

رَبِّ الْعَالَمِينَ



# مبانی بینایی کامپیوٹر

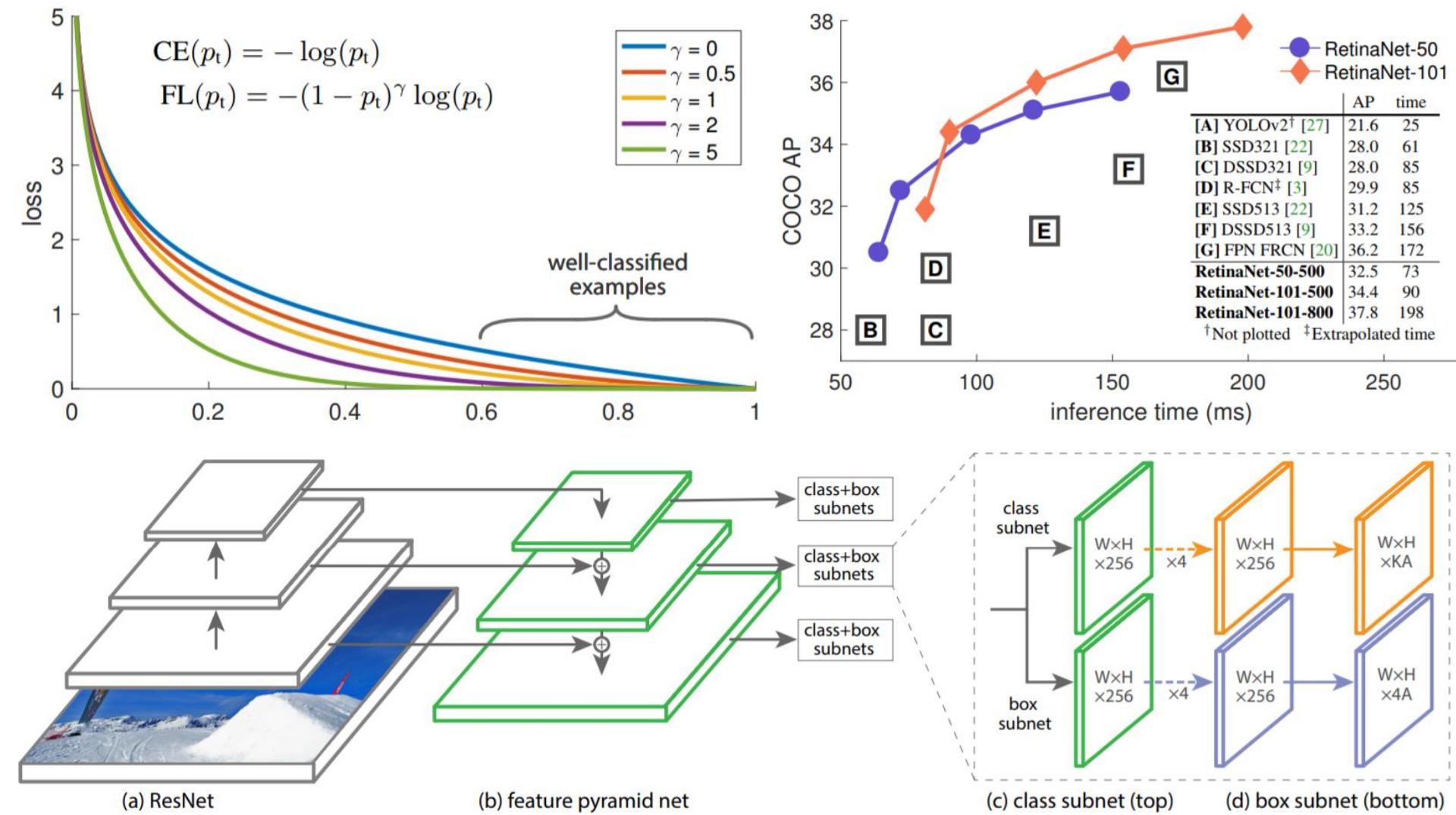
مدرس: محمدرضا محمدی

۱۴۰۱

# تشخيص اشياء

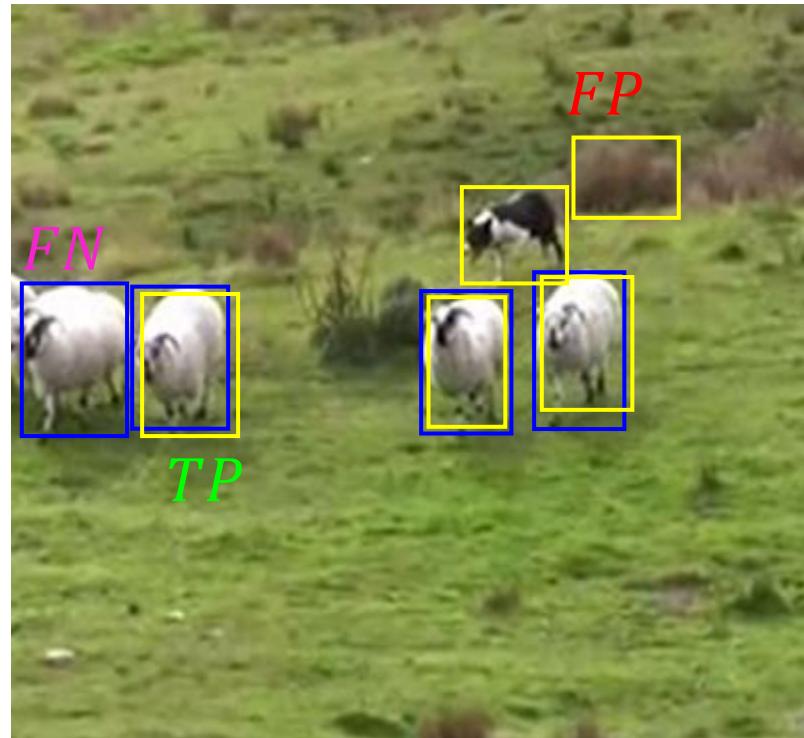
# Object Detection

# RetinaNet



# دقت متوسط (AP)

- در یک تصویر تشخیص‌های متفاوتی داریم که طبق شکل زیر تعریف می‌شوند:



$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

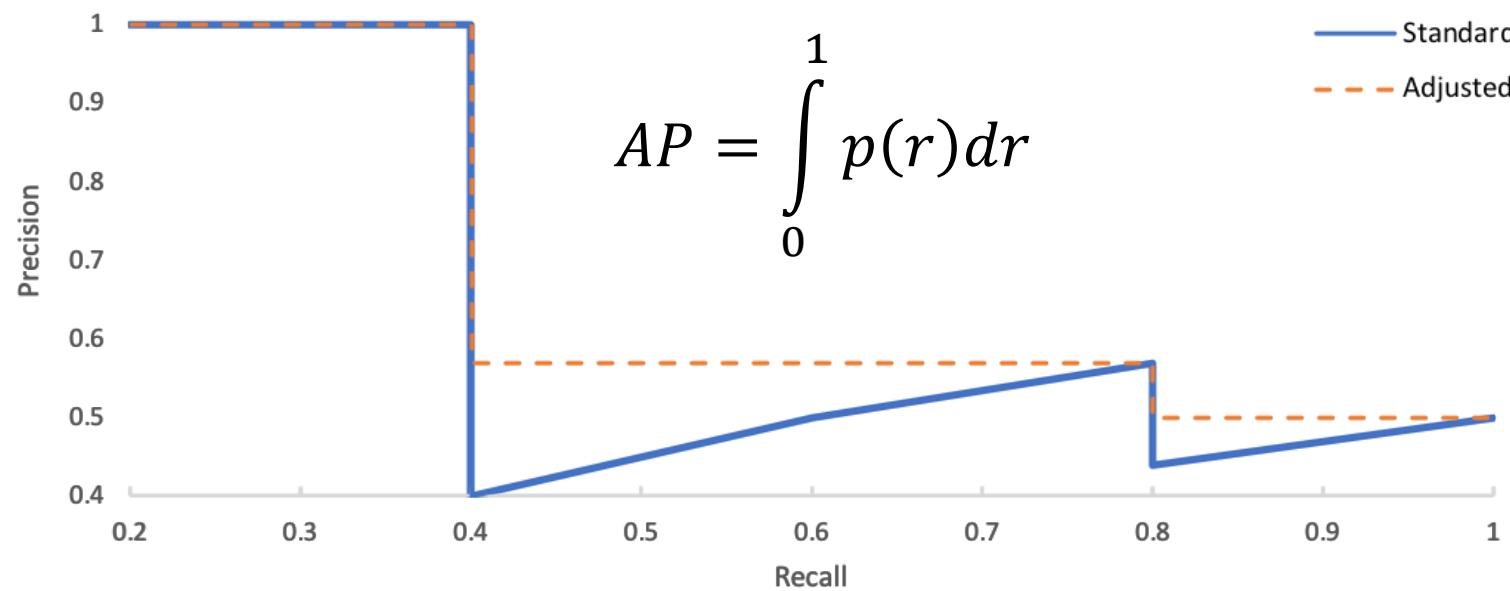
$$F_1 = 2 \frac{Precision \times Recall}{Precision + Recall}$$

- حد آستانه برای پذیرش یک تشخیص را چند قرار دهیم؟

# دقت متوسط (AP)

- تشخیص‌ها را بر اساس امتیاز آنها مرتب می‌کنیم و دقت متوسط را محاسبه می‌کنیم

Rank	Score
1	0.99
2	0.96
3	0.91
4	0.89
5	0.84
6	0.72
7	0.56
8	0.38
9	0.25
10	0.09



# میانگین دقت متوسط (mAP)

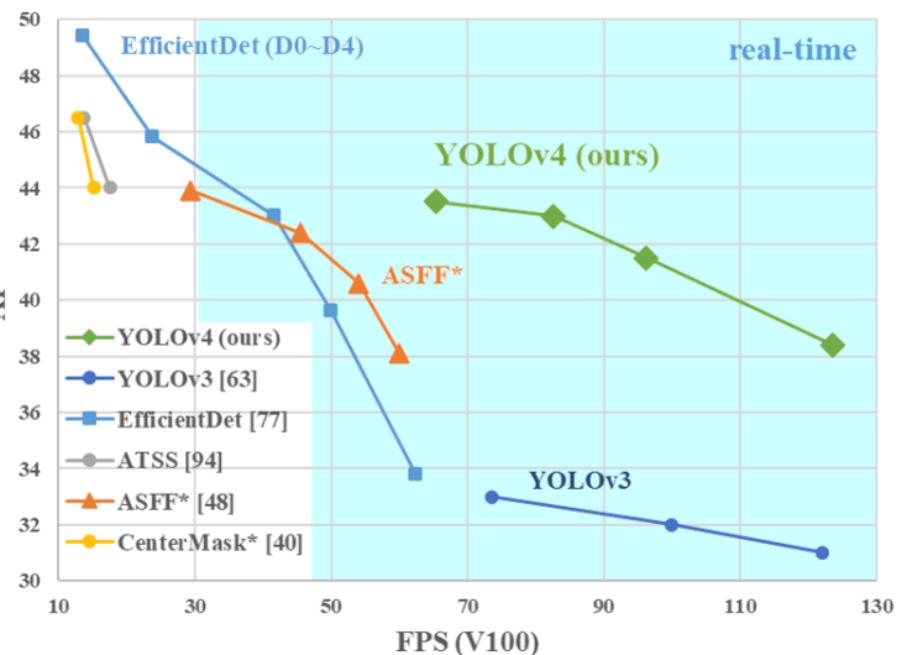
- در مسائل تشخیص چندین شیء متفاوت، دقت متوسط برای هر شیء به طور جداگانه محاسبه شده و میانگین آن برای تمام کلاس‌ها گزارش می‌شود

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast R-CNN [5]	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster R-CNN [15]	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
YOLO [14]	07++12	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
SSD300 [11]	07++12	72.4	85.6	80.1	70.5	57.6	46.2	79.4	76.1	89.2	53.0	77.0	60.8	87.0	83.1	82.3	79.4	45.9	75.9	69.5	81.9	67.5
SSD512 [11]	07++12	74.9	87.4	82.3	75.8	59.0	52.6	81.7	81.5	90.0	55.4	79.0	59.8	88.4	84.3	84.7	83.3	50.2	78.0	66.3	86.3	72.0
ResNet [6]	07++12	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
YOLOv2 544	07++12	73.4	86.3	82.0	74.8	59.2	51.8	79.8	76.5	90.6	52.1	78.2	58.5	89.3	82.5	83.4	81.3	49.1	77.2	62.4	83.8	68.7

# روش‌های دیگر محاسبه AP

- در مجموعه داده COCO شش روش برای دیگر برای محاسبه AP پیشنهاد شده است:
  - : همان مقدار قبل به ازای  $\text{IoU}=0.50$  است  $\text{AP}_{50}$  -
  - : همان مقدار قبل به ازای  $\text{IoU}=0.75$  است  $\text{AP}_{75}$  -
  - : میانگین در  $\text{IoU}=0.50:0.05:0.95$  است  $\text{AP}$  -
  - ( $\text{area} < 32^2 \text{ px}$ ) برای اشیاء کوچک  $\text{AP}_{\text{S}}$  -
  - ( $32^2 \text{ px} < \text{area} < 96^2 \text{ px}$ ) برای اشیاء متوسط  $\text{AP}_{\text{M}}$  -
  - ( $96^2 \text{ px} < \text{area}$ ) برای اشیاء بزرگ  $\text{AP}_{\text{L}}$  -

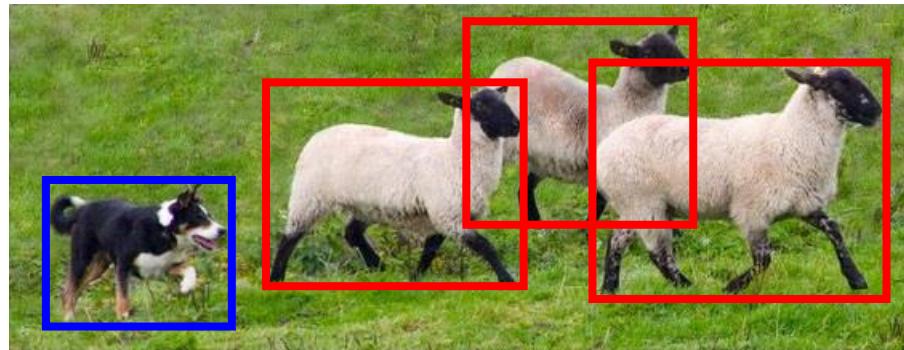
## MS COCO Object Detection



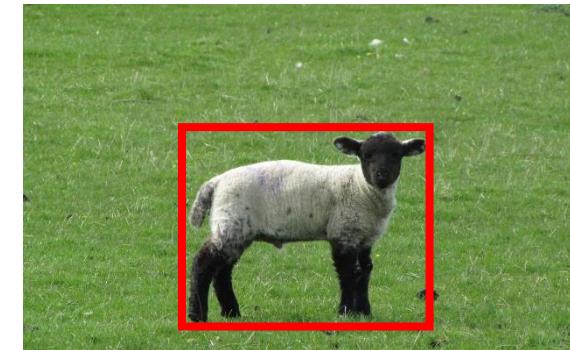
Method	Backbone	Size	FPS	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
<b>YOLOv4: Optimal Speed and Accuracy of Object Detection</b>									
YOLOv4	CSPDarknet-53	416	38 (M)	41.2%	62.8%	44.3%	20.4%	44.4%	56.0%
YOLOv4	CSPDarknet-53	512	31 (M)	43.0%	64.9%	46.5%	24.3%	46.1%	55.2%
YOLOv4	CSPDarknet-53	608	23 (M)	43.5%	65.7%	47.3%	26.7%	46.7%	53.3%
<b>Learning Rich Features at High-Speed for Single-Shot Object Detection [84]</b>									
LRF	VGG-16	300	76.9 (M)	32.0%	51.5%	33.8%	12.6%	34.9%	47.0%
LRF	ResNet-101	300	52.6 (M)	34.3%	54.1%	36.6%	13.2%	38.2%	50.7%
LRF	VGG-16	512	38.5 (M)	36.2%	56.6%	38.7%	19.0%	39.9%	48.8%
LRF	ResNet-101	512	31.3 (M)	37.3%	58.5%	39.7%	19.7%	42.8%	50.1%
<b>YOLOv3: An incremental improvement [63]</b>									
YOLOv3	Darknet-53	320	45 (M)	28.2%	51.5%	29.7%	11.9%	30.6%	43.4%
YOLOv3	Darknet-53	416	35 (M)	31.0%	55.3%	32.3%	15.2%	33.2%	42.8%
YOLOv3	Darknet-53	608	20 (M)	33.0%	57.9%	34.4%	18.3%	35.4%	41.9%
YOLOv3-SPP	Darknet-53	608	20 (M)	36.2%	60.6%	38.2%	20.6%	37.4%	46.1%
<b>SSD: Single shot multibox detector [50]</b>									
SSD	VGG-16	300	43 (M)	25.1%	43.1%	25.8%	6.6%	25.9%	41.4%
SSD	VGG-16	512	22 (M)	28.8%	48.5%	30.3%	10.9%	31.8%	43.5%
<b>Single-shot refinement neural network for object detection [95]</b>									
RefineDet	VGG-16	320	38.7 (M)	29.4%	49.2%	31.3%	10.0%	32.0%	44.4%
RefineDet	VGG-16	512	22.3 (M)	33.0%	54.5%	35.5%	16.3%	36.3%	44.3%
<b>Parallel Feature Pyramid Network for Object Detection [34]</b>									
PFPNet-R	VGG-16	320	33 (M)	31.8%	52.9%	33.6%	12%	35.5%	46.1%
PFPNet-R	VGG-16	512	24 (M)	35.2%	57.6%	37.9%	18.7%	38.6%	45.9%
<b>Focal Loss for Dense Object Detection [45]</b>									
RetinaNet	ResNet-50	500	13.9 (M)	32.5%	50.9%	34.8%	13.9%	35.8%	46.7%
RetinaNet	ResNet-101	500	11.1 (M)	34.4%	53.1%	36.8%	14.7%	38.5%	49.1%
RetinaNet	ResNet-50	800	6.5 (M)	35.7%	55.0%	38.5%	18.9%	38.9%	46.3%
RetinaNet	ResNet-101	800	5.1 (M)	37.8%	57.5%	40.8%	20.2%	41.1%	49.2%

# مسئله‌های بینایی کامپیووتر

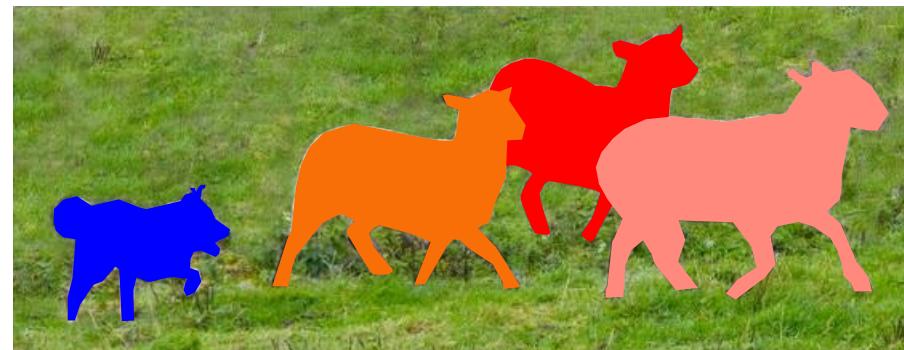
تشخیص اشیاء (Object Detection)



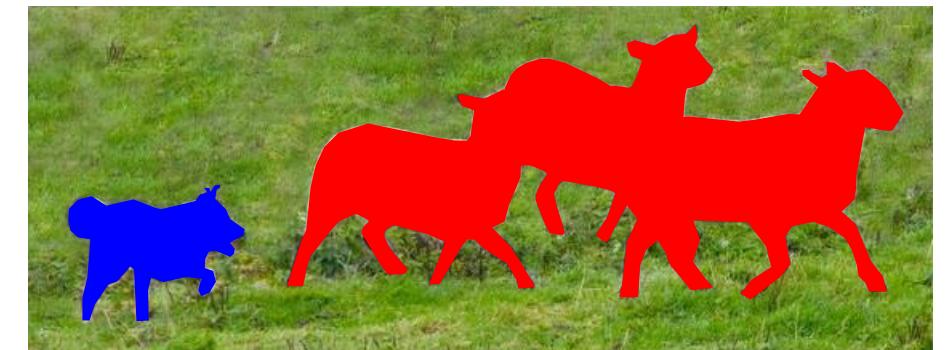
دسته‌بندی + مکان‌یابی



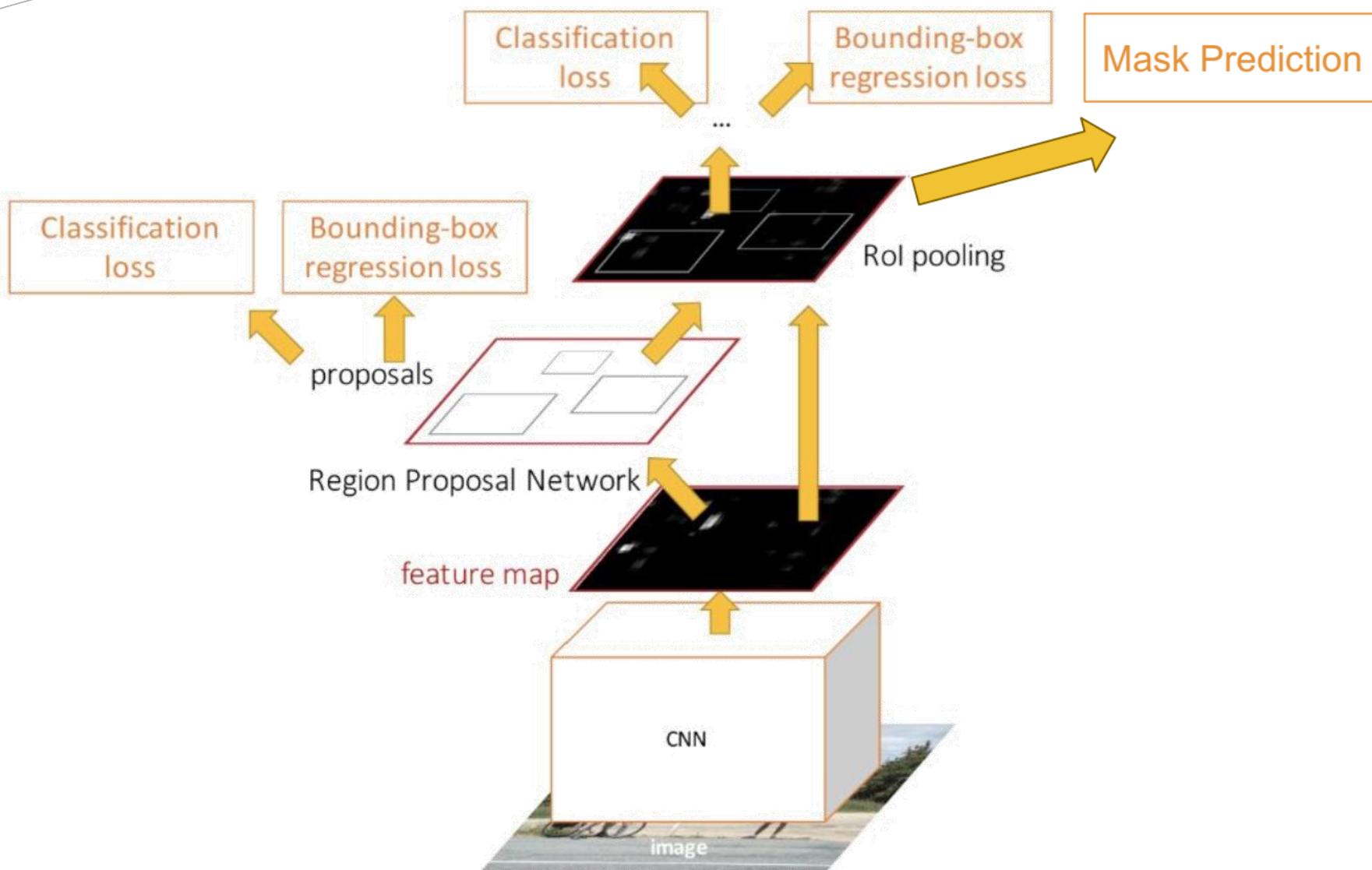
بخش‌بندی نمونه‌ها (Instance Segmentation)



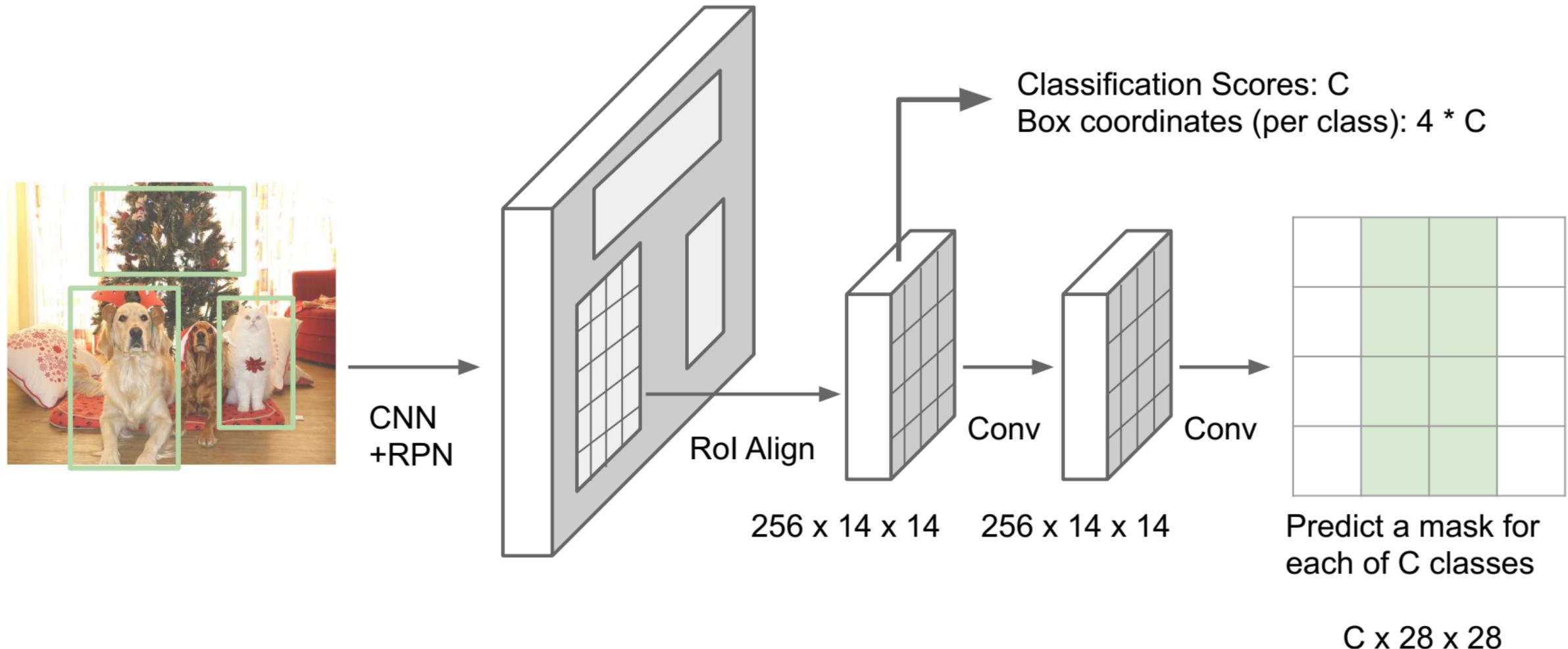
بخش‌بندی معنایی (Semantic Segmentation)



# Mask R-CNN



# Mask R-CNN





رديابي اشياء

Object Tracking

# ردیابی اشیاء



# ردیابی اشیاء



# تشخیص و ردیابی اشیاء



## چالش‌های ردیابی اشیاء

- اشیاء کوچک دارای ویژگی‌های دیداری اندک
- حرکت‌های بسیار سریع و نامنظم
- پس زمینه پیچیده
- انسداد، خروج از دید دوربین و بازگشت
- وجود اشیاء مشابه در همسایگی



# چالش‌های ردیابی اشیاء

- اشیاء کوچک دارای ویژگی‌های دیداری اندک
- حرکت‌های بسیار سریع و نامنظم
- پس زمینه پیچیده
- انسداد، خروج از دید دوربین و بازگشت
- وجود اشیاء مشابه در همسایگی
- ردیابی همزمان چندین شیء متحرک
- ردیابی میان دوربین‌های مختلف



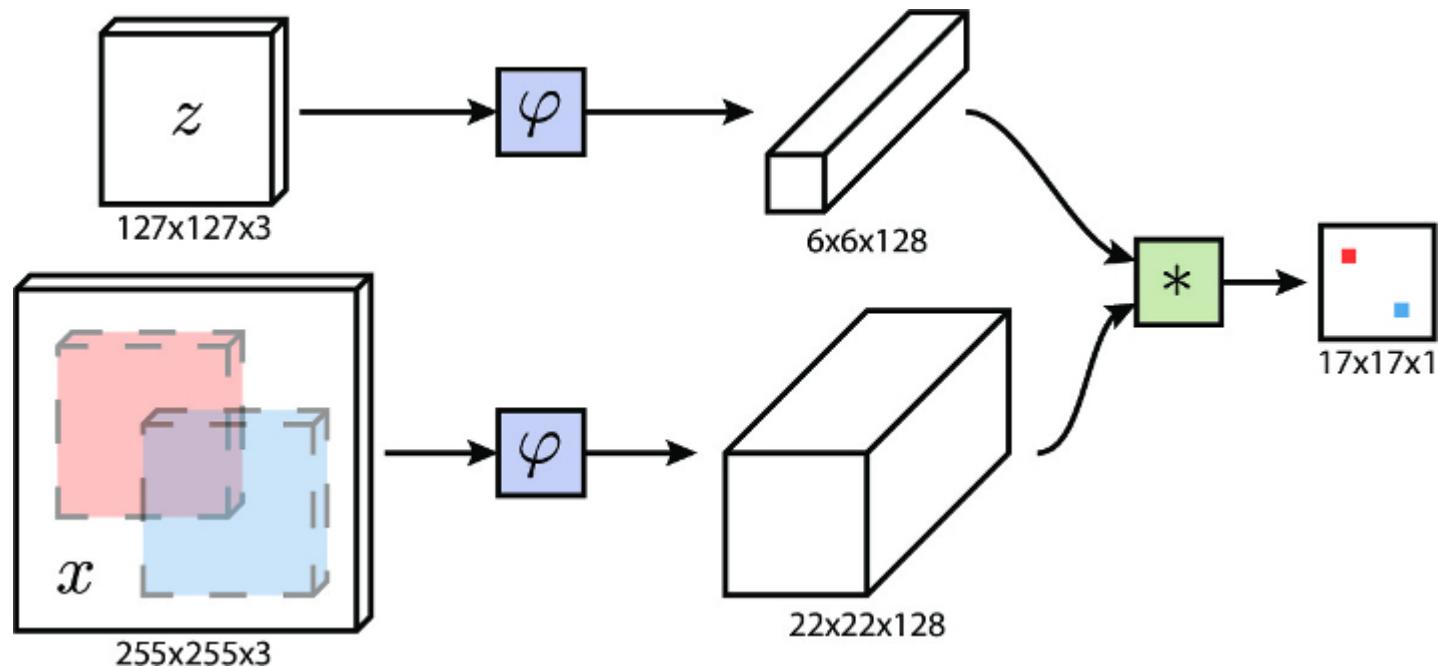
# ردیابی یک شیء

- برای پیدا کردن محل شیء در فریم بعد می‌توان از تطبیق کلیشه استفاده کرد
- برای اشیاء ساده و نرخ تصویربرداری بالا می‌تواند عملکرد مناسبی داشته باشد

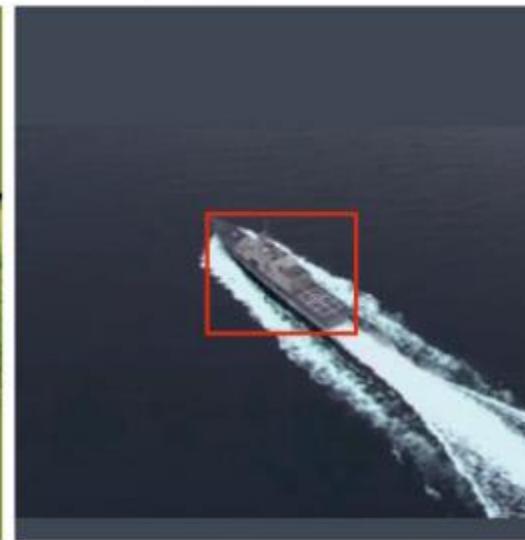
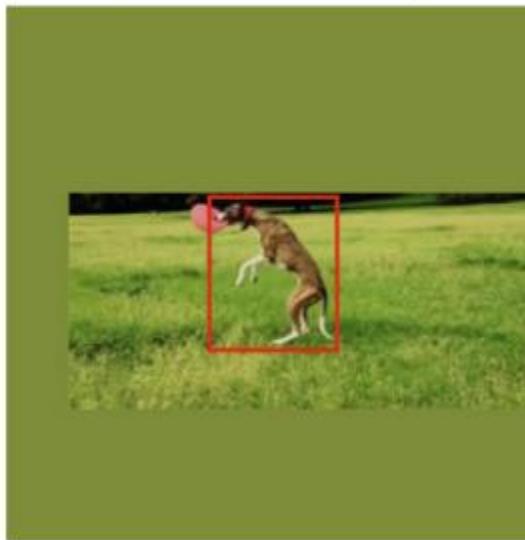
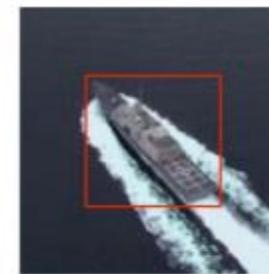


# SiamFC

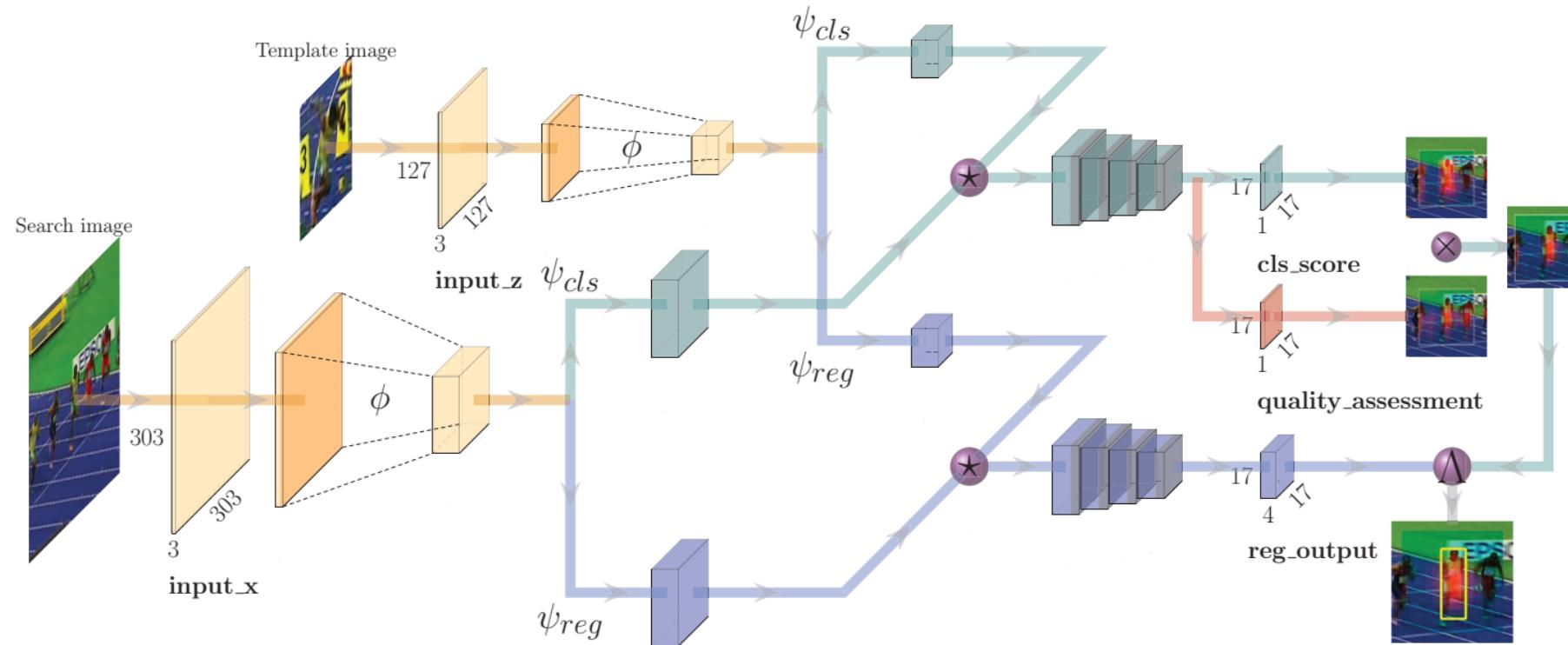
- برای مقایسه دقیق‌تر، می‌توان با استفاده از شبکه‌های عمیق بازنمایی معنایی سطح بالا از تصاویر استخراج کرد و سپس مقایسه را انجام داد



# SiamFC



# SiamFC++



- : feature extractor
- : classification branch
- : regression branch
- : quality assessment
- : operation
- $\star$ : cross-correlation
- $\times$ : element-wise production
- $\Lambda$ : argmax (taking left w.r.t. right)

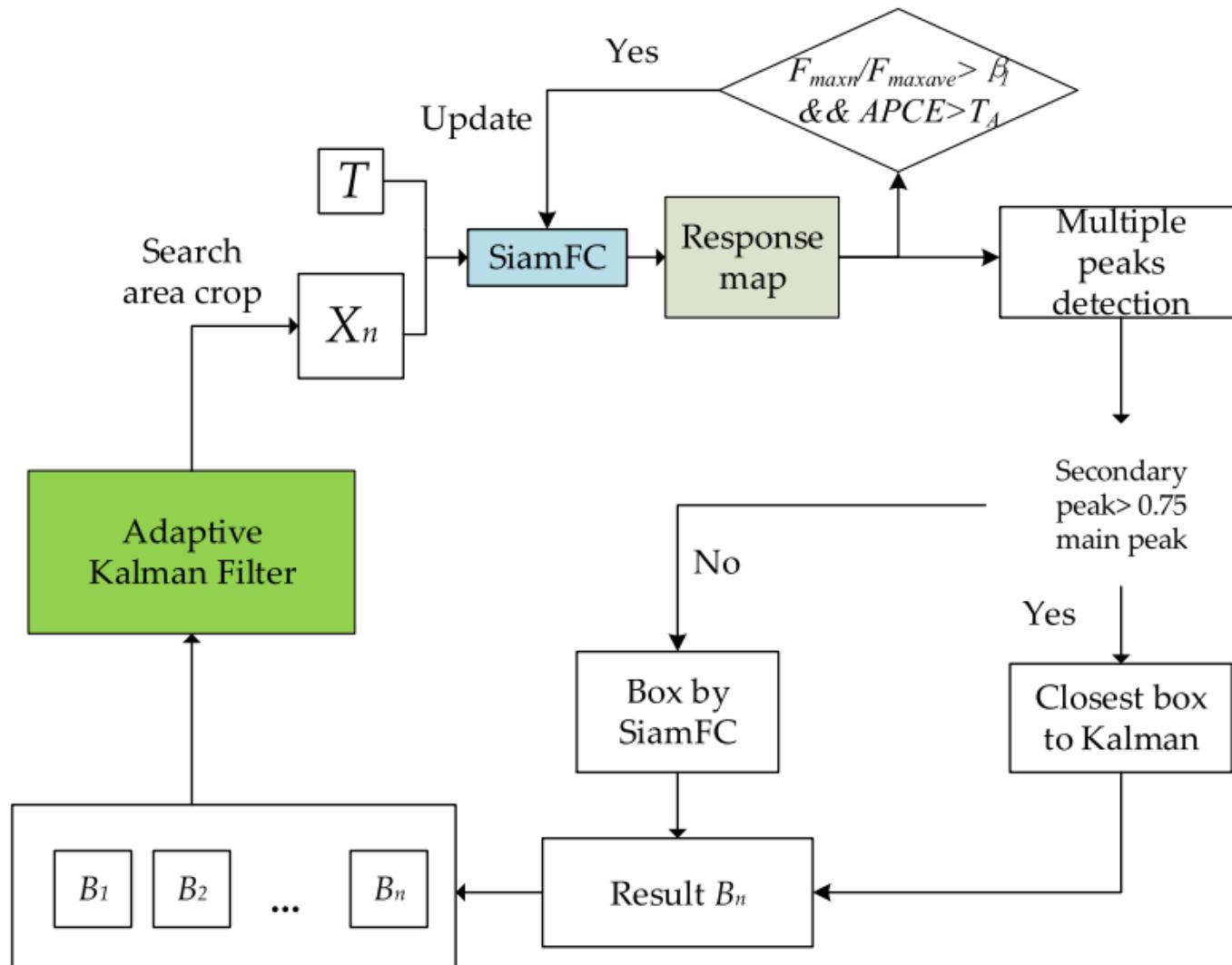
# ردیابی با پیش‌بینی حرکت

- استفاده از پیش‌بینی حرکت، به خصوص در حضور اشیاء مشابه، می‌تواند کمک قابل توجهی به ردیابی نماید



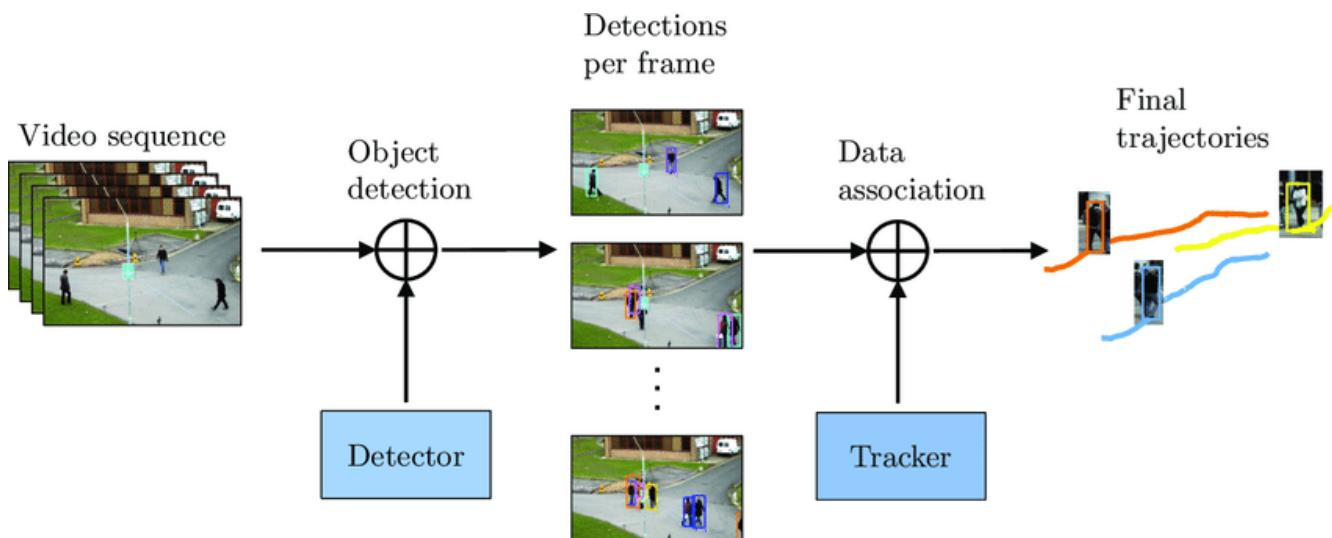
- می‌توان برای افزایش سرعت، حتی ناحیه جستجو را محدود کرد
- از جمله الگوریتم‌های پرکاربرد برای پیش‌بینی محل شیء می‌توان به فیلتر Kalman اشاره کرد

# رديابي با پيش‌بيني حرکت



# ردیابی با تشخیص

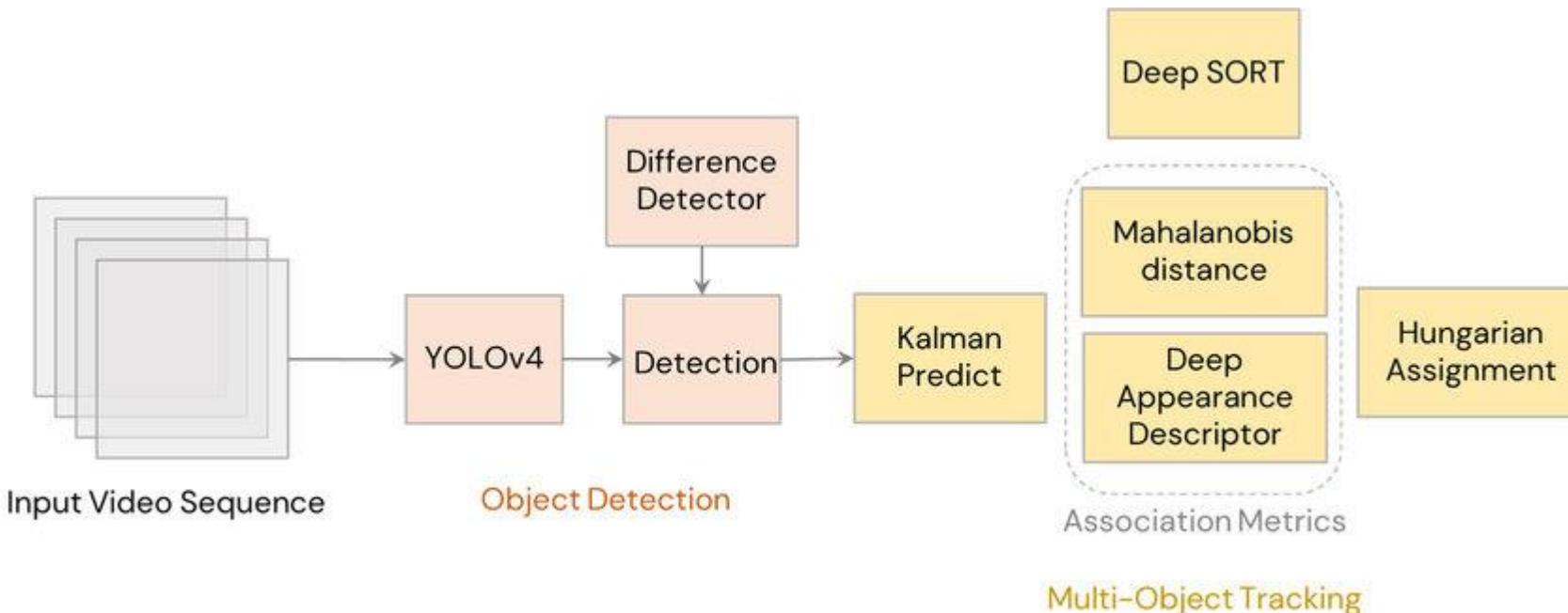
- این خانواده از روش‌ها با معرفی روش **SORT** مطرح شدند
- این روش (و دیگر روش‌های مبتنی بر همین رویکرد) از سه گام اساسی تشکیل می‌شوند:
  - تشخیص اشیاء
  - **Faster R-CNN**
  - قربات‌سنگی (مبتنی بر ویژگی‌های مکانی و بصری)



- فقط از **Kalman** و **IoU** استفاده می‌کند
- انتساب
- **Hungarian**

# DeepSORT

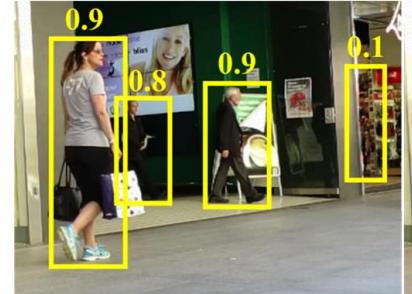
- در یک شبکه عمیق برای سنجش قرابت بصری اشیاء استفاده می‌شود
- در برخی از مقالات بعد از DeepSORT فقط به استخراج توصیفگرهای ظاهری بهتر پرداخته شده است



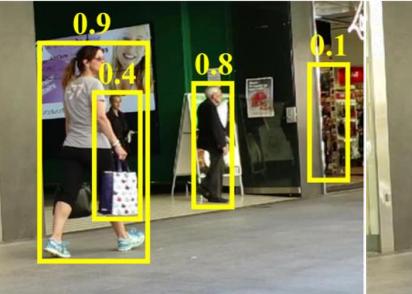
# ByteTrack

- برای بهبود ردیابی اشیاء در فریم‌هایی که به خوبی قابل تشخیص نیستند، تشخیص‌های دارای امتیاز کم هم می‌توانند استفاده شوند در صورتیکه به خوبی با تشخیص‌های قوی در فریم‌های قبل منطبق شوند

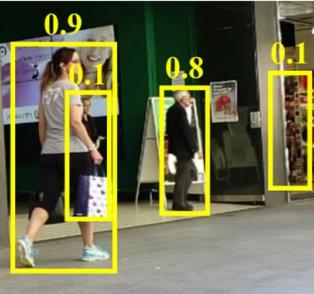
Frame  $t_1$



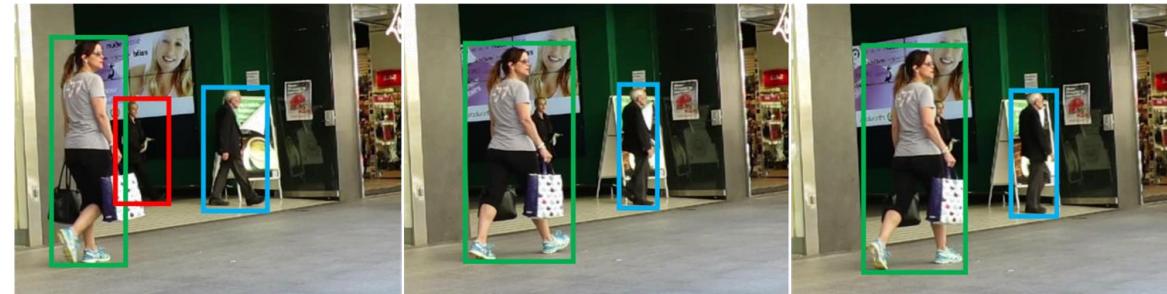
Frame  $t_2$



Frame  $t_3$



(a) detection boxes



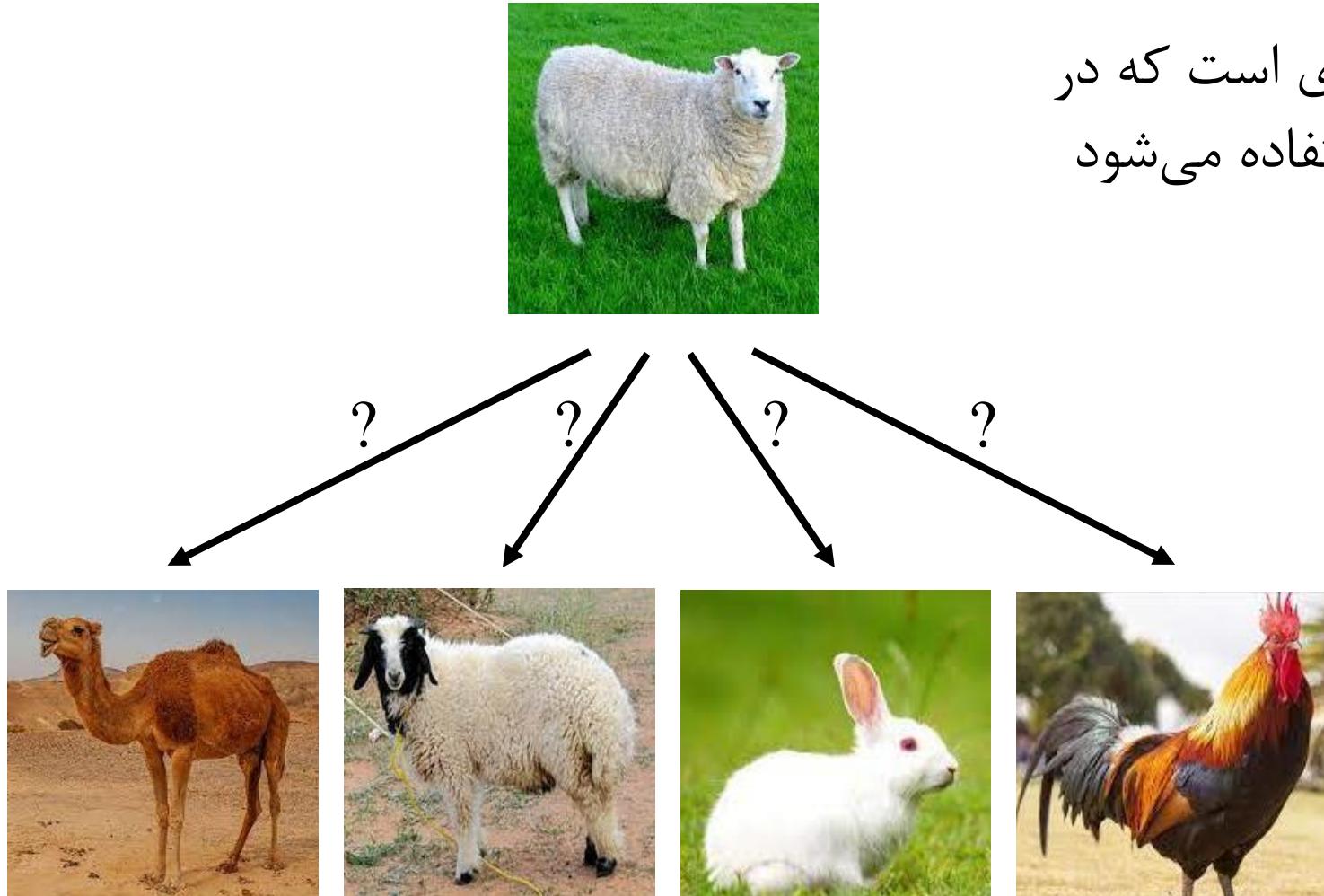
(b) tracklets by associating high score detection boxes



(c) tracklets by associating every detection box

# یادگیری تک نمونه (One Shot Learning)

- یادگیری تک نمونه یک مسئله دسته بندی است که در آن تنها از یک نمونه برای هر کلاس استفاده می شود



Test Image

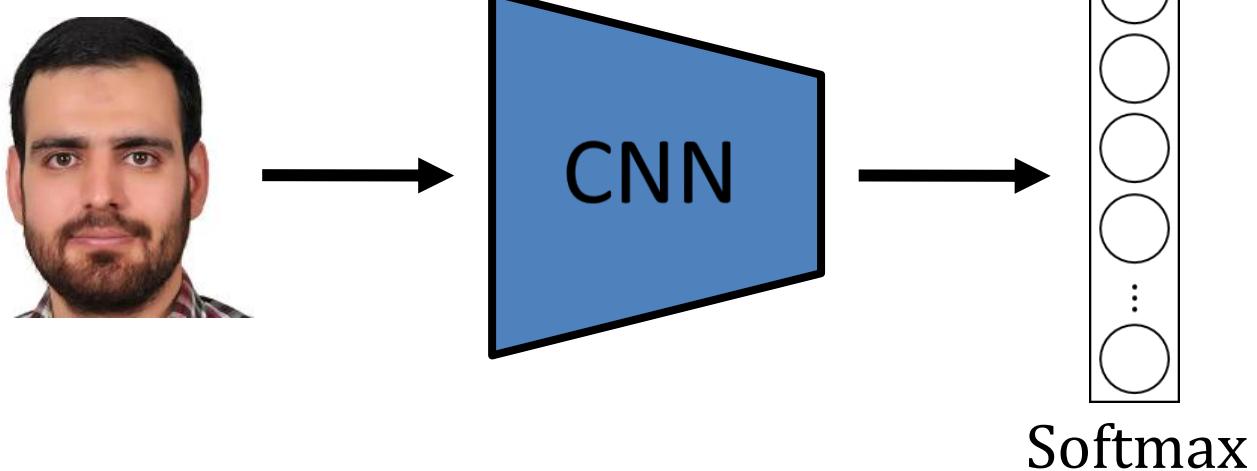


Support Set

ਣ	ਹ	ਫ	ਠ	ਕ
ਕੁ	ਰ	ਥ	ਮ	ਅ
ਲ	ਖ	ਲੁ	ਥ	ਜ
ਸ	ਕ	ਤ	ਸ	ਚ
ਤ	ਪ	ਡ	ਏ	ਧ

# بازشناسی چهره تکنمونه

- یادگیری تنها از یک تصویر برای بازشناسی افراد
- آیا ConvNet + Softmax برای بازشناسی چهره تکنمونه مناسب است?
  - داده کافی برای آموزش یک شبکه عصبی قوی وجود ندارد
  - اگر یک نفر جدید اضافه شود؟
- بجای آن، یک تابع "شباخت" را آموزش می‌دهیم

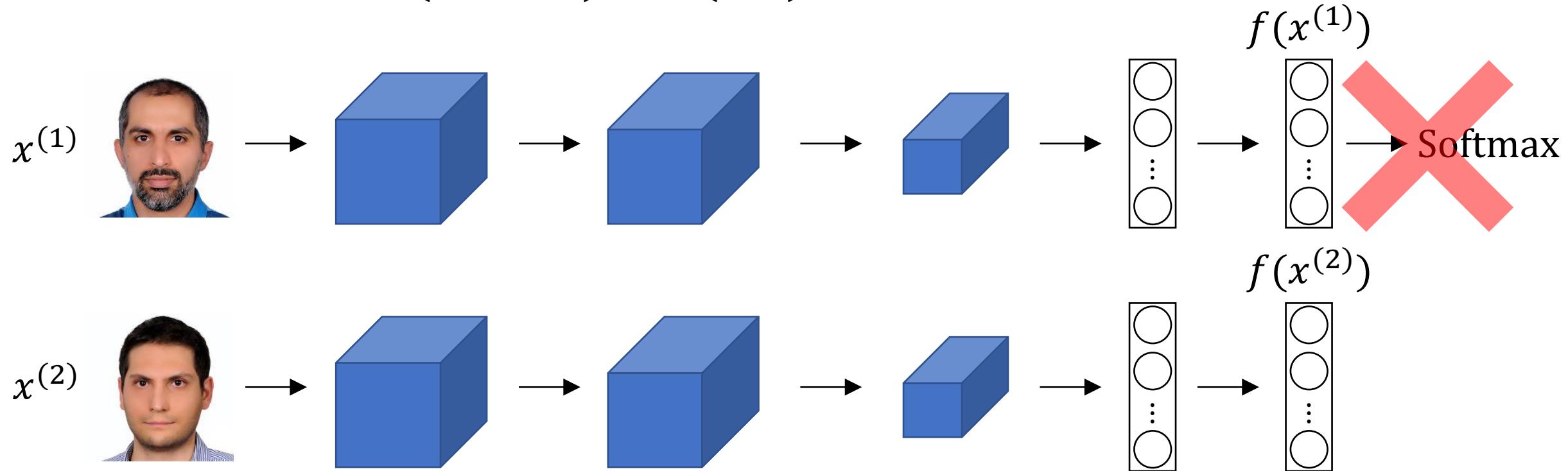


# یادگیری تابع شباخت



• درجه تفاوت بین دو تصویر  $d(img1, img2)$

$$d(x^{(1)}, x^{(2)}) = \|f(x^{(1)}) - f(x^{(2)})\|_2^2$$



# یادگیری تابع شباخت

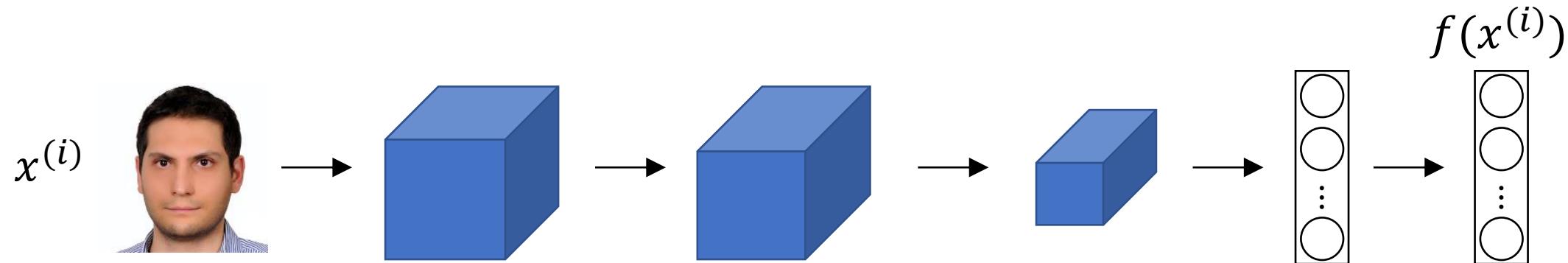
•  $d(img1, img2)$ : درجه تفاوت بین دو تصویر

$$d(x^{(1)}, x^{(2)}) = \|f(x^{(1)}) - f(x^{(2)})\|_2^2$$

• پارامترهای شبکه آموزش می‌بینند تا:

- اگر  $x^{(i)}$  و  $x^{(j)}$  مربوط به یک نفر باشند،  $d(x^{(i)}, x^{(j)})$  عدد کوچکی باشد

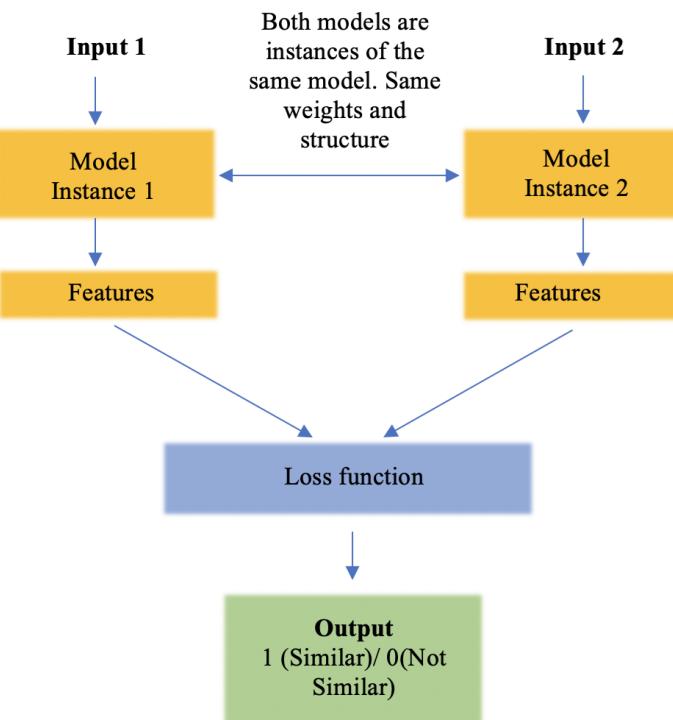
- اگر  $x^{(i)}$  و  $x^{(j)}$  مربوط به افراد متفاوتی باشند،  $d(x^{(i)}, x^{(j)})$  عدد بزرگی باشد



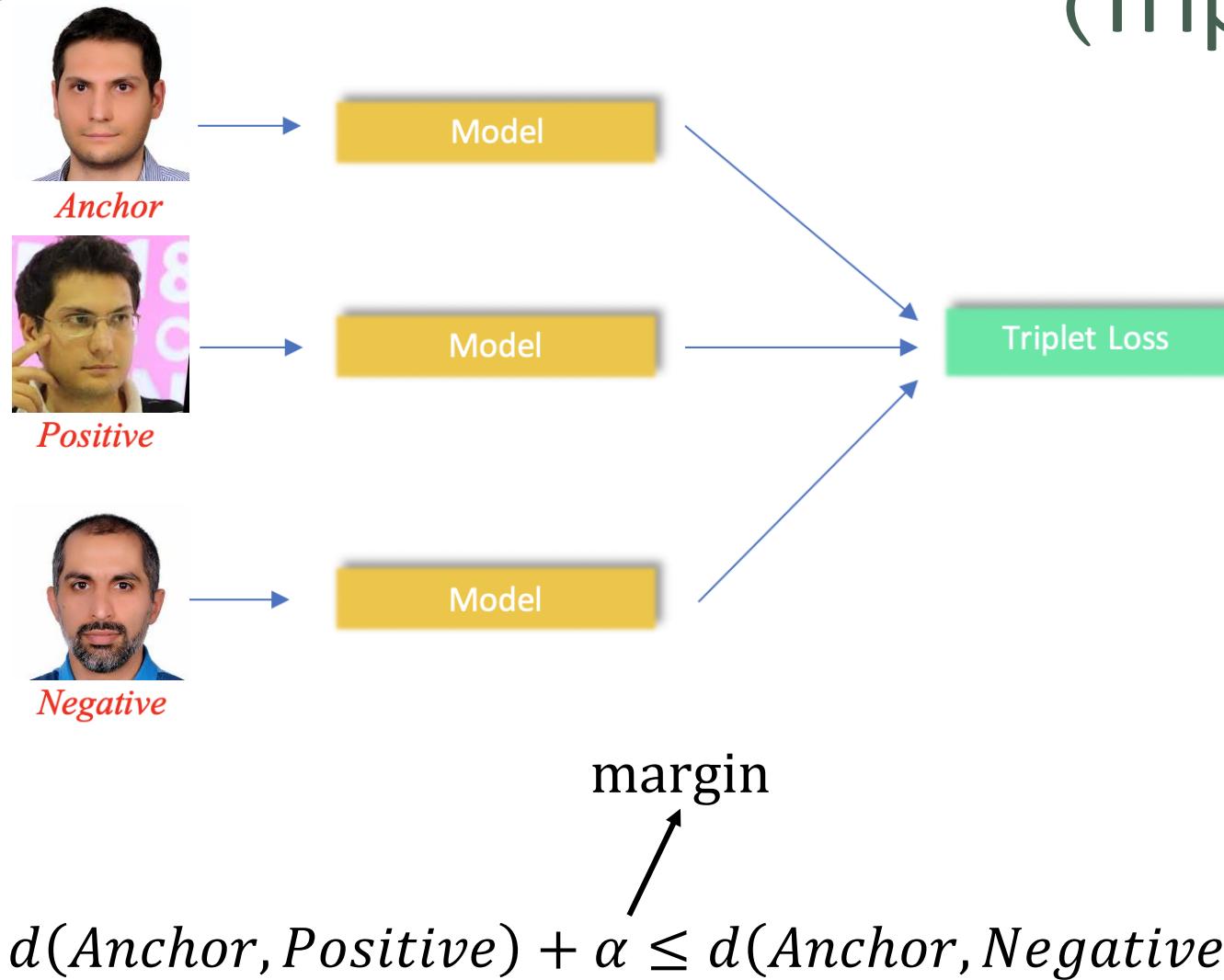
# شبکه Siamese

$x^{(1)}$	$x^{(2)}$	$y$
		0
		1
		0
		1

- یک شبکه Siamese کلاسی از شبکه‌های عصبی است که شامل یک یا چند شبکه یکسان است



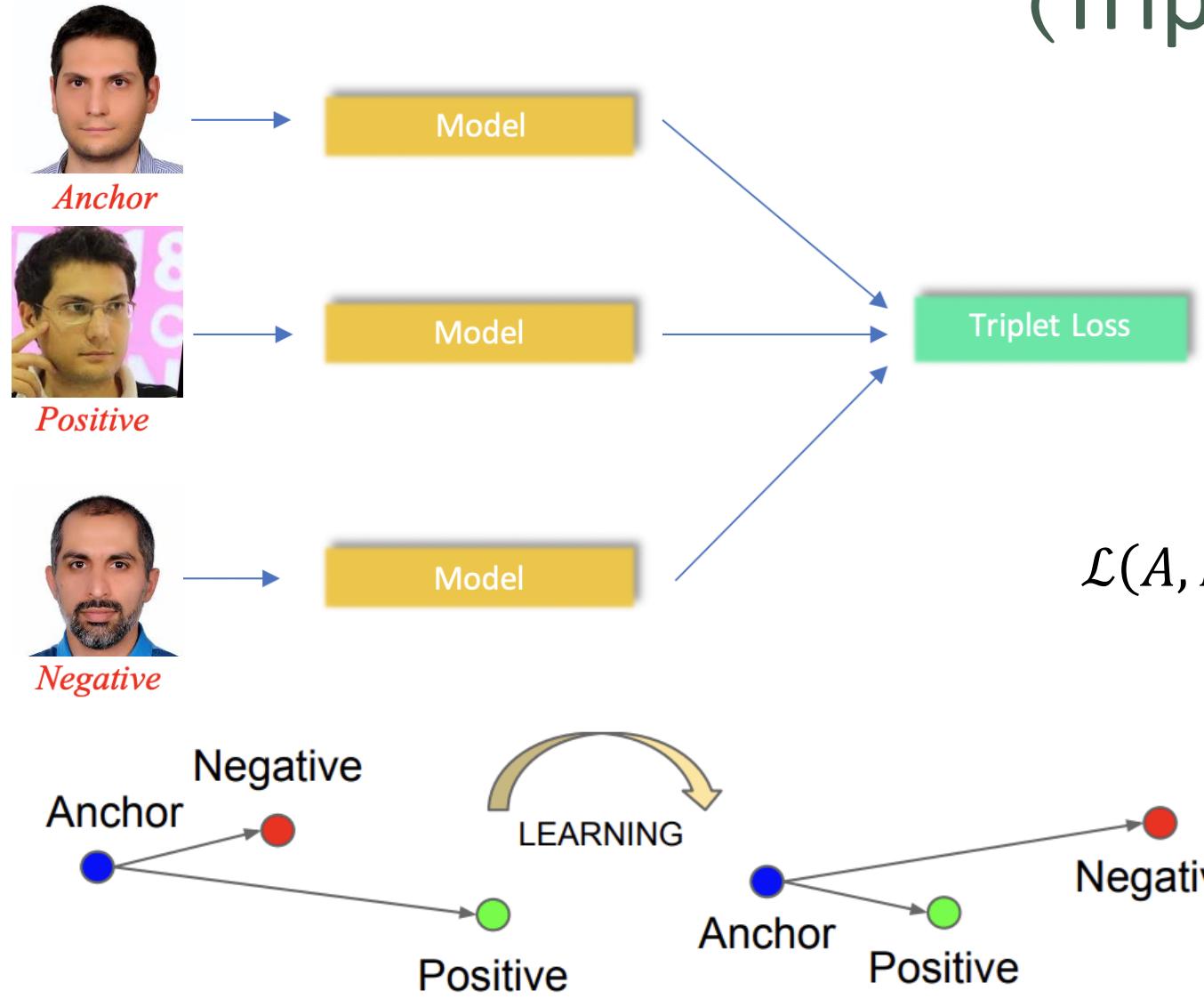
# تابع ضرر سه تایی (Triplet Loss)



- مدل سه ورودی می‌گیرد:  
Negative و Positive از Anchor -



# تابع ضرر سه‌تایی (Triplet Loss)



$$J = \sum_{i=1}^M \mathcal{L}(A^{(i)}, P^{(i)}, N^{(i)})$$

# تابع ضرر سه‌تایی (Triplet Loss)

$$d(A, P) + \alpha \leq d(A, N)$$

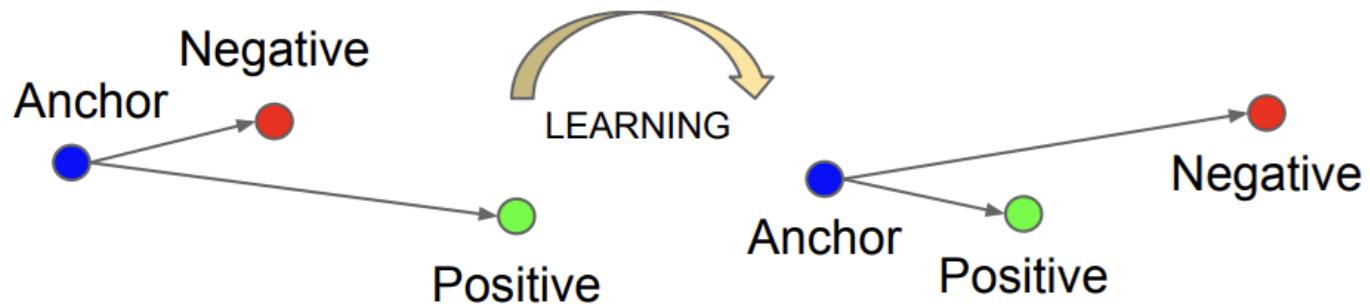
$$d(A, P) - d(A, N) + \alpha \leq 0$$

$$\mathcal{L}(A, P, N) = \max(d(A, P) - d(A, N) + \alpha, 0)$$

$$J = \sum_{i=1}^M \mathcal{L}(A^{(i)}, P^{(i)}, N^{(i)})$$

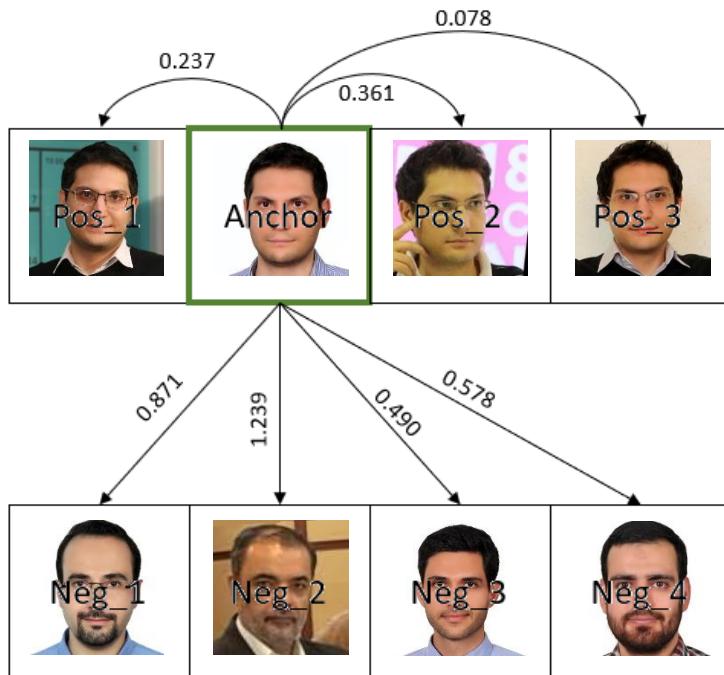
- برای نمونه‌های مربوط به یک کلاس، جانمایی‌هایی تولید می‌شود که فاصله کمی داشته باشند

- این سیستم مستقیماً یک نگاشت از تصاویر چهره به فضای فشرده اقلیدسی را می‌آموزد که در آن فاصله‌ها مستقیماً با معیار تشابه چهره‌ها مطابقت دارند



# انتخاب سه تایی ها

- سه تایی هایی که برای آموزش مدل استفاده می شوند باید با دقت انتخاب شوند
- اگر به صورت تصادفی انتخاب شوند
  - شرط  $d(A,P) + \alpha \leq d(A,N)$  به راحتی برآورده می شود
  - تابع ضرر کوچک خواهد بود و به روزرسانی مدل به کندی انجام خواهد شد
- سه تایی ها به صورت آنلاین و بر حسب فاصله فعلی با یکدیگر تولید می شوند
  - به آنها مثبت سخت (hard negative) و منفی سخت (hard positive) گفته می شود



# FaceNet

- CASIA-Webface (10K ids/0.5M images)
  - VGG2 (9K ids/3.31M images)
  - Glint360K (360K ids/17M images)
- 
- LFW (5749 ids/13233 images/6K pairs)

Model name	LFW accuracy	Training dataset	Architecture
<a href="#">20180408-102900</a>	0.9905	CASIA-WebFace	<a href="#">Inception ResNet v1</a>
<a href="#">20180402-114759</a>	0.9965	VGGFace2	<a href="#">Inception ResNet v1</a>

خدا با چنان کن سرانجام کار  
تو خشنود باشی و ما رستگار