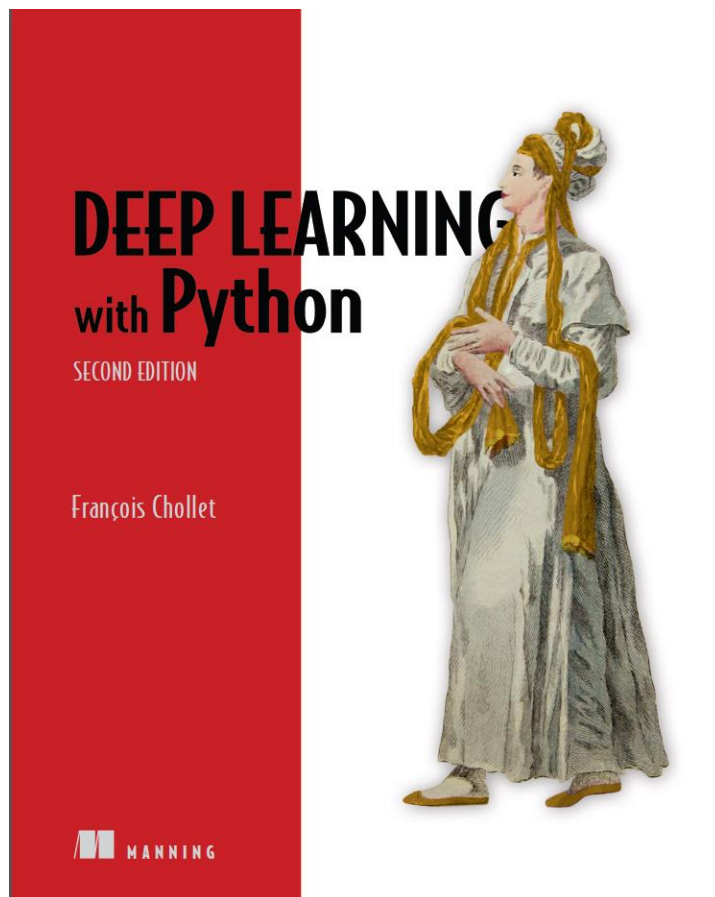


# بینایی کامپیوتر (COMPUTER VISION)



- مبانی یادگیری عمیق (Deep Learning)
- سعید محقق / زمستان 1400

# بینایی کامپیوتر

- بینایی کامپیوتر: استخراج اطلاعات معنی‌دار از تصاویر و ویدیوهای دیجیتال

- مدل مناسب برای آنالیز تصاویر ← شبکه عصبی کانولوشنی (CNN)



Convolutional Neural Network

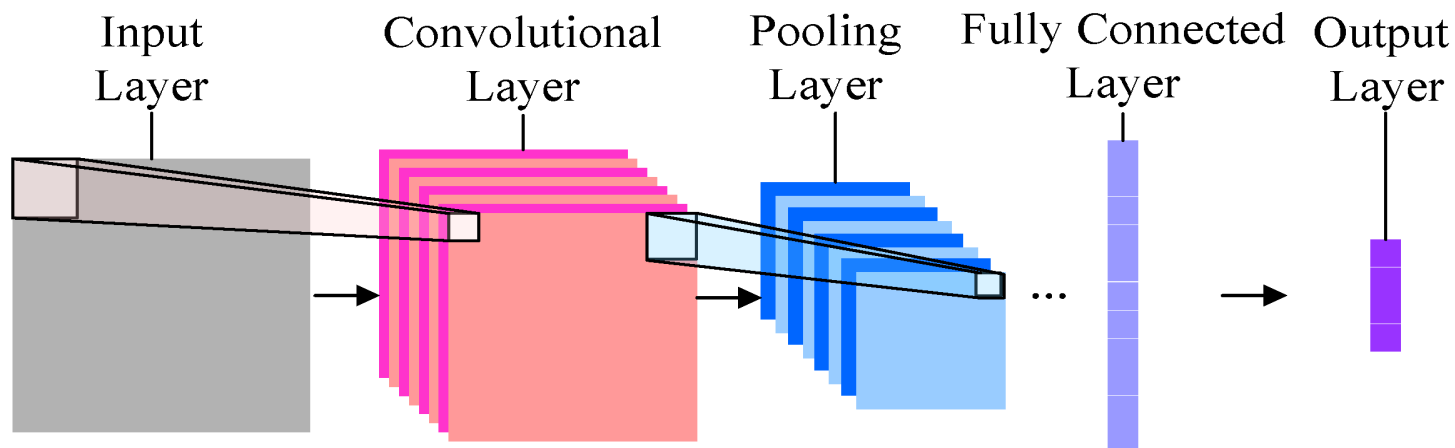
# ویژگی‌های مدل‌های CNN

استفاده از لایه‌های کانولوشن (conv) و کاهش اندازه

استفاده از وزن‌های پنجره‌ای (فیلترها / کرنل‌ها)

استفاده از چند فیلتر در هر لایه (ایجاد چند خروجی متفاوت)

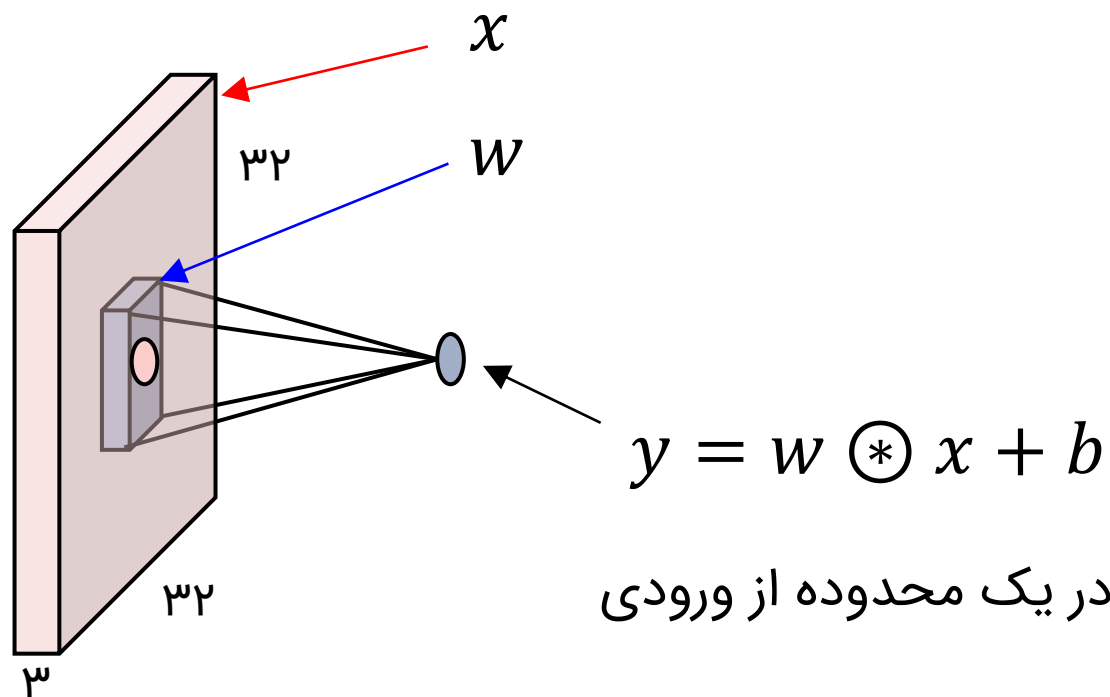
مناسب برای پردازش داده‌های تصویری



## لایه کانولوشن (Conv)

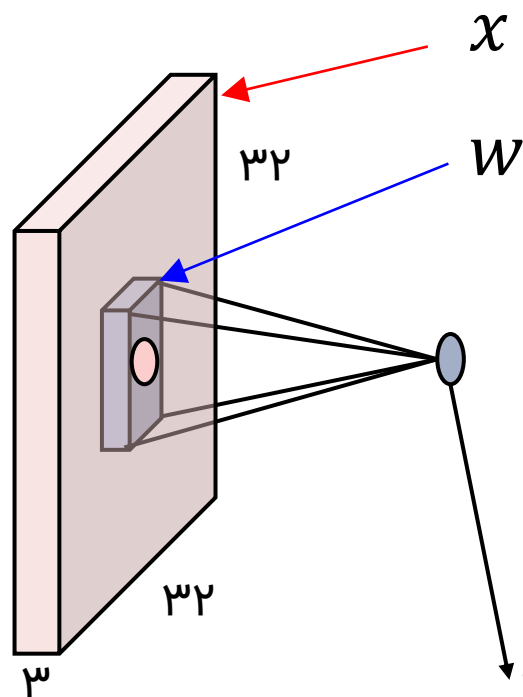
- کانولوشن تصویر ورودی و فیلتر

- مثال کانولوشن در تصاویر: <http://aishack.in/tutorials/image-convolution-examples>



## لایه کانولوشن (Conv)

- کانولوشن تصویر ورودی و فیلتر
- لغزاندن پنجره بر روی تصویر و محاسبه کانولوشن



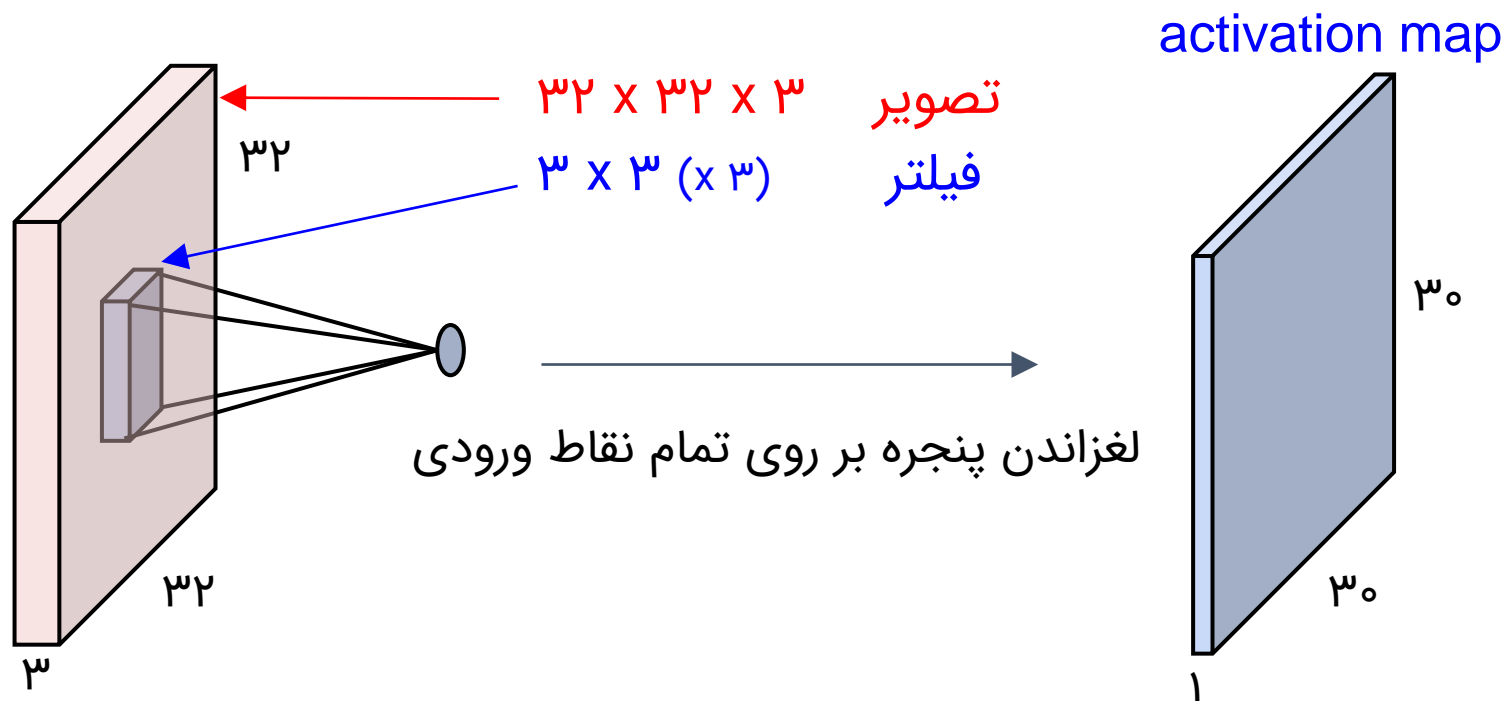
$$\begin{matrix} W & & X \\ \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} & \odot & \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \end{matrix} \rightarrow \begin{matrix} W & & X \\ \begin{bmatrix} i & h & g \\ f & e & d \\ c & b & a \end{bmatrix} & \cdot & \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$(i \cdot 1) + (h \cdot 2) + (g \cdot 3) + (f \cdot 4) + (e \cdot 5) + (d \cdot 6) + (c \cdot 7) + (b \cdot 8) + (a \cdot 9).$$

## لایه کانولوشن (Conv)

- یک فیلتر  $\leftarrow$  یک خروجی

- نمایش نحوه کانولوشن: [https://github.com/vdumoulin/conv\\_arithmetic](https://github.com/vdumoulin/conv_arithmetic)



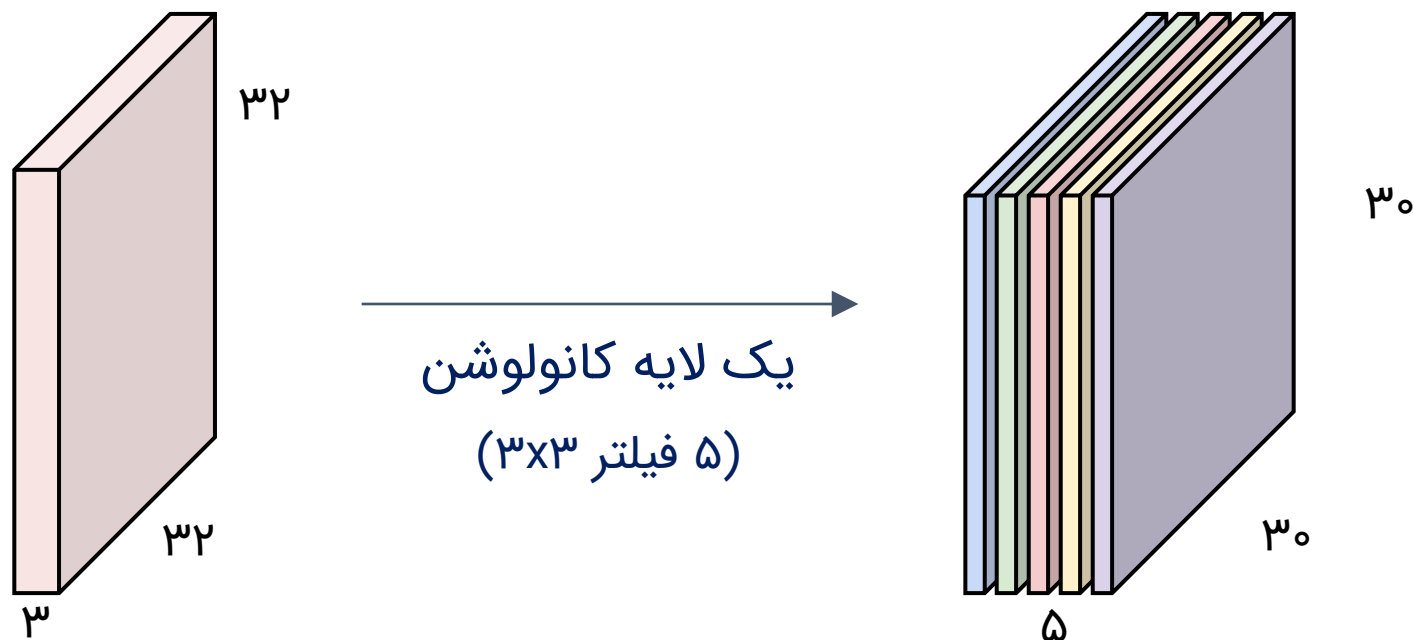
## لایه کانولوشن (Conv)

- چند فیلتر در یک لایه



## لایه کانولوشن (Conv)

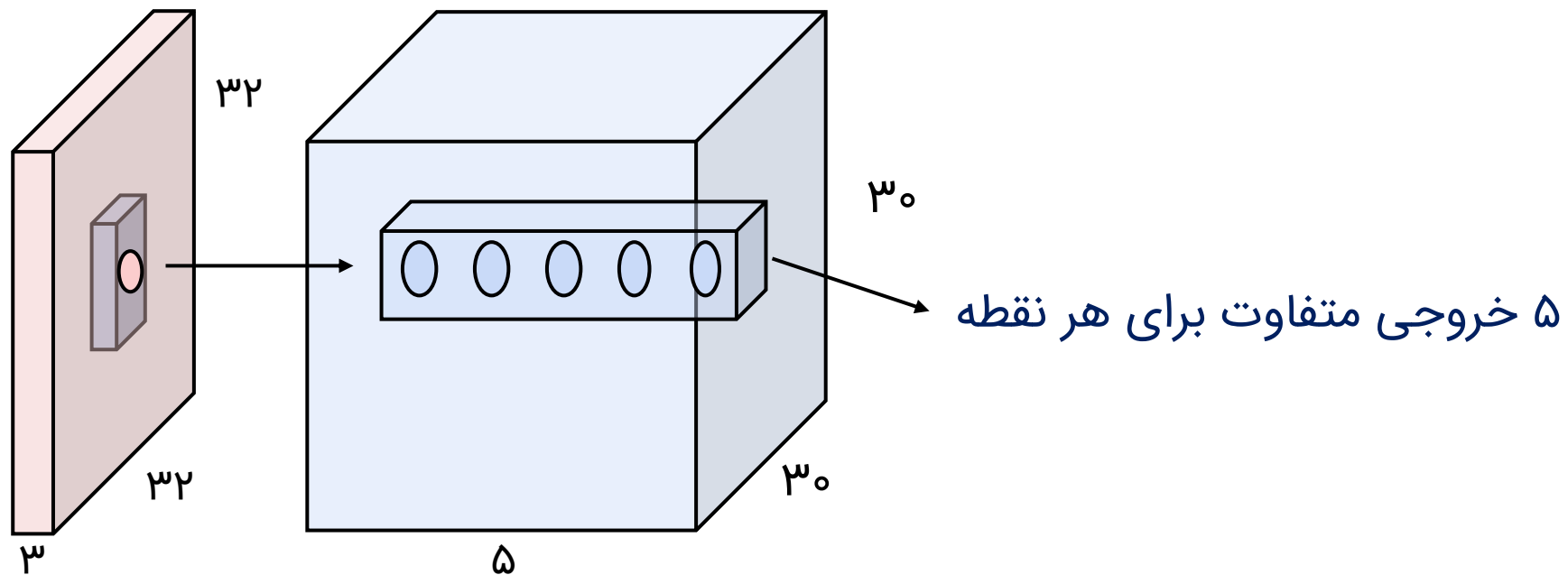
- چند فیلتر در یک لایه





## لایه کانولوشن (Conv)

- ویژگی‌های مختلف از هر نقطه

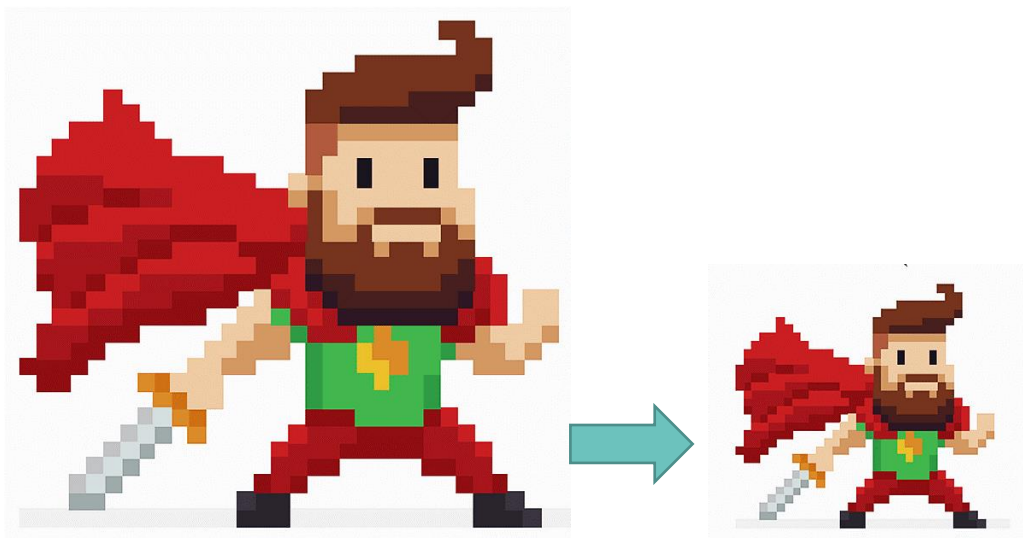


## لایه کاهش اندازه

هدف ← کاهش حجم محاسبات / استفاده از فیلترهای بیشتر

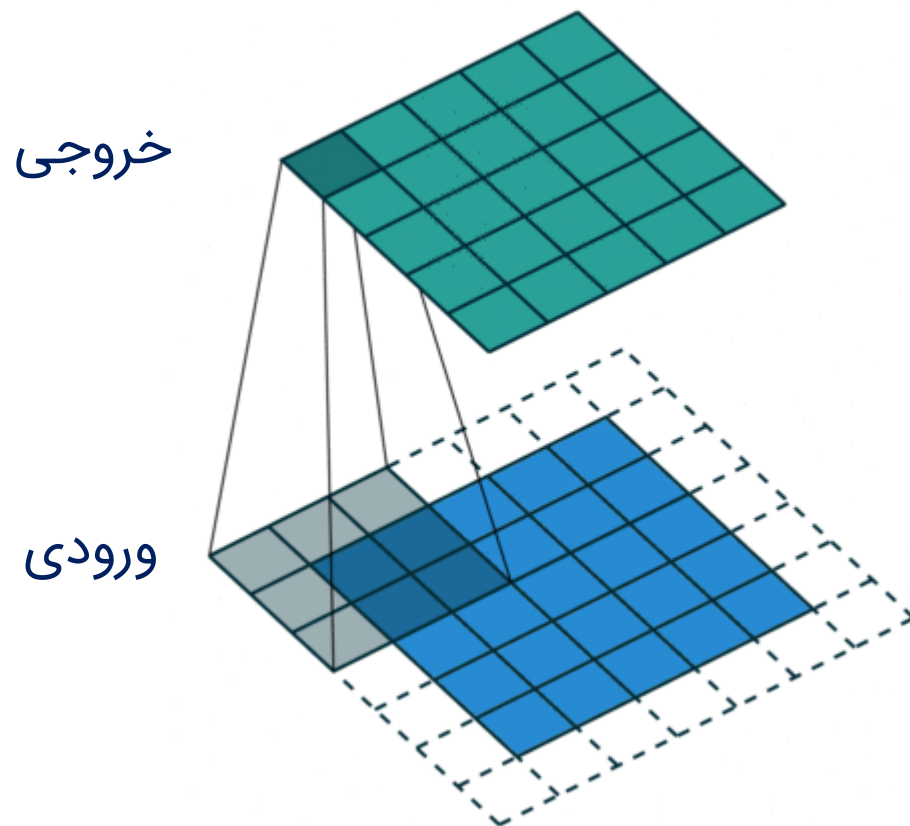
روش‌های کاهش اندازه:

- استفاده از گام (Stride) برای فیلتر کانولوشن
- استفاده از لایه Pooling
- استفاده از Dilation در فیلتر کانولوشن



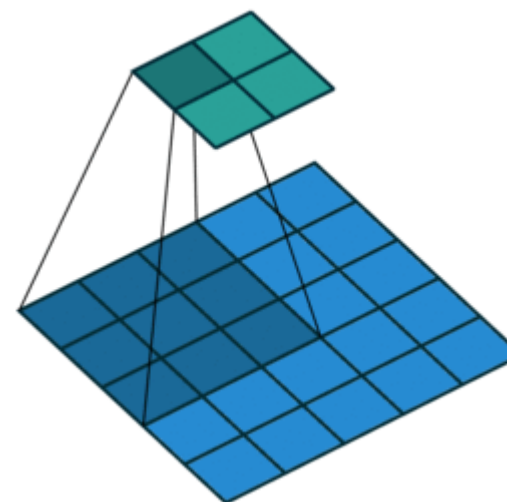
## کاهش اندازه

- با استفاده از گام (stride)



Conv + padding

سایز خروجی = سایز ورودی

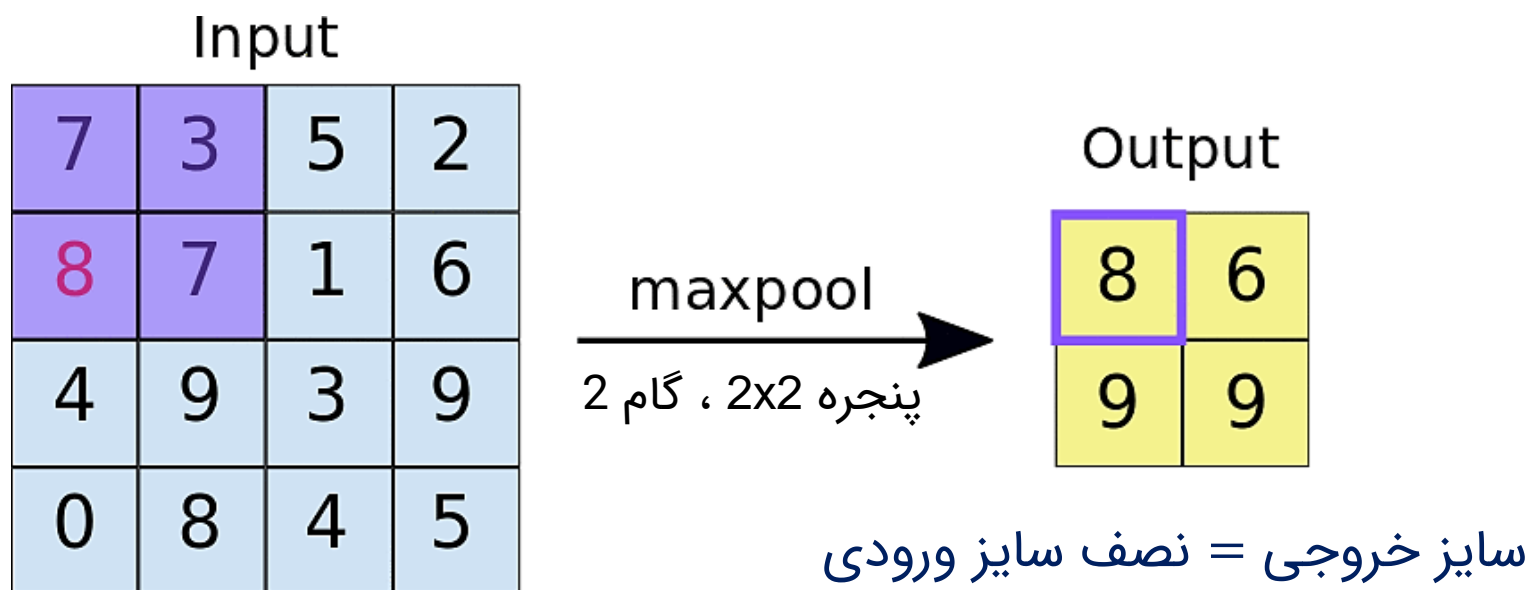


Conv + stride

سایز خروجی  $\approx$  نصف سایز ورودی

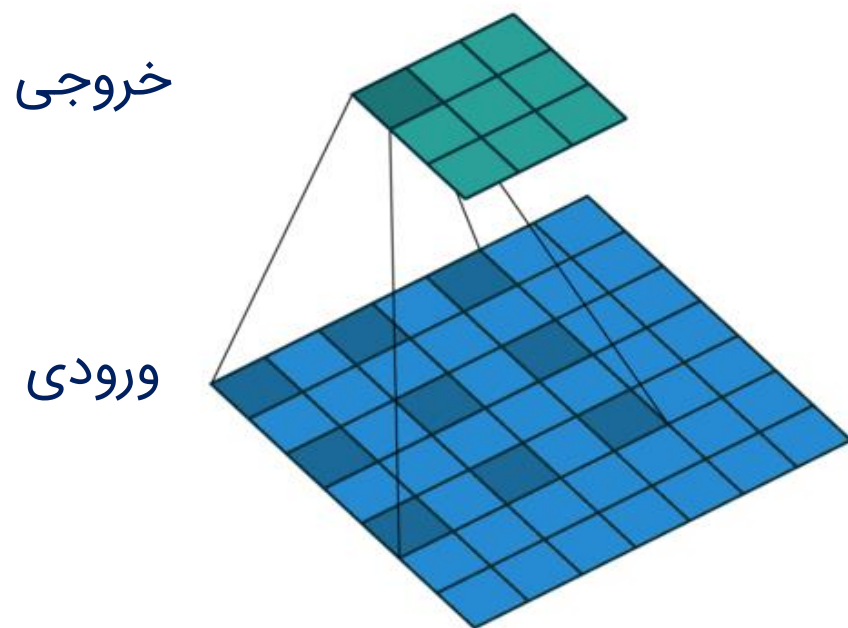
## کاهش اندازه

- با استفاده از pooling



## کاهش اندازه

- با استفاده از dilation



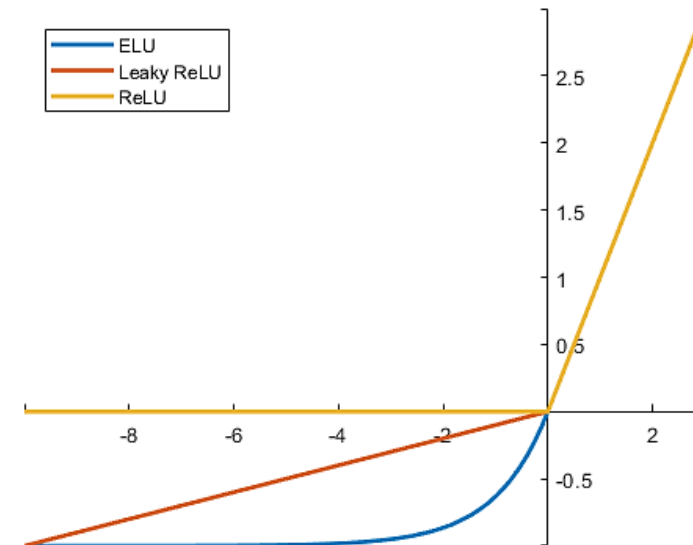
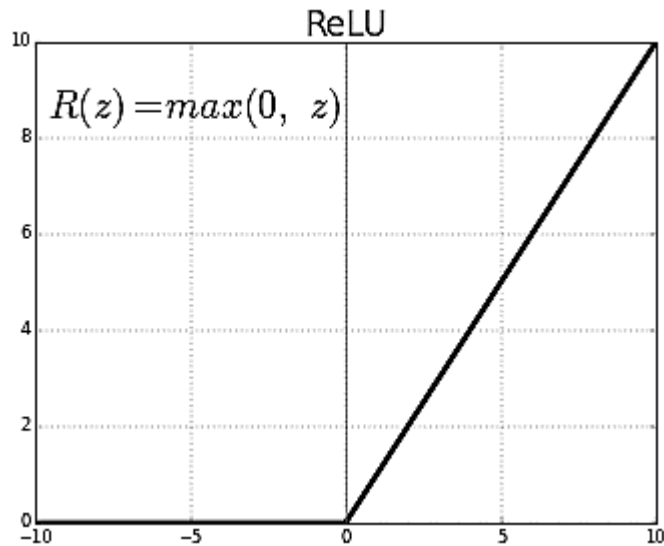
فیلتر  $3 \times 3$

Dilation rate = 2

سایز خروجی  $\approx$  نصف سایز ورودی

# تابع غیر خطی

## • تابع ReLU (Rectified Linear Unit)



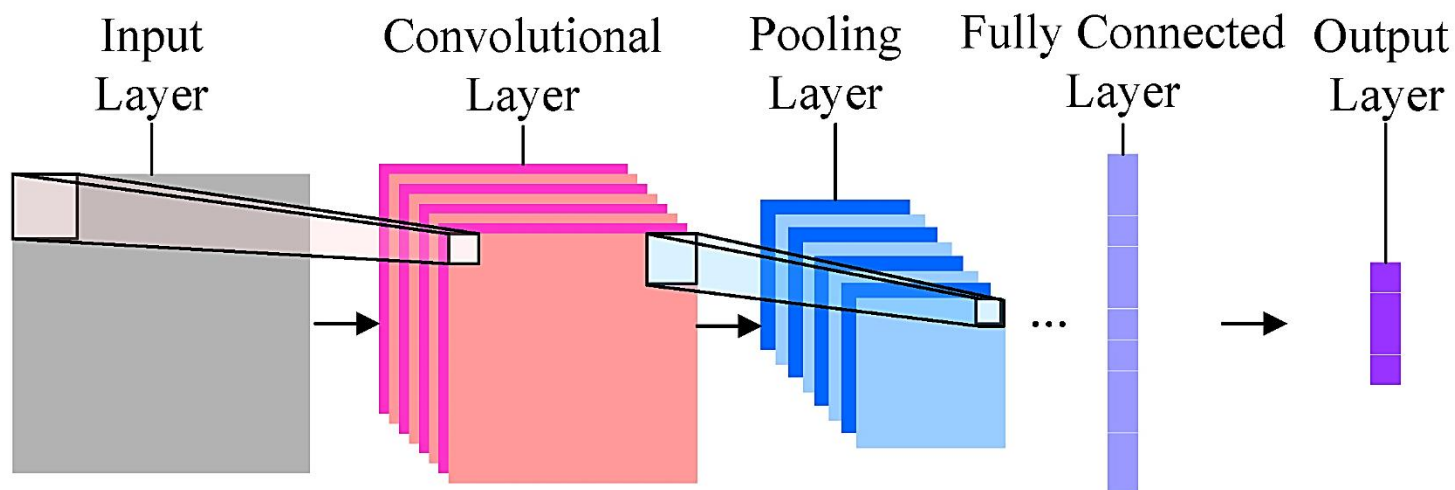
$$y = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \quad \leftarrow \text{ReLU} \\ ax & x < 0 \quad \leftarrow \text{PReLU / Leaky ReLU} \\ a(e^x - 1) & x < 0 \quad \leftarrow \text{ELU} \end{cases}$$

# نحوه نمایش مدل کانولوشنی

ترکیب لایه‌های Conv، Pooling و Fully Connected

کاربرد

• کلاس‌بندی (Classification)



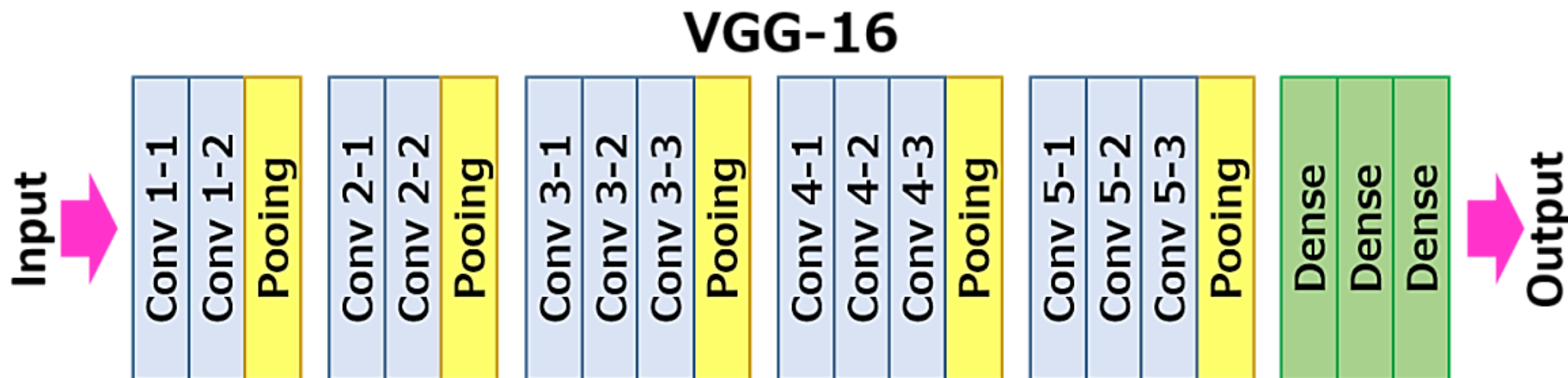
کار عملی با شبکه CNN: <https://poloclub.github.io/cnn-explainer/>

## نحوه نمایش مدل کانولوشنی

کاربرد

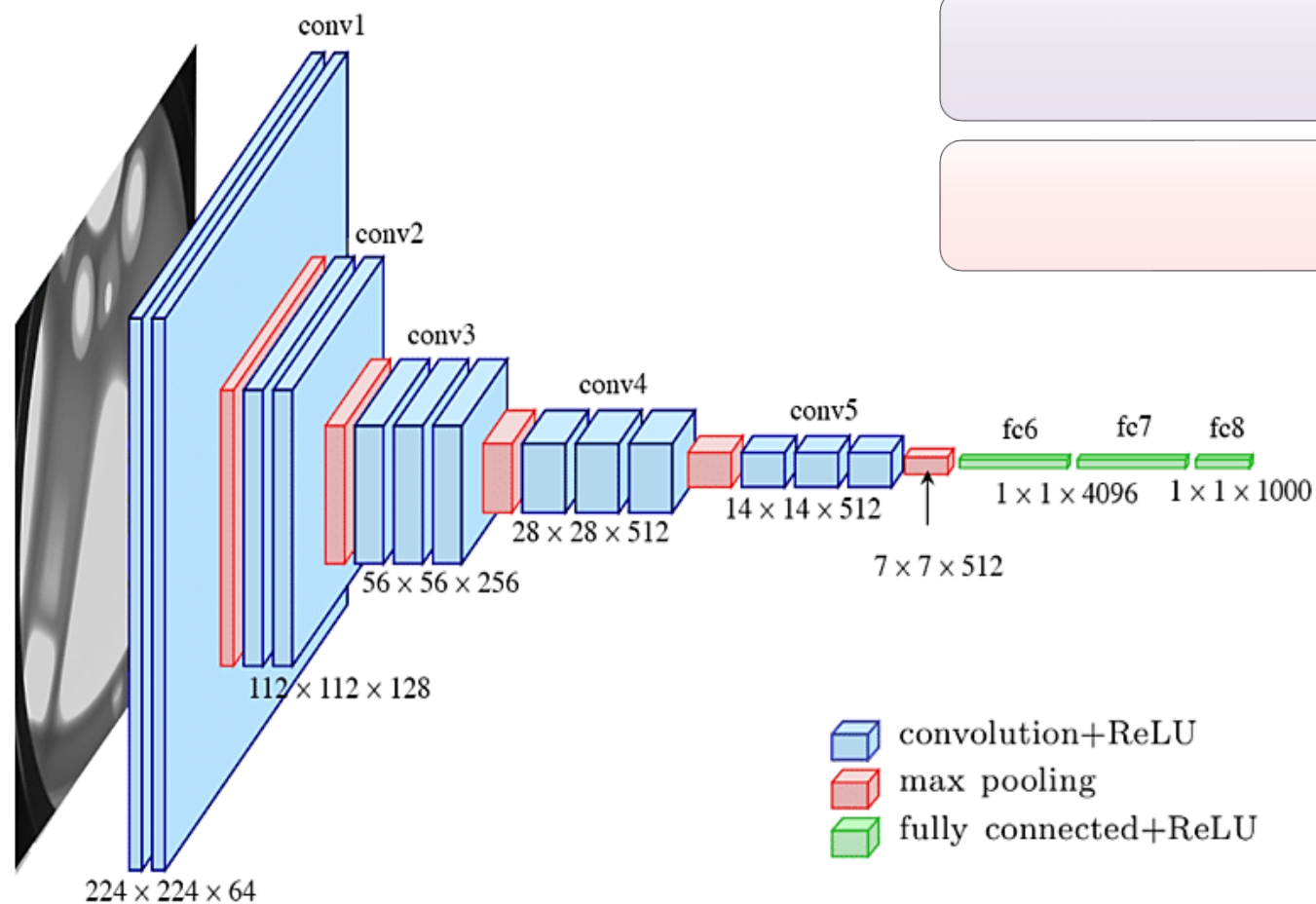
مدل VGG16

کلاس‌بندی (Classification)





# نحوه نمایش مدل کانولوشنی



مدل VGG16

کاربرد

• کلاس‌بندی (Classification)

# معماری‌های مختلف CNN

## Classification

- LeNet
- VGG
- GoogLeNet
- ResNet
- ...

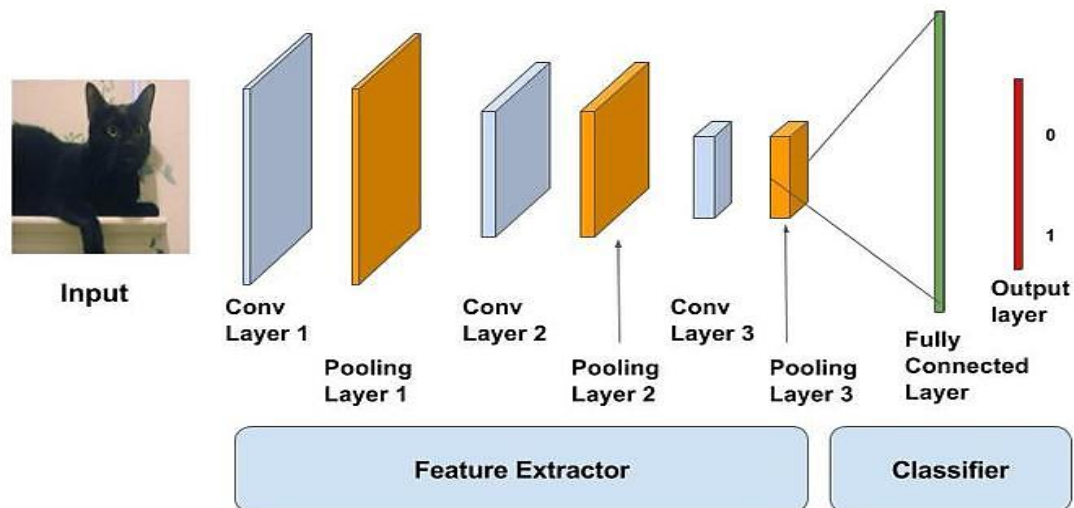
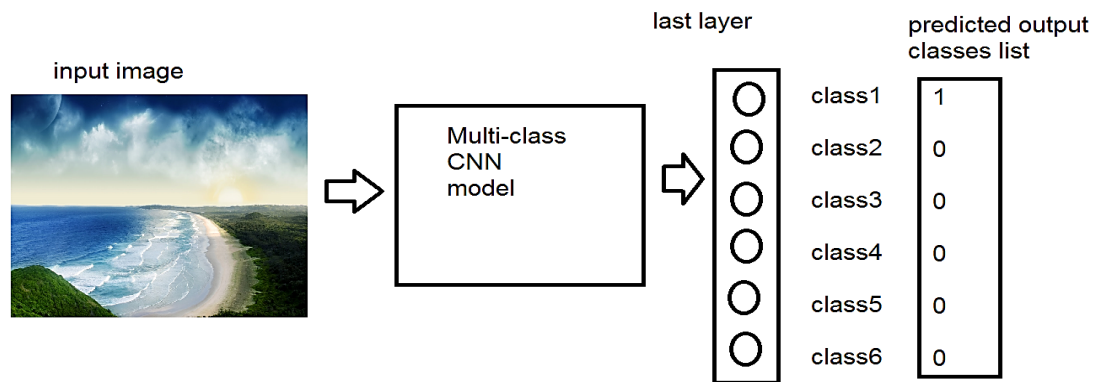
## Object Detection

- R - CNN
- YOLO
- MTCNN
- ...

## Segmentation

- Mask R - CNN
- U - Net
- ...

# کلاس بندی (Classification)



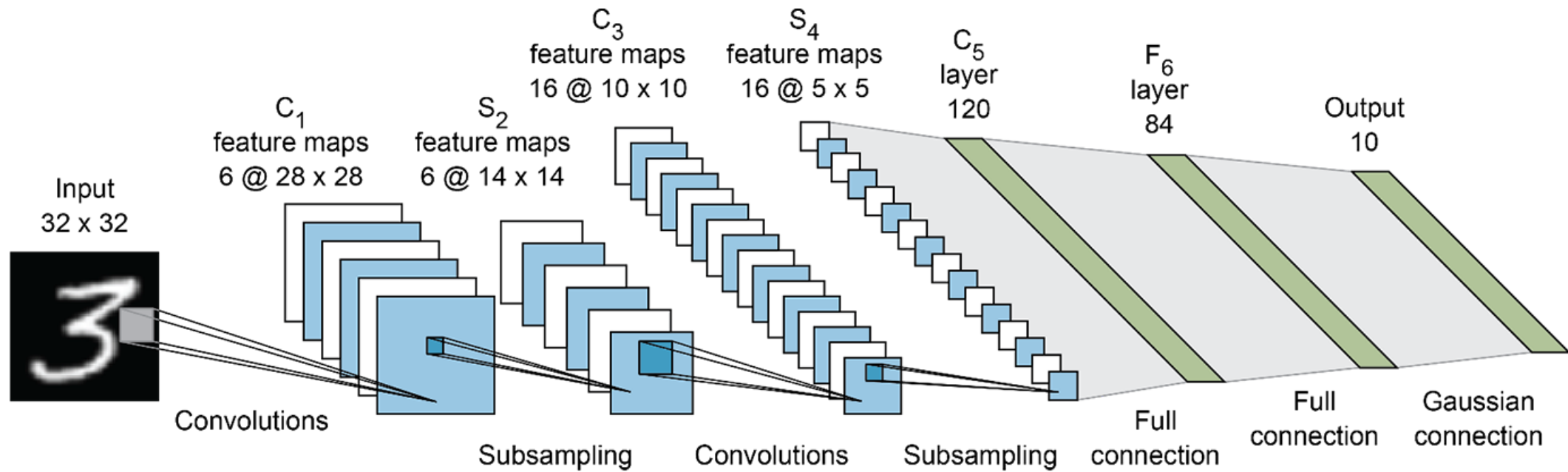
تشخیص  
کلاس تصویر  
ورودی

شماره کلاس  
ورودی

خروجی مدل

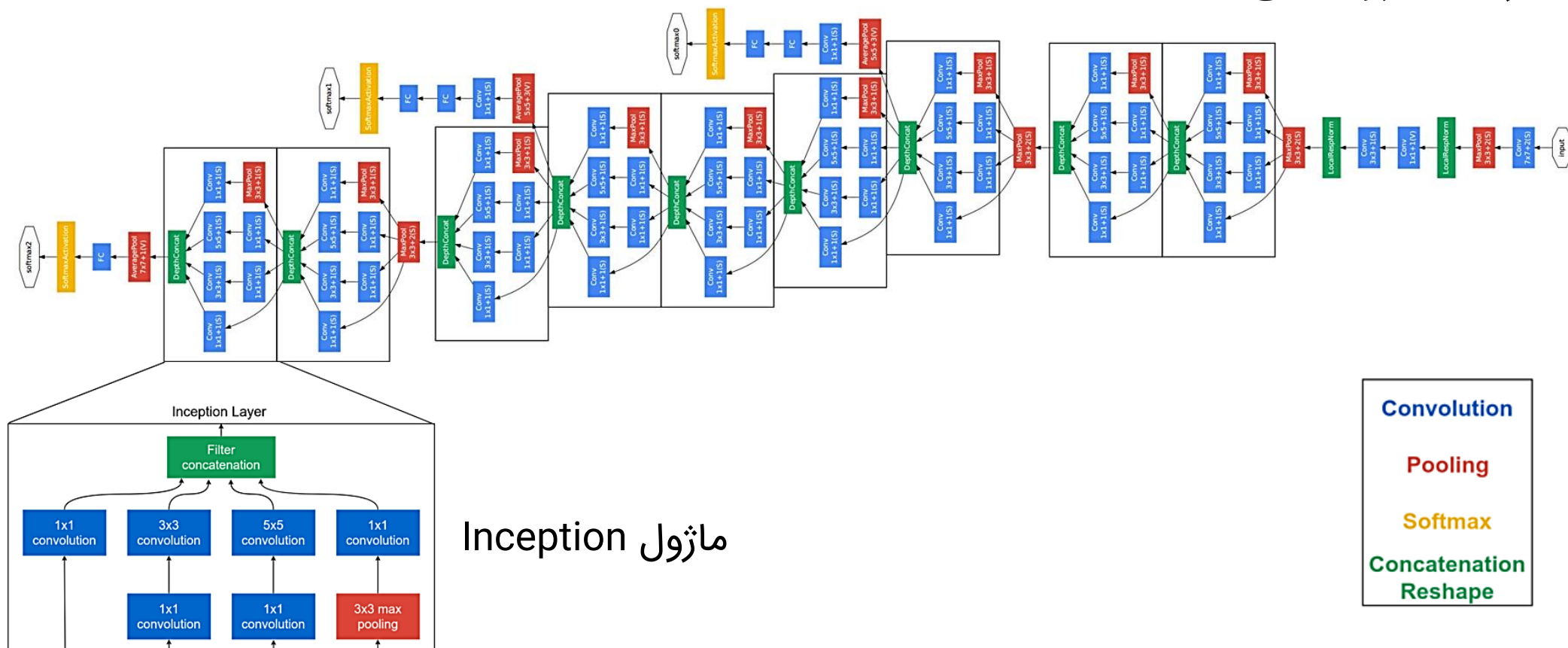
# مدل LeNet

• اولین نمونه موفق



# مدل GoogLeNet

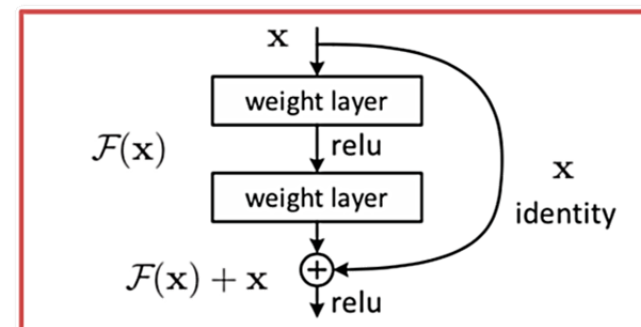
• با معرفی ماژول‌های Inception



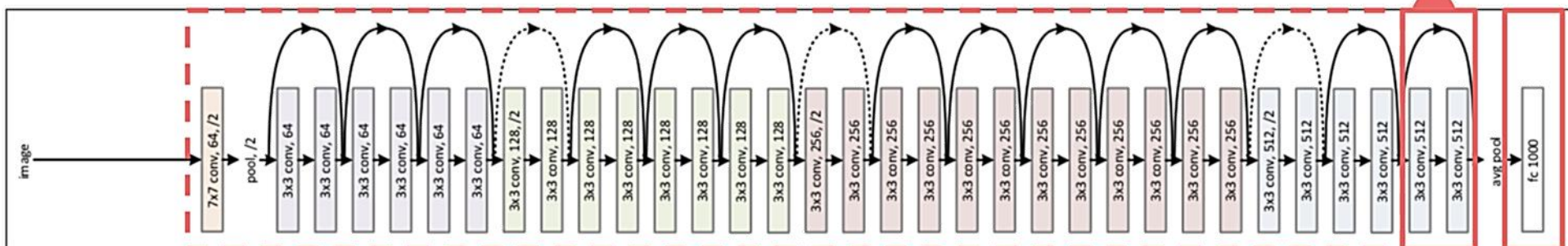
ماژول Inception

# مدل ResNet

- قابلیت استفاده از تعداد لایه‌های زیاد



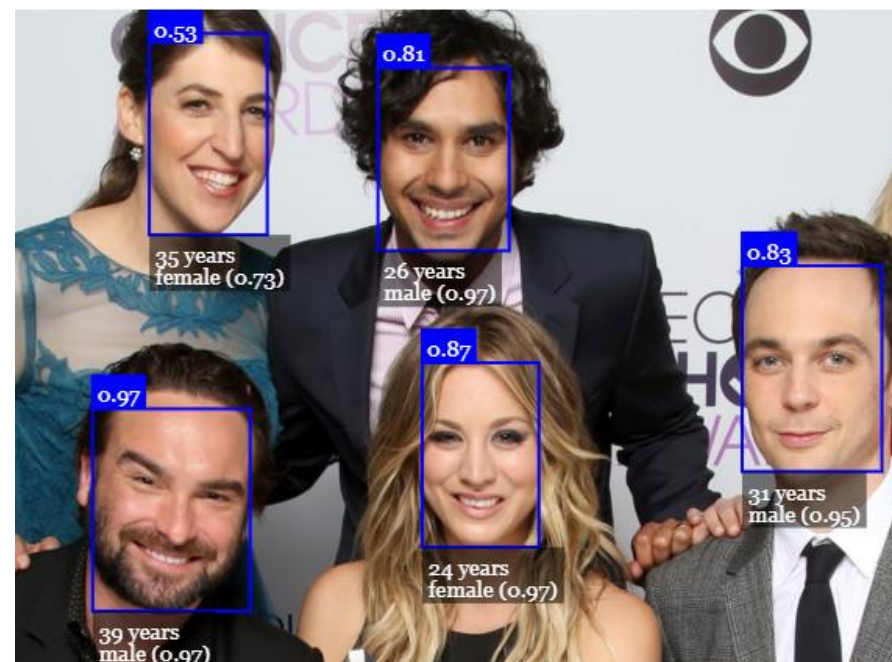
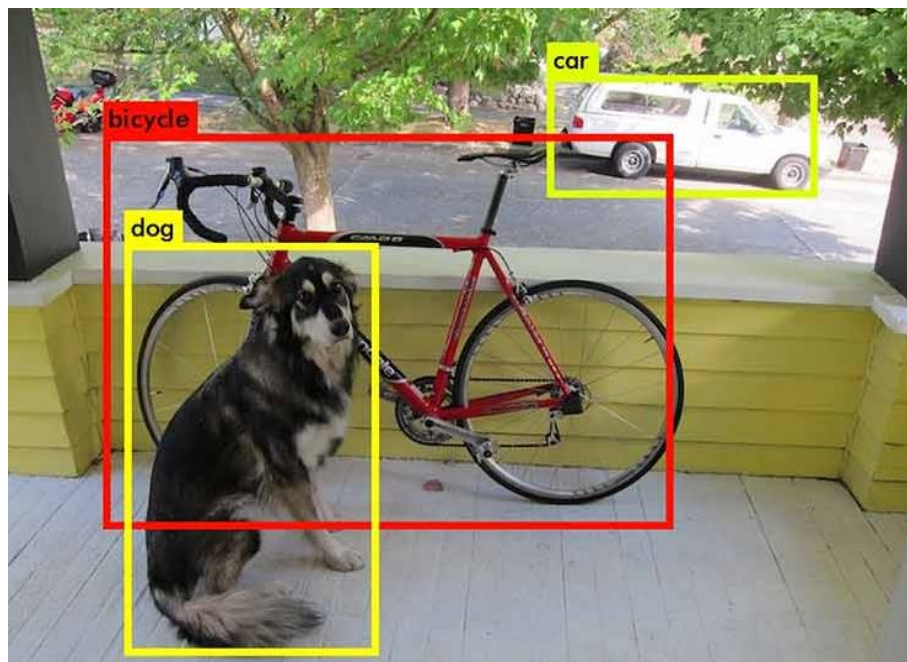
ResNet50 Diagram





## تشخیص اشیا / چهره (Object / Face Detection)

- تشخیص کلاس سوژه‌های موجود در تصویر + محل قرارگیری آن‌ها
- خروجی مدل: کلاس سوژه + مختصات کادر



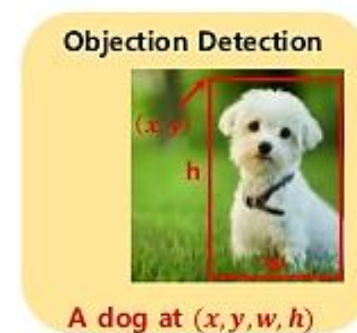
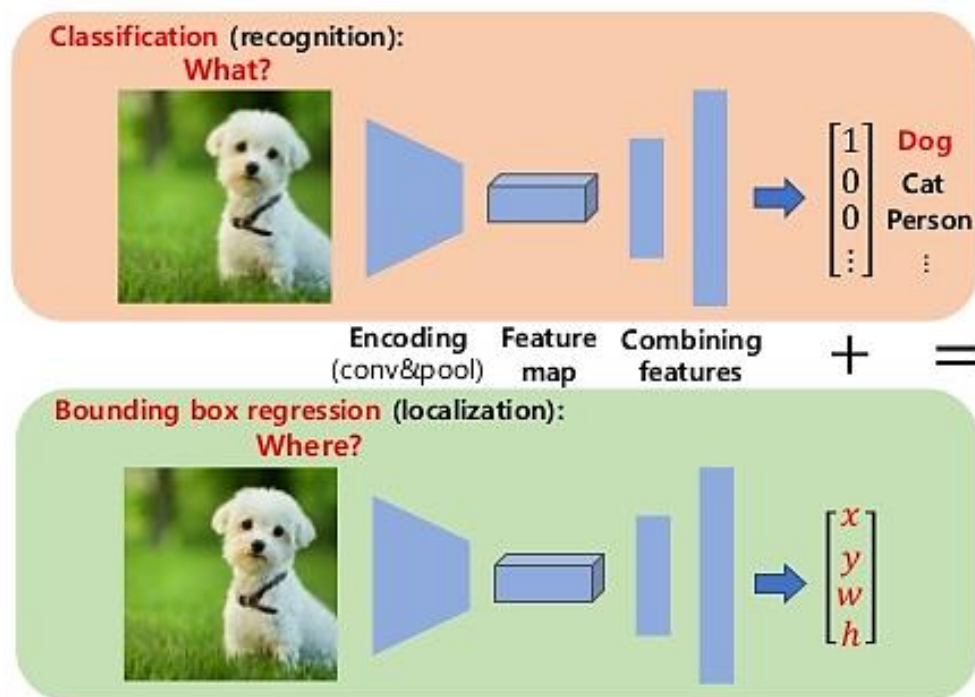
# مدل R - CNN

تشخیص اشیا (Classification) + محل قرارگیری در تصویر (Regression)

مدل‌های ارتقایافته:

• Fast R - CNN

• Faster R - CNN



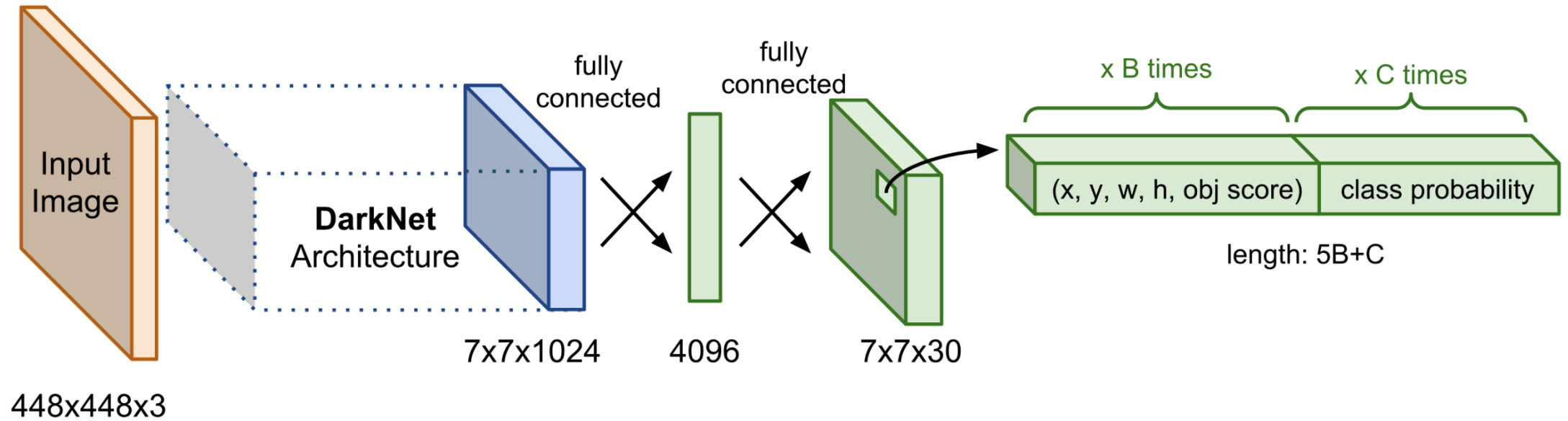
**Bounding box information**

- $(x, y)$ : top left corner position
- $w$  = width
- $h$  = height



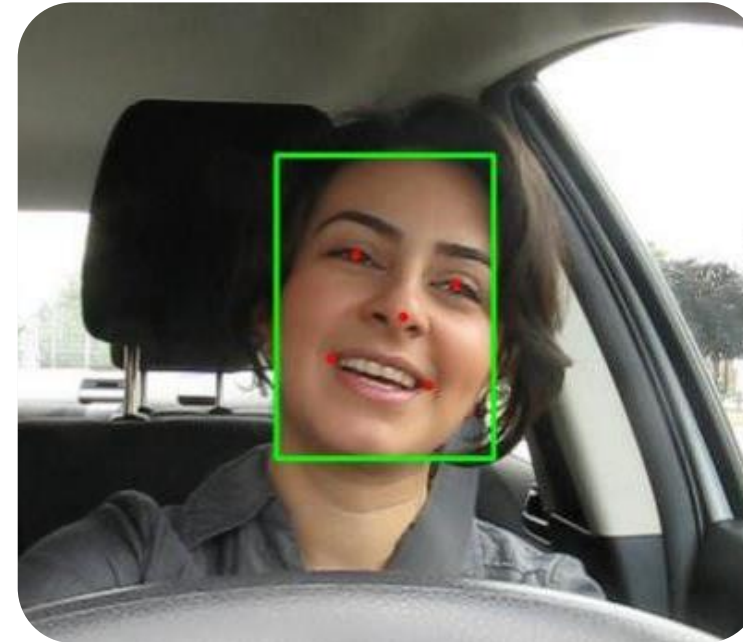
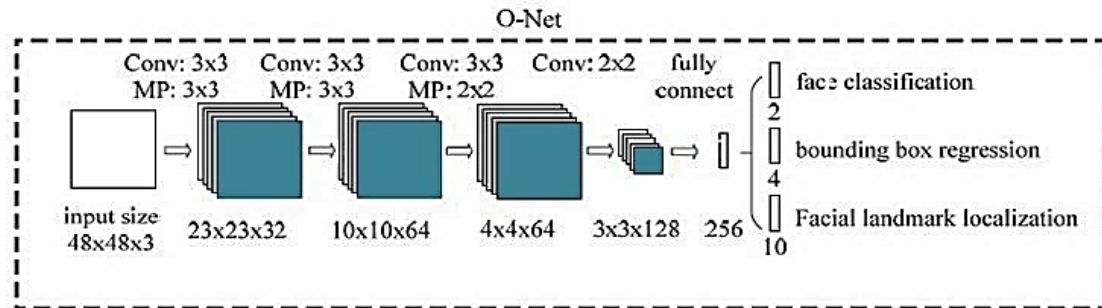
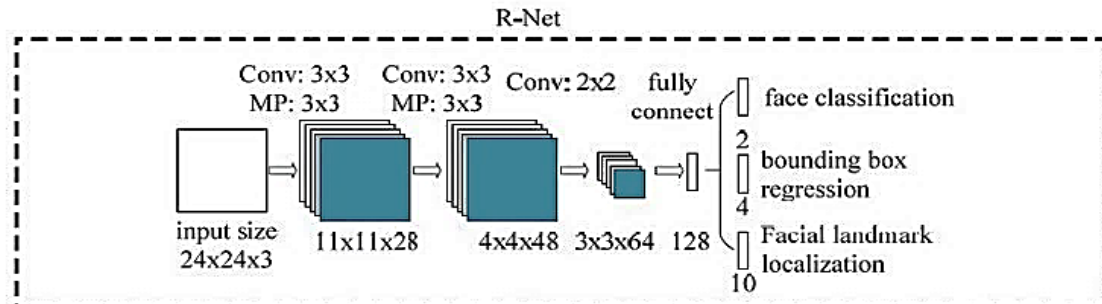
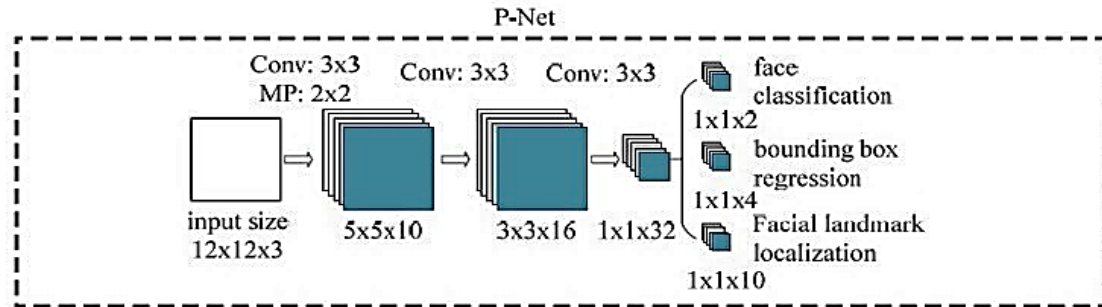
# مدل YOLO

- سرعت بالا ← مناسب برای تشخیص سریع (Realtime detection)



# مدل MTCNN

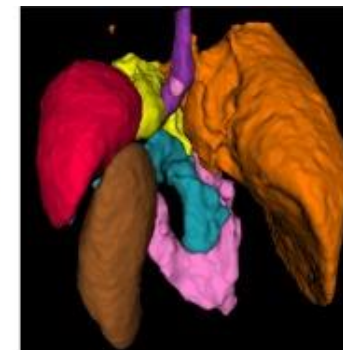
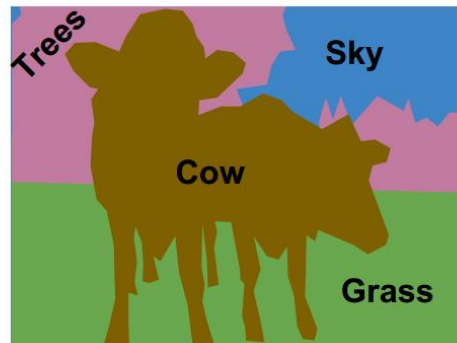
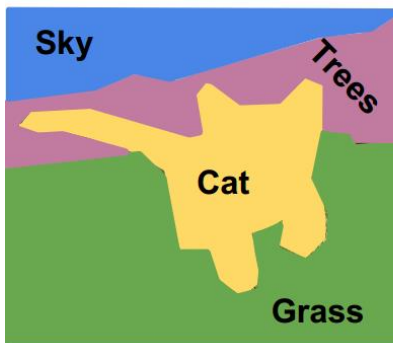
• کاربرد: تشخیص چهره



# بخش‌بندی (Segmentation)

خروجی مدل  
• تصویر

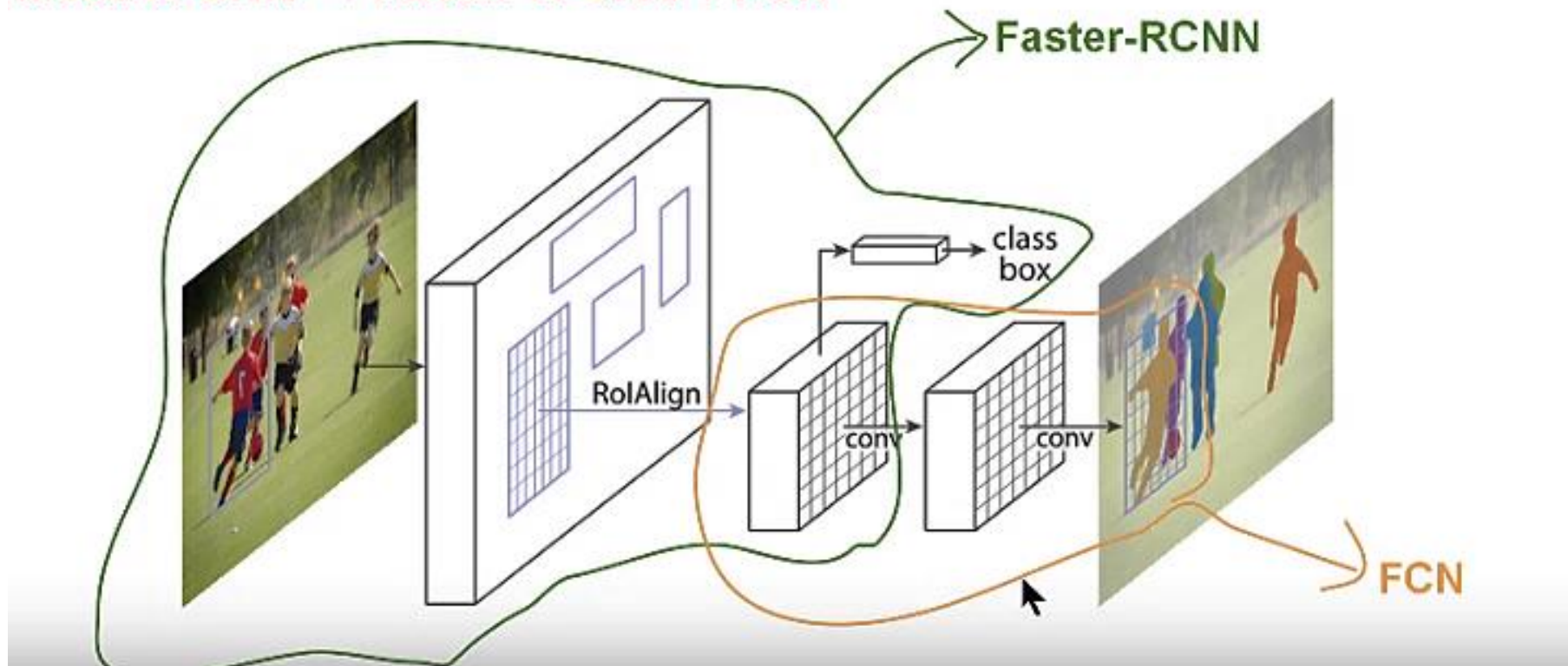
تشخیص کلاس هر نقطه ← مرزبندی دقیق سوژه



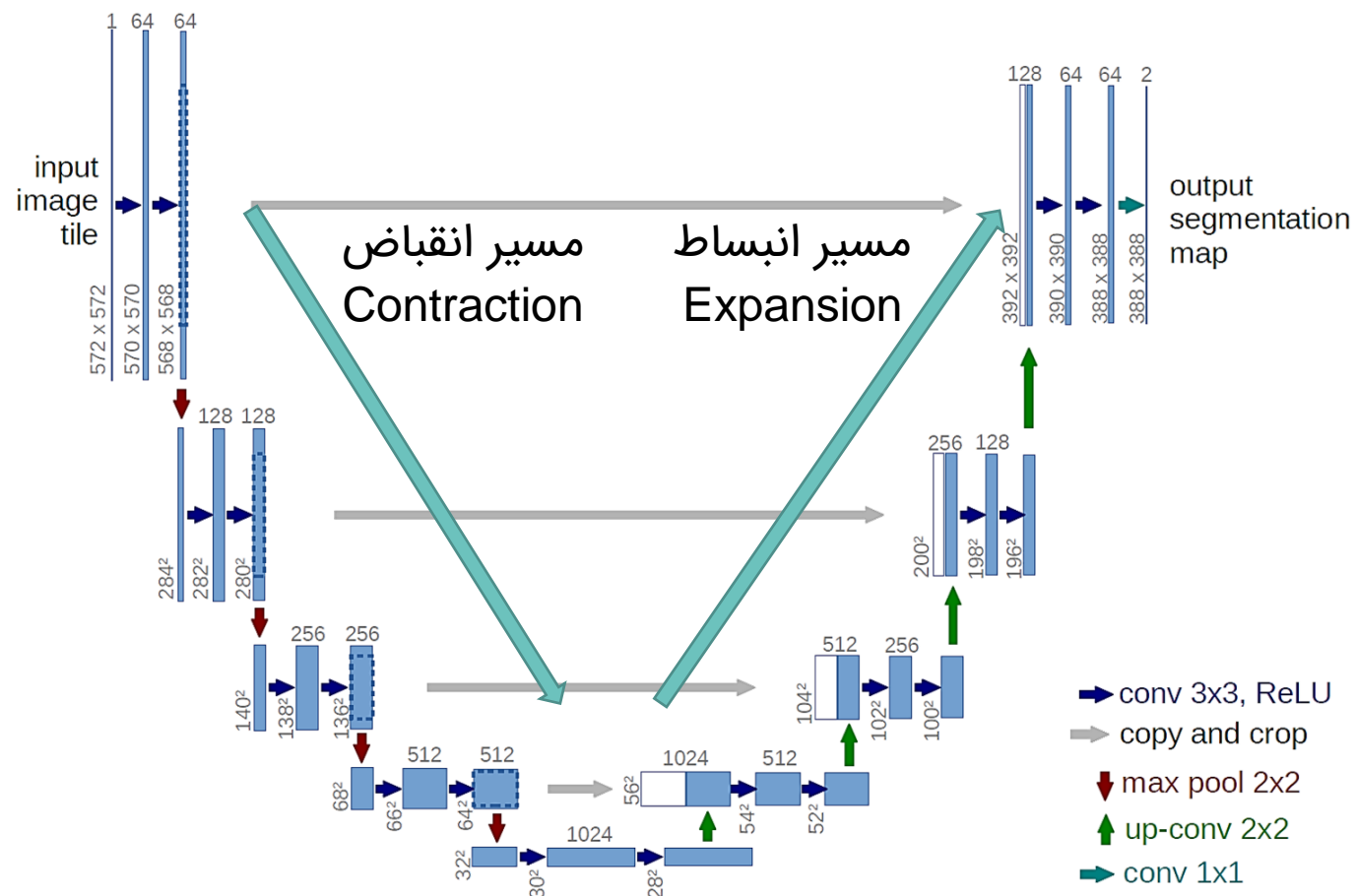
# مدل Mask R - CNN

- تشخیص اشیا + بخش‌بندی (Segmentation)

Mask R-CNN → Faster R-CNN + FCN



# مدل U - Net



• مناسب برای تصاویر پزشکی

