
بسم الله الرحمن الرحيم

مبانی بینایی کامپیوتر

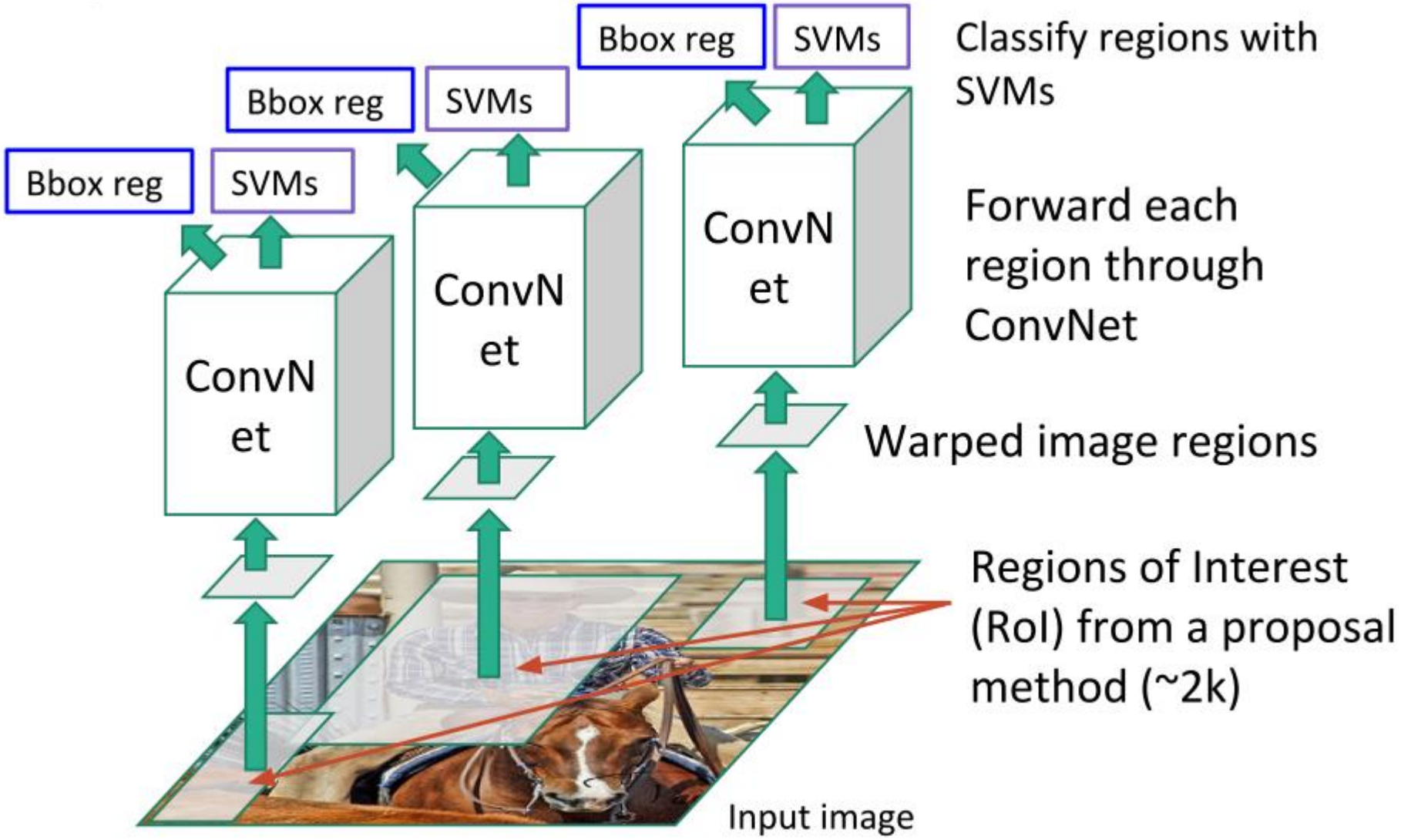
مدرس: محمدرضا محمدی

۱۳۹۹

تشخیص اشیاء

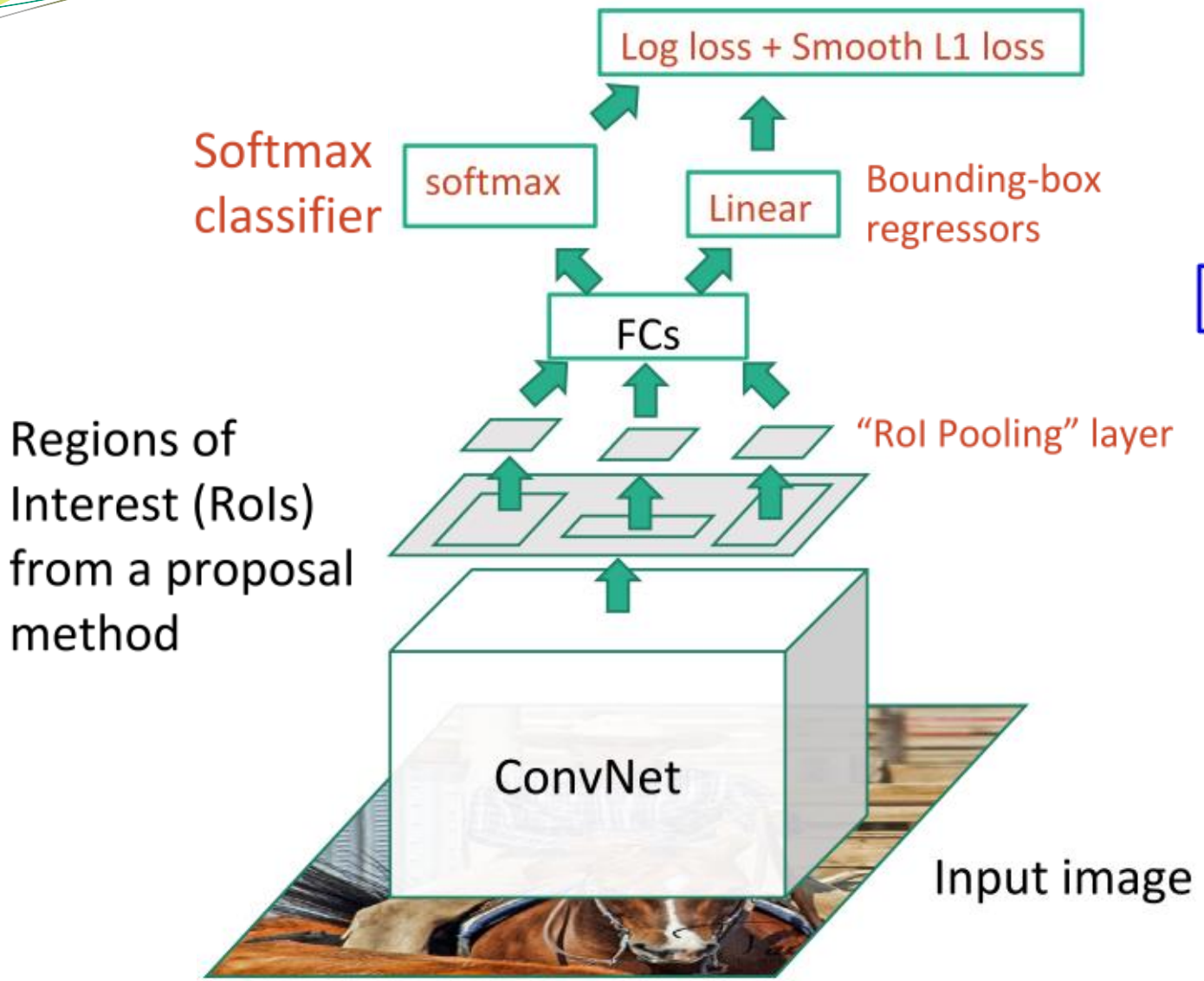
Object Detection

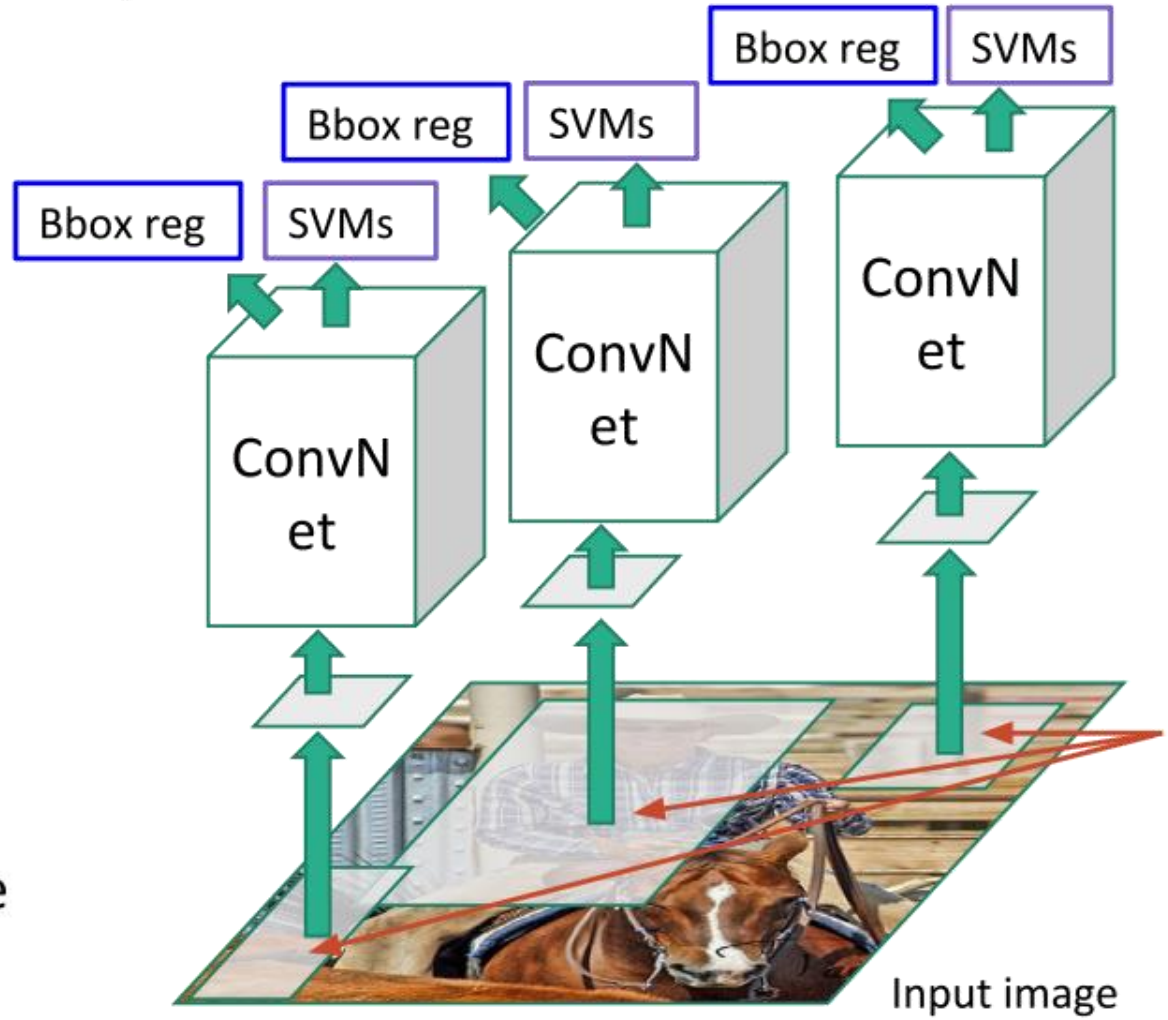
R-CNN



Girshick, Ross, et al. "Region-based convolutional networks for accurate object detection and segmentation." *IEEE transactions on pattern analysis and machine intelligence* (2015)

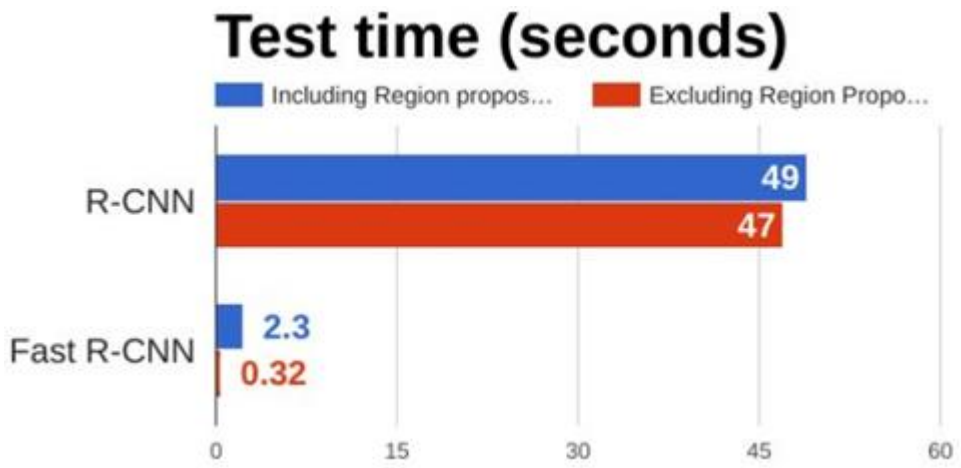
Fast R-CNN



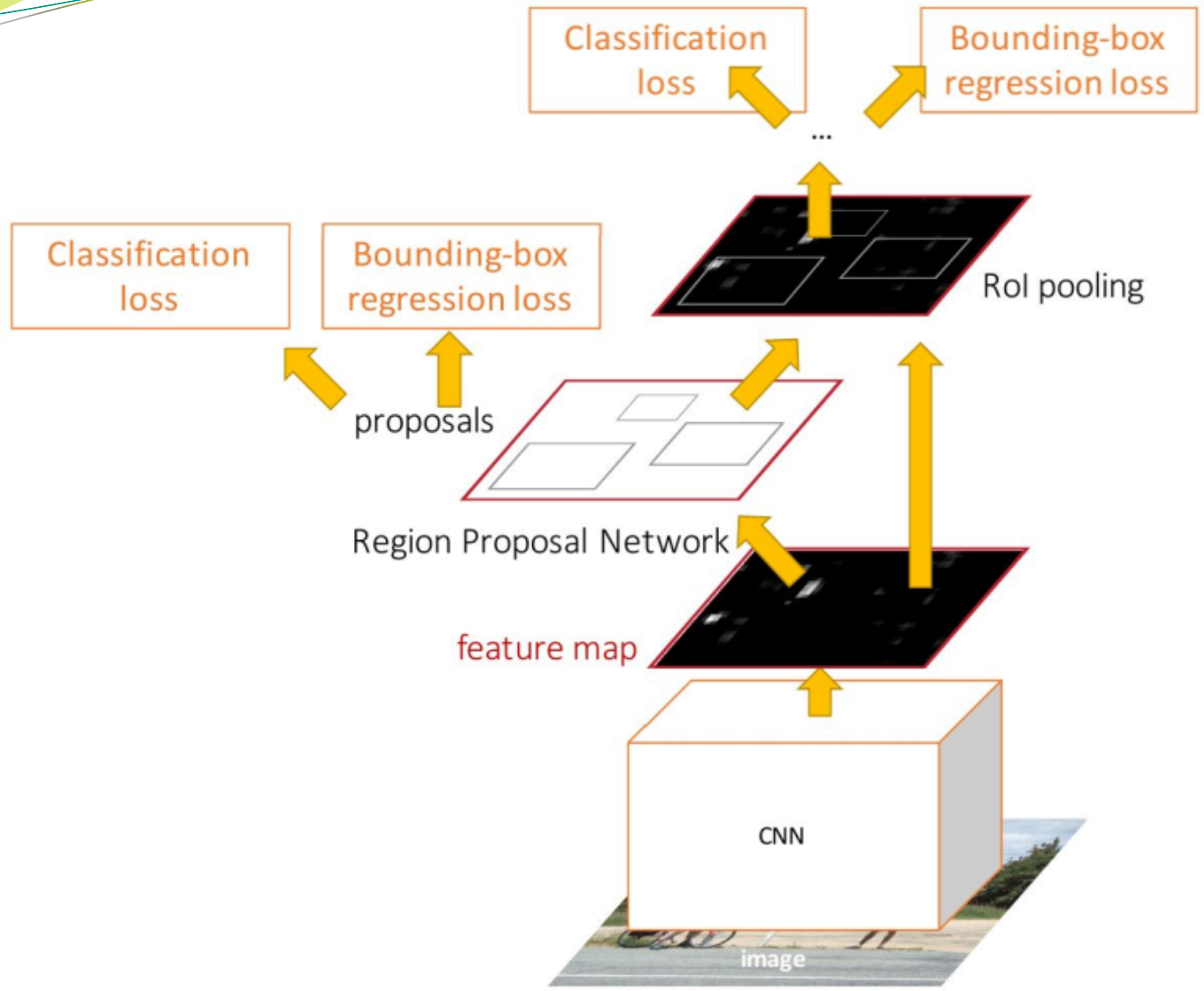


Girshick, Ross. "Fast r-cnn." *Proceedings of the IEEE international conference on computer vision*. (2015).

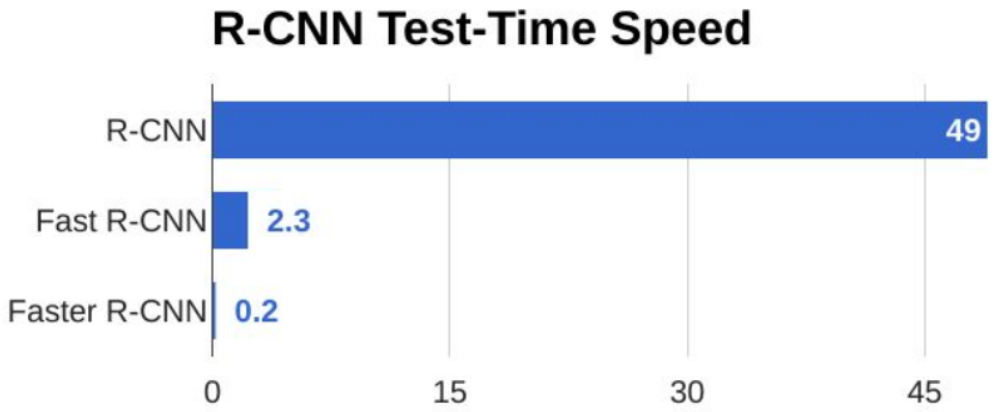
مقایسه زمانی



Faster R-CNN



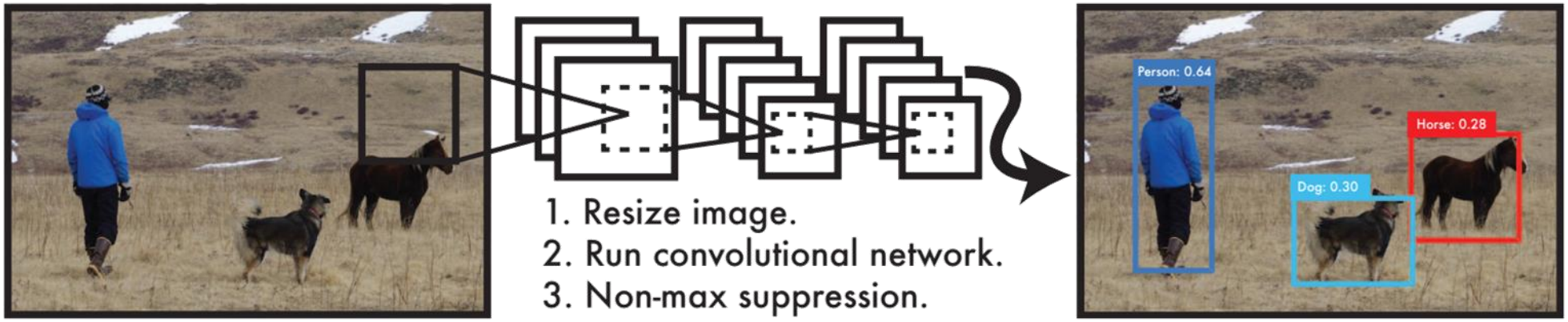
- ناحیه‌های پیشنهادی هم توسط بخشی از شبکه (RPN) تولید می‌شوند
- آموزش شبکه end-to-end انجام می‌شود



Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *IEEE transactions on pattern analysis and machine intelligence* (2016)

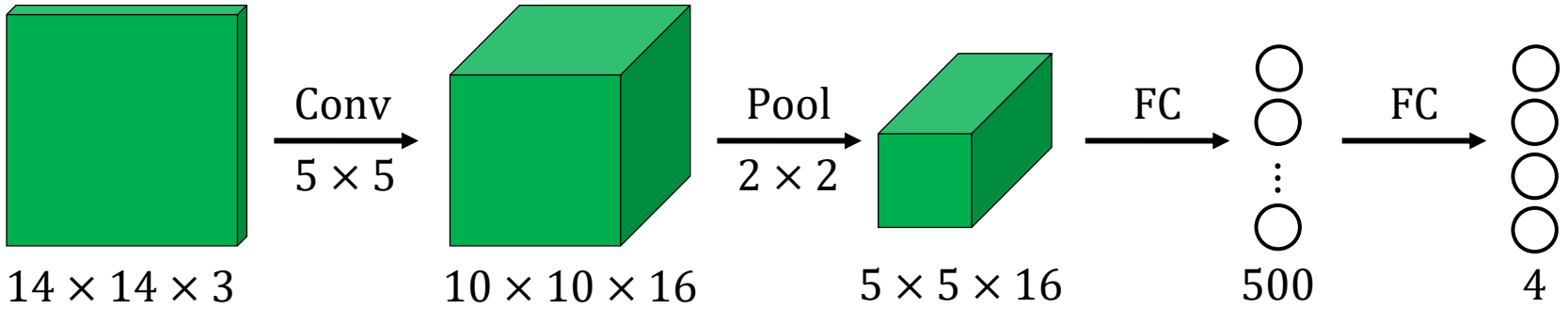
تشخیص اشیاء یک مرحله‌ای

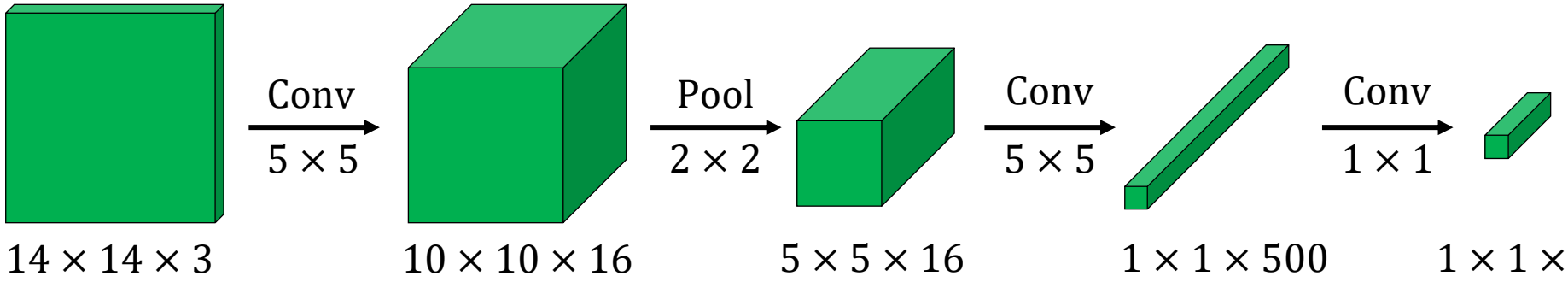
- برخی از الگوریتم‌های جدیدتر مانند YOLO، SSD و RetinaNet بدون بخش مستقیمی برای تولید ناحیه‌های پیشنهادی توسعه یافته‌اند
- این الگوریتم‌ها مبتنی بر پنجره لغزان هستند
  - به نوعی، توسعه روش دسته‌بندی و مکان‌یابی هستند



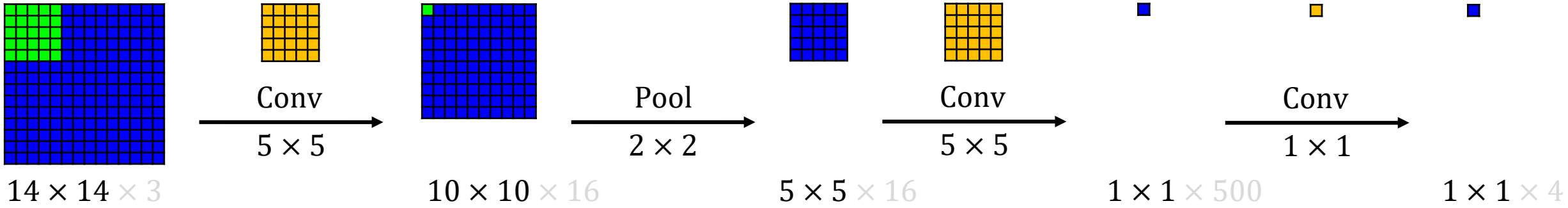
Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2016).

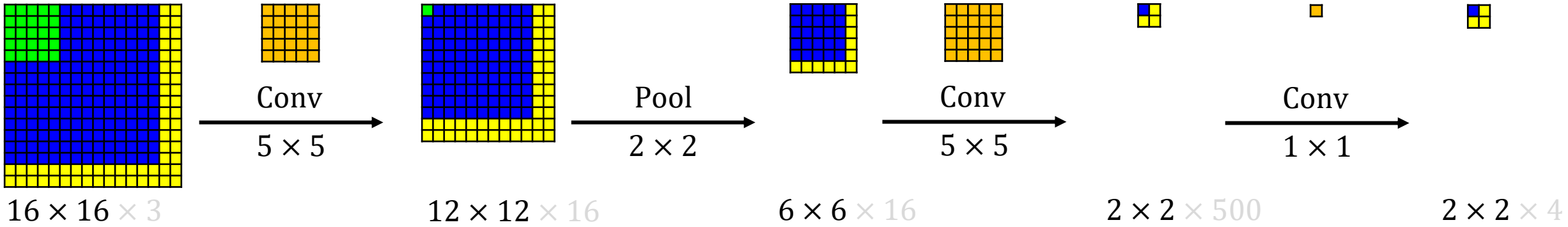
تبدیل لایه‌های FC به Conv



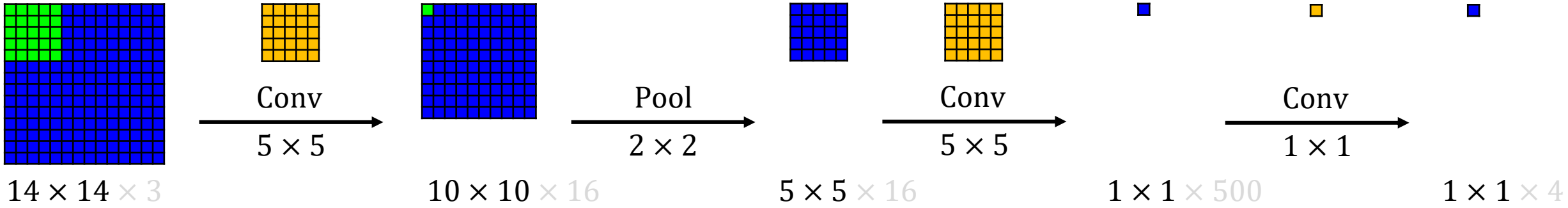


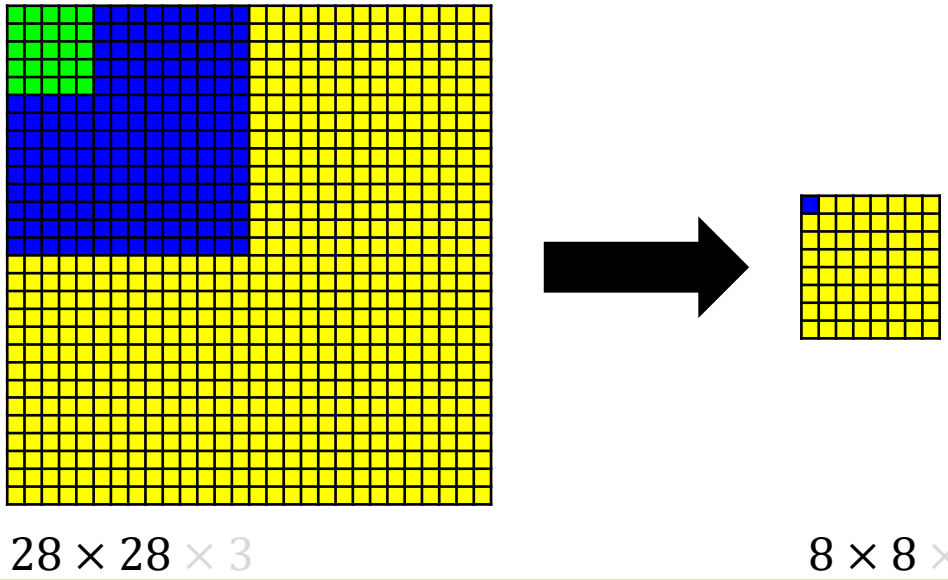
پیاده‌سازی کانولوشنی پنجره لغزان

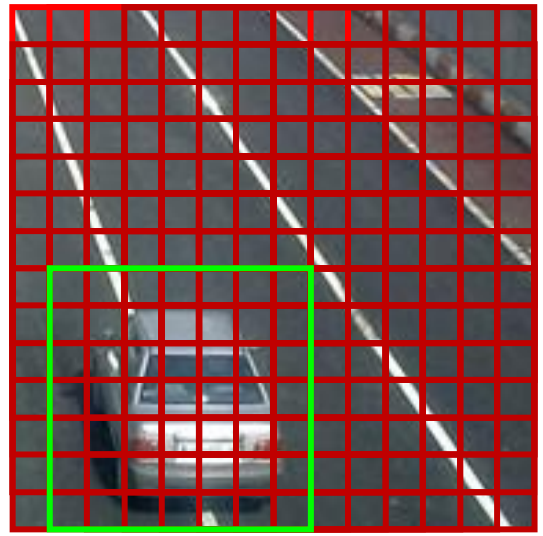




پیاده‌سازی کانولوشنی پنجره لغزان







تطبیق کلیشه

- روش تطبیق کلیشه ساده‌ترین روش برای یافتن یک شیء در تصویر است
- در این روش، یک کلیشه (template) از شیء مورد نظر ساخته شده و مشابهت (یا فاصله) هر ناحیه از تصویر نسبت به آن سنجیده می‌شود

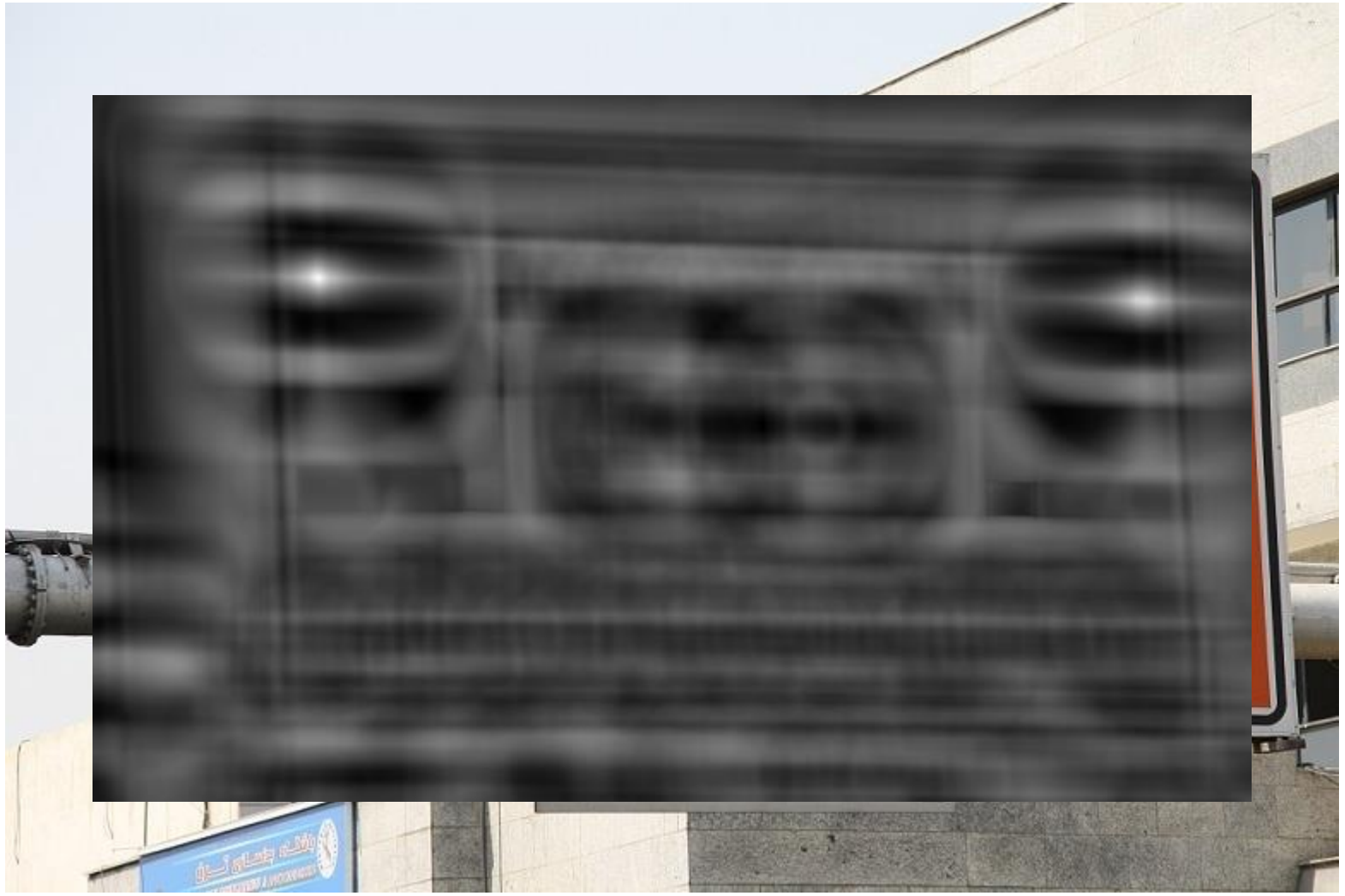
```
result = cv2. matchTemplate(image, templ, method[, mask])

// image:          Image where the search is running. It must be 8-bit or 32-bit floating-point (W × H)
// templ:          Searched template. It must be not greater than the source image and have the same data type (w × h)
// method:         Parameter specifying the comparison method (cv2.TM_SQDIFF, ..., cv2.TM_CCOEFF_NORMED)
// mask:           Optional mask. It must have the same size as templ. If the data type is CV_8U, the mask is interpreted
                   as a binary mask, for data type CV_32F, the mask values are used as weights
// result:         Map of comparison results. It must be single-channel 32-bit floating-point ((W − w + 1) × (H − h + 1))
```

تطبیق کلیشه


کنترل مکانیزه
محدوده طرح ترافیک
۱۷-۶:۳۰
به استثناءِ وسائط نقلیه مجاز و ایام تعطیل
A شهید مطهری

تطبیق کلیشه



تطبیق کلیشه


MARIO
001200
×00
WORLD
1-1
TIME
348


MARIO
001200
×00
WORLD
1-1
TIME
348

تطبیق کلیشه

- method
  - TM_SQDIFF

$$R(x,y) = \sum_{x',y'} (T(x',y') - I(x+x',y+y'))^2$$

  - TM_SQDIFF_NORMED

$$R(x,y) = \frac{\sum_{x',y'} (T(x',y') - I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}}$$

  - TM_CCORR

$$R(x,y) = \sum_{x',y'} (T(x',y') \cdot I(x+x',y+y'))$$

  - TM_CCORR_NORMED

$$R(x,y) = \frac{\sum_{x',y'} (T(x',y') \cdot I(x+x',y+y'))}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}}$$

  - TM_CCOEFF

$$R(x,y) = \sum_{x',y'} (T'(x',y') \cdot I'(x+x',y+y'))$$

  - TM_CCOEFF_NORMED

$$R(x,y) = \frac{\sum_{x',y'} (T'(x',y') \cdot I'(x+x',y+y'))}{\sqrt{\sum_{x',y'} T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2}}$$

$$T'(x',y') = T(x',y') - 1/(w \cdot h) \cdot \sum_{x'',y''} T(x'',y'')$$

$$I'(x+x',y+y') = I(x+x',y+y') - 1/(w \cdot h) \cdot \sum_{x'',y''} I(x+x'',y+y'')$$

خواندن پلاک

- برای خواندن کاراکترهای پلاک، می‌توان تطبیق کلیشه هر کاراکتر با تصویر را محاسبه کرد

خواندن پلاک

- برای خواندن کاراکترهای پلاک، می‌توان تطبیق کلیشه هر کاراکتر با تصویر را محاسبه کرد

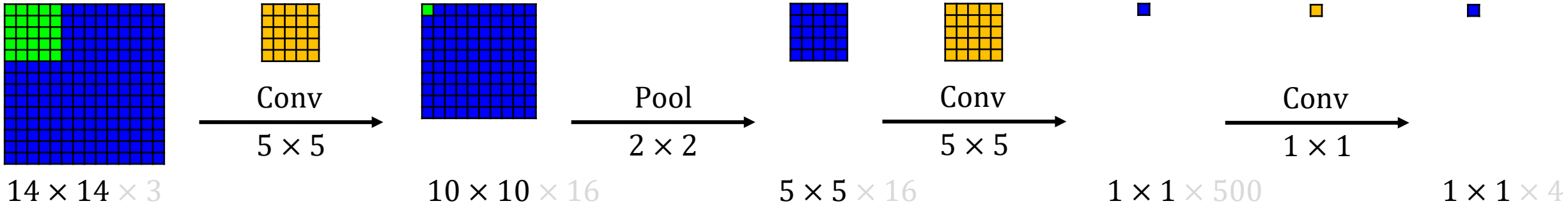
خواندن پلاک

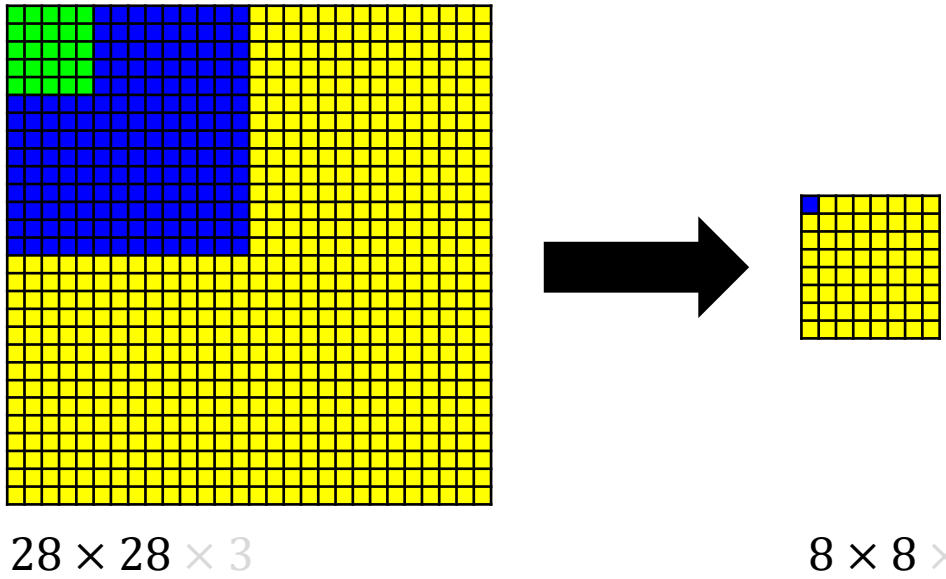
- برای خواندن کاراکترهای پلاک، می‌توان تطبیق کلیشه هر کاراکتر با تصویر را محاسبه کرد
- سپس، مکان‌هایی که نتیجه تطبیق کلیشه برای آنها بیش از حدی باشد را به عنوان مکان آن کاراکتر در نظر می‌گیریم

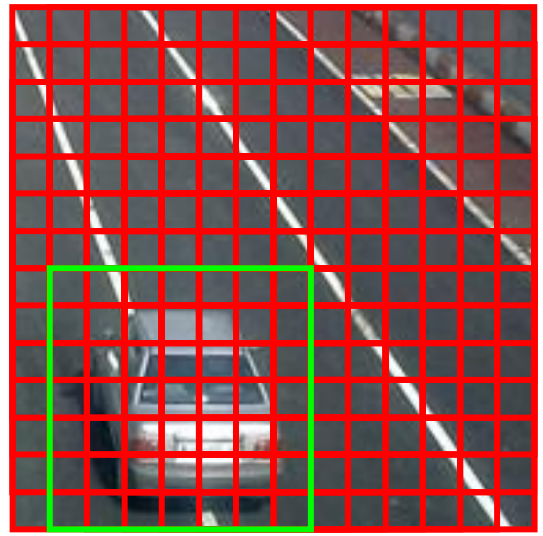
خواندن پلاک

- برای خواندن کاراکترهای پلاک، می‌توان تطبیق کلیشه هر کاراکتر با تصویر را محاسبه کرد
- سپس، مکان‌هایی که نتیجه تطبیق کلیشه برای آنها بیش از حدی باشد را به عنوان مکان آن کاراکتر در نظر می‌گیریم
- برای تشخیص کاراکترهای با ابعاد مختلف می‌توان از کلیشه‌های با ابعاد مختلف استفاده نمود

پیاده‌سازی کانولوشنی پنجره لغزان

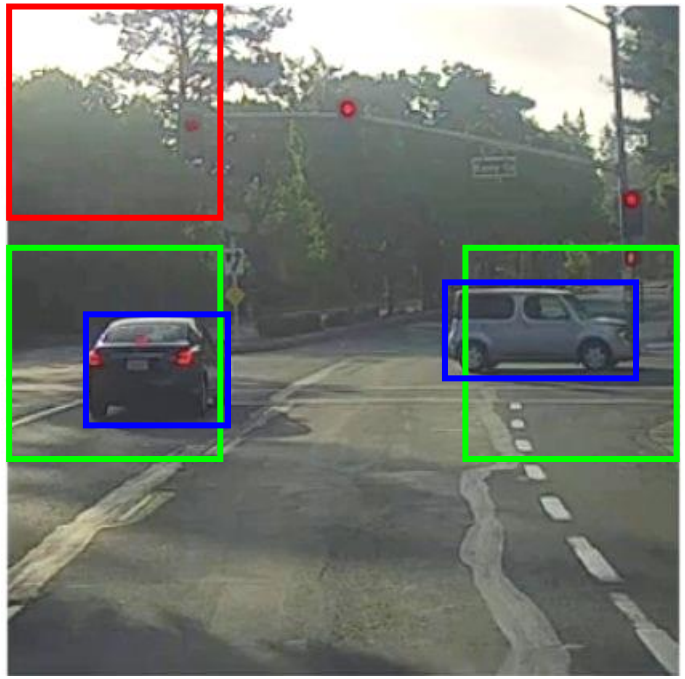






تخمین محدوده

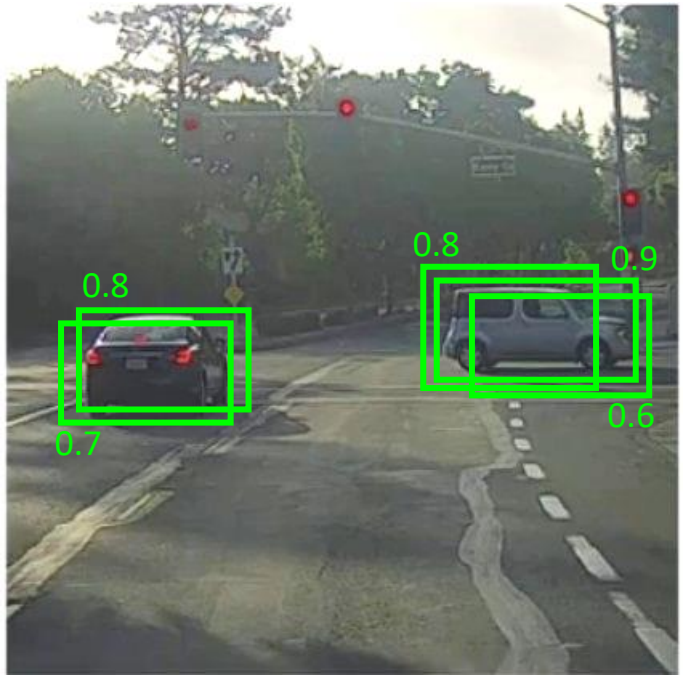
- برای هر نقطه علاوه بر امتیاز کلاس‌ها، می‌توان محدوده دقیق‌تر شیء مورد نظر را آموزش داد



$$\begin{bmatrix} p_c \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \Rightarrow \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

حذف مقادیر غیربیشینه

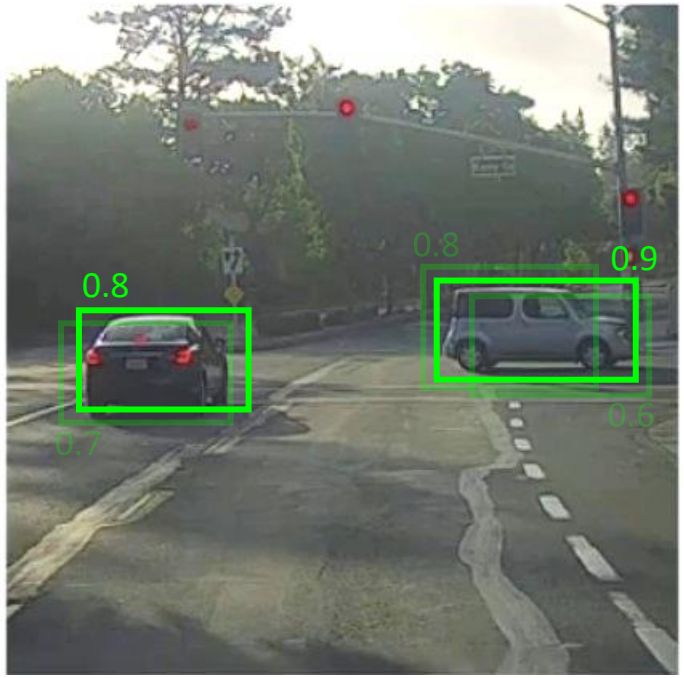
- ممکن است یک شیء در چند نقطه تشخیص داده شود
- از میان ناحیه‌های دارای اشتراک زیاد، یکی انتخاب می‌شود



- پاسخ‌هایی که دارای $p_c < 0.5$ هستند حذف می‌شوند
- تکرار
  - بیشترین p_c انتخاب می‌شود
  - پاسخ‌هایی که دارای IoU بیش از ۰.۵ با این پاسخ هستند حذف می‌شوند

حذف مقادیر غیربیشینه

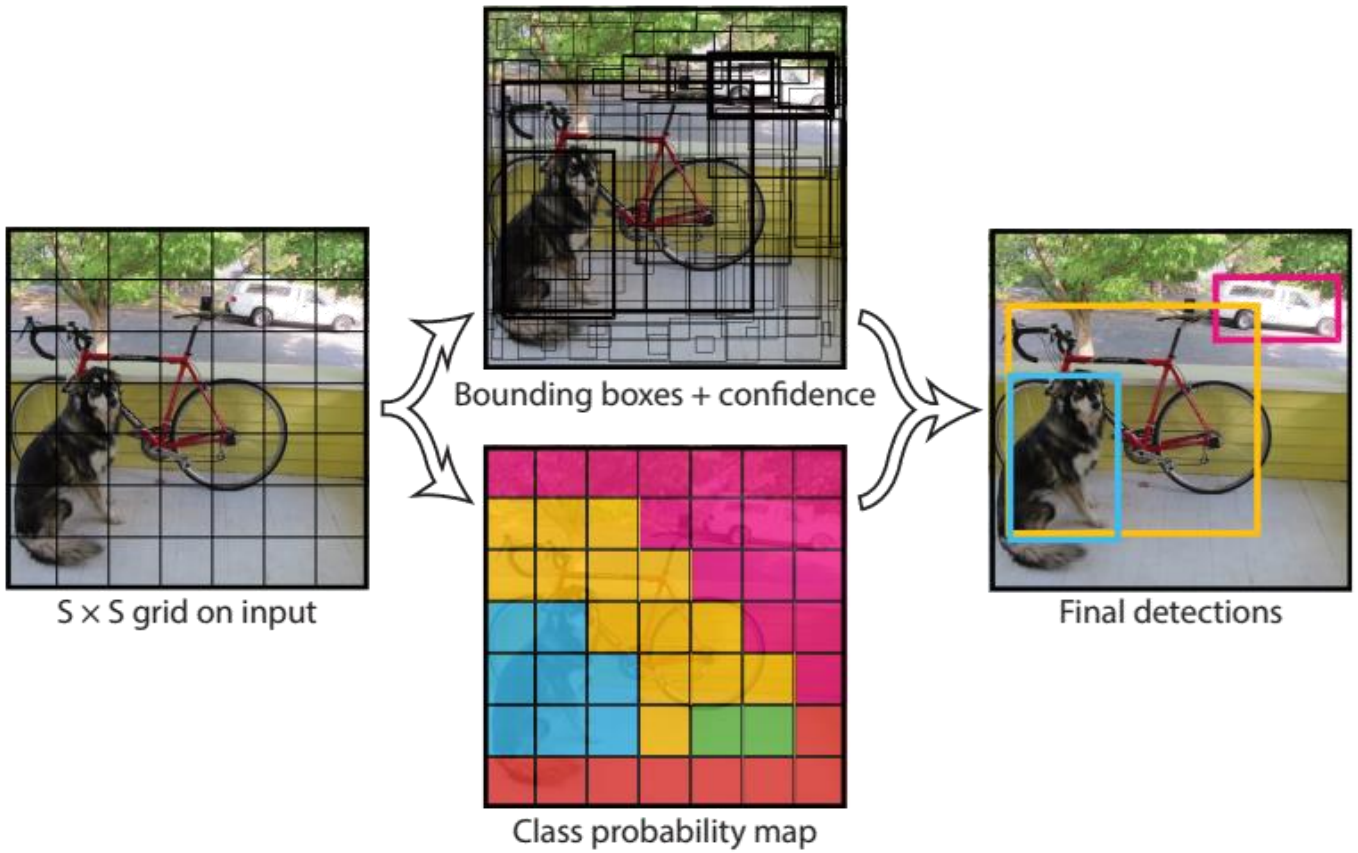
- ممکن است یک شیء در چند نقطه تشخیص داده شود
- از میان ناحیه‌های دارای اشتراک زیاد، یکی انتخاب می‌شود



- پاسخ‌هایی که دارای $p_c < 0.5$ هستند حذف می‌شوند
- تکرار
  - بیشترین p_c انتخاب می‌شود
  - پاسخ‌هایی که دارای IoU بیش از ۰.۵ با این پاسخ هستند حذف می‌شوند

YOLO: You Only Look Once

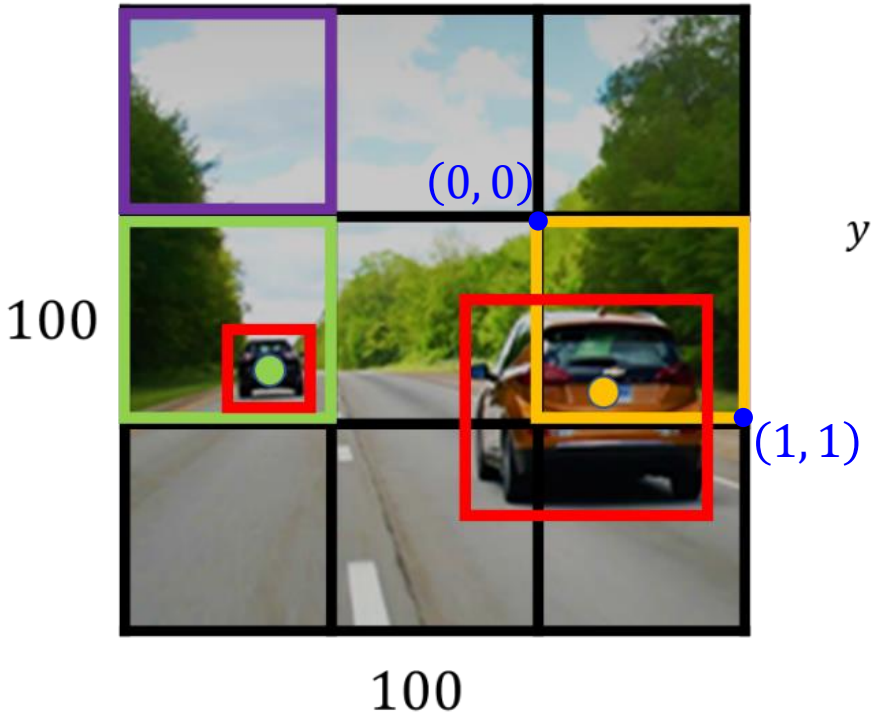
- در روش YOLO، تصویر ورودی به تعدادی ناحیه کوچک تقسیم می‌شود و برای هر ناحیه یک دسته‌بند و یک تابع رگرسیون طراحی می‌شود



Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2016).

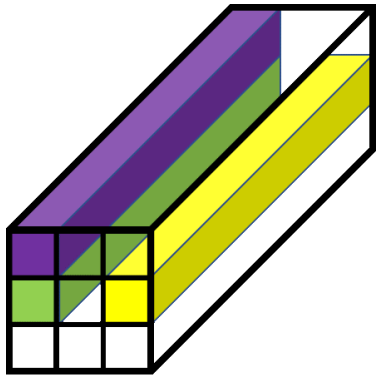
YOLO

Labels for training for each grid cell:



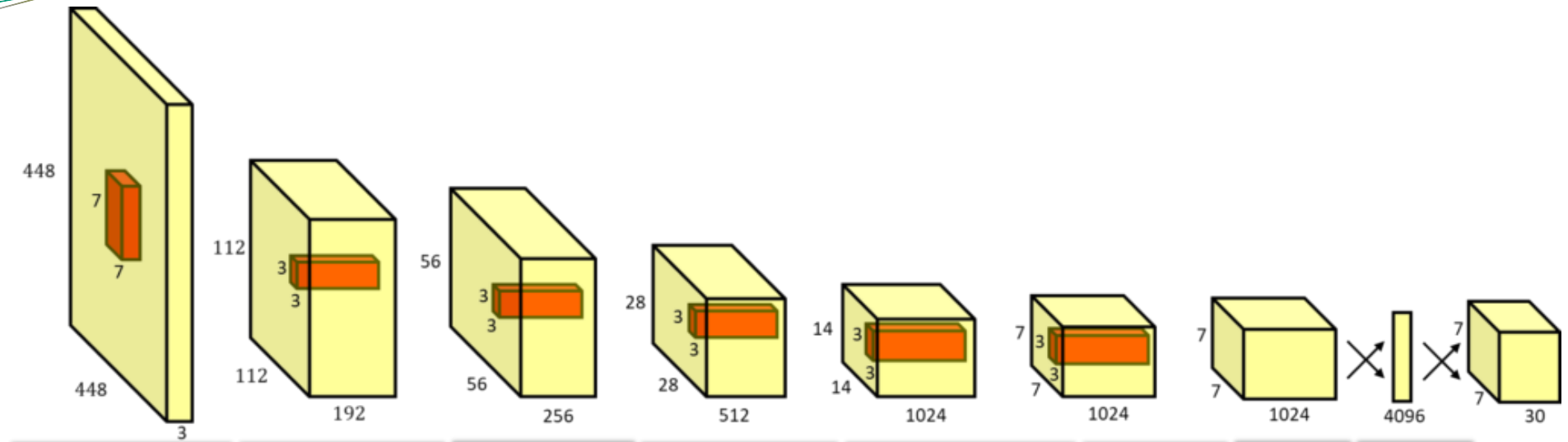
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \quad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

→ bounding box

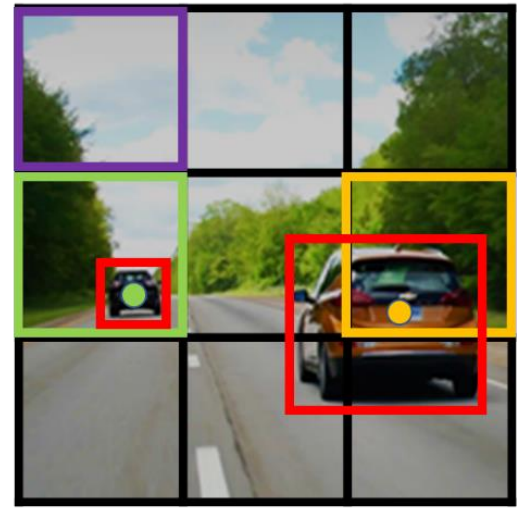


$3 \times 3 \times 8$

YOLO



Conv. Layer $7 \times 7 \times 64 - s - 2$ Maxpool Layer $2 \times 2 - s - 2$	Conv. Layer $3 \times 3 \times 192$ Maxpool Layer $2 \times 2 - s - 2$	Conv. Layers $1 \times 1 \times 128$ $3 \times 3 \times 256$ $1 \times 1 \times 256$ $3 \times 3 \times 512$ Maxpool Layer $2 \times 2 - s - 2$	Conv. Layers $\left.\begin{matrix}1 \times 1 \times 256\\3 \times 3 \times 512\end{matrix}\right\} \times 4$ $1 \times 1 \times 512$ $3 \times 3 \times 1024$ Maxpool Layer $2 \times 2 - s - 2$	Conv. Layers $\left.\begin{matrix}1 \times 1 \times 512\\3 \times 3 \times 1024\end{matrix}\right\} \times 4$ $3 \times 3 \times 1024$ $3 \times 3 \times 1024 - s - 2$	Conv. Layers $3 \times 3 \times 1024$ $3 \times 3 \times 1024$	Conn Layer	Conn Layer



100

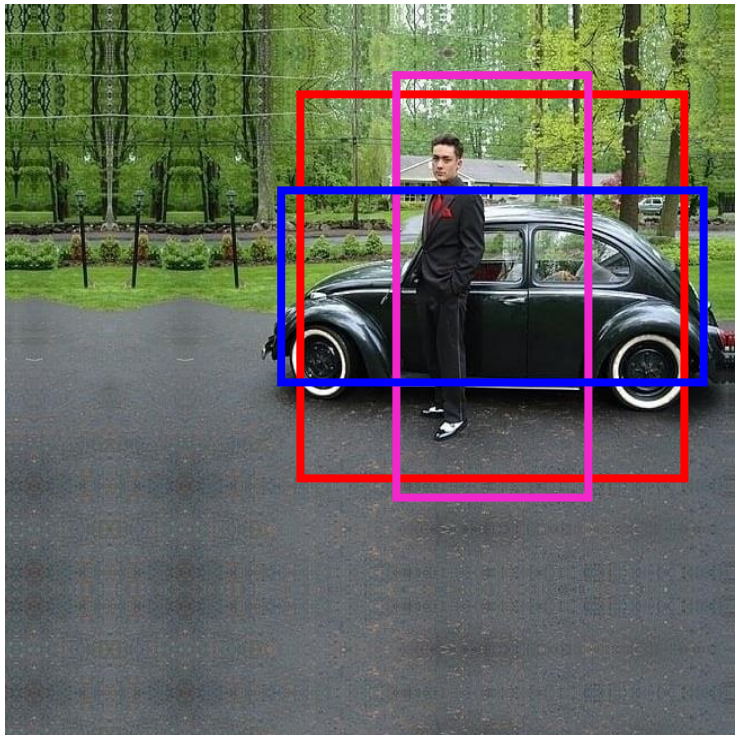
100

YOLO

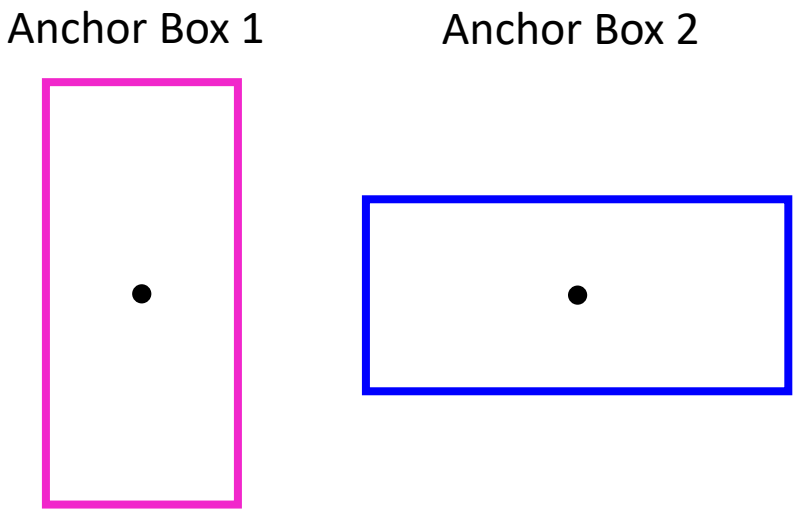
Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Anchor Boxes

- در هر ناحیه تنها یک شیء قابل تشخیص است
- برای اضافه کردن امکان تشخیص چند شیء، می‌توان در هر ناحیه چند خروجی قرار داد



$$\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ c_2 \\ p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

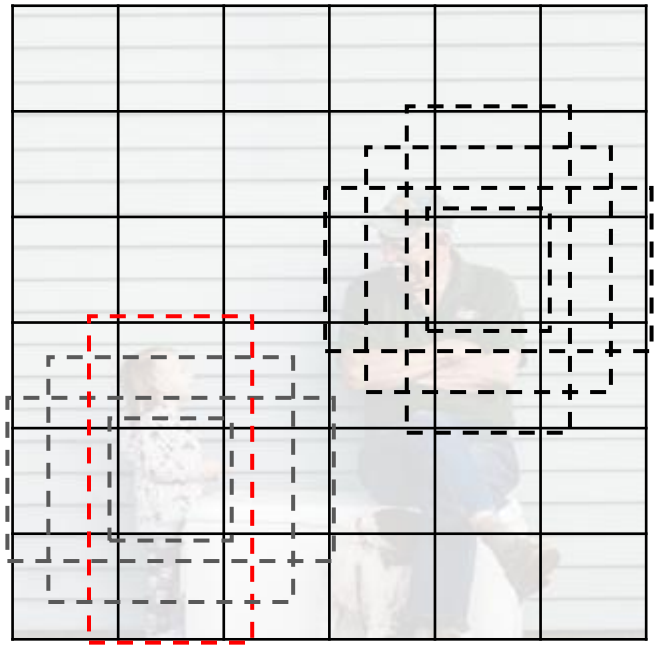


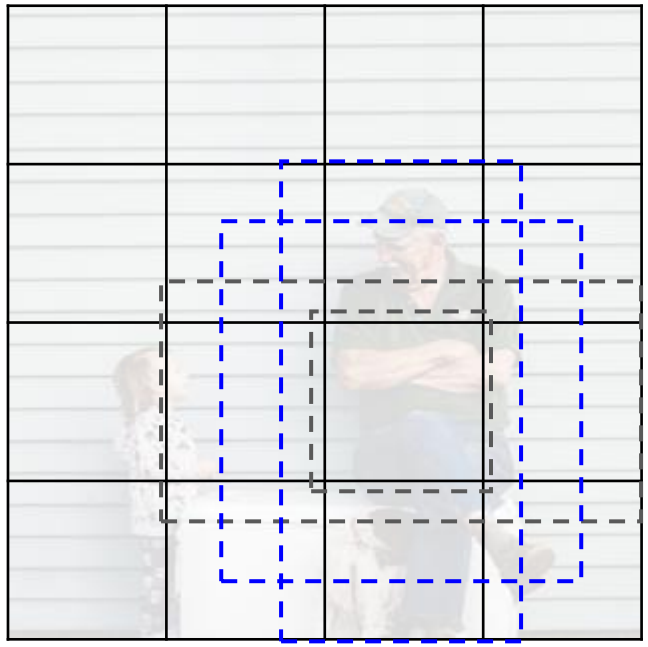
SSD: Single Shot MultiBox Detector

- روش YOLO یک روش آموزش end-to-end شبکه است که نیازی به تولید ناحیه پیشنهادی و تغییر ابعاد آنها ندارد
- سرعت روش YOLO از روش‌هایی که از بخش تولید ناحیه‌های پیشنهادی استفاده می‌کنند بهتر است اما دقت پائین‌تری دارد
- در مقاله SSD بهبودهایی داده شده است که در ضمن افزایش سرعت، دقت نیز افزایش یافته است (به خصوص برای اشیاء کوچک)
- مهمترین نوآوری SSD آن است که برای تشخیص اشیاء و محل آنها از چند لایه استفاده کرده است تا اشیاء با ابعاد مختلف قابل تشخیص باشند

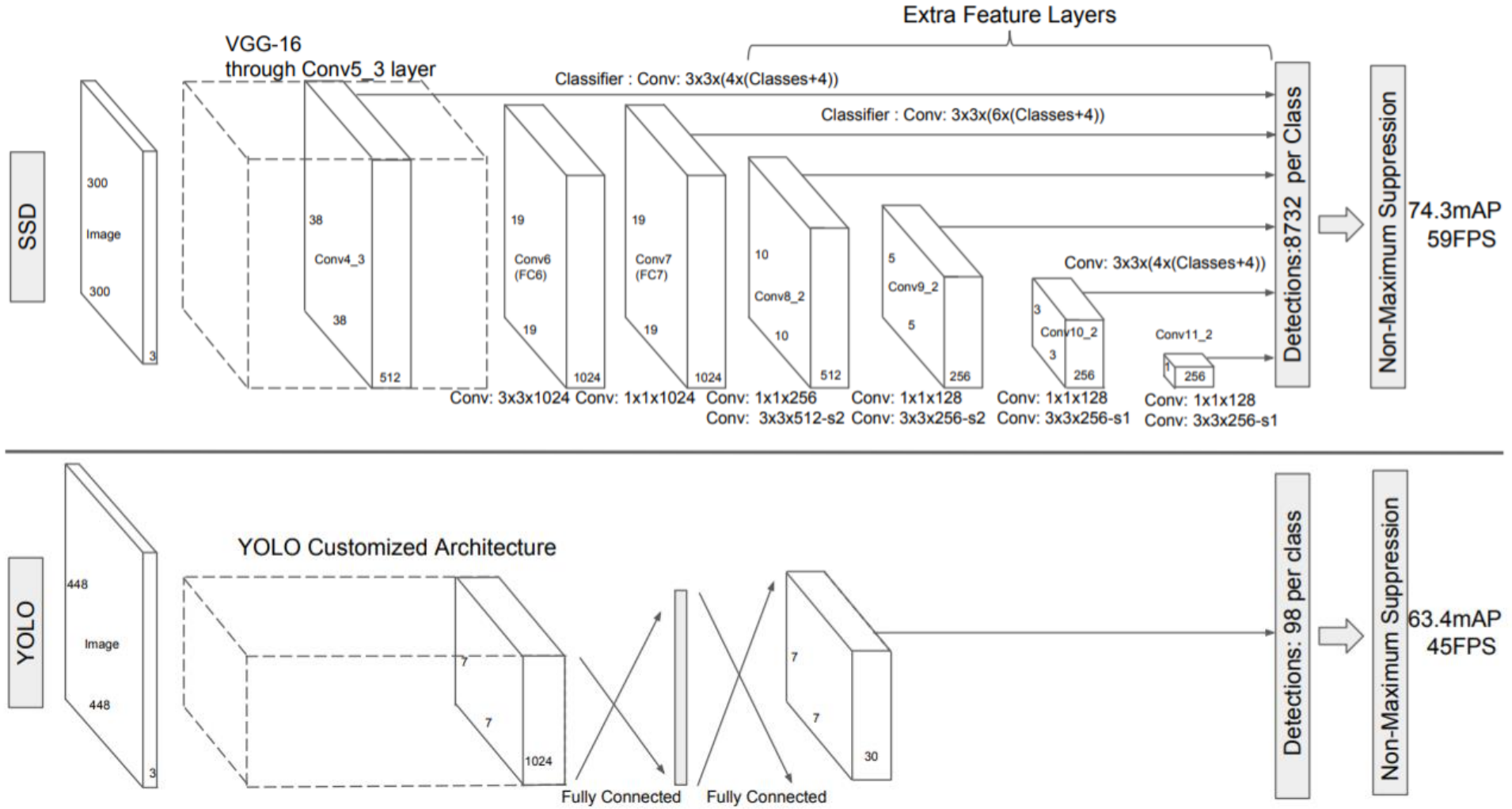
Liu, W., et al. "Ssd: Single shot multibox detector“, *European conference on computer vision*, (2016).

SSD





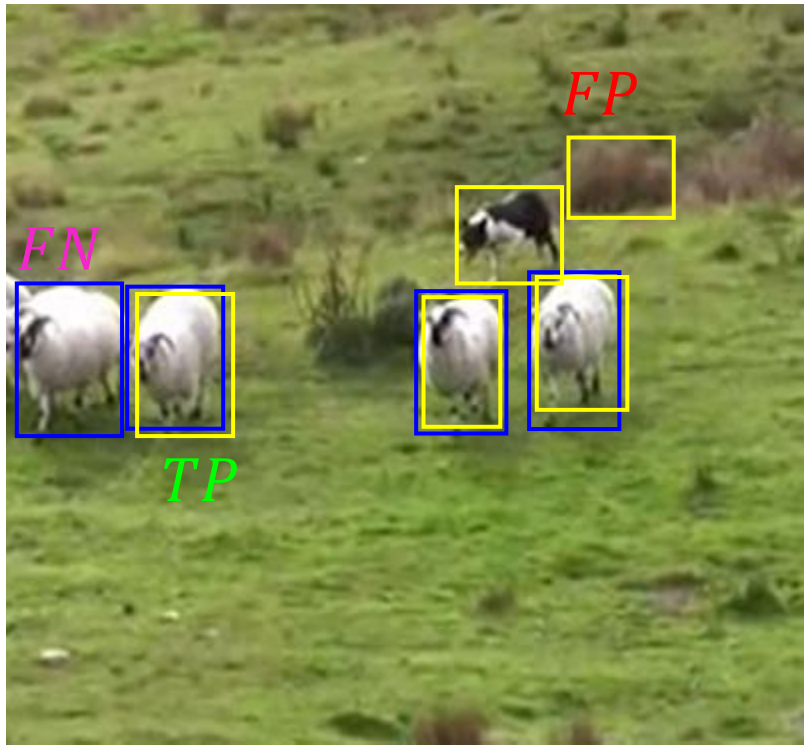
SSD vs YOLO



http://cvmldlcnns.blogspot.com/2017/05/ssd-single-shot-multibox-detector.html

دقت متوسط (AP)

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



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

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

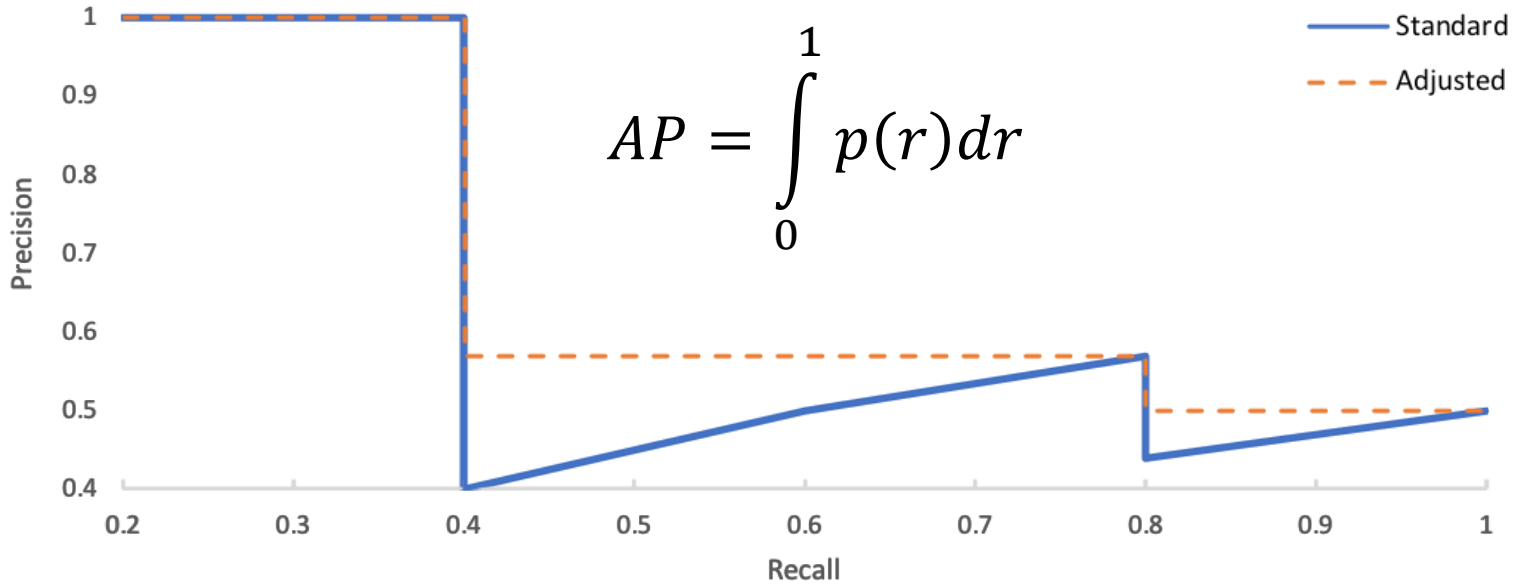
$$F_1 = 2\frac{Precision \times Recall}{Precision + Recall}$$

- 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



$$AP = \int_0^1 p(r)dr$$

https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173

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

Redmon, J., and Farhadi, A., "YOLO9000: better, faster, stronger“, *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2017).

به پایان آمد این دفتر

حکایت همچنان باقیست ...