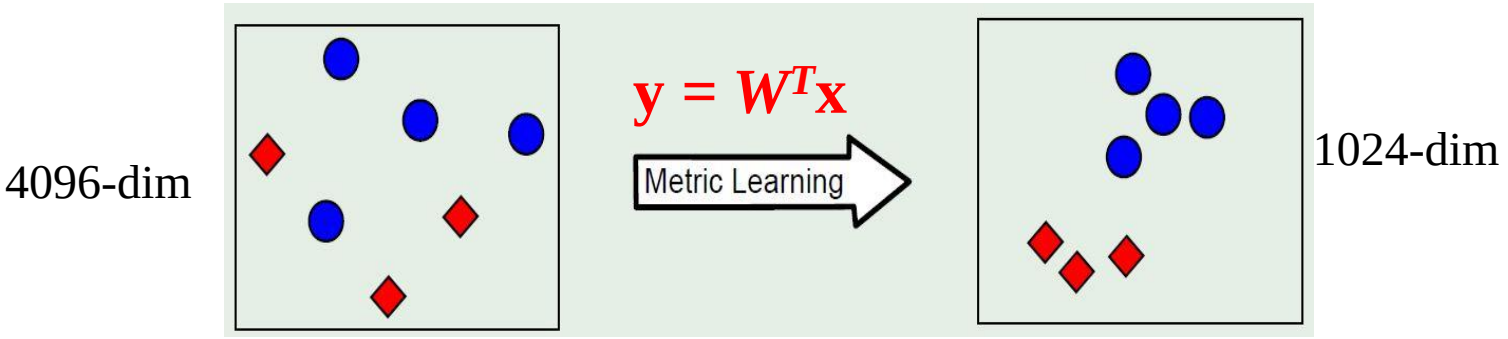
Omkar M Parkhi, et al. "Deep Face Recognition," British Machine Vision Association, 2015

Learning Task Specific Embedding Features

- Learning embedding by minimizing triplet loss

$$\sum_{(a,p,n)\in T} \max\{0, \alpha - \|\mathbf{x}_a - \mathbf{x}_n\|_2^2 + \|\mathbf{x}_a - \mathbf{x}_p\|_2^2\}$$

- Learning a linear projection $\boldsymbol{W}$ from 4096 to 1024 dimensions with SGD, network is frozen, 10 epochs, fixed learning rate 0.25



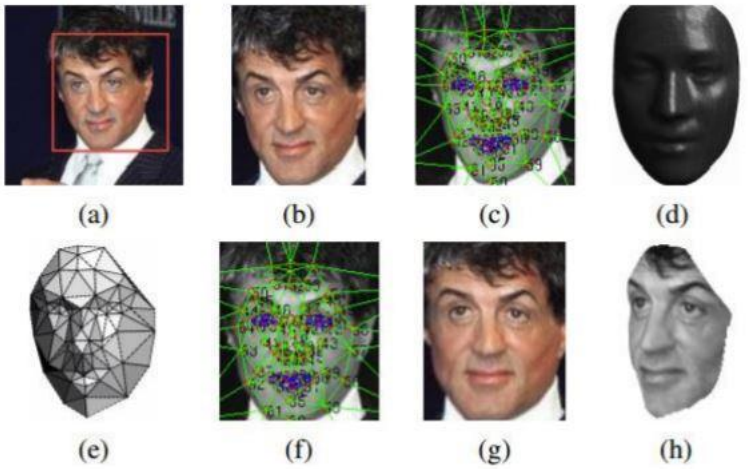
Verification & Identification of VGGFace on IJB-A Data

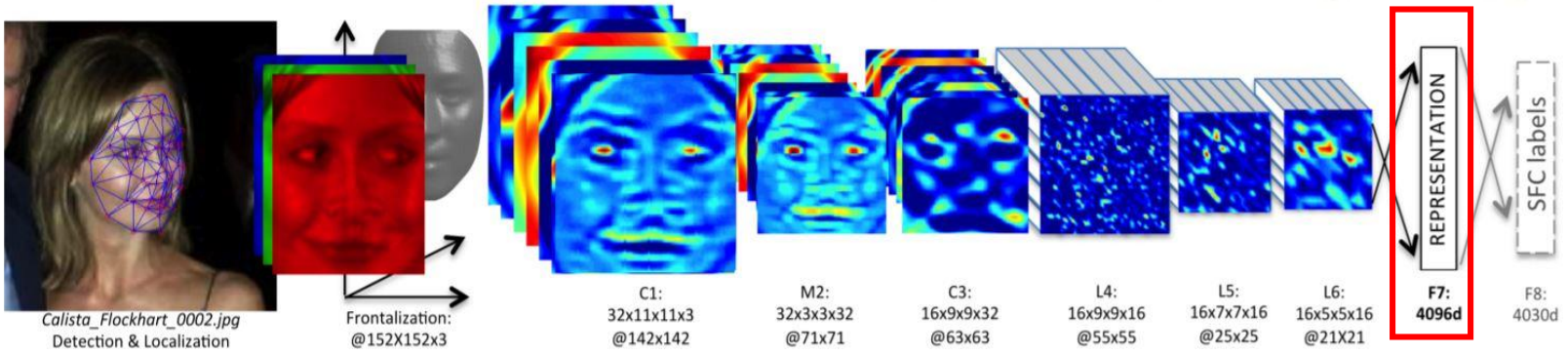
Method	IJB-A Verification (TAR@FAR)			IJB-A Identification			
	0.001	0.01	0.1	FPIR=0.01	FPIR=0.1	Rank=1	Rank=10
TDFF [146]	0.979±0.004	0.991±0.002	0.996±0.001	0.946±0.047	0.987±0.003	0.992±0.001	0.998±0.001
L2-softmax [109]	0.943±0.005	0.970±0.004	0.984±0.002	0.915±0.041	0.956±0.006	0.973±0.005	0.988±0.003
DA-GAN [56]	0.930±0.005	0.976±0.007	0.991±0.003	0.890±0.039	0.949±0.009	0.971±0.007	0.989±0.003
VGGface2 [39]	0.921±0.014	0.968±0.006	0.990±0.002	0.883±0.038	0.946±0.004	0.982±0.004	0.994±0.001
TDFF [146]	0.919±0.006	0.961±0.007	0.988±0.003	0.878±0.035	0.941±0.010	0.964±0.006	0.992±0.002
NAN [83]	0.881±0.011	0.941±0.008	0.979±0.004	0.817±0.041	0.917±0.009	0.958±0.005	0.986±0.003
All-In-One Face [100]	0.823±0.020	0.922±0.010	0.976±0.004	0.792±0.020	0.887±0.014	0.947±0.008	0.988±0.003
Template Adaptation [121]	0.836±0.027	0.939±0.013	0.979±0.004	0.774±0.049	0.882±0.016	0.928±0.010	0.986±0.003
TPE [81]	0.813±0.020	0.900±0.010	0.964±0.005	0.753±0.030	0.863±0.014	0.932±0.010	0.977±0.005

FAR (False Accept Rate): is the probability that the system incorrectly accepts a non-authorized person.
FRR (False Reject Rate): is the probability that the system incorrectly rejects an authorized person.
TAR (True accept rate): is the probability that the system correctly accepts an authorized person.
FPIR (N,T,L)=(Num. searches where one or more enrolled candidates are returned at or above threshold, T)/(Num. searches attempted)

FaceBook DeepFace

- Facial alignment with 3D modeling
- Yaniv Taigman, et. al, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," IEEE CVPR 2014



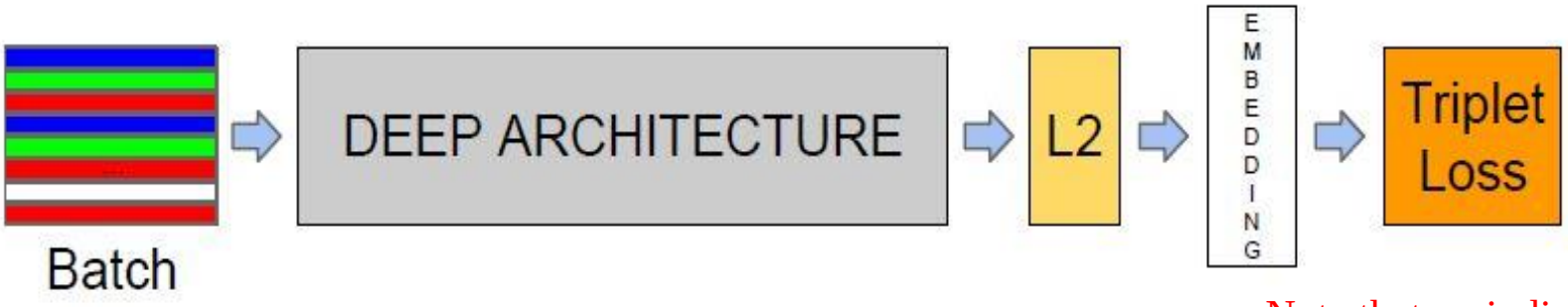


4030 classes

Figure 2. **Outline of the *DeepFace* architecture**. A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than 120 million parameters, where more than 95% come from the local and fully connected layers.

Google FaceNet

- Inception: use of multi-scale convolutional and pooling layers in parallel and concatenate into a single feature extraction
- In training data 40 faces are selected per identity per mini batch. Additionally, randomly sampled negative faces are added to each mini-batch. No cross-entropy loss needed



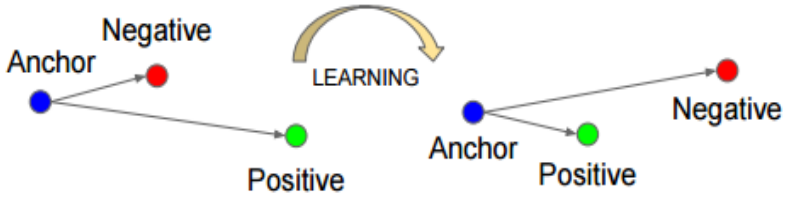
Florian Schroff, et al., "FaceNet: A Unified Embedding for Face Recognition and Clustering," arXiv, June 2015

Note that no individual ID (face recognition) is performed

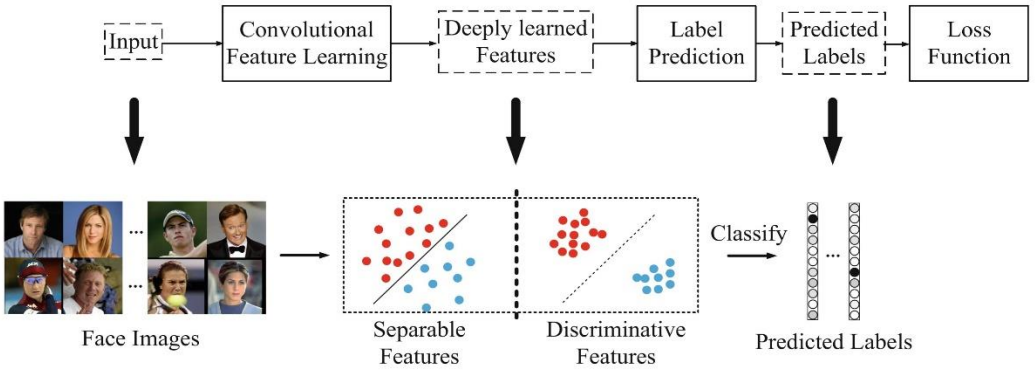
Google Triplet Loss Learning

- Training: minimizing triplet loss

$$\sum_{(a,p,n)\in T} \max\{0, \alpha - \|\mathbf{x}_a - \mathbf{x}_n\|_2^2 + \|\mathbf{x}_a - \mathbf{x}_p\|_2^2\}$$



- "a" an anchor face image, "p" as a positive $p \neq a$ and negative "n" examples of the anchor's identity.



Verification Performance

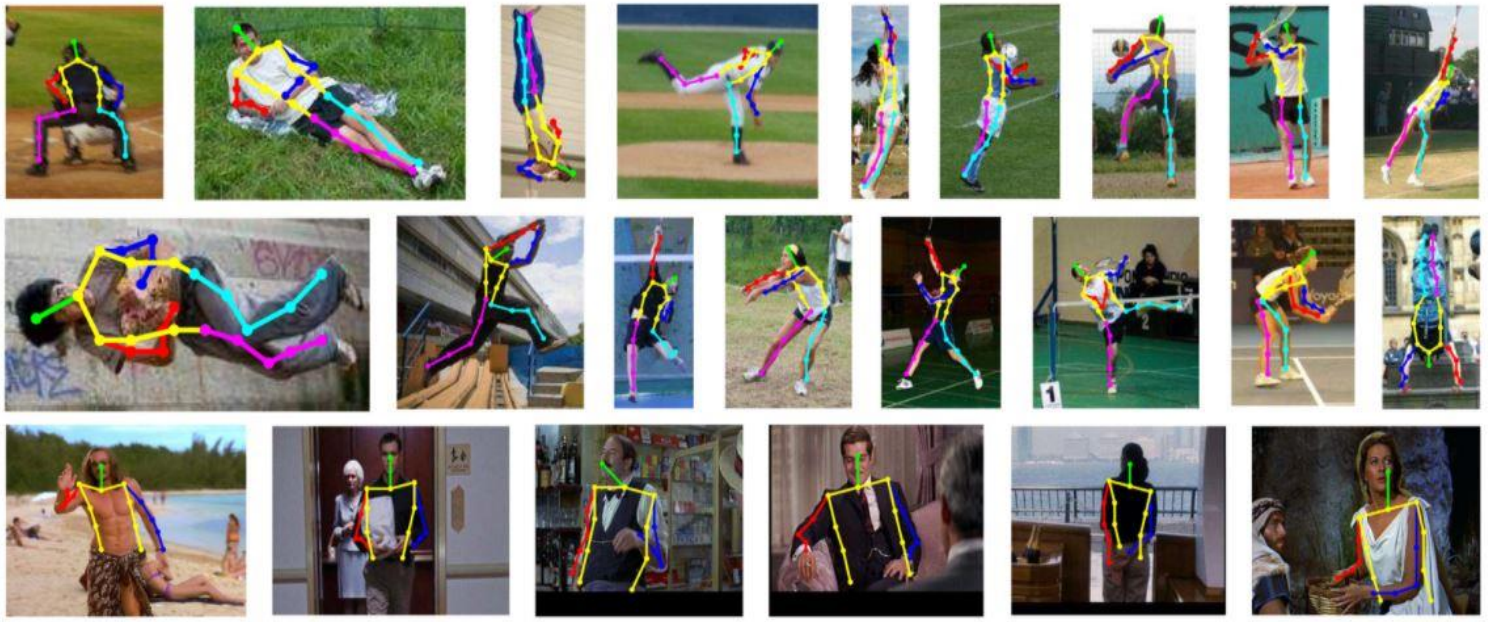
No.	Method	# Training Images	# Networks	Accuracy (%)
1	Fisher Vector Faces	-	-	93.10
2	Facebook DeepFace	7 M	3	97.35
3	DeepFace Fusion	500 M	5	98.37
4	CUHK DeepID-2,3	Full	200	99.47
5	Google FaceNet	200 M	1	98.87
6	FaceNet+ Alignment	200 M	1	99.63
7	VGG Face	2.6 M	1	98.95

Facebook: DeepFace, Google: FaceNet, Oxford: VGG Face

Human Pose Estimation

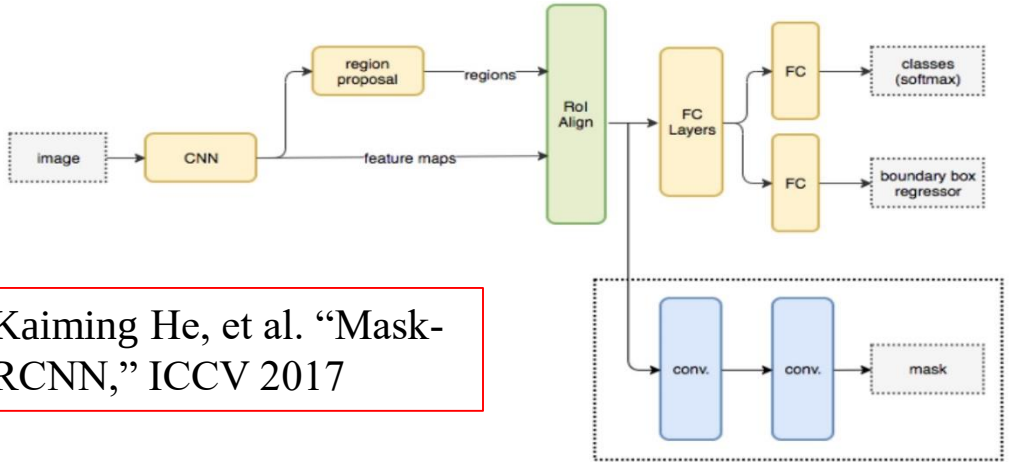
Human Pose Estimation

- Human Pose Estimation (HPE): the problem of 2D/3D localization of human joints (also known as keypoints - elbows, wrists, etc) in images or videos. It is also defined as the search for a specific pose in space of all articulated poses.

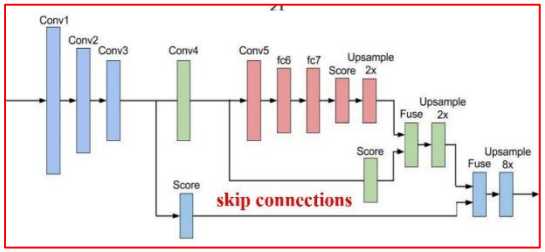


Mask R-CNN for HPE

- Suppose a semantically segmented "human" object in mask R-CNN can belong to one among K classes. The segmentation branch outputs **K** binary masks of size **m x m**, where each binary mask represents all object's parts belonging to that class alone.
- Regarded as the **top-down** approach, the person detection stage is performed in parallel to the part detection stage, i.e., keypoint detection and person detection are independent of each other.



Kaiming He, et al. "Mask-RCNN," ICCV 2017



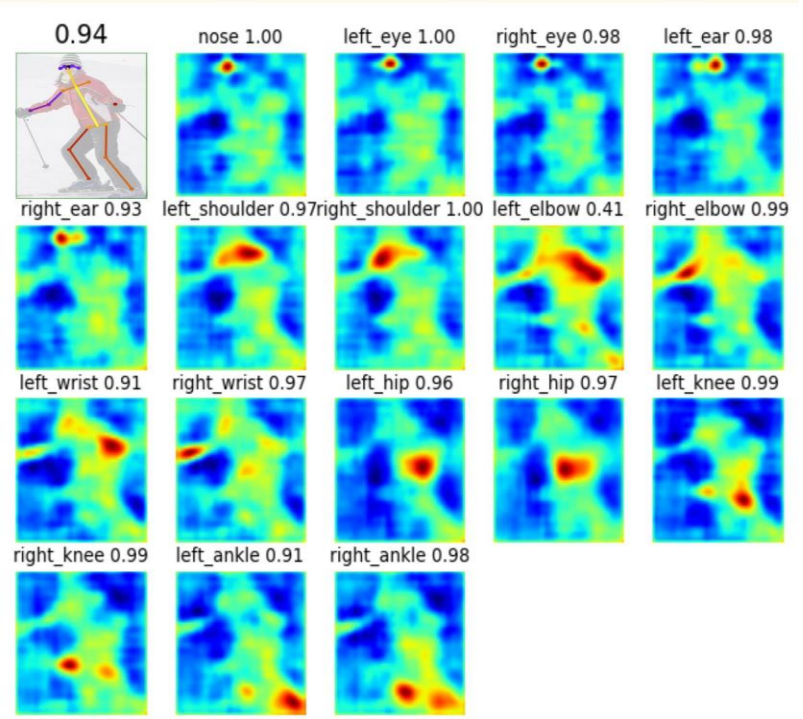
FCN

Human Part Segmentation

- We can extract key points belonging to every person in the image by modeling each type of keypoint as a distinct mask class and treating this like a segmentation.

Mask R-CNN

- Keypoint = 1-hot mask
- Human pose = 17 keypoints
- Represent pose as 17 masks



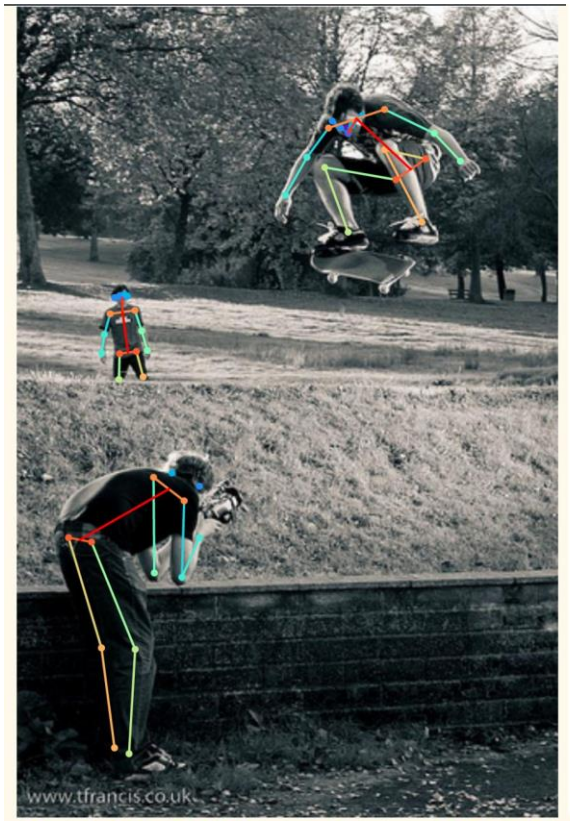
COCO Keypoint DataSet & Performance Evaluation

- Dataset. The COCO dataset contains over 200,000 images and 250,000 person instances labeled with 17 keypoints.
- Object Keypoint Similarity (OKS): instead IOU, nor CE

$$\frac{\sum_i \exp(-d_i^2/2s^2k_i^2)\delta(v_i>0)}{\sum_i \delta(v_i>0)}$$

  - d_i is the Euclidean distance between the detected keypoint and the corresponding ground truth,
  - v_i is the visibility flag of the ground truth,
  - s is the object scale, and
  - k_i is a per-keypoint constant that controls falloff.
  - AP50 (AP at OKS = 0.50)

HPE of Mask R-CNN



Mask R-CNN results on COCO

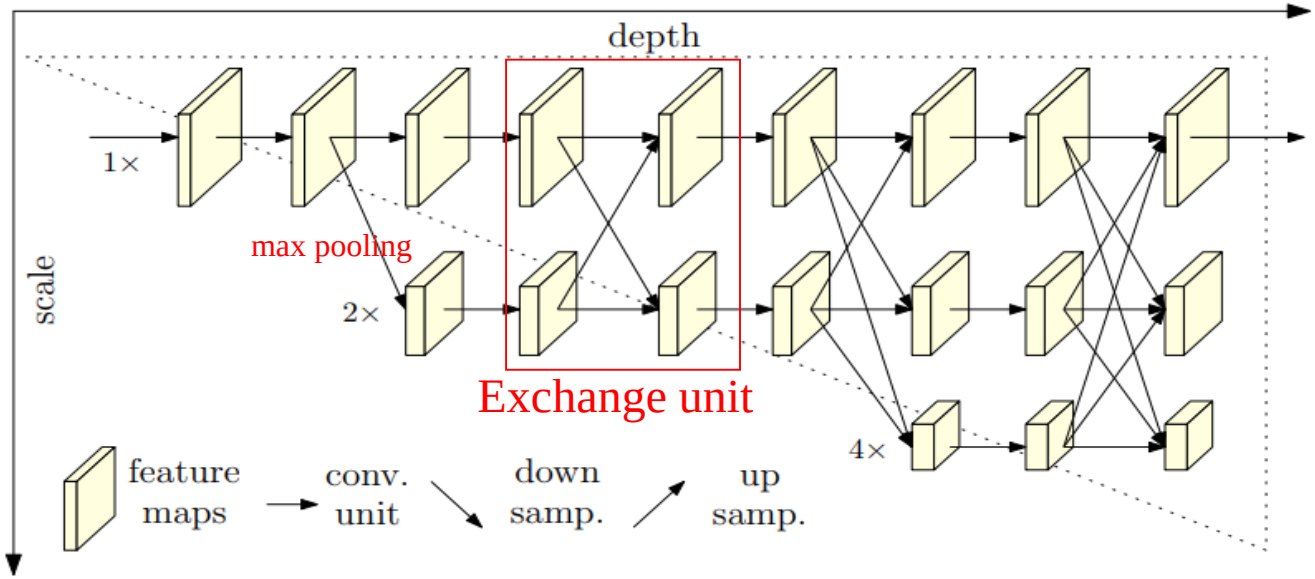
Mask R-CNN results on COCO

Performance

	AP^{bb}_{person}	AP^{mask}_{person}	AP^{kp}
Faster R-CNN	52.5	-	-
Mask R-CNN, mask-only	**53.6**	**45.8**	-
Mask R-CNN, keypoint-only	50.7	-	64.2
Mask R-CNN, keypoint & mask	52.0	45.1	**64.7**

Deep High-Resolution (HR) Network for HPE

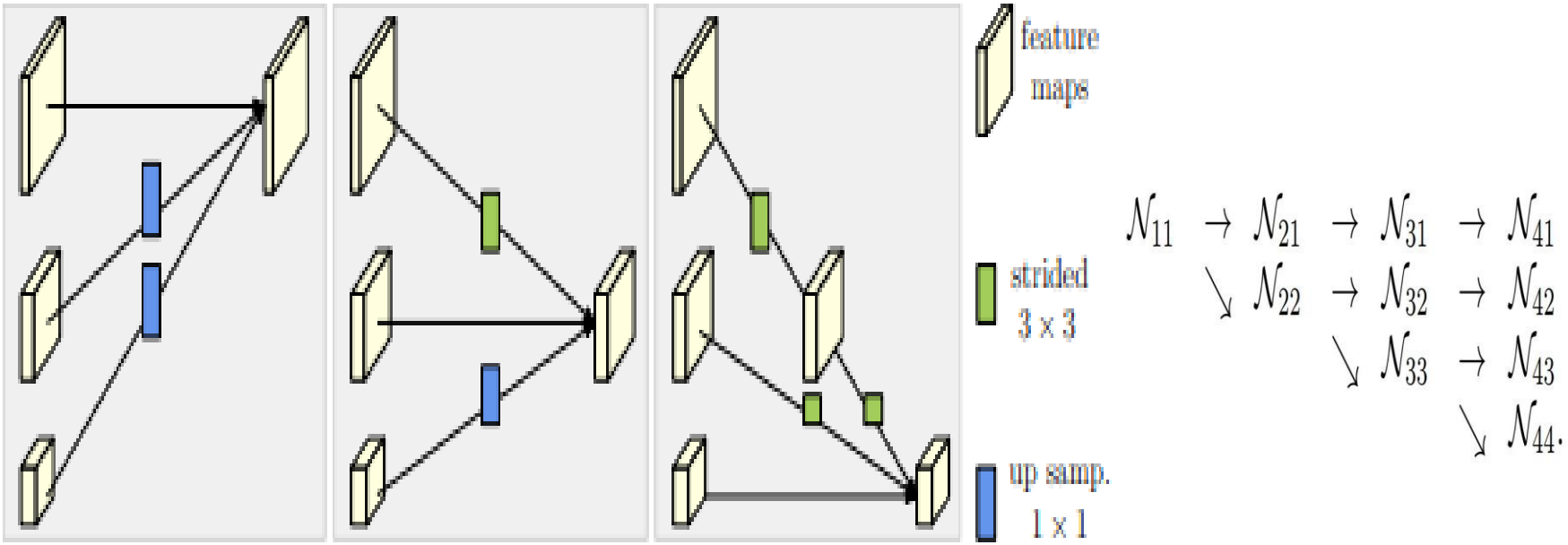
- Parallel high-to-low resolution subnetworks with repeated information exchange across multi-resolution subnetworks (multi-scale fusion).
- The horizontal and vertical directions correspond to the depth of the network and the scale of the feature maps, respectively



Ke Sun et al., "Deep High-Resolution Representation Learning for Human Pose Estimation," CVPR 2019

Deep High-Resolution (HR) Network for HPE

- Combinations of max pooling, strided convolution and up-sampling for a new stage.



Ke Sun et al., "Deep High-Resolution Representation Learning for Human Pose Estimation," CVPR 2019

Top-Down HRNet Performance

Figure 4. Qualitative results of some example images in the MPII (top) and COCO (bottom) datasets: containing viewpoint and appearance change, occlusion, multiple persons, and common imaging artifacts.

Top-Down HPE: Multi-person's keypoints are estimated through using the deep features similar to two-stage Mask R-CNN.

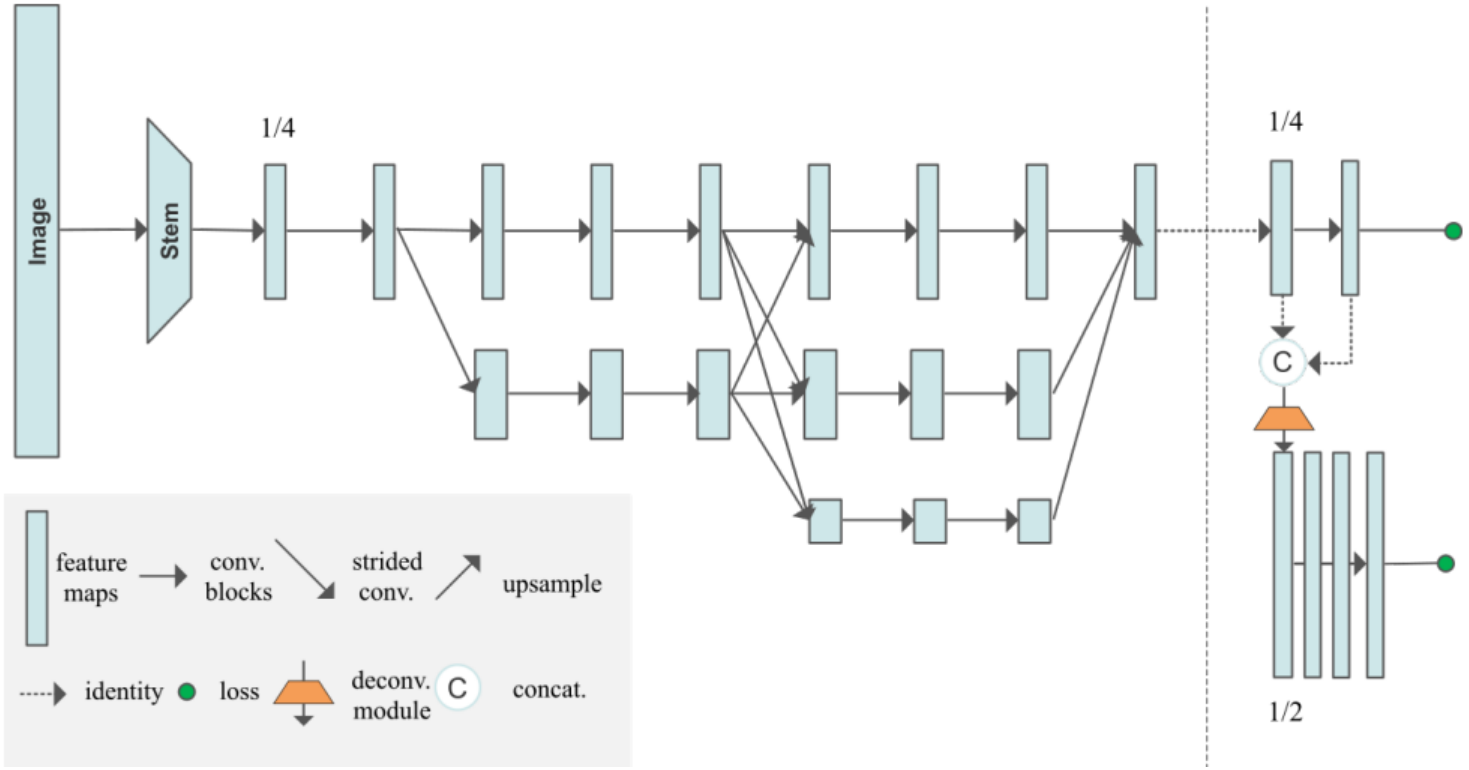
Top-Down HRNet Performance

Table 2. Comparisons on the COCO test-dev set. #Params and FLOPs are calculated for the pose estimation network, and those for human detection and keypoint grouping are not included.

Method	Backbone	Input size	#Params	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Bottom-up: keypoint detection and grouping										
OpenPose [6]	–	–	–	–	61.8	84.9	67.5	57.1	68.2	66.5
Associative Embedding [39]	–	–	–	–	65.5	86.8	72.3	60.6	72.6	70.2
PersonLab [46]	–	–	–	–	68.7	89.0	75.4	64.1	75.5	75.4
MultiPoseNet [33]	–	–	–	–	69.6	86.3	76.6	65.0	76.3	73.5
Top-down: human detection and single-person keypoint detection										
Mask-RCNN [21]	ResNet-50-FPN	–	–	–	63.1	87.3	68.7	57.8	71.4	–
G-RMI [47]	ResNet-101	353 × 257	42.6M	57.0	64.9	85.5	71.3	62.3	70.0	69.7
Integral Pose Regression [60]	ResNet-101	256 × 256	45.0M	11.0	67.8	88.2	74.8	63.9	74.0	–
G-RMI + extra data [47]	ResNet-101	353 × 257	42.6M	57.0	68.5	87.1	75.5	65.8	73.3	73.3
CPN [11]	ResNet-Inception	384 × 288	–	–	72.1	91.4	80.0	68.7	77.2	78.5
RMPE [17]	PyraNet [77]	320 × 256	28.1M	26.7	72.3	89.2	79.1	68.0	78.6	–
CFN [25]	–	–	–	–	72.6	86.1	69.7	78.3	64.1	–
CPN (ensemble) [11]	ResNet-Inception	384 × 288	–	–	73.0	91.7	80.9	69.5	78.1	79.0
SimpleBaseline [72]	ResNet-152	384 × 288	68.6M	35.6	73.7	91.9	81.1	70.3	80.0	79.0
HRNet-W32	HRNet-W32	384 × 288	28.5M	16.0	74.9	92.5	82.8	71.3	80.9	80.1
HRNet-W48	HRNet-W48	384 × 288	63.6M	32.9	**75.5**	**92.5**	**83.3**	**71.9**	**81.5**	**80.5**
HRNet-W48 + extra data	HRNet-W48	384 × 288	63.6M	32.9	**77.0**	**92.7**	**84.5**	**73.4**	**83.1**	**82.0**

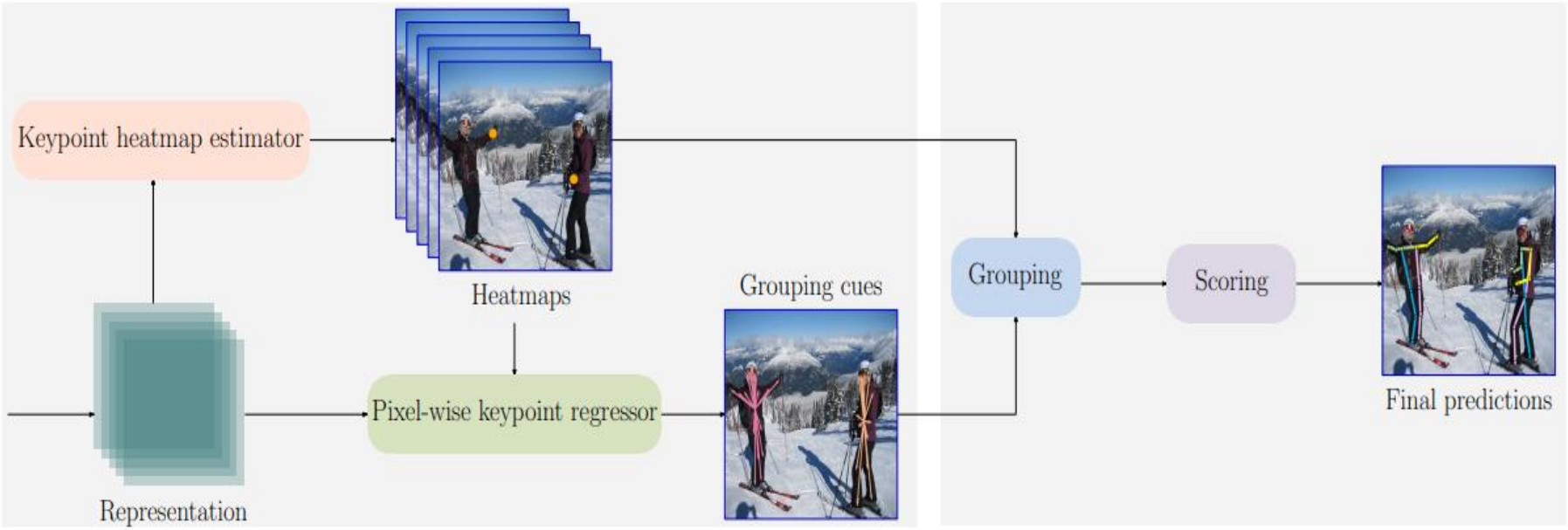
HRNet-W32 and HRNet-W48, where 32 and 48 represent the widths (C) of the high-resolution subnetworks in last three stages, respectively

HigherHRNet: A Bottom-Up HPE



The network uses HRNet as backbone, followed by one or more deconvolution modules to generate multi-resolution and high-resolution heatmaps. Multi-resolution supervision is used for training

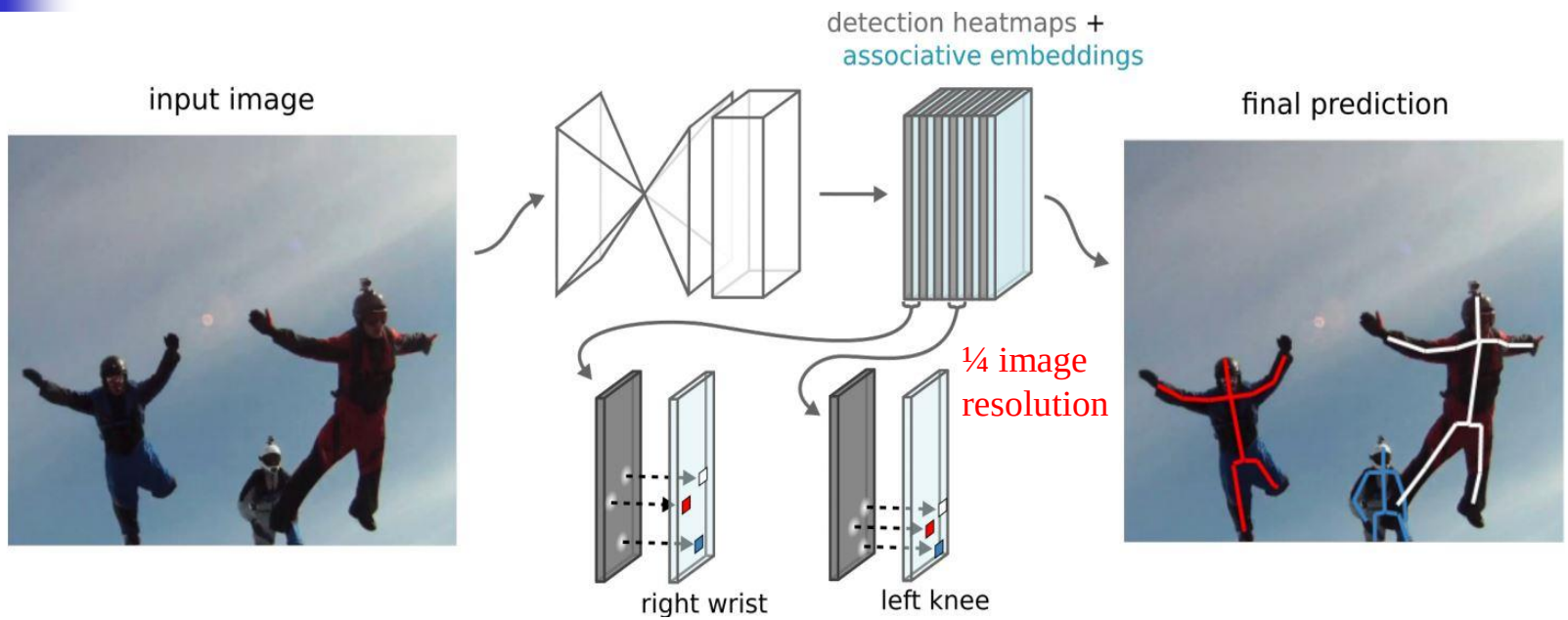
HigherHRNet: A Bottom-Up HPE



Associative embedding is used for keypoint grouping. The grouping process clusters identity-free keypoints into individuals by grouping keypoints whose tags have small L2 distance

A. Newell, et al., "Associative embedding: End-to-end learning for joint detection and grouping," NIPS 2017

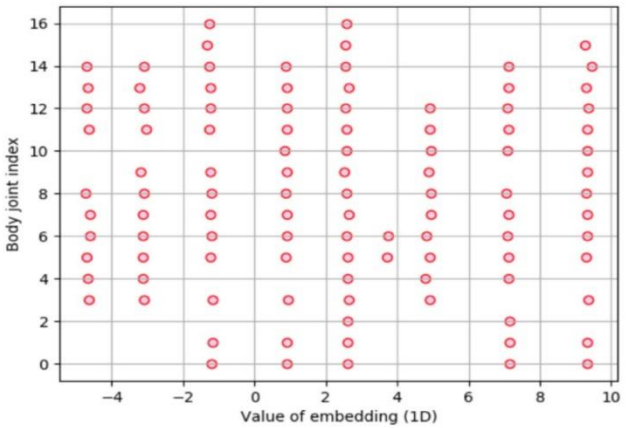
HigherHRNet: A Bottom-Up HPE



The network produces a tag at each pixel location for each joint. In other words, each joint heatmap has a corresponding "tag" heatmap. To parse detections into individual people, we use non-maximum suppression to get the peak detections for each joint and retrieve their corresponding tags at the same pixel location

A. Newell, et al., "Associative embedding: End-to-end learning for joint detection and grouping," NIPS 2017

Associative Embeddings (Tagmaps) of HigherHRNet



- Assuming all *K* joints are annotated, the reference embedding for the *n-th* person and grouping loss

$$L_g(h,T) = \frac{1}{N}\sum_n\sum_k \left(\bar{h}_n - h_k(x_{nk},)\right)^2 + \frac{1}{N^2}\sum_n\sum_{n'} \exp\{-\frac{1}{2\sigma^2}\left(\bar{h}_n - \bar{h}_{n'}\right)^2\}$$

Grouping Loss

Same person, similar embeddings among keypoints

different person, larger embeddings distance among keypoints

Reference Embedding

$$\bar{h}_n = \frac{1}{K}\sum_k h_k(x_{nk})$$

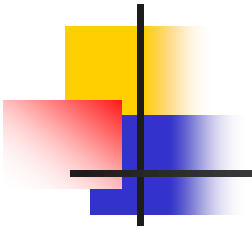
HigherHRNet: A Bottom-Up HPE

Method	Backbone	Input size	#Params	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
w/o multi-scale test									
OpenPose [3]†	-	-	-	-	61.8	84.9	67.5	57.1	68.2
Hourglass [30]	Hourglass	512	277.8M	206.9	56.6	81.8	61.8	49.8	67.0
PersonLab [33]	ResNet-152	1401	68.7M	405.5	66.5	88.0	72.6	62.4	72.3
PifPaf [22]	-	-	-	-	66.7	-	-	62.4	72.9
Bottom-up HRNet‡	HRNet-W32	512	28.5M	38.9	64.1	86.3	70.4	57.4	73.9
HigherHRNet (Ours)	HRNet-W32	512	28.6M	47.9	66.4	87.5	72.8	61.2	74.2
HigherHRNet (Ours)	HRNet-W48	640	63.8M	154.3	**68.4**	**88.2**	**75.1**	**64.4**	**74.2**
w/ multi-scale test									
Hourglass [30]	Hourglass	512	277.8M	206.9	63.0	85.7	68.9	58.0	70.4
Hourglass [30]†	Hourglass	512	277.8M	206.9	65.5	86.8	72.3	60.6	72.6
PersonLab [33]	ResNet-152	1401	68.7M	405.5	68.7	89.0	75.4	64.1	75.5
HigherHRNet (Ours)	HRNet-W48	640	63.8M	154.3	**70.5**	**89.3**	**77.2**	**66.6**	**75.8**

† Indicates using refinement.

‡ Our implementation, not reported in [38, 40]

Table 1. Comparisons with bottom-up methods on the **COCO2017 test-dev** set. All GFLOPs are calculated at single-scale. For PersonLab [33], we only calculate its backbone's #Params and GFLOPs. Top: w/o multi-scale test. Bottom: w/ multi-scale test. *It is worth noting that our results are achieved without refinement.*



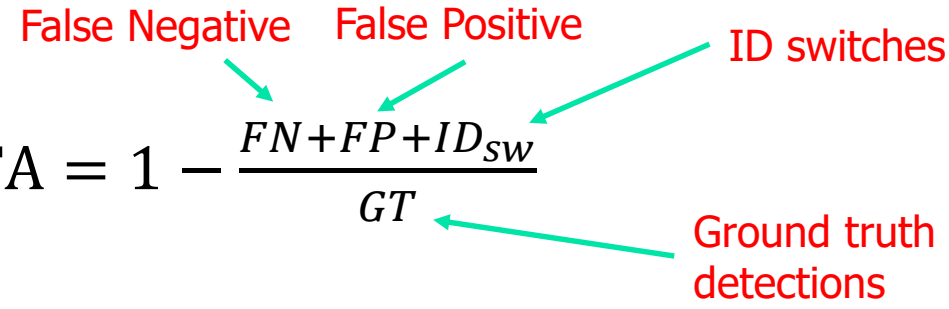
Multiple Objects Tracking (MOT) in Videos

Video Object Tracking

- Single target/visual tracking in a video
- Multiple object tracking (MOT)

Tracking Metrics: MOTA

- MOTA: multiple object tracking accuracy



- Characteristics of MOTA
  - FN and FP are based on detection results, not tracking, more sensitive to detection threshold.
  - Compared with FN and FP, ID_{sw} sometimes has a smaller portion of influence, which cannot show tracker's performance.

Tracking Metrics: IDF_1

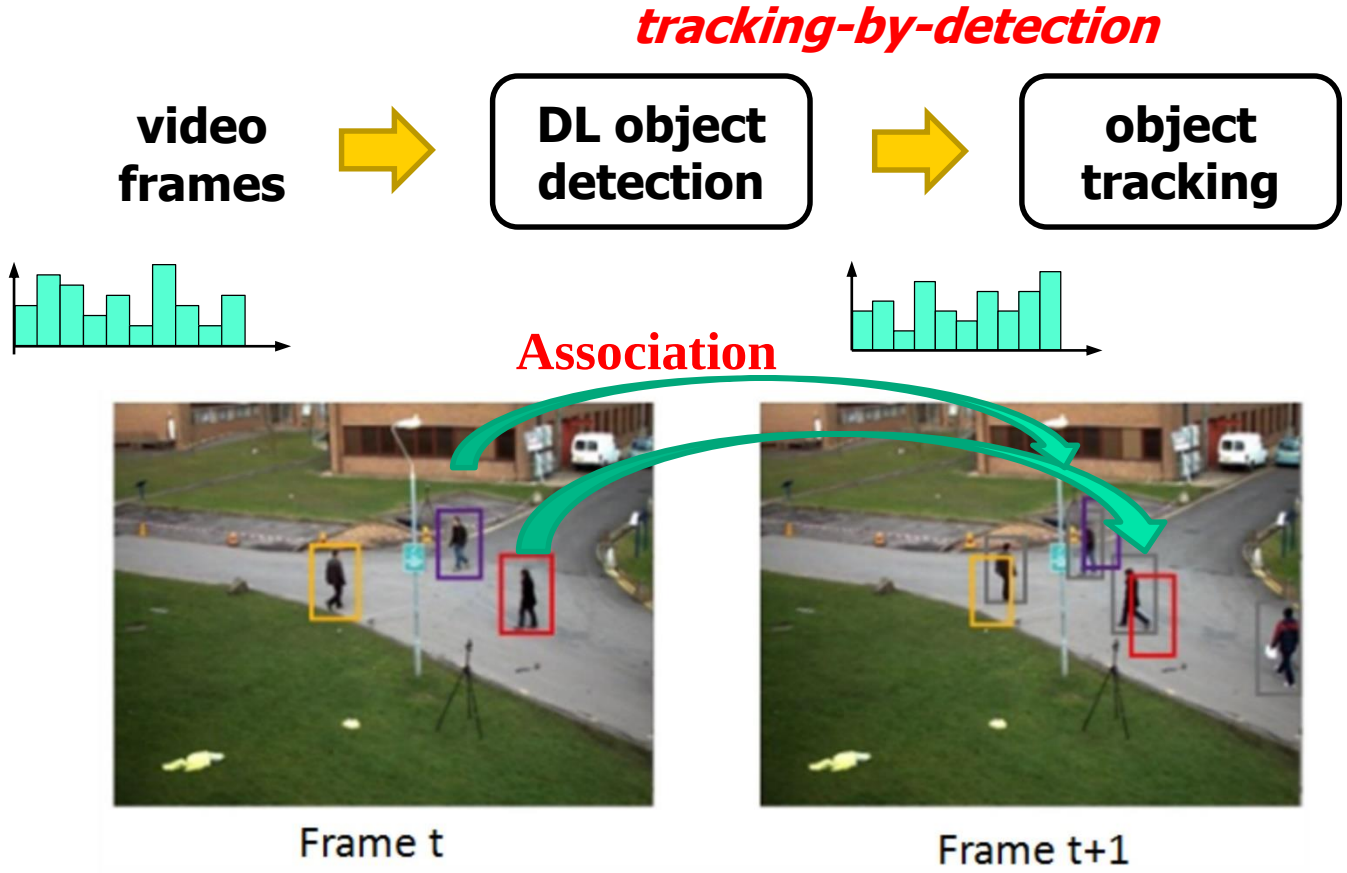
- A measure between ground truth trajectories and computed trajectories. (the comparison is based on trajectory, not detections)
- $\text{IDFN} = \sum_\tau \sum_{t \in \mathcal{T}_\tau} m(\tau, \gamma_m(\tau), t, \Delta)$ How many GT bboxes are missed
- $\text{IDFP} = \sum_\gamma \sum_{t \in \mathcal{T}_\gamma} m(\tau_m(\gamma), \gamma, t, \Delta)$ How many created bboxes are incorrect
- $\text{IDTP} = \sum_\tau \text{len}(\tau) - \text{IDFN} = \sum_\gamma \text{len}(\gamma) - \text{IDFP}$

τ: ground truth trajectory.
$\gamma_m(\tau)$: computed trajectory that best matches τ.
γ: computed trajectory.
$\tau_m(\gamma)$: ground truth trajectory that best matches γ.
t: frame index.
Δ: IOU threshold that judges whether computed bbox matches ground truth bbox. Usually set Δ=0.5.
$m(\cdot)$: if there is a mis-match at t, $m(\cdot) = 1$. Otherwise, $m(\cdot) = 0$.

$$\text{IDF}_1 = \frac{2\text{IDTP}}{2\text{IDTP} + \text{IDFP} + \text{IDFN}}$$

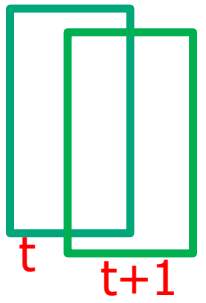
MOT under Deep Learning Detections

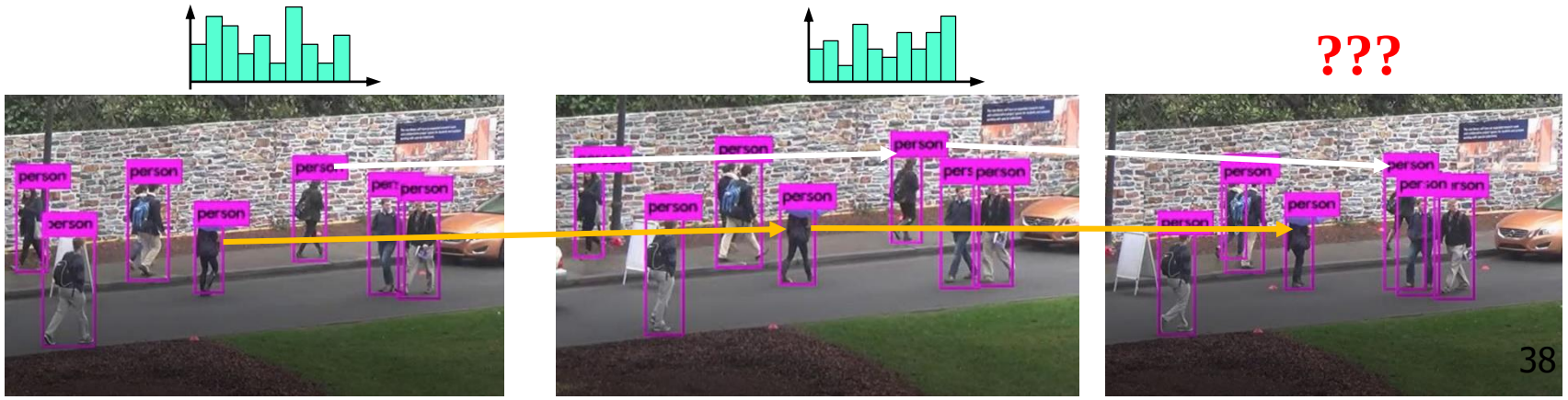
- Features: color, texture, gradient, CNN, etc



Tracking by Detection

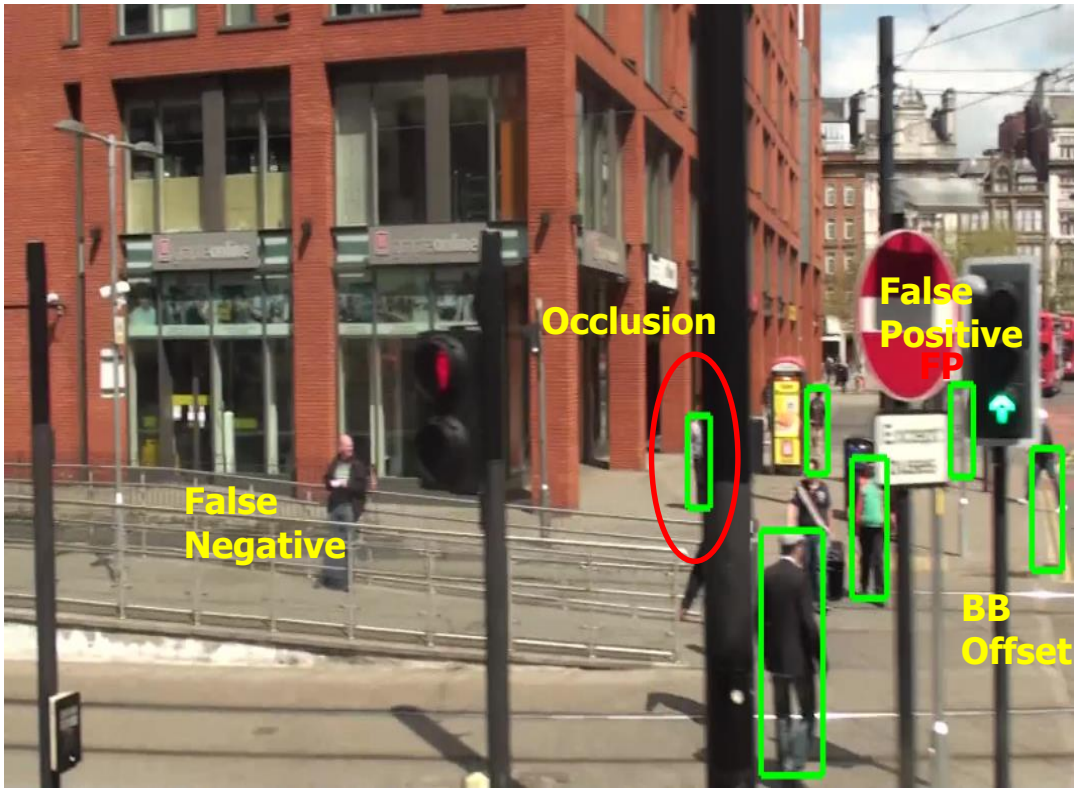
- Detected object association
  - Appearance feature similarity.
  - Intersection over union (IOU) of bounding boxes w/wo motion compensation)





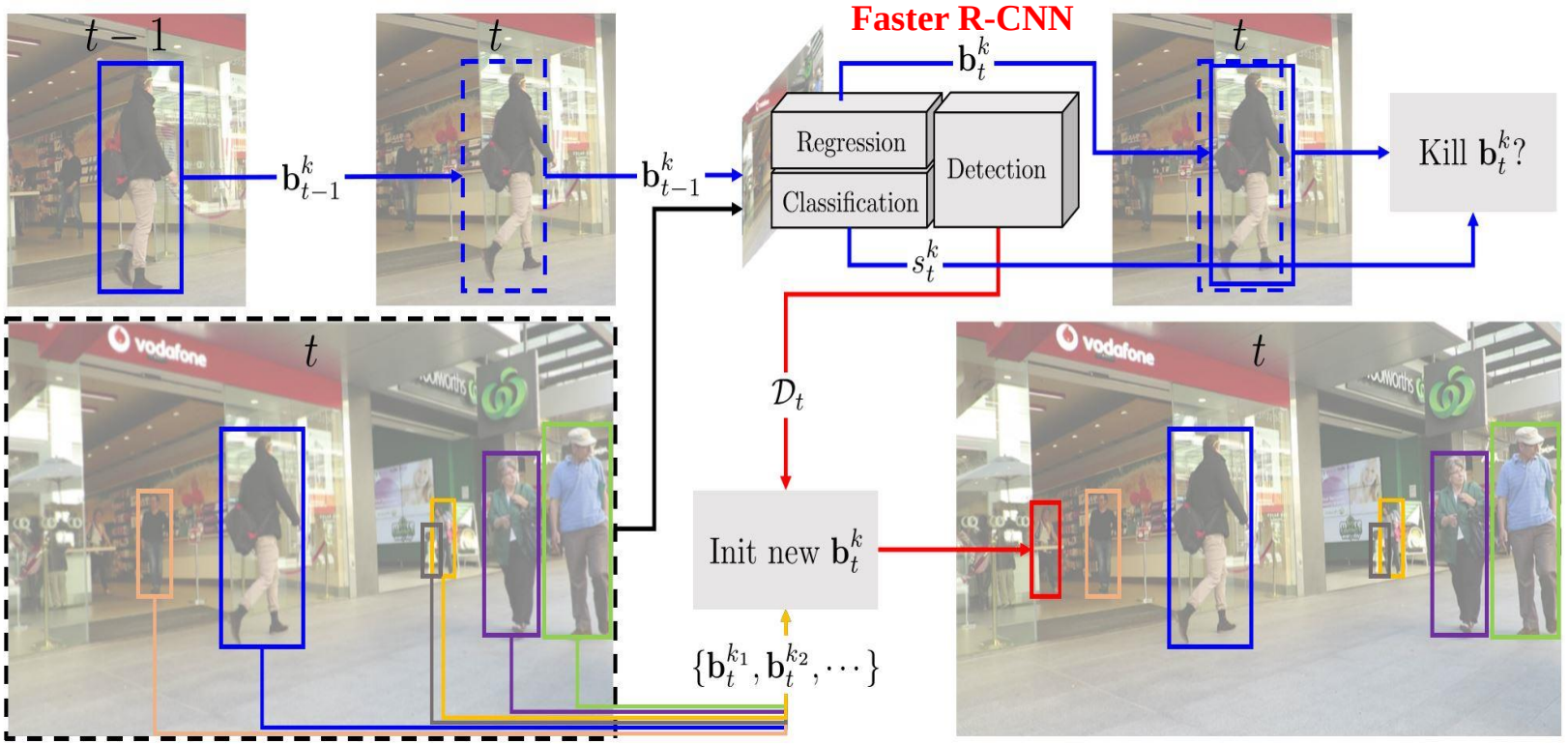
Tracking by Detection Challenges

- Detectors are never perfect
- Features are never invariant in time/space
- Association algorithms are never robust



Tracktor: Association via RoI Pooling+Bbox Regression

- Bounding Box Regression: applying RoI pooling on the features of the current frame but with the previous bounding box coordinates. [P. Bergmann, et al., "Tracking without bells and whistles," IEEE ICCV 2019]

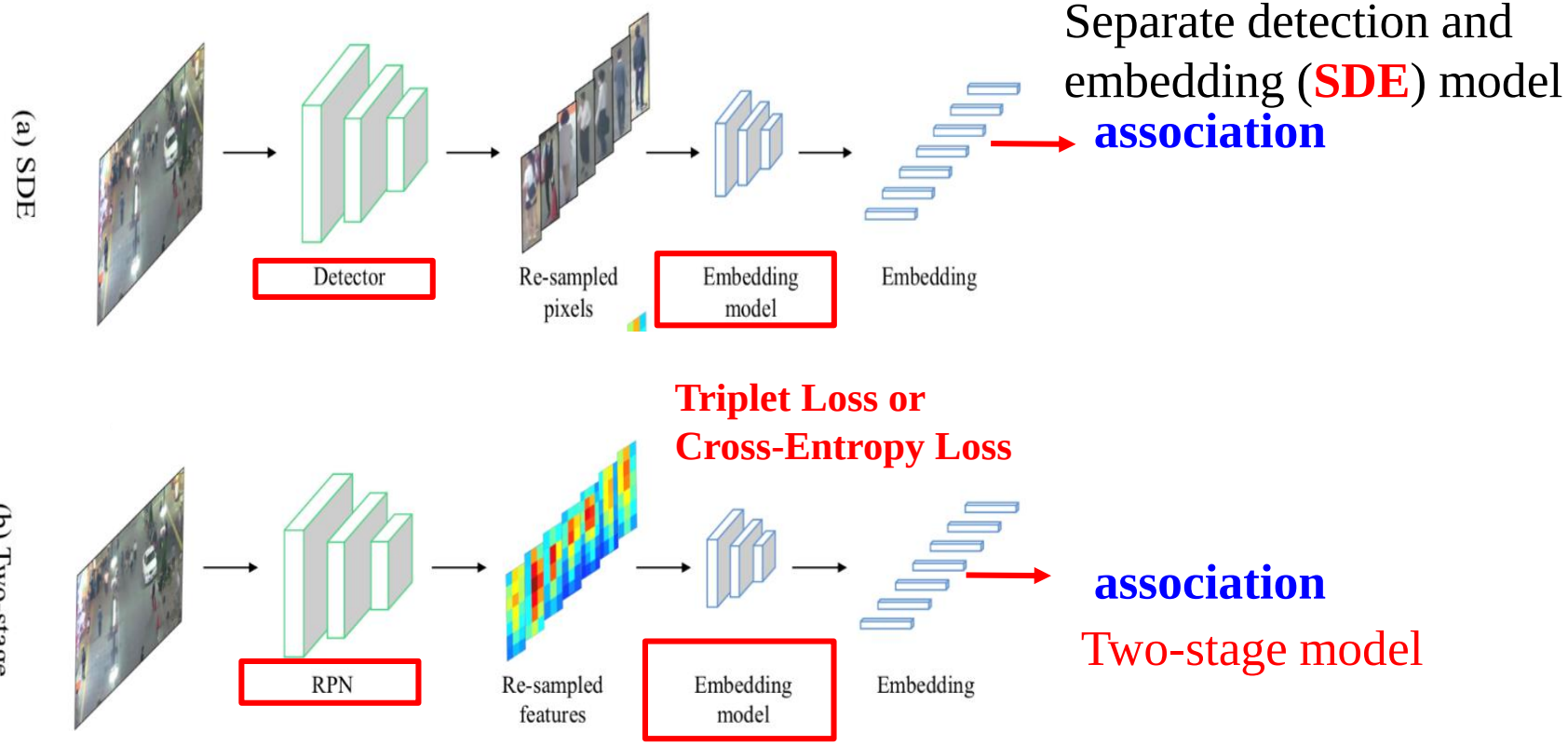


Performance of Tracktor

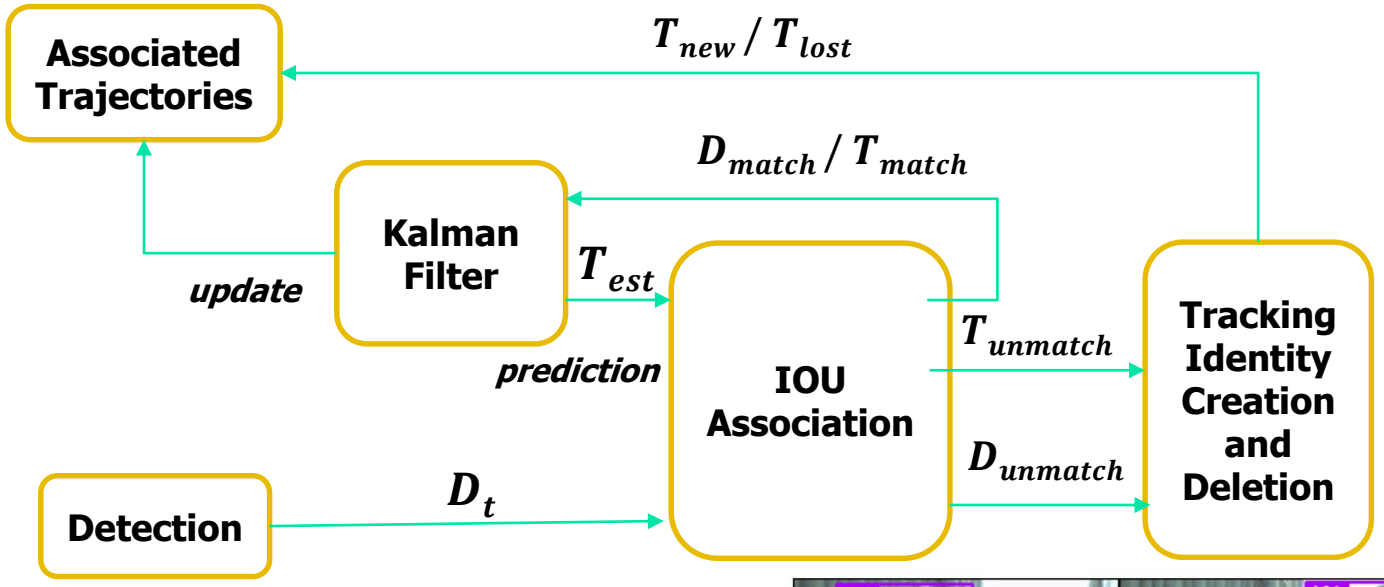
	Method	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
MOT17	Tracktor++	**53.5**	52.3	19.5	36.6	**12201**	248047	2072
	eHAF [58]	51.8	**54.7**	**23.4**	37.9	33212	**236772**	1834
	FWT [23]	51.3	47.6	21.4	35.2	24101	247921	2648
	jCC [30]	51.2	54.5	20.9	37.0	25937	247822	**1802**
	MOTDT17 [9]	50.9	52.7	17.5	35.7	24069	250768	2474
	MHT_DAM [32]	50.7	47.2	20.8	36.9	22875	252889	2314
MOT16	Tracktor++	**54.4**	**52.5**	19.0	**36.9**	**3280**	**79149**	682
	HCC [44]	49.3	50.7	17.8	39.9	5333	86795	**391**
	LMP [59]	48.8	51.3	18.2	40.1	6654	86245	481
	GCRA [43]	48.2	48.6	12.9	41.1	5104	88586	821
	FWT [23]	47.8	44.3	**19.1**	38.2	8886	85487	852
	MOTDT [9]	47.6	50.9	15.2	38.3	9253	85431	792
2D MOT 2015	Tracktor++	**44.1**	46.7	18.0	**26.2**	6477	**26577**	1318
	AP_HWDPL_p [8]	38.5	**47.1**	8.7	37.4	**4005**	33203	586
	AMIR15 [56]	37.6	46.0	15.8	26.8	7933	29397	1026
	JointMC [30]	35.6	45.1	**23.2**	39.3	10580	28508	457
	RAR15pub [17]	35.1	45.4	13.0	42.3	6771	32717	**381**

Embedding Extractions

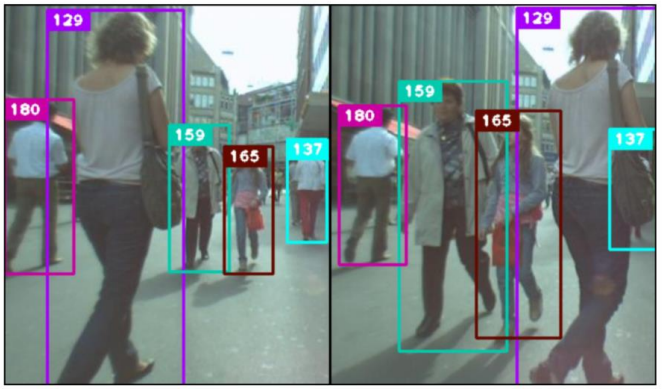
Tracking-by-detection: detection + association



Simple Online and Real-Time Tracking (SORT)



- Only detections' BBoxs from the previous and current frame are presented to the tracker.



Alex Bewley, et al., "Simple Online and Realtime Tracking," ICIP 2016

Deep Embedding Feature Association (DeepSORT)

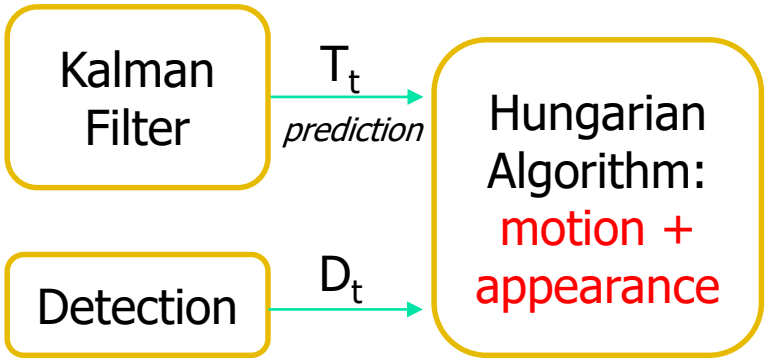
- Input detections D_t no longer contain only bounding box information, but also appearance feature descriptor (2 Conv + 6 Residual blocks), $\{\boldsymbol{r}_j\}$

$D_t = [x, y, w, h, confidence, feature]$

$T = [x, y, r, h, \dot{u}, \dot{v}, \dot{r}, \dot{h}]$, aspect ratio

$T_t = FT_{t-1}$

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & dt & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & dt & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & dt \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Nicolai Wojke†, Alex Bewley, Dietrich Paulus, "Simple Online and Realtime Tracking with Deep Association Metric," ICIP 2017

Motion and Appearance Association

- Mahalanobis distance between predicted Kalman states and newly arrived measurements:

$$d^{(1)}(i,j) = \left(\boldsymbol{d}_\mathrm{j} - \boldsymbol{y}_i\right)^T \boldsymbol{S}_i^{-1}\left(\boldsymbol{d}_\mathrm{j} - \boldsymbol{y}_i\right)$$

- Keep a gallery $R_k = \{\boldsymbol{r}_k^{(i)}\}_{k=1}^{L_k}$ of the last $L_k = 100$ associated appearance descriptors for each track k.
- A Cosine Distance metric is also used into the assignment

$$d^{(2)}(i,j) = min\left\{1 - \boldsymbol{r}_j^T \boldsymbol{r}_k^{(i)} \,|\boldsymbol{r}_k^{(i)} \in \mathcal{R}_i\right\}, \quad \|\boldsymbol{r}_j\| = 1$$

- Combine both metrics with a weighted sum (for Hungarian)

$$c_{ij} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j)$$

Target Association – Assignment Problem

- Assignment Problem: Let *C* be an *nxn* matrix representing the costs of each of *n* workers to perform any of *n* jobs. The assignment problem is to assign jobs to workers so as to minimize the total cost.

$$C(i,j) = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} p & q & r & s \end{array} \\ \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \\ 3 & 6 & 9 & 12 \\ 4 & 8 & 12 & 16 \end{bmatrix} \end{array}$$

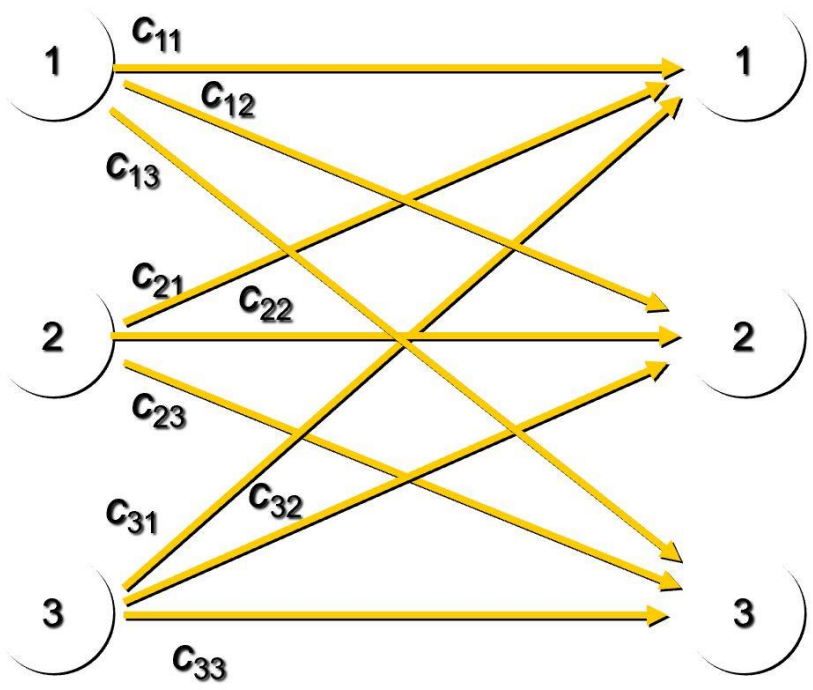
```
Workers = {a,b,c,d}
   Jobs = {p,q,r,s}

An arbitrary assignment
  A = {(a,q),(b,s),(c,r),(d,p)}

Total cost = 23
```

- number of workers does not equal the number of jobs — add dummy workers/jobs with 0 assignment costs as needed
- worker *i* cannot do job *j* — assign $c_{ij} = +\infty$
- Can we find the minimal cost assignment?
- Remember that each assignment must be unique in its row and column.

Assignment: A Linear Programming Problem



$$\text{Min} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}$$

(Sum of assignments from a source should be exactly equal to 1):

$$\sum_{j=1}^{n} X_{ij} = 1 \qquad \text{For } i=1,2,\ldots,n$$

(Sum of assignments to a destination should be equal to the demanded quantity by that destination):

$$\sum_{i=1}^{n} X_{ij} = 1 \qquad \text{For } j=1,2,\ldots,n$$

(Quantities to be assigned can be either 0 or 1):

$$X_{ij} = 0 \text{ or } 1 \qquad \text{For all } i \text{ and } j.$$

Hungarian Method Example

		Machine			
		1	2	3	
Job	1	5	7	9	1
	2	14	10	12	1
	3	15	13	16	1
		1	1	1	

- **Step 1:** Select the smallest value in each row. Subtract this value from each value in that row
- **Step 2:** Do the same for the columns that do not have any zero value.

Hungarian Method Example

		Machine		
		1	2	3
Job	1	**5**	7	9
	2	14	**10**	12
	3	15	**13**	16

		Machine		
		1	2	3
Job	1	0	2	4
	2	4	0	**2**
	3	2	0	3

	Machine		
	1	2	3
1	**0**	2	2
2	4	0	**0**
3	2	**0**	1

If not finished, continue with other columns.

Hungarian Method Example

- **Step 3:** Assignments are made at zero values.
  - Therefore, we assign job 1 to machine 1; job 2 to machine 3, and job 3 to machine 2.
  - Total cost is 5+12+13 = 30.
  - It is not always possible to obtain a feasible assignment as in here.

One More Example

	1	2	3	4
1	***1***	4	6	3
2	9	***7***	10	9
3	***4***	5	11	7
4	8	7	8	***5***

	1	2	**3**	4
1	0	3	**5**	2
2	2	0	**3**	2
3	0	1	**7**	3
4	3	2	**3**	0

	1	2	3	4
1	***0***	3	2	2
2	2	***0***	0	2
3	0	1	4	3
4	3	2	***0***	0

- A feasible assignment is not possible at this moment.
- In such a case, The procedure is to draw a ***minimum*** number of ***lines*** through some of the rows and columns, ***Such that all zero values are crossed out***.

One More Example

	1	2	3	4
1	0	3	2	2
2	2	0	0	2
3	0	1	4	3
4	3	2	0	0

- The next step is to select the smallest uncrossed out element. This element is ***subtracted from every uncrossed out element*** and ***added to every element at the intersection*** of two lines.

	1	2	3	4
1	***0***	2	1	1
2	3	0	***0***	2
3	0	***0***	3	2
4	4	2	0	***0***

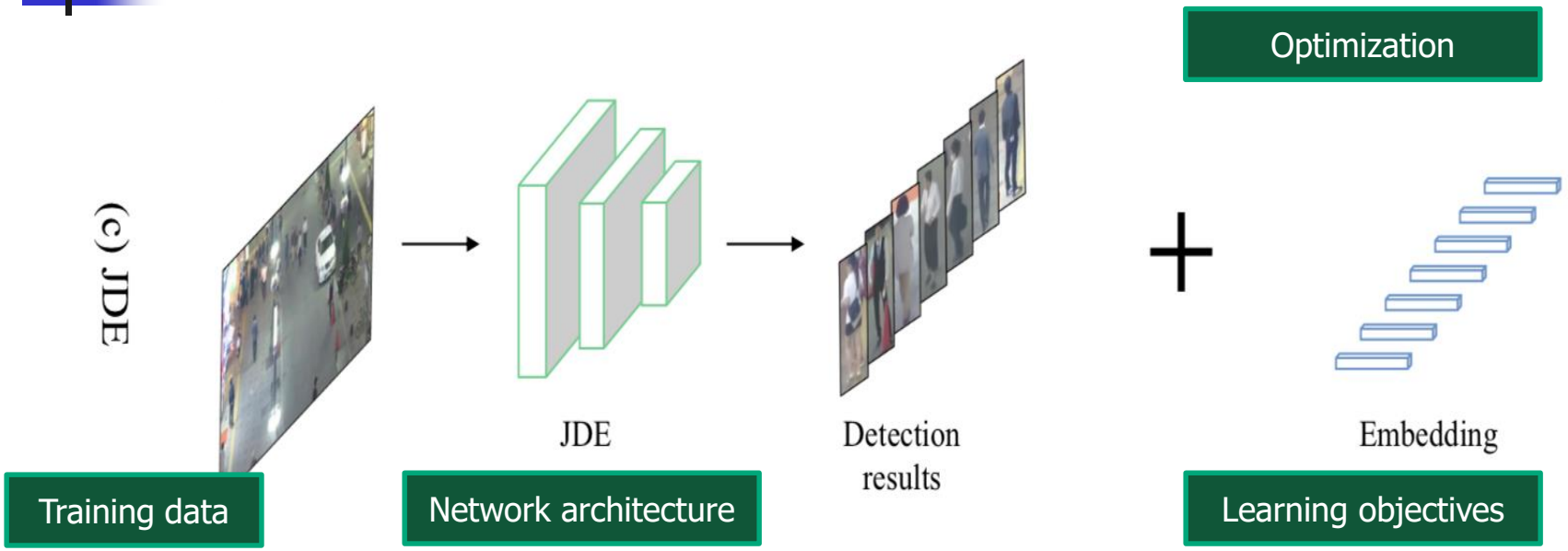
DeepSORT Performance

- A wide residual CNN, trained on a large-scale person Re-ID dataset with over 1,100,000 images of 1,261 pedestrians, deep metric learning in a people tracking context.

		MOTA ↑	MOTP ↑	MT ↑	ML ↓	ID ↓	FM ↓	FP ↓	FN ↓	Runtime ↑
KDNT [16]*	BATCH	68.2	79.4	41.0%	19.0%	933	1093	11479	45605	0.7 Hz
LMP_p [17]*	BATCH	**71.0**	**80.2**	**46.9%**	21.9%	434	**587**	7880	**44564**	0.5 Hz
MCMOT_HDM [18]	BATCH	62.4	78.3	31.5%	24.2%	1394	1318	9855	57257	35 Hz
NOMTwSDP16 [19]	BATCH	62.2	79.6	32.5%	31.1%	**406**	642	**5119**	63352	3 Hz
EAMTT [20]	**ONLINE**	52.5	78.8	19.0%	34.9%	910	**1321**	**4407**	81223	12 Hz
POI [16]*	**ONLINE**	**66.1**	79.5	**34.0%**	20.8%	805	3093	5061	**55914**	10 Hz
SORT [12]*	**ONLINE**	59.8	**79.6**	25.4%	22.7%	1423	1835	8698	63245	**60 Hz**
Deep SORT (Ours)*	**ONLINE**	61.4	79.1	32.8%	**18.2%**	**781**	2008	12852	56668	40 Hz

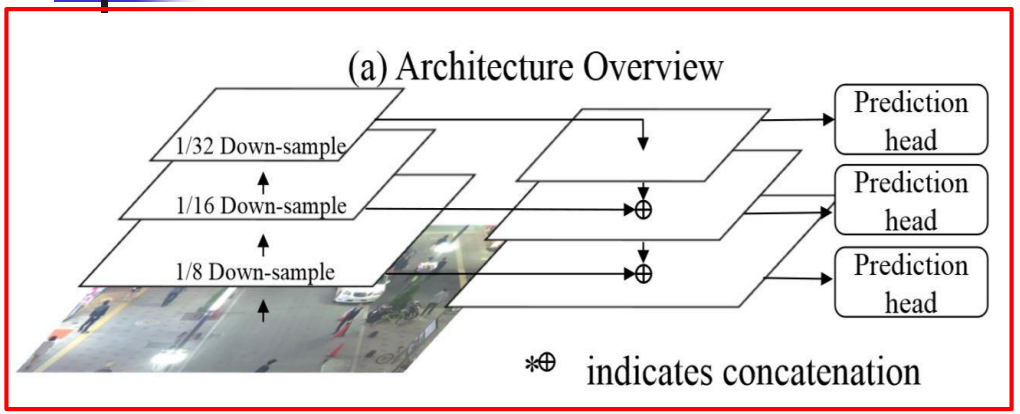
MOT-16 performance, with a non-standard POI detector

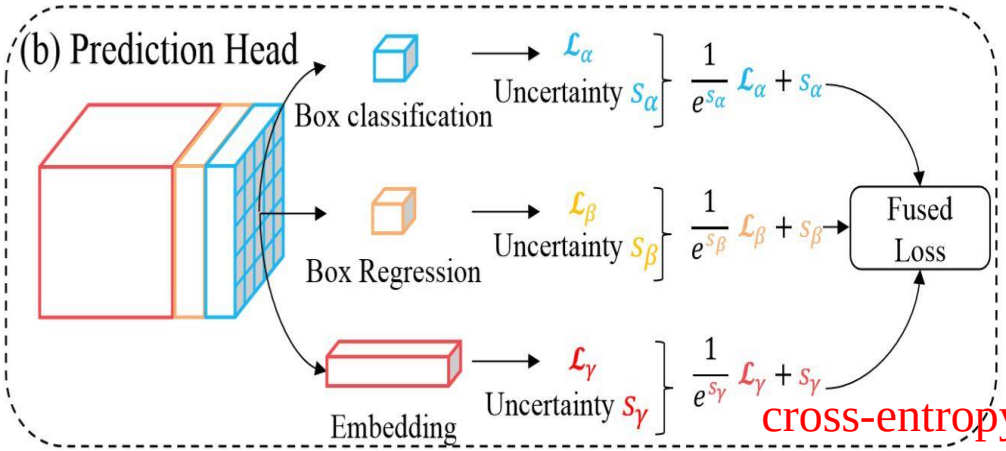
Joint Detection and Embedding (JDE)



- A single network to *simultaneously* output detection results and the corresponding appearance embeddings.
- Improve computational efficiency (near real-time)
- As accurate as state-of-the-art SDE

JDE Architecture



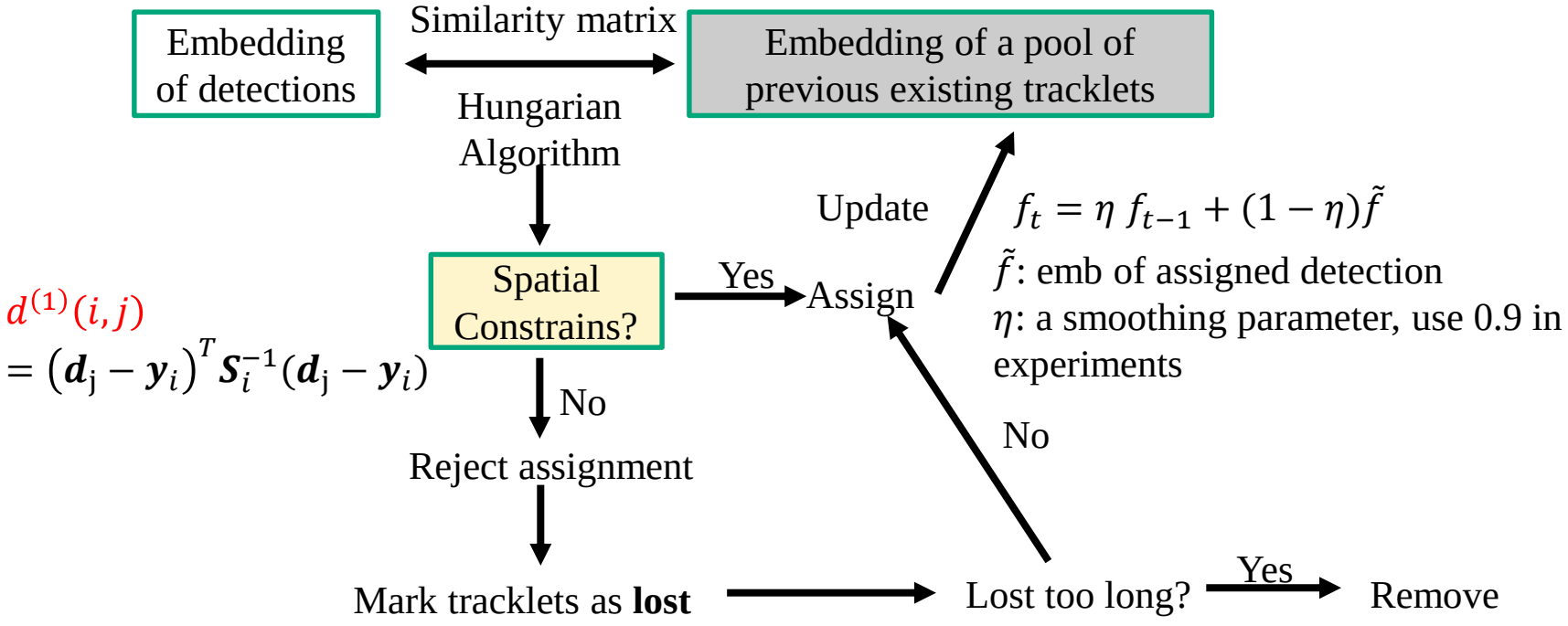


- Feature Pyramid Network [1] + RPN detection based
- Mainly for pedestrians, anchor box aspect ratio 1:3
- Each scale 4 anchor boxes
- Embedding Dimension 64, 128
- Task dependent uncertainty "s_α , s_β, s_r" in the loss

cross-entropy loss

[1] T.Y. Lin et. al, "Feature pyramid net- works for object detection". In CVPR 2017.

Online Association



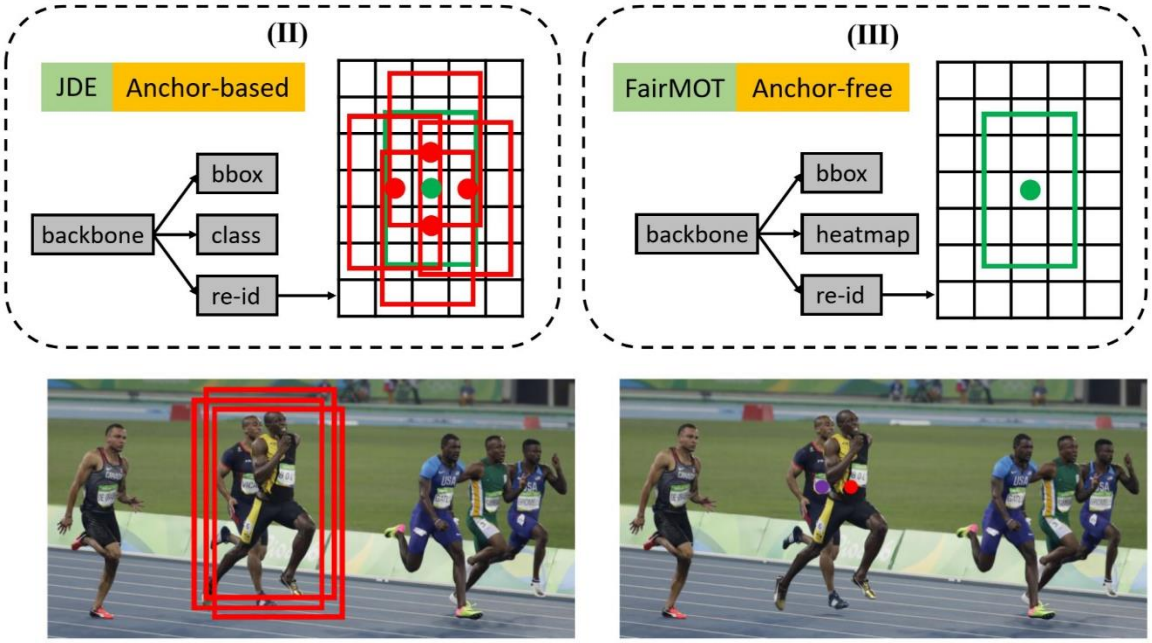
JDE Tracking Performance

- Use additional data for training, so compare under 'private protocol'
- High IDs → low IDF1, 18.8 FPS on MOT-16 (using YOLOv3)

Method	Det	Emb	#box	#id	MOTA	IDF1	MT	ML	IDs	FPSD	FPSA	FPS
DeepSORT_2	FRCNN	WRN	429K	1.2k	61.4	62.2	32.8	18.2	781	<15*	17.4	<8.1
RAR16wVGG	FRCNN	Inception	429K	-	63.0	63.8	39.9	22.1	**482**	<15*	1.6	<1.5
TAP	FRCNN	MRCNN	429K	-	64.8	**73.5**	**40.6**	22.0	794	<15*	18.2	<8.2
CNNMTT	FRCNN	5-Layer	429K	0.2K	65.2	62.2	32.4	21.3	946	<15*	11.2	<6.4
POI	FRCNN	QAN	429K	16K	**66.1**	65.1	34.0	21.3	805	<15*	9.9	<6
JDE-864(ours)	JDE	-	270K	8.7K	62.1	56.9	34.4	**16.7**	1,608	**34.3**	81.0	**24.1**
JDE-1088(ours)	JDE	-	270K	8.7K	64.4	55.8	35.4	20.0	1,544	24.5	**81.5**	18.8

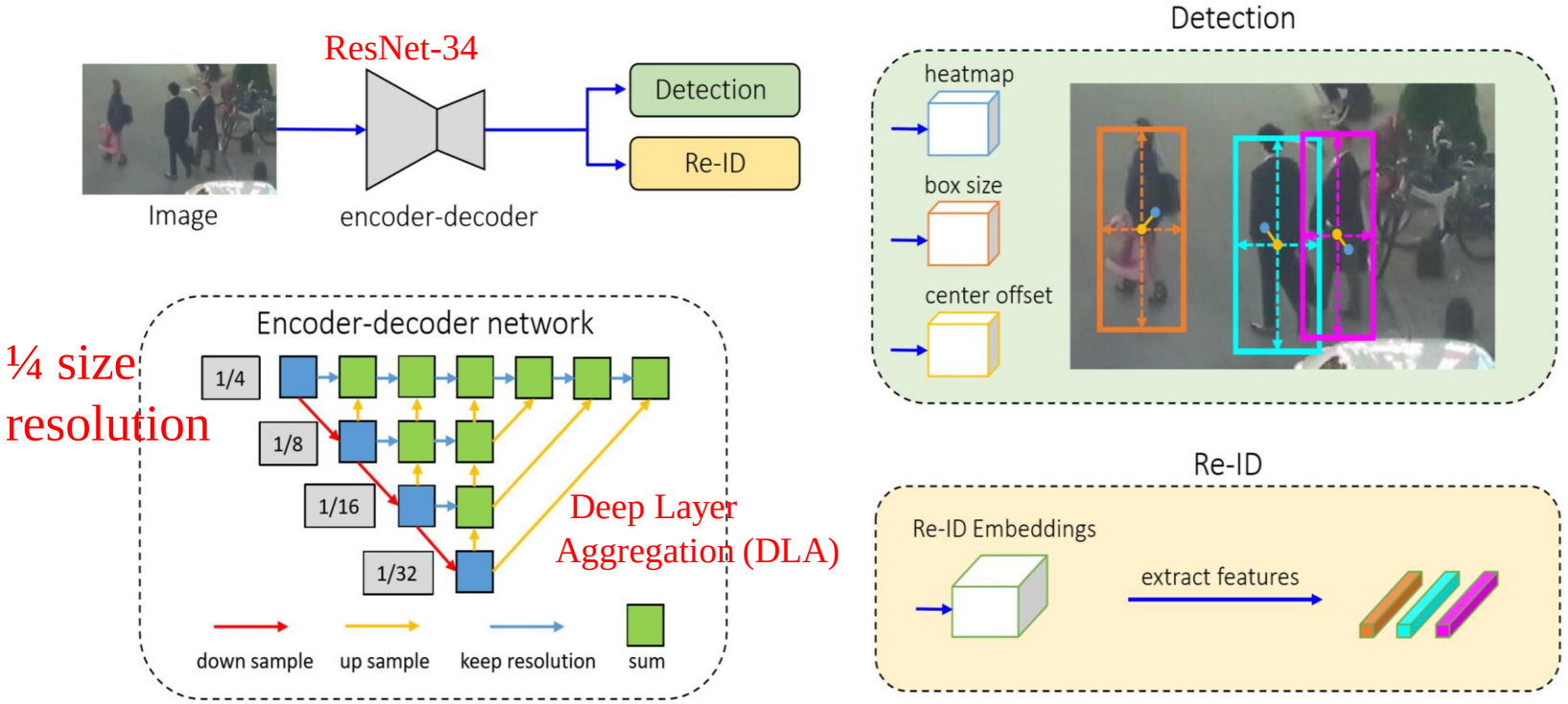
FairMOT

- Red boxes are positive anchors and green boxes are target objects.
- JDE extracts re-ID embedding at the centers of all positive anchors, three different anchors are responsible for predicting the same identity.
- FairMOT extracts re-ID features at the object center.



FairMOT

- Learning lower-dimensional features is better for one-shot MOT



Yifu Zhang, et al., "FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking," arXiv:2004.01888

HeatMap Head and Loss

- Responsible for estimating the locations of the object centers
- GT Heatmap Respor

$$M_{xy} = \sum_{i=1}^{N} \exp^{-\frac{(x-\tilde{c}_x^i)^2+(y-\tilde{c}_y^i)^2}{2\sigma_c^2}}$$

- The total loss functions

$$L_{\text{total}} = \frac{1}{2}\left(\frac{1}{e^{w_1}} L_{\text{detection}} + \frac{1}{e^{w_2}} L_{\text{identity}} + w_1 + w_2\right)$$

Task dependent uncertainty

$$L_{\text{detection}} = L_{\text{heat}} + L_{\text{box}}$$

$$L_{\text{identity}} = -\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbf{L}^i(k)\log(\mathbf{p}(k))$$

$$L_{\text{heat}} = -\frac{1}{N}\sum_{xy}\begin{cases}(1-\hat{M}_{xy})^{\alpha}\log(\hat{M}_{xy}), & M_{xy}=1;\\ (1-M_{xy})^{\beta}(\hat{M}_{xy})^{\alpha}\log(1-\hat{M}_{xy}) & \text{otherwise,}\end{cases}$$

$$L_{\text{box}} = \sum_{i=1}^{N} \|\mathbf{o}^i - \hat{\mathbf{o}}^i\|_1 + \|\mathbf{s}^i - \hat{\mathbf{s}}^i\|_1$$

Center and size offset

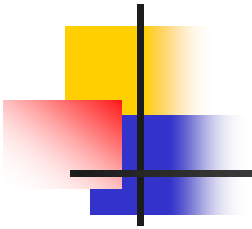
Online Association of FairMOT

- Similar to DeepSORT: Link the detected boxes of the subsequent frames to the existing tracklets according to their cosine distances computed on Re-ID features and their box overlap by Hungarian assignment.
- Kalman Filter is used to predict the locations of the tracklets in the current frame. If it is too far from the linked detection, we set the corresponding cost to infinity which effectively prevents from linking the detections with large motion.
- Similar to JDE: The appearance features of the trackers are updated in each time step to handle appearance variations.

FairMOT Performance

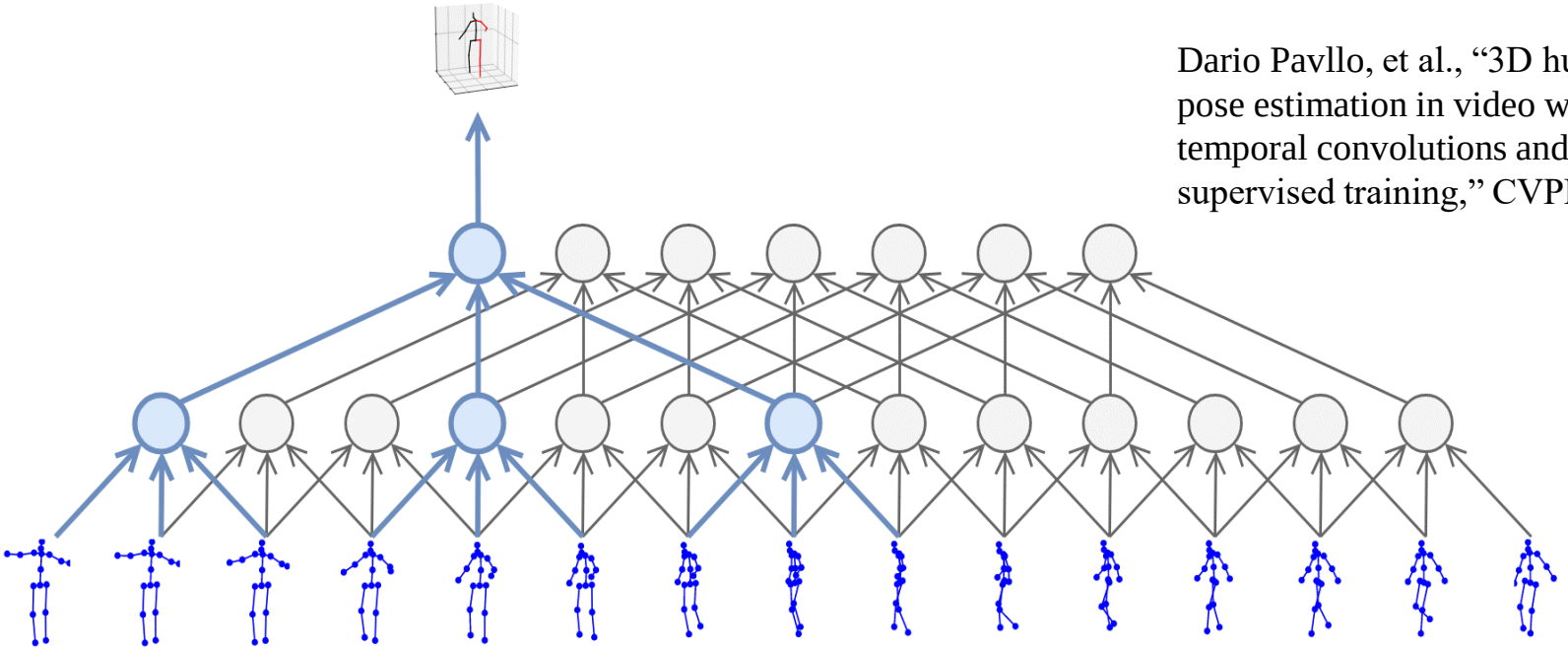
Dataset	Tracker	MOTA↑	IDF1↑	MT↑	ML↓	IDs↓	FPS↑
MOT15	MDP_SubCNN [25]	47.5	55.7	30.0%	18.6%	628	<1.7
	CDA_DDAL [64]	51.3	54.1	36.3%	22.2%	544	<1.2
	EAMTT [65]	53.0	54.0	35.9%	19.6%	7538	<4.0
	AP_HWDPL [66]	53.0	52.2	29.1%	20.2%	708	6.7
	RAR15 [7]	56.5	61.3	45.1%	14.6%	**428**	<3.4
	TubeTK* [44]	58.4	53.1	39.3%	18.0%	854	5.8
	FairMOT (Ours)*	**60.6**	**64.7**	**47.6%**	**11.0%**	591	**30.5**
MOT16	EAMTT [65]	52.5	53.3	19.9%	34.9%	910	<5.5
	SORTwHPD16 [1]	59.8	53.8	25.4%	22.7%	1423	<8.6
	DeepSORT_2 [2]	61.4	62.2	32.8%	18.2%	781	<6.4
	RAR16wVGG [7]	63.0	63.8	39.9%	22.1%	**482**	<1.4
	VMaxx [67]	62.6	49.2	32.7%	21.1%	1389	<3.9
	TubeTK* [44]	64.0	59.4	33.5%	19.4%	1117	1.0
	JDE* [14]	64.4	55.8	35.4%	20.0%	1544	18.5
	TAP [6]	64.8	**73.5**	38.5%	21.6%	571	<8.0
	CNNMTT [5]	65.2	62.2	32.4%	21.3%	946	<5.3
	POI [4]	66.1	65.1	34.0%	20.8%	805	<5.0
	CTrackerV1* [68]	67.6	57.2	32.9%	23.1%	1897	6.8
	FairMOT (Ours)*	**74.9**	72.8	**44.7%**	**15.9%**	1074	**25.9**
MOT17	SST [69]	52.4	49.5	21.4%	30.7%	8431	<3.9
	TubeTK* [44]	63.0	58.6	31.2%	19.9%	4137	3.0
	CTrackerV1* [68]	66.6	57.4	32.2%	24.2%	5529	6.8
	CenterTrack* [70]	67.3	59.9	34.9%	24.8%	**2898**	22.0
	FairMOT (Ours)*	**73.7**	**72.3**	**43.2%**	**17.3%**	3303	**25.9**
MOT20	FairMOT (Ours)*	**61.8**	**67.3**	**68.8%**	**7.6%**	**5243**	**13.2**

Average Precision (AP) for evaluating detection performance, and True Positive Rate (TPR) at a false accept rate of 0.1 for rigorously evaluating re-ID features with ground-truth detections



3D Human Pose Estimation from Videos

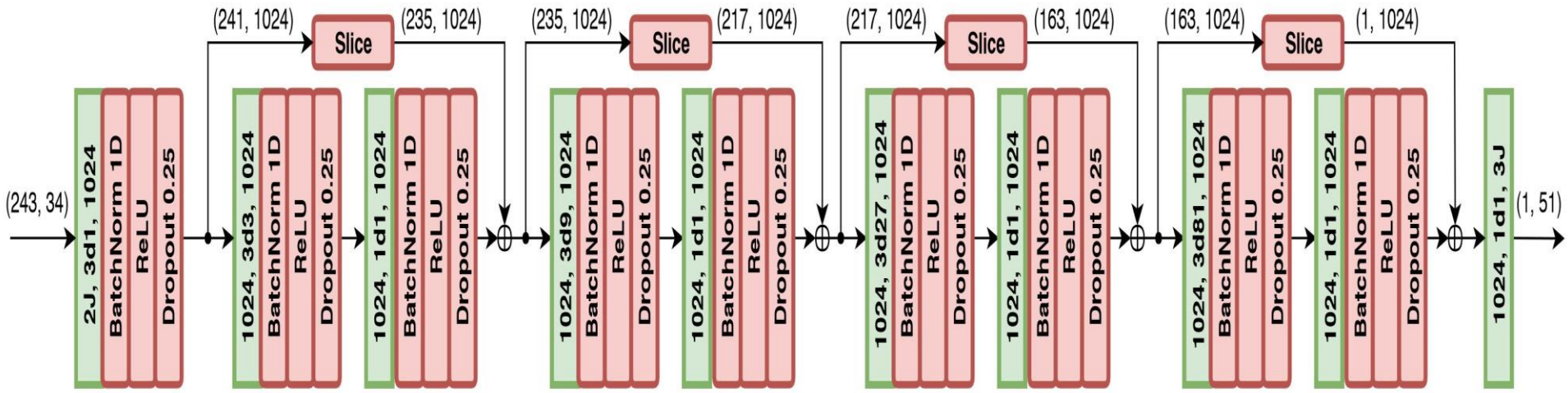
Facebook VideoPose3D



Dario Pavllo, et al., "3D human pose estimation in video with temporal convolutions and semi-supervised training," CVPR 2019

- Temporal convolutional model takes 2D keypoint sequences as input and generates 3D pose estimates as output.
- Dilated temporal convolutions to capture long-term information.

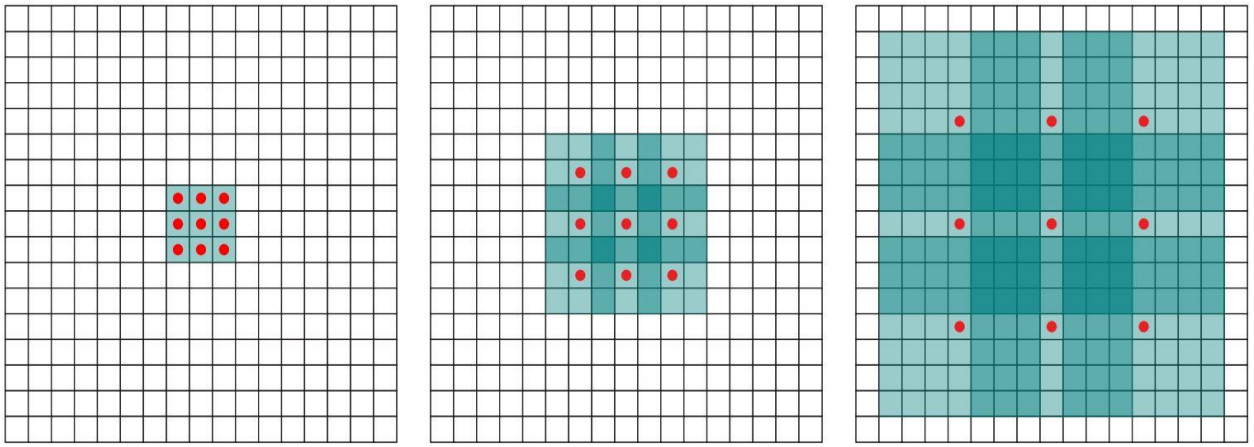
CNN Architecture



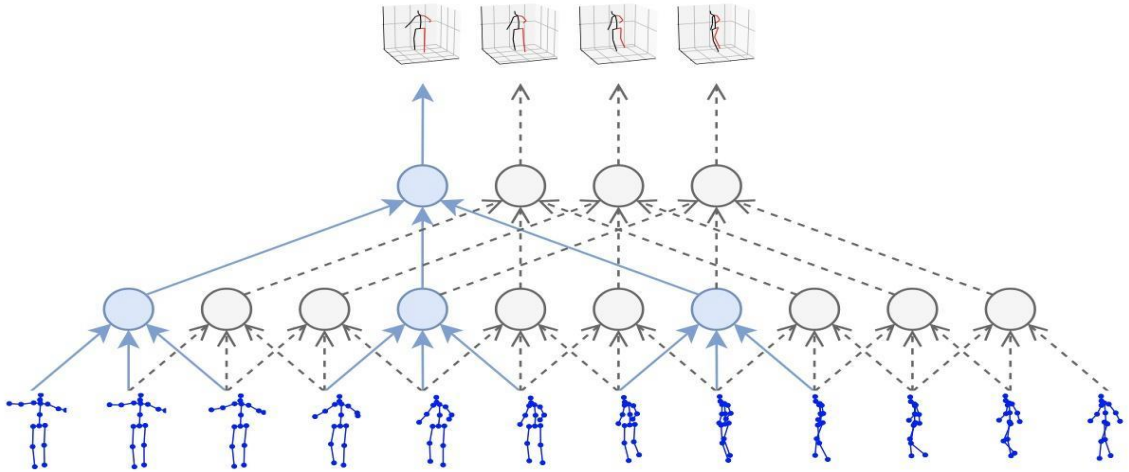
- The input consists of 2D keypoints for a receptive field of 243 frames (B = 4 ResNet blocks) with J = 17 joints.
- Temporal convolutional layers are in green where 2J, 3d1(3^b), 1024 denotes $2 \cdot J$ input channels, kernels of size 3 with dilation 1 (3^b), and 1024 output channels.

Dilated Convolution

2D dilated convolution



1D dilated convolution



Temporal Dilated Convolutional Model

- **A temporal convolutional architecture**
  - offers precise control over the temporal receptive field
  - beneficial to model temporal dependencies for the task of 3D pose estimation.
- **Dilated convolutions**
  - model long-term dependencies
  - while at the same time maintaining efficiency
- **Batch processing:** all frames in the input sequence using both past and future data to exploit temporal information.
- **Real-time scenarios:** also experiment with causal convolutions, i.e., convolutions that only have access to past frames.

Performance

	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Pavlakos *et al.* [41] CVPR'17 (∗)	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Tekin *et al.* [52] ICCV'17	54.2	61.4	60.2	61.2	79.4	78.3	63.1	81.6	70.1	107.3	69.3	70.3	74.3	51.8	63.2	69.7
Martinez *et al.* [34] ICCV'17 (∗)	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Sun *et al.* [50] ICCV'17 (+)	52.8	54.8	54.2	54.3	61.8	67.2	53.1	53.6	71.7	86.7	61.5	53.4	61.6	47.1	53.4	59.1
Fang *et al.* [10] AAAI'18	50.1	54.3	57.0	57.1	66.6	73.3	53.4	55.7	72.8	88.6	60.3	57.7	62.7	47.5	50.6	60.4
Pavlakos *et al.* [40] CVPR'18 (+)	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8	71.1	56.6	52.9	60.9	44.7	47.8	56.2
Yang *et al.* [58] CVPR'18 (+)	51.5	58.9	50.4	57.0	62.1	65.4	49.8	52.7	69.2	85.2	57.4	58.4	43.6	60.1	47.7	58.6
Luvizon *et al.* [33] CVPR'18 (∗)(+)	49.2	51.6	47.6	50.5	51.8	60.3	48.5	51.7	61.5	70.9	53.7	48.9	57.9	44.4	48.9	53.2
Hossain & Little [16] ECCV'18 (†)(∗)	48.4	50.7	57.2	55.2	63.1	72.6	53.0	51.7	66.1	80.9	59.0	57.3	62.4	46.6	49.6	58.3
Lee *et al.* [27] ECCV'18 (†)(∗)	**40.2**	49.2	47.8	52.6	50.1	75.0	50.2	**43.0**	**55.8**	73.9	54.1	55.6	58.2	43.3	43.3	52.8
Ours, single-frame	47.1	50.6	49.0	51.8	53.6	61.4	49.4	47.4	59.3	67.4	52.4	49.5	55.3	39.5	42.7	51.8
Ours, 243 frames, causal conv. (†)	45.9	48.5	44.3	47.8	51.9	57.8	46.2	45.6	59.9	68.5	50.6	46.4	51.0	34.5	35.4	49.0
Ours, 243 frames, full conv. (†)	45.2	**46.7**	43.3	**45.6**	**48.1**	**55.1**	44.6	44.3	57.3	**65.8**	**47.1**	**44.0**	**49.0**	32.8	**33.9**	**46.8**
Ours, 243 frames, full conv. (†)(∗)	45.1	47.4	**42.0**	46.0	49.1	56.7	**44.5**	44.4	57.2	66.1	47.5	44.8	49.2	**32.6**	34.0	47.1

- Human3.6M contains 3.6 million video frames for 11 subjects, of which seven are annotated with 3D poses. Each subject performs 15 actions that are recorded using four synchronized cameras at 50 Hz.
- Mean per-joint position error (MPJPE) in millimeters: mean Euclidean distance between predicted joint positions and ground-truth joint positions