
بسم الله الرحمن الرحيم

مبانی بینایی کامپیوتر

مدرس: محمدرضا محمدی

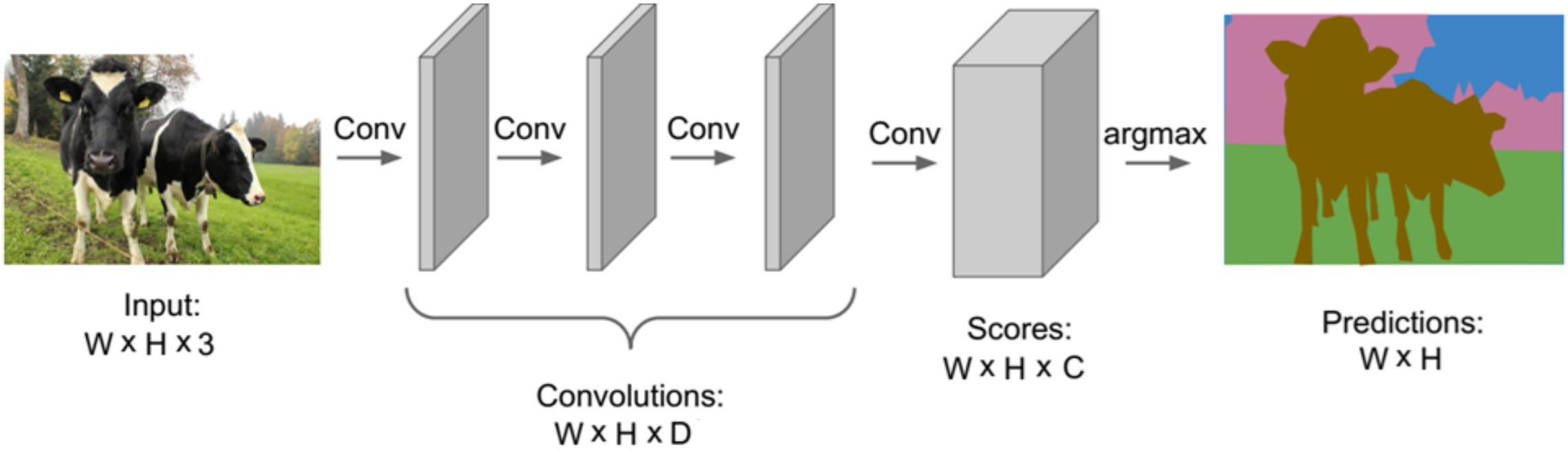
بهار ۱۴۰۳

ناحیه‌بندی معنایی

Semantic Segmentation

شبکه‌های کاملا کانولوشنی

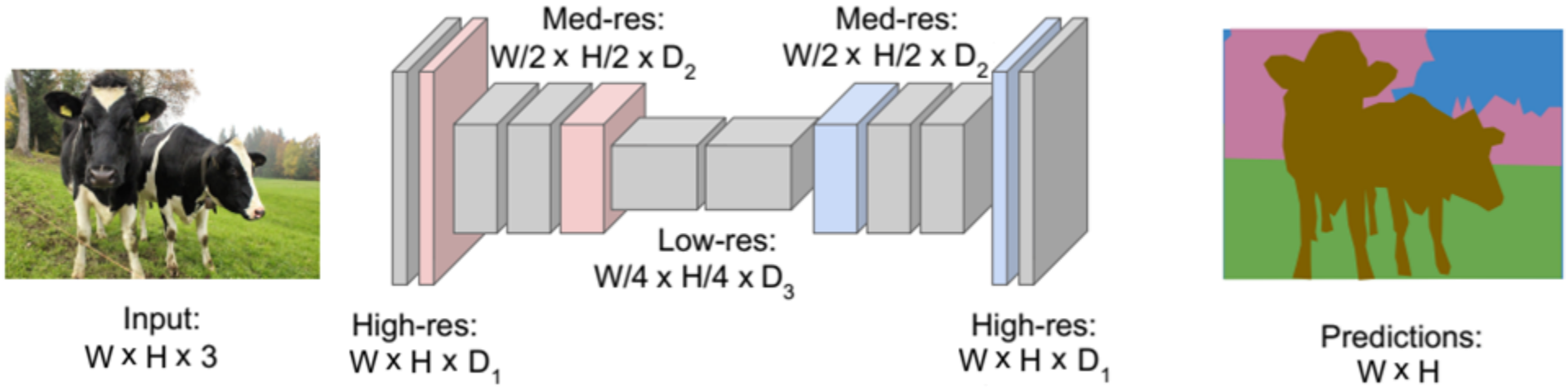
- برای داشتن میدان تاثیر بزرگ، باید از فیلترهای بزرگ و/یا عمق زیاد استفاده کنیم
- حافظه مورد نیاز و حجم محسابات بسیار زیاد خواهد بود!



https://towardsdatascience.com/semantic-segmentation-in-the-era-of-neural-networks-703bf93e5ee1

شبکه‌های کاملا کانولوشنی

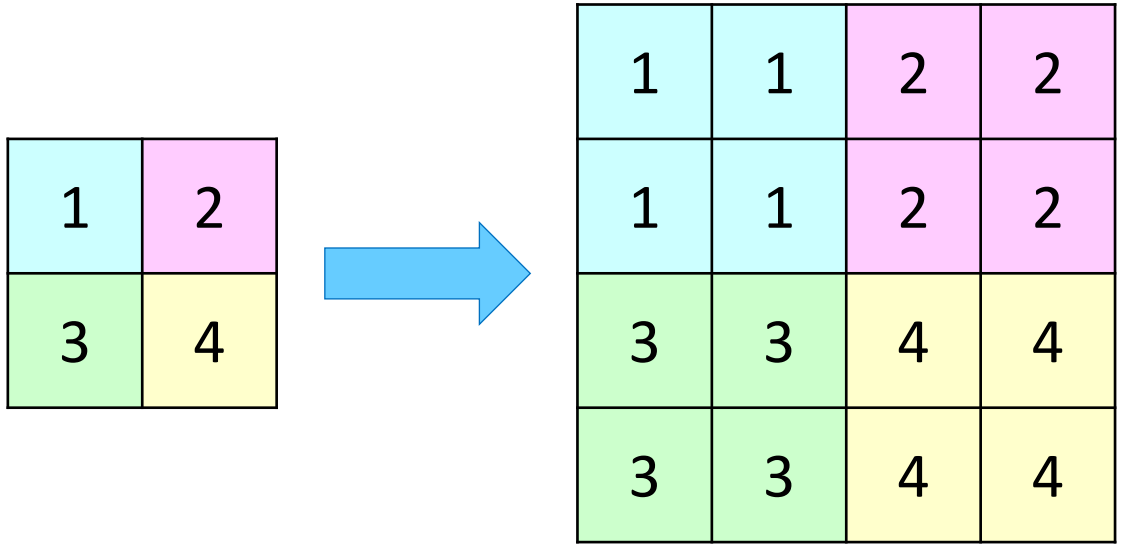
- می‌توان مشابه با شبکه‌های متداول، ابعاد مکانی را در طول شبکه کاهش داد و مجدد ابعاد مکانی را به صورت تدریجی افزایش داد
- برای کاهش ابعاد مکانی می‌توان از Pooling استفاده کرد
- چطور می‌توان ابعاد را افزایش داد؟



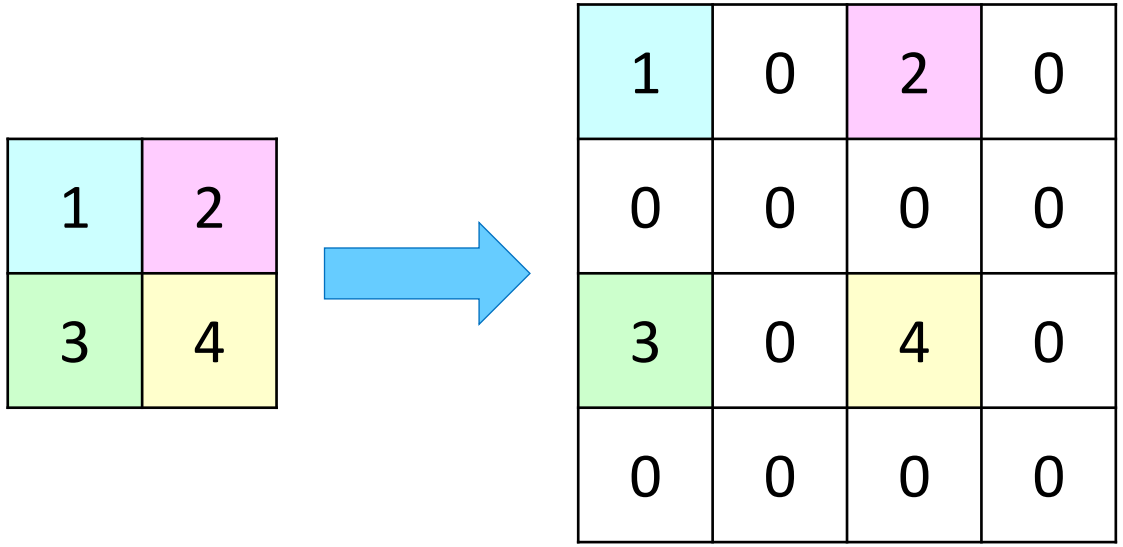
http://cs231n.stanford.edu/slides/2020/lecture_12.pdf

افزایش ابعاد (Unpooling)

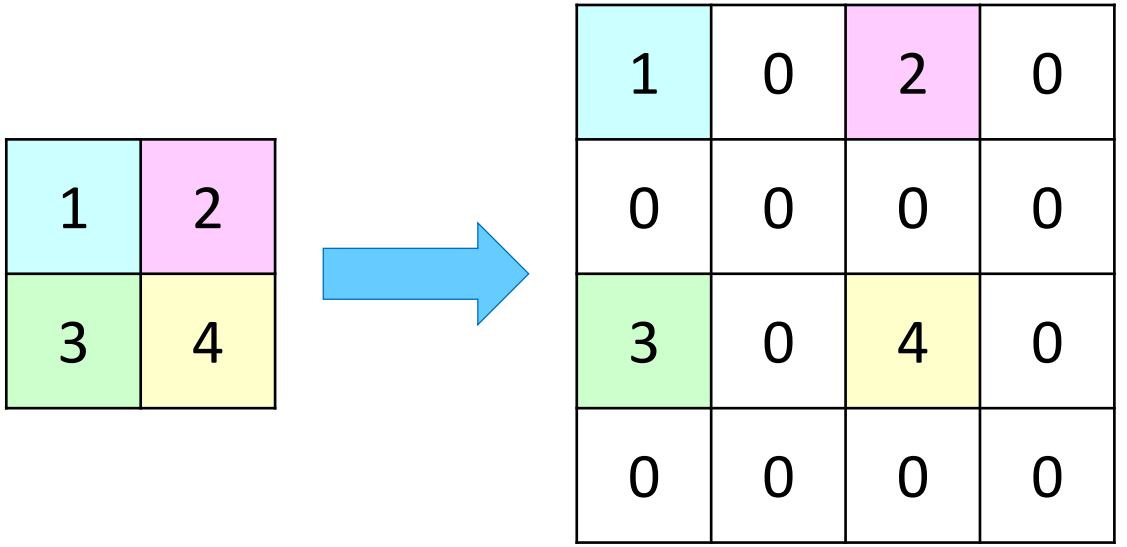
نزدیک‌ترین همسایه



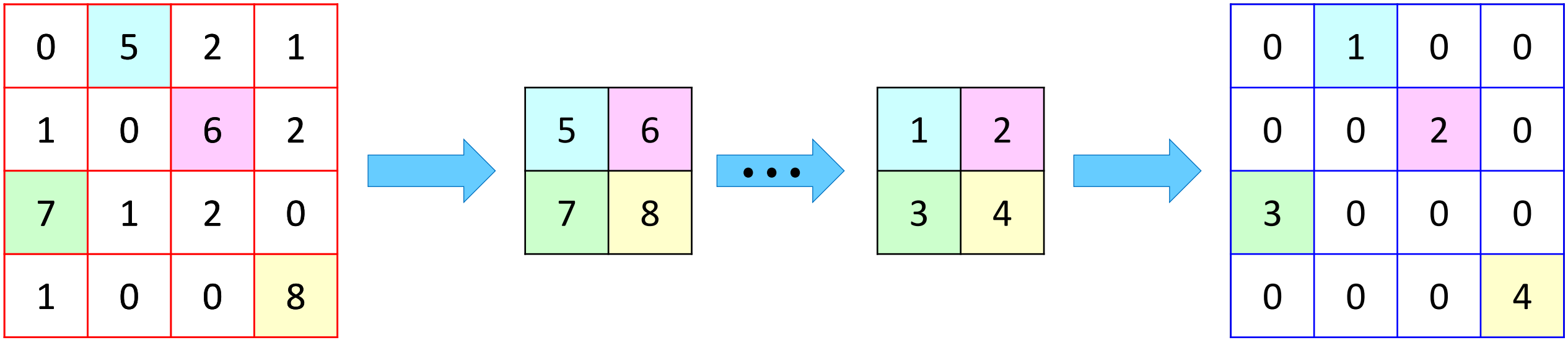
Bed of Nails



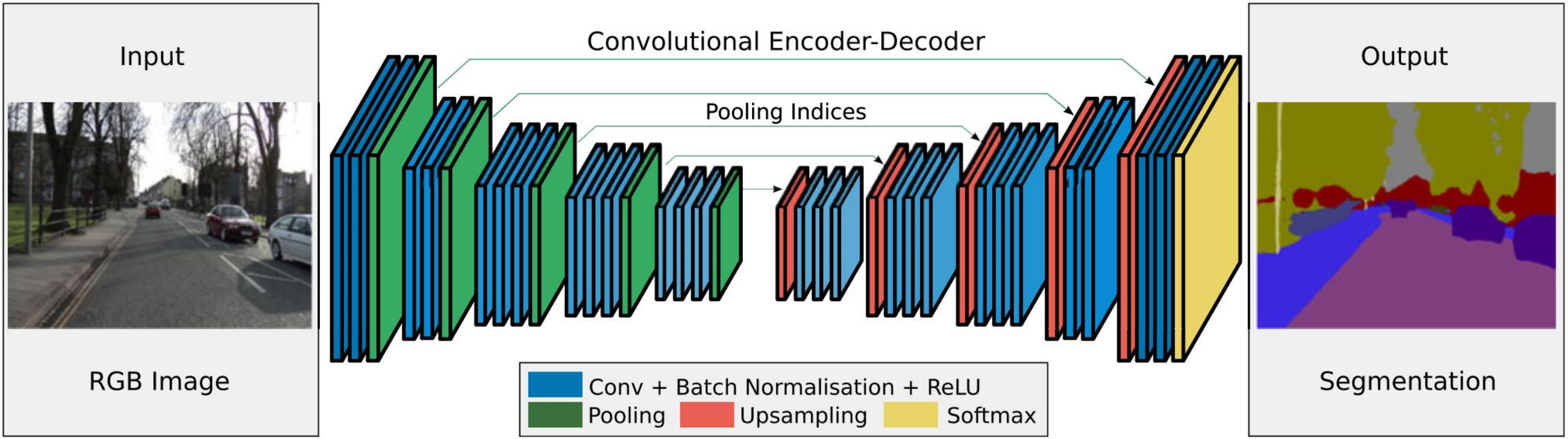
افزایش ابعاد (Max Unpooling)



افزایش ابعاد (Max Unpooling)

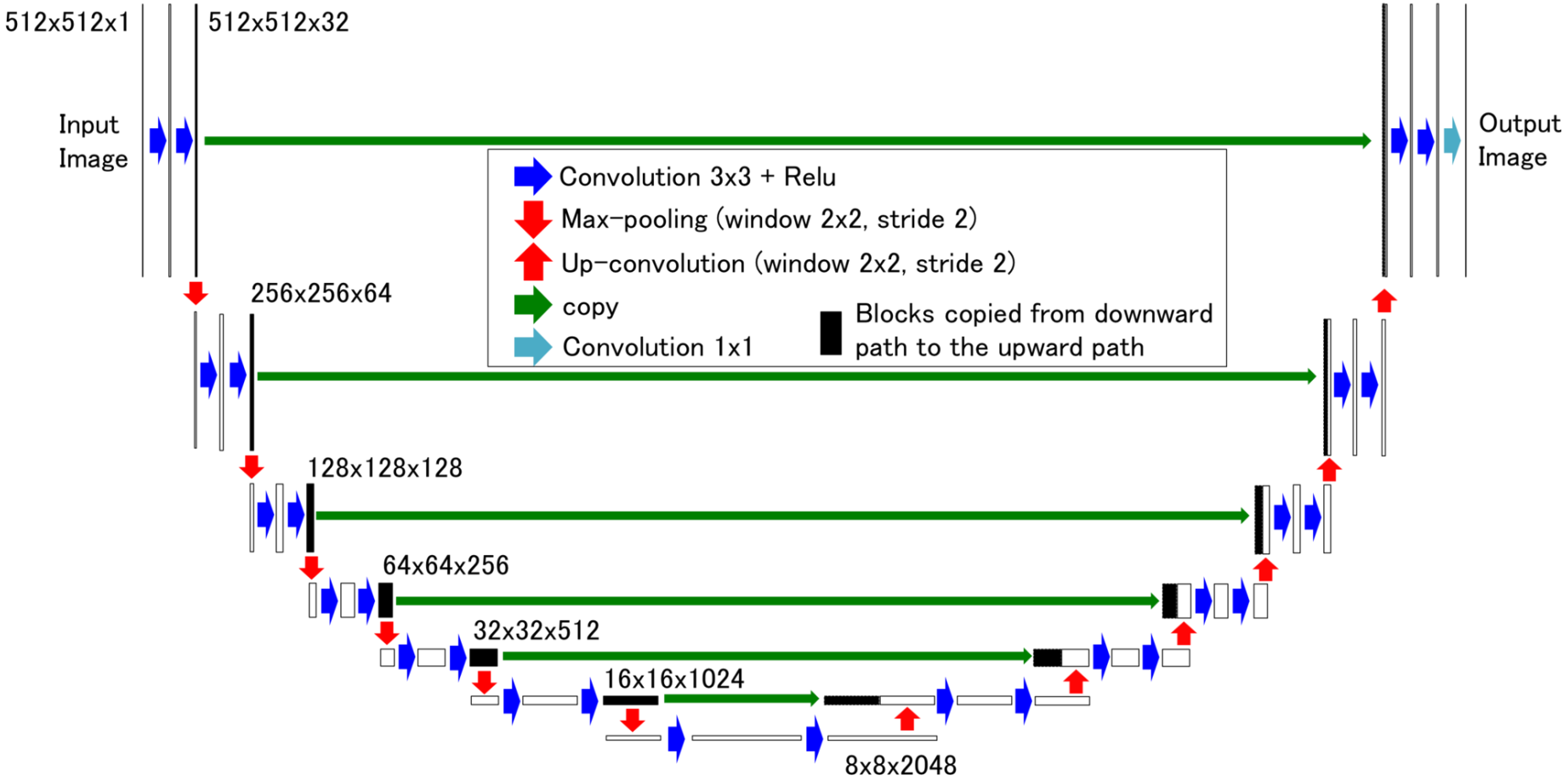


SegNet



Badrinarayanan, V., et al., "Segnet: A deep convolutional encoder-decoder architecture for image segmentation“, *IEEE transactions on pattern analysis and machine intelligence*, 2017

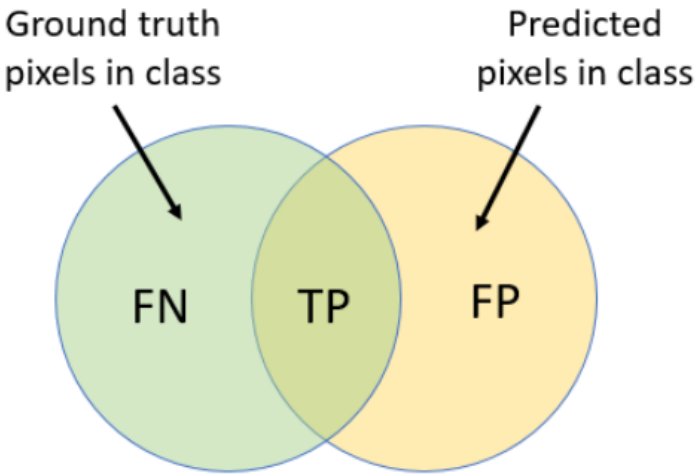
U-Net



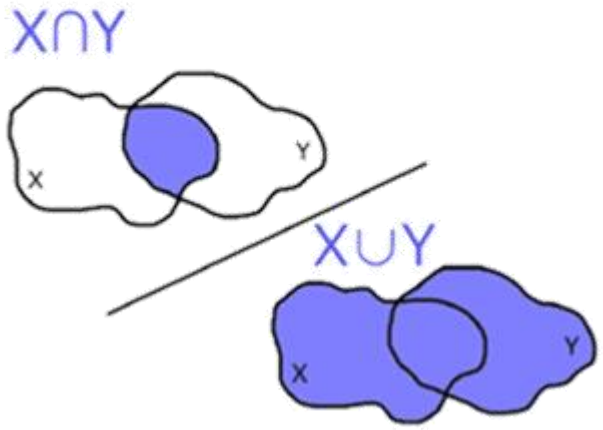
Ronneberger, O., et al., "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*, 2015

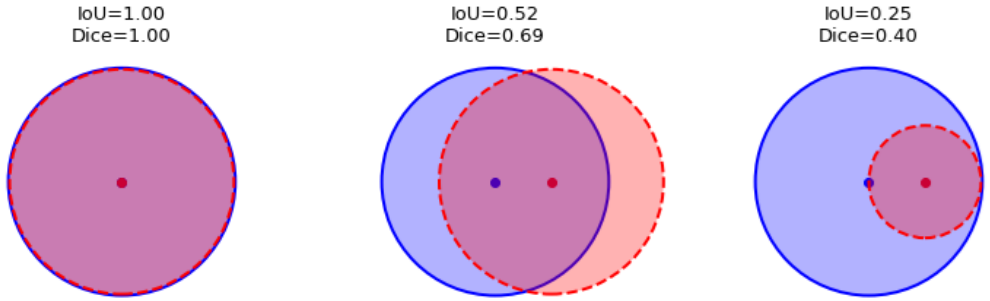
```
img_size = (200, 200, 3)
num_classes = 10
filters = 32

# function that defines 2D transposed convolutional (Deconvolutional) layer
def deconv(x, filters, kernel_size=3):
    x = keras.layers.UpSampling2D(size=(2,2), interpolation='nearest')(x)
    x = keras.layers.Conv2D(filters=filters, kernel_size=kernel_size, activation='relu', padding='same')(x)
    return x

# define input layer
input = keras.layers.Input(shape=img_size)

# begin with contraction part
conv1 = keras.layers.Conv2D(filters=filters*1, kernel_size=5, activation='relu', padding='same')(input)
pool1 = keras.layers.MaxPool2D(pool_size=(2,2), strides=(2,2))(conv1)

conv2 = keras.layers.Conv2D(filters=filters*2, kernel_size=5, activation='relu', padding='same')(pool1)
pool2 = keras.layers.MaxPool2D(pool_size=(2,2), strides=(2,2))(conv2)

conv3 = keras.layers.Conv2D(filters=filters*4, kernel_size=5, activation='relu', padding='same')(pool2)

# expansive path
up4 = deconv(conv3, filters*2)
up4 = keras.layers.Concatenate()([conv2, up4])
conv4 = keras.layers.Conv2D(filters=filters*2, kernel_size=5, activation='relu', padding='same')(up4)

up5 = deconv(conv4, filters*1)
up5 = keras.layers.Concatenate()([conv1, up5])
conv5 = keras.layers.Conv2D(filters=filters*1, kernel_size=5, activation='relu', padding='same')(up5)

# define output layer
output = keras.layers.Conv2D(filters=num_classes, kernel_size=1, activation='softmax')(conv5)

# create model
model = keras.models.Model(inputs=input, outputs=output)
```

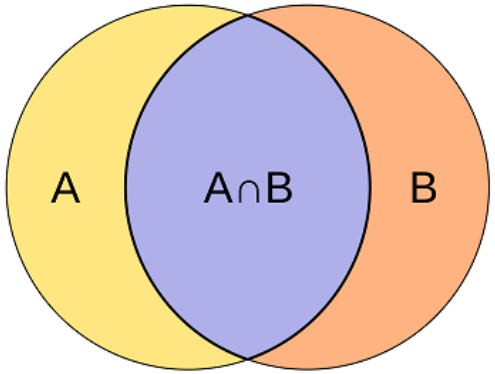
ارزیابی ناحیه‌بندی



$$\text{IoU} = \frac{\text{Intersection } X \cap Y}{\text{Union } X \cup Y}$$







$$\text{Dice coefficient}(A, B) = \frac{2 \times |A \cap B|}{|A| + |B|}$$

Table 1: Segmentation results. Intersection over Union (IoU) and Dice coefficient (Dice) are in % and inference time (Time) is in *ms*.

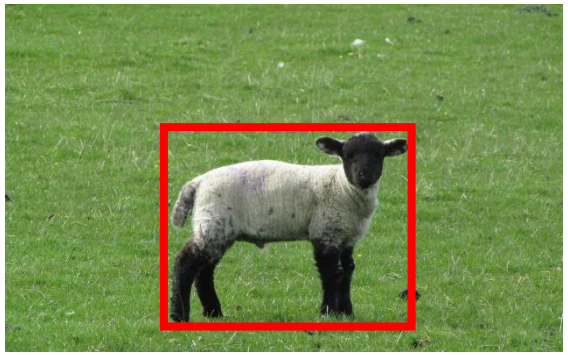
Model	IOU	Dice	Time
U-Net	73.18	83.06	30
TernausNet-11	74.94	84.43	51
TernausNet-16	73.83	83.05	60
AlbuNet-34	**75.35**	**84.98**	**21**

تشخیص اشیاء

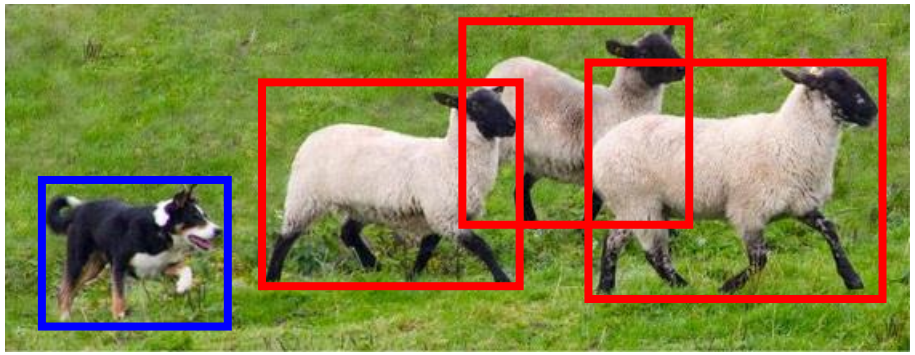
Object Detection

مسئله‌های بینایی کامپیوتر

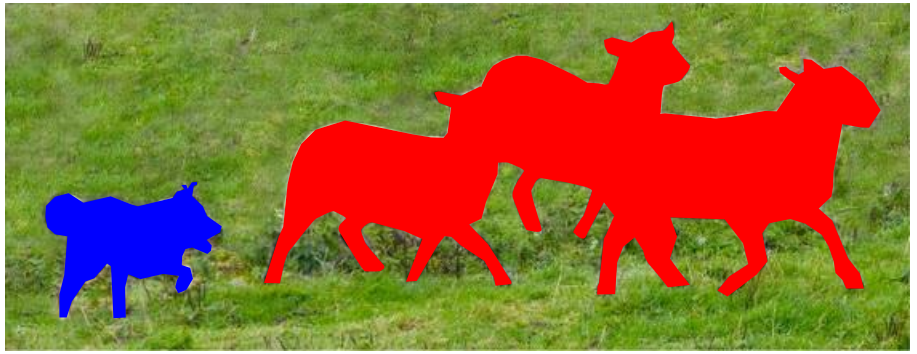
دسته‌بندی + مکان‌یابی



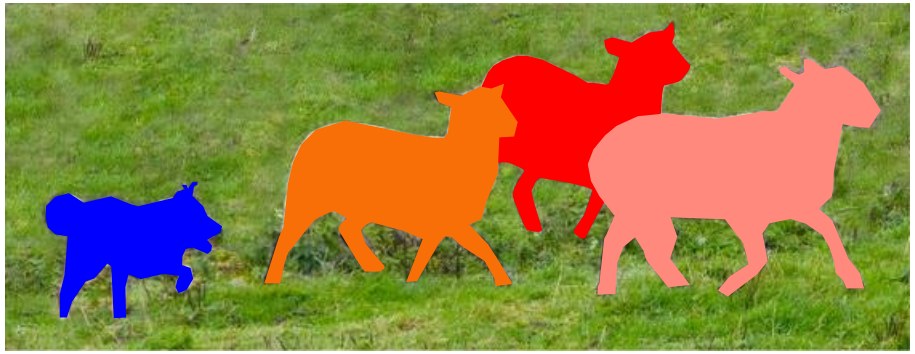
تشخیص اشیاء (Object Detection)



ناحیه‌بندی معنایی (Semantic Segmentation)



ناحیه‌بندی نمونه‌ها (Instance Segmentation)



دسته‌بندی و مکان‌یابی