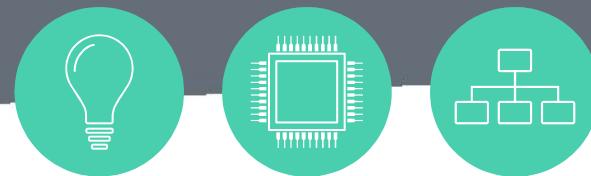


# یادگیری ماشین

وحید زهتاب



- با اقتباس از:

[1] Ba et al., Lecture 4 - CSC2516 - 2023

[2] MIT 6S191 - Lecture 3 - 2024



# مباحث این جلسه

بینایی ماشین

حدودیت‌های شبکه‌های عصبی

شبکه‌های عصبی پیچشی (CNN)

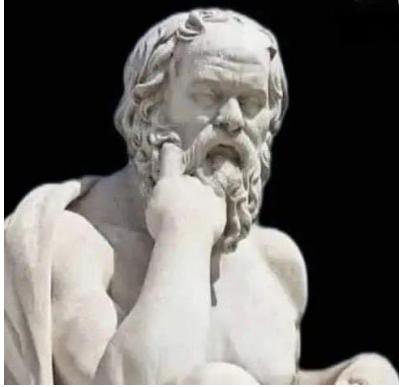
بررسی مدل‌های مختلف



# هدف

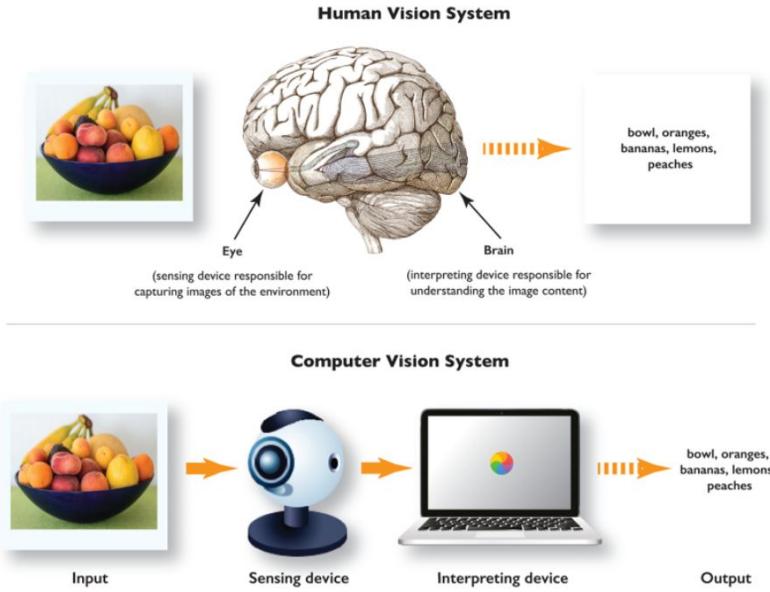
“Understanding a question  
is half an answer.”

— Socrates

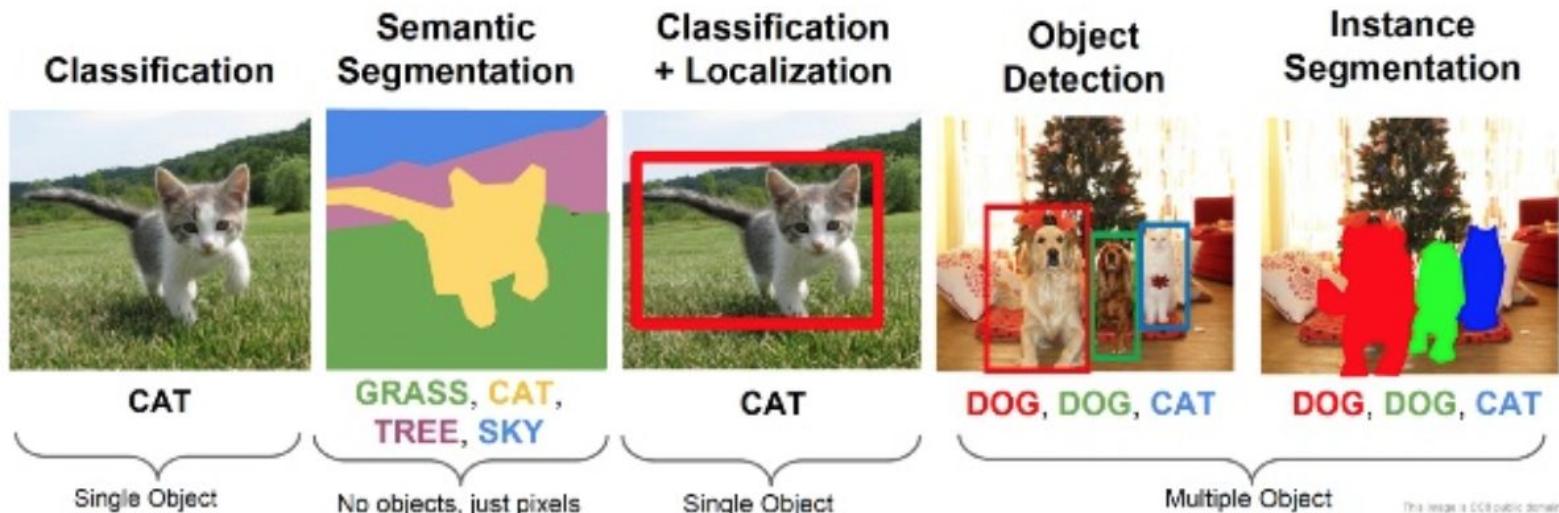


# (Computer Vision) بینایی ماشین

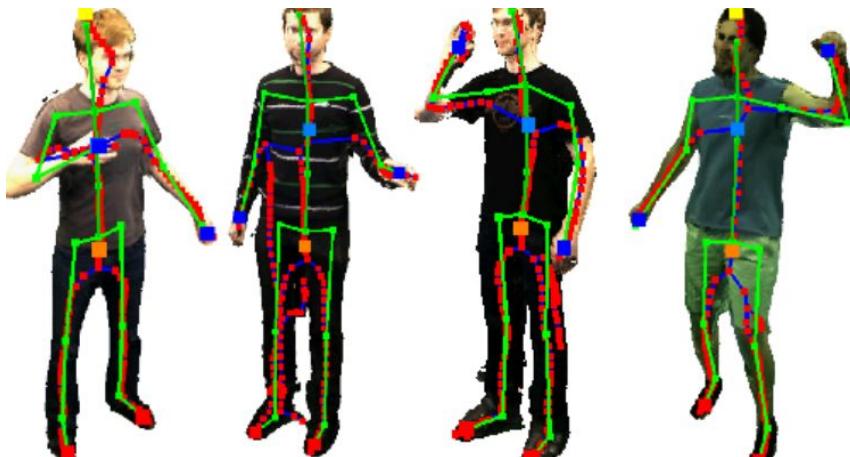
- تعریف: استخراج خودکار، تجزیه و تحلیل تصویر یا دنباله ای از تصاویر



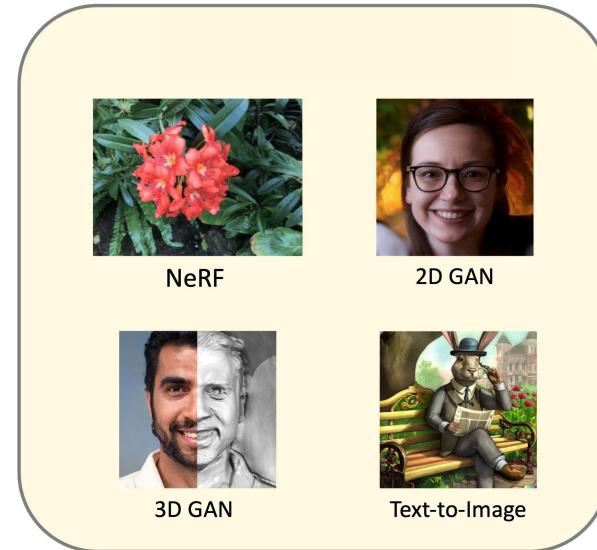
# چند وظیفه در بینایی ماشین



# چند وظیفه دیگر در بینایی ماشین



Pose estimation



Generative Models



# کامپیوٹر چہ می بیند؟

What you see



Input Image

What you both see

167	163	174	168	160	162	128	163	172	162	166	156
185	182	163	74	75	62	93	17	110	210	180	154
180	180	80	14	94	6	10	93	48	108	158	181
206	198	6	124	131	111	120	204	166	15	54	180
194	68	157	251	237	239	239	238	237	87	71	201
172	198	207	233	233	214	220	239	228	98	74	296
188	88	179	209	186	215	211	158	139	76	20	169
189	97	168	84	16	168	134	11	31	62	23	148
199	168	181	193	168	227	178	143	182	186	36	190
205	174	156	252	236	231	149	178	228	43	95	294
190	216	116	149	236	187	46	150	79	38	218	241
190	224	147	198	227	210	127	102	36	101	255	224
190	214	173	46	128	143	96	90	2	109	249	215
187	196	235	78	1	81	47	0	6	217	256	211
183	202	237	148	0	0	12	108	200	138	243	236
195	206	123	207	177	121	128	200	175	18	94	218

Input Image + values

What the computer "sees"

157	153	174	168	150	162	129	181	172	141	155	166
185	182	163	74	75	62	93	17	110	210	180	154
180	180	80	14	94	6	10	93	48	106	159	181
206	198	6	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	238	237	87	71	201
172	195	207	233	233	214	220	229	228	98	74	296
188	88	179	209	186	215	211	158	139	75	20	169
189	97	168	84	16	168	134	11	31	62	22	148
199	168	181	193	168	227	178	143	182	186	36	190
205	174	156	252	236	231	149	178	228	43	95	294
190	216	116	149	236	187	46	150	79	38	218	241
190	224	147	198	227	210	127	102	36	101	255	224
190	214	173	46	128	143	96	90	2	109	249	215
187	196	235	78	1	81	47	0	6	217	256	211
183	202	237	148	0	0	12	108	200	138	243	236
195	206	123	207	177	121	128	200	175	13	96	218

Pixel intensity values  
("pix-el"=picture-element)

Levin / Image Processing & Computer Vision

An image is just a matrix of numbers [0,255]  
i.e., 1080x1080x3 for an RGB image



# کامپیوٹر چگونہ رنگی می بیند؟



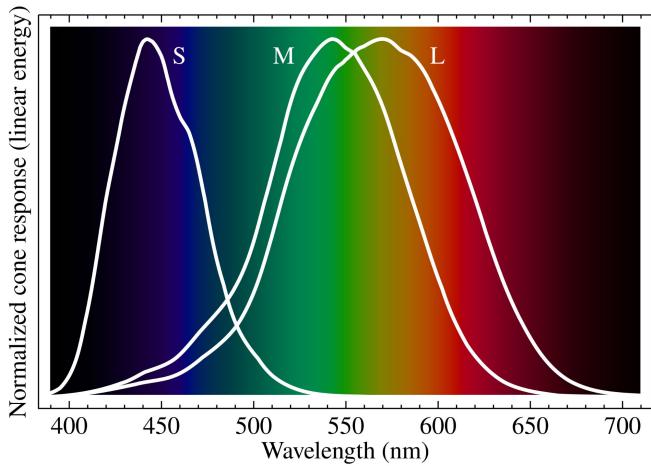
Red



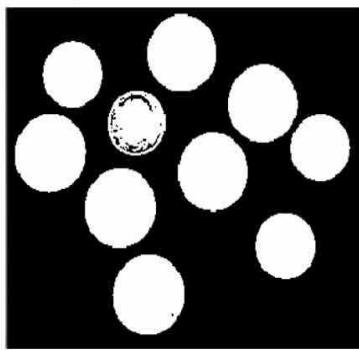
Green



Blue



# پردازش تصویر دیجیتال



**Image Thresholding**

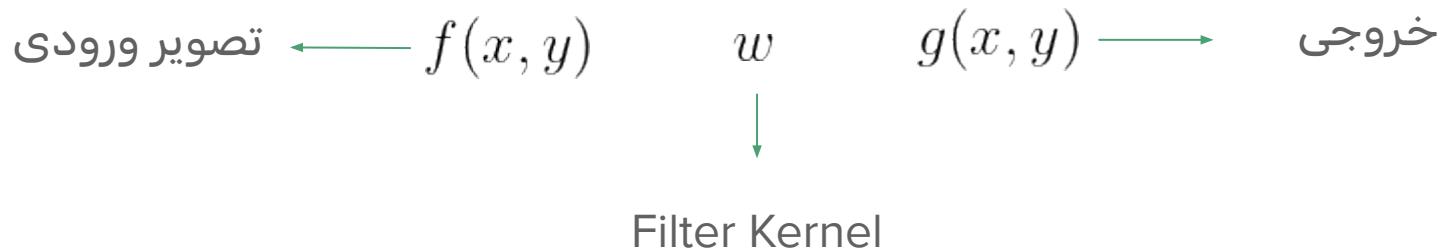


**Edge Detection**



# Convolution Filter

$$g(x, y) = w * f(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b w(i, j)f(x - i, y - j)$$



# Convolution Representation

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

مطالعه‌ی بیشتر و توضیحات



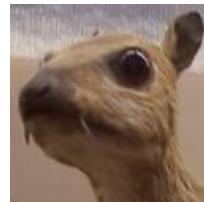
# فیلتر های پردازش تصویر سنتی



فیلتر تشخیص لبه

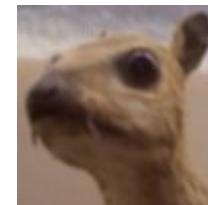
$$= \begin{bmatrix} 0 & -1 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \parallel$$

فیلتر  
sharpener



blur فیلتر

$$* \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$

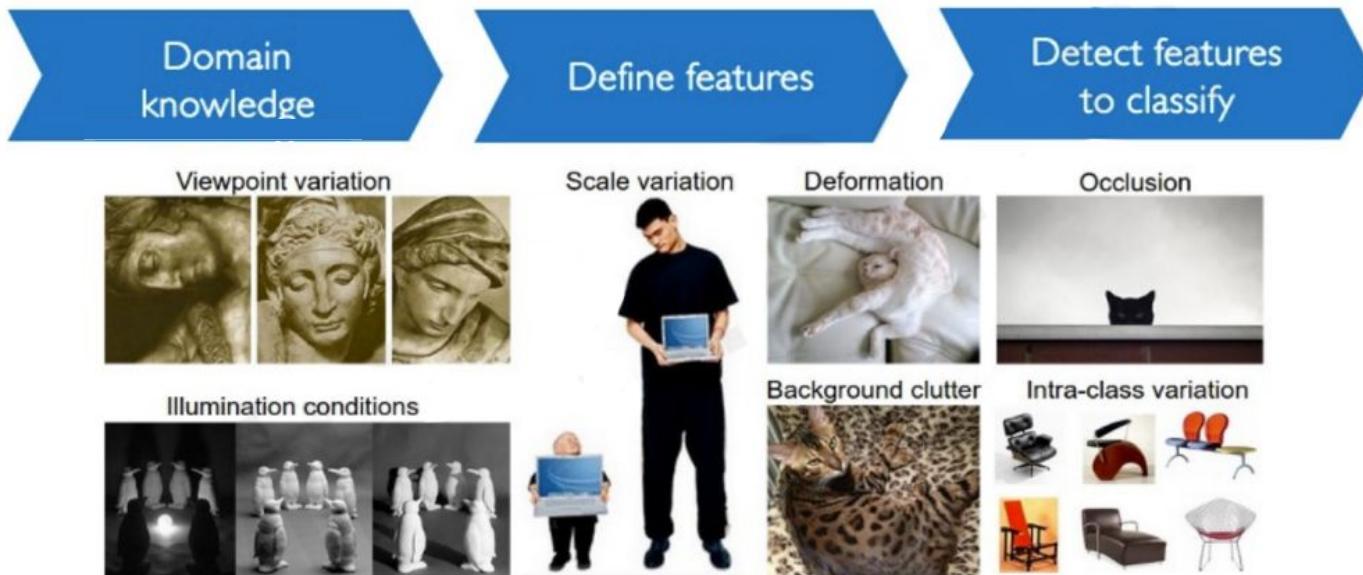


مثال تعاملی: آشنایی با فیلترها

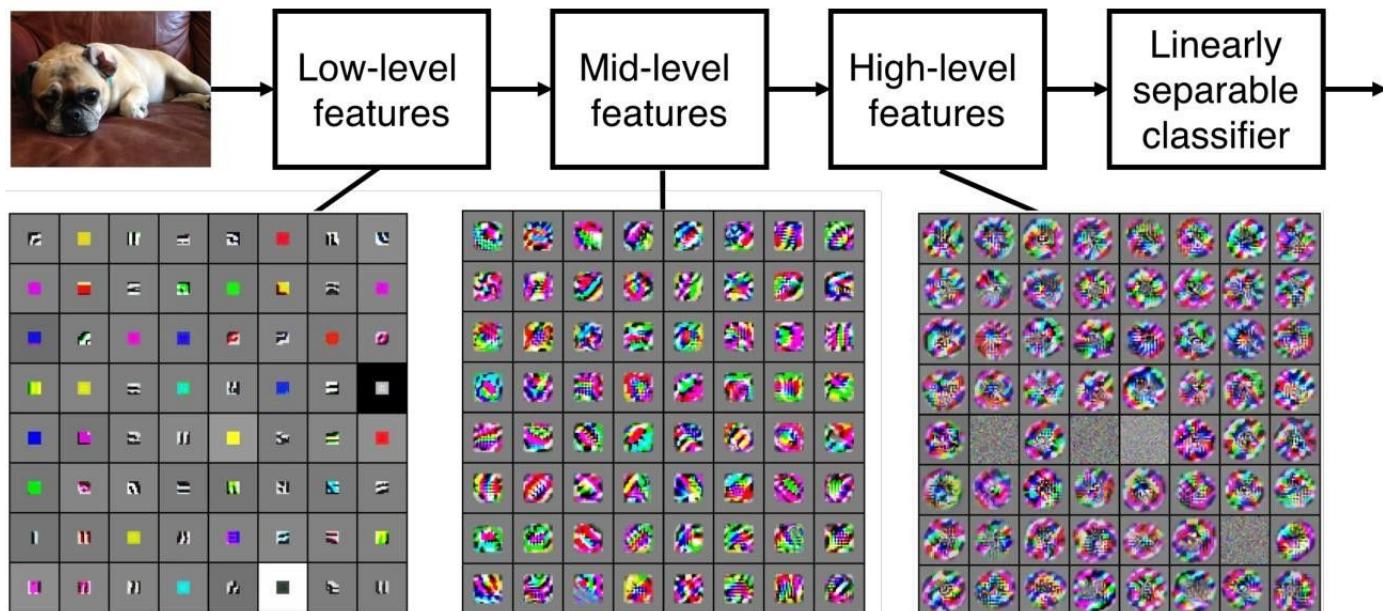


# مشکل پردازش تصویر سنتی

- استخراج ویژگی (Feature) به صورت دستی



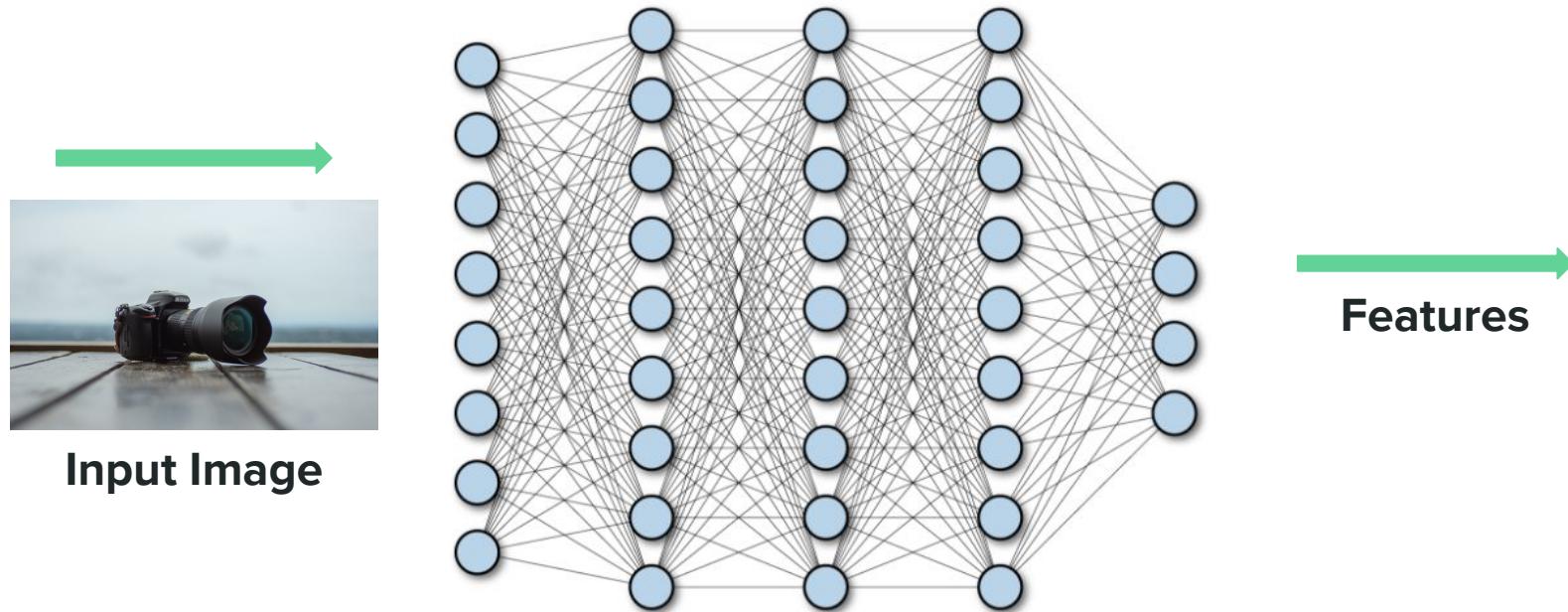
# استخراج نمایشی از ویژگی



آیا میتوانیم این ویژگی ها را یاد بگیریم؟

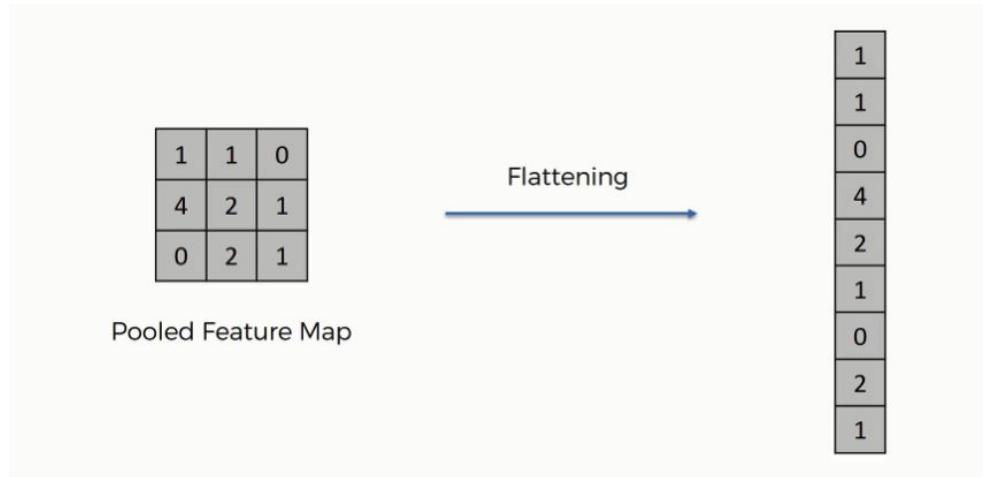


# استفاده از شبکه‌های عصبی چند لایه



# استفاده از شبکه‌های عصبی چند لایه

- تصاویر دو بعدی به صورت یک بردار یک بعدی رفتار می‌شوند (flatten)
- روابط مکانی در نظر گرفته نمی‌شوند



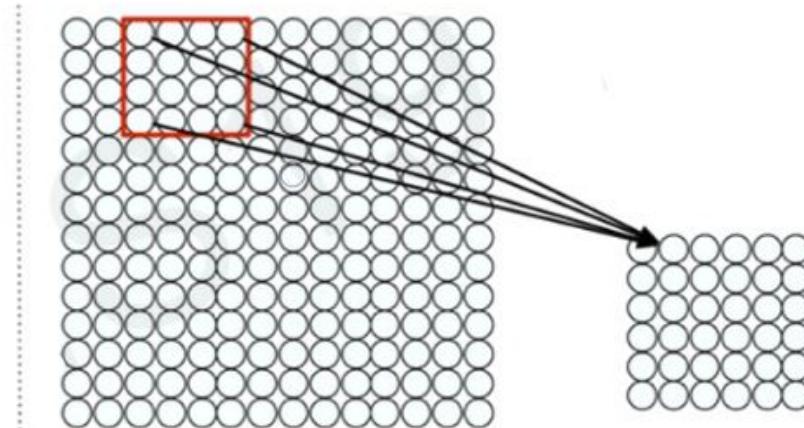
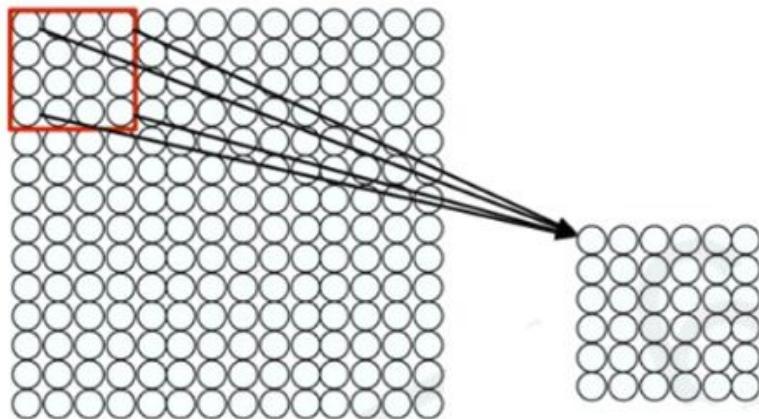
# نیازمندی های شبکه

- شبکه های عصبی نسبت به شیفت متغیر هستند
- شبکه های میخواهیم که نسبت به شیفت رفتار ثابتی داشته باشند



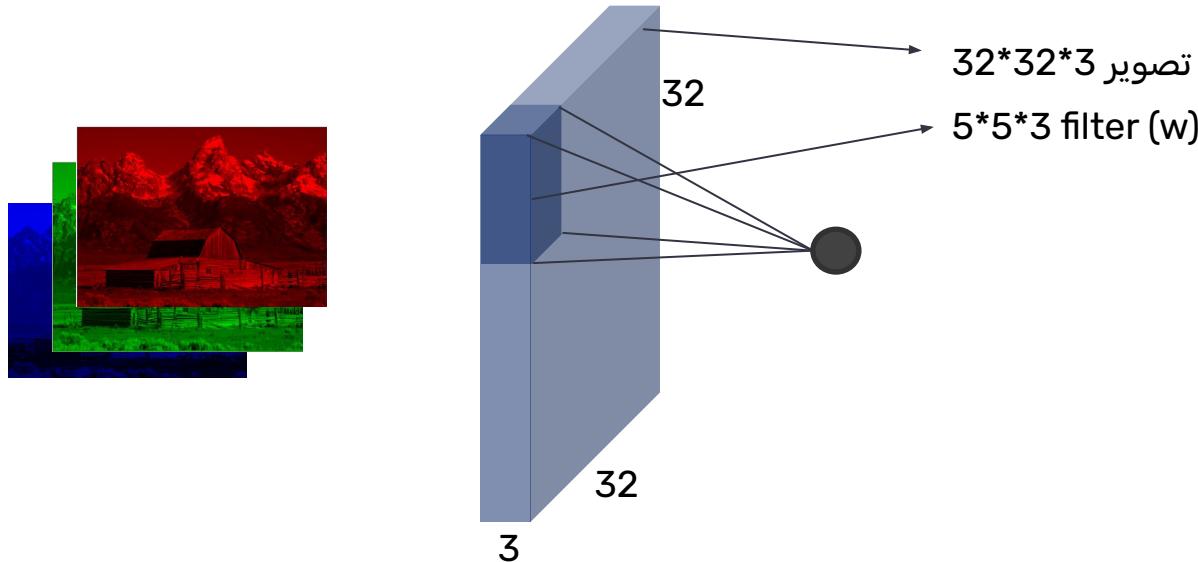
# استفاده از اطلاعات مکانی

- ایده وصل کردن پچ های ورودی به نورون های لایه های مخفی
- کانولوشن؟

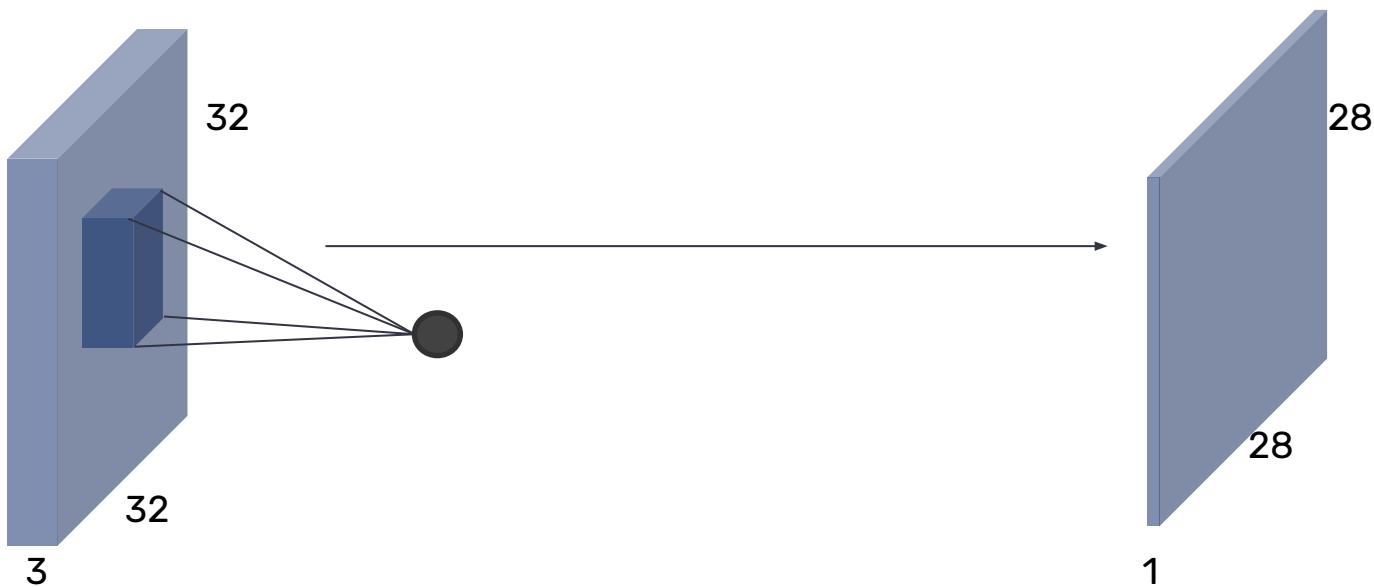


# شبکه های عصبی پیچشی (CNN)

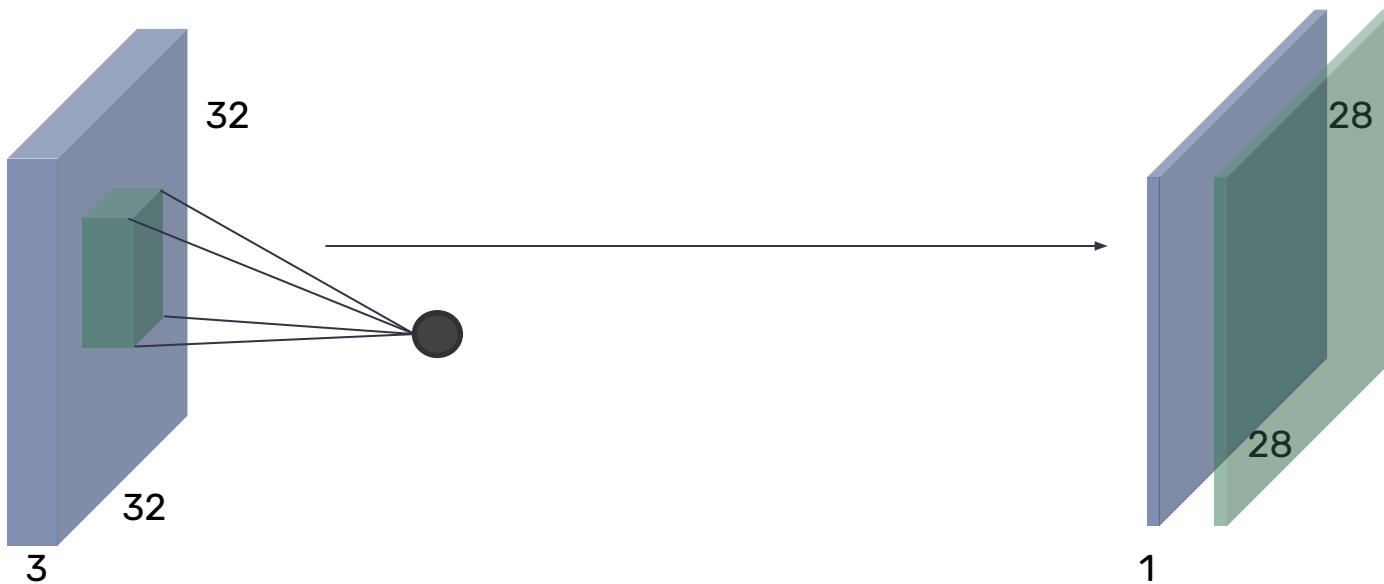
- بجای طراحی فیلترهای خاص به صورت محاسباتی و تحلیلی طراحی فیلتر را به شبکه های عصبی واگذار کنیم؟



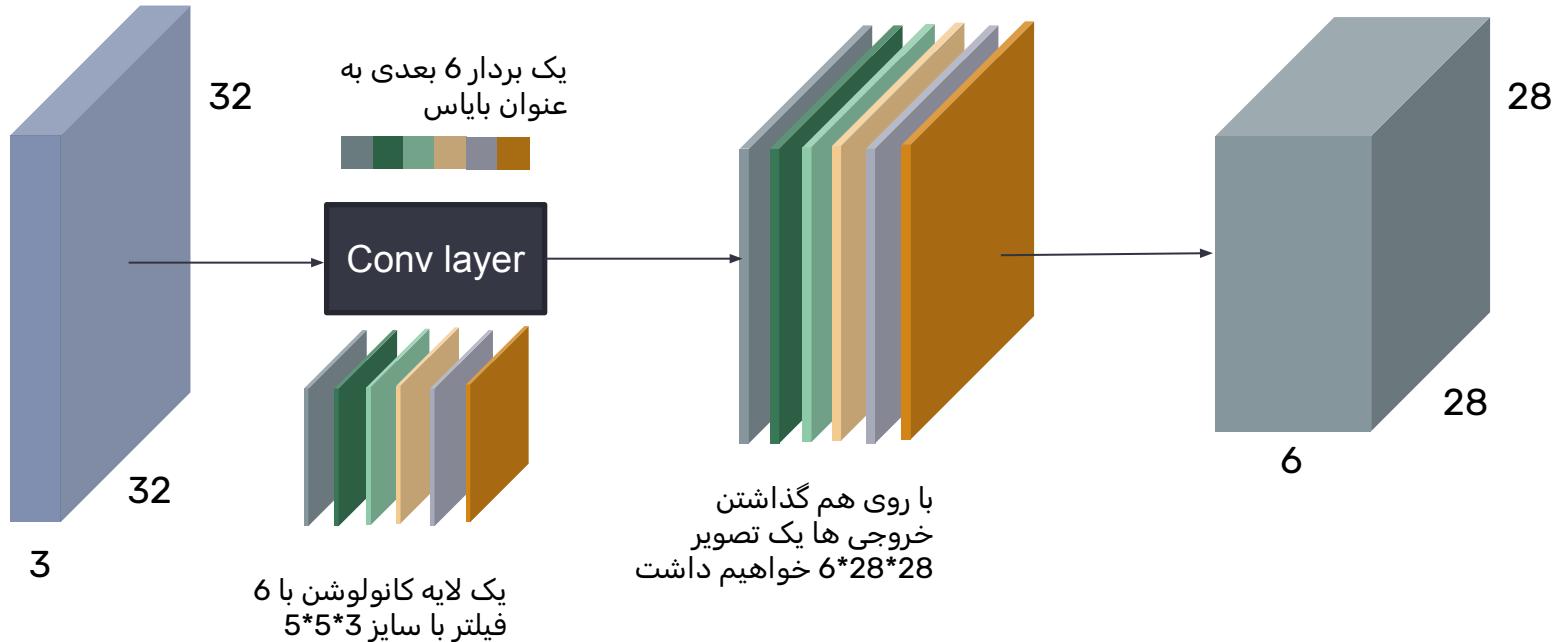
# ساخت فیلتر توسط مدل



# استفاده از فیلتر های متنوع

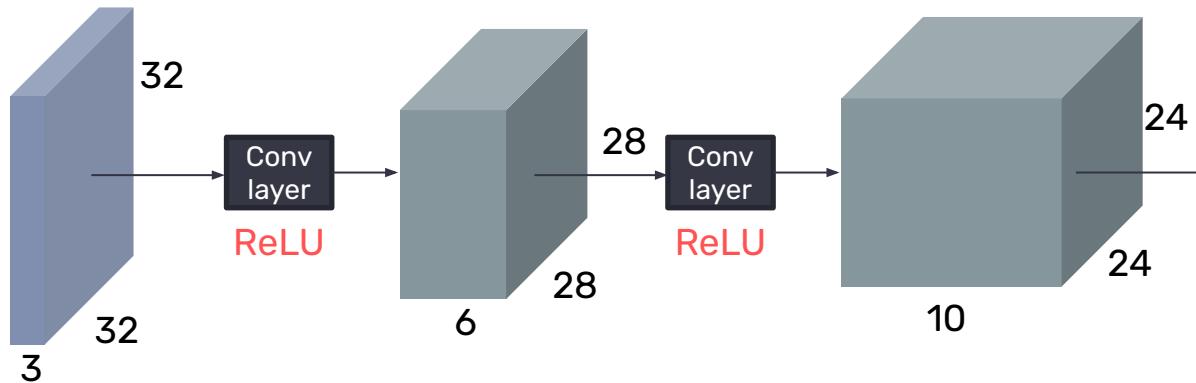


# یک لایه شبکه عصبی پیچشی

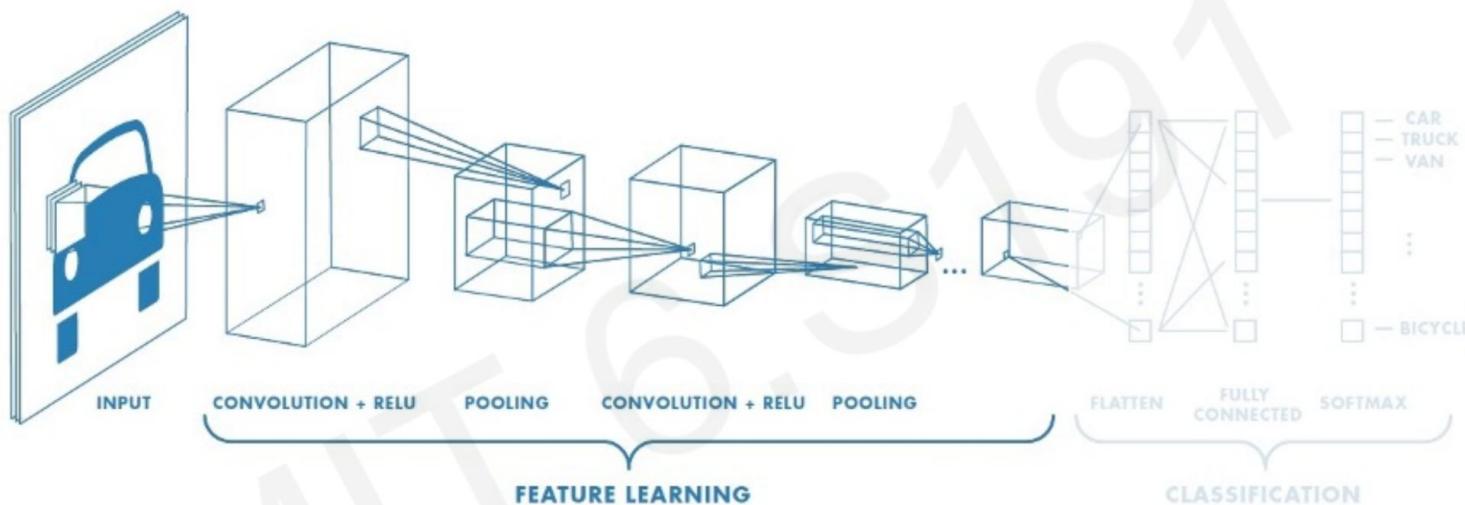


# یک شبکه عصبی پیچشی ساده

- متشکل از چند لایه کانولوشن که مابین آنها از توابع اکتیویشن استفاده می‌شود.



# One Network to Rule them All



1. Learn features in input image through **convolution**
2. Introduce **non-linearity** through activation function (real-world data is non-linear!)
3. Reduce dimensionality and preserve spatial invariance with **pooling**



# شبکه های عصبی پیچشی (CNN)

شبکه های عصبی پیچشی بطور کلی از سه نوع لایه تشکیل شده اند

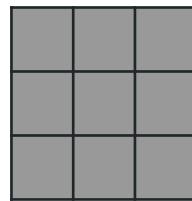
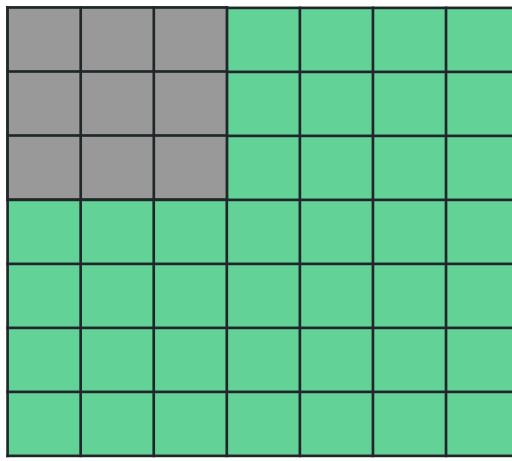
- لایه های پیچشی (Convolution Layer)
- لایه های تجمیع (Pooling Layer) : به منظور کاهش بعد در راستای طول و عرض
- لایه های کاملاً متصل (Fully-Connected Layer) : به منظور طبقه بندی و  
وظایف دیگر



# تنظیمات کانولوشن - پرش

فیلتر با سایز پرش (stride) متفاوت

Stride = 2



فیلتر  $3 \times 3$

وروودی  $7 \times 7$

○ فیلتر با سایز شیفت یک را بررسی کردیم

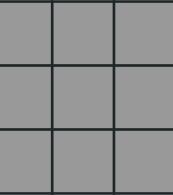
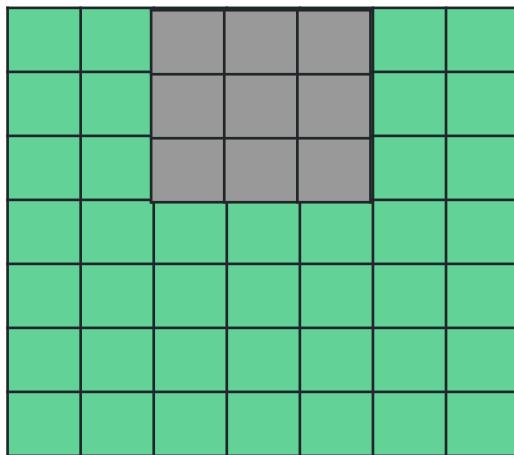


○ شیفت به اندازه دو

# تنظیمات کانولوشن - پرش

فیلتر با سایز پرش (stride) متفاوت

Stride = 2



فیلتر  $3 \times 3$

○ فیلتر با سایز شیفت یک را بررسی کردیم

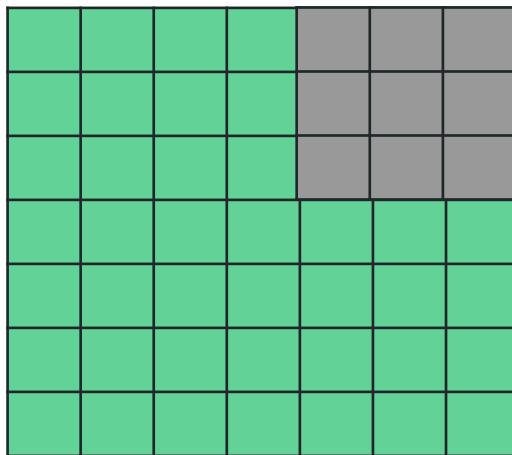


○ شیفت به اندازه دو

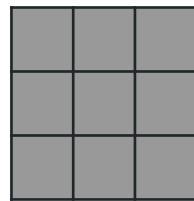
# تنظیمات کانولوشن - پرش

فیلتر با سایز پرش(stride) متفاوت

Stride = 2



ورودي  
 $7 \times 7$



فیلتر  
 $3 \times 3$

○ فیلتر با سایز شیفت یک را بررسی کردیم

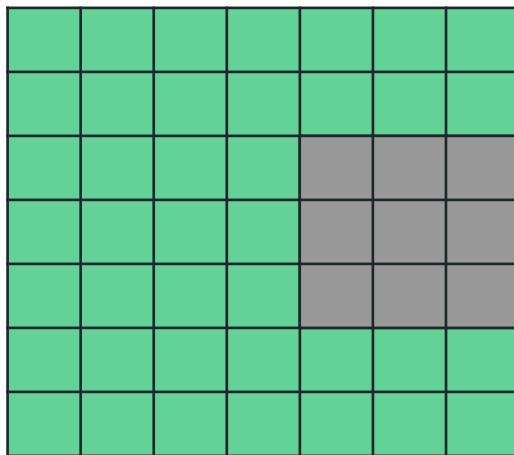


○ شیفت به اندازه دو

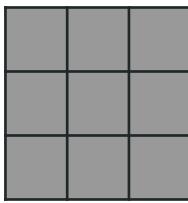
# تنظیمات کانولوشن - پرش

فیلتر با سایز پرش (stride) متفاوت

Stride = 2



7\*7  
ورودی



فیلتر 3\*3

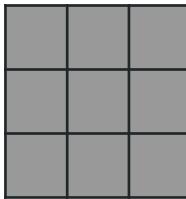
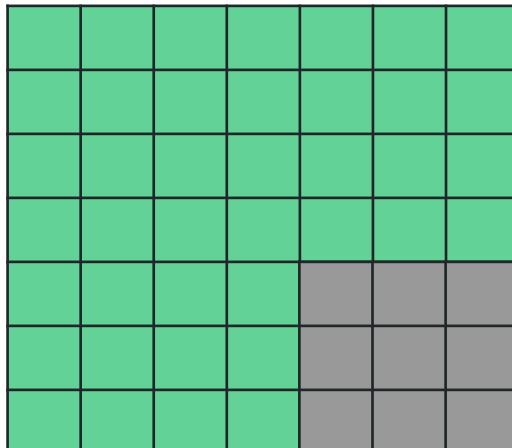
○ فیلتر با سایز شیفت یک را بررسی کردیم



○ شیفت به اندازه دو

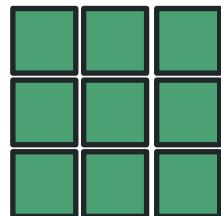
# تنظیمات کانولوشن - پرش

Stride = 2

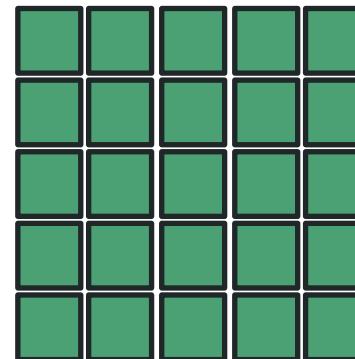


فیلتر  $3 \times 3$

- فیلتر با سایز شیفت یک را بررسی کردیم.



خروجی  $3 \times 3$   
stride=2



خروجی  $5 \times 5$   
stride=1

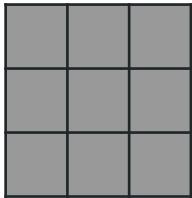
فیلتر با سایز پرش (stride) متفاوت

- شیفت به اندازه دو

# تنظیمات کانولوشن - Padding

0	0	0	0	0	0	0	0	0
0								0
0								0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0

ورودي  
7\*7



فیلتر 3\*3

Stride = 1  
Padding = 1  
N = 7  
f = 3

M = 7

$$M = \frac{N + 2P - f}{stride} + 1$$

$$Input = N \times N$$

$$Padded\ Input = (N + 2P) \times (N + 2P)$$

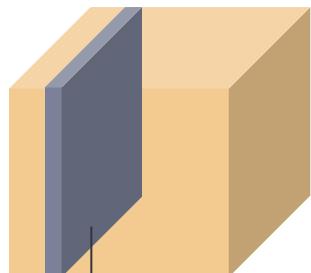
$$Filter = f \times f$$

$$Output = M \times M$$

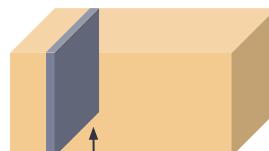


# Pooling layer

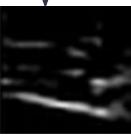
$224 \times 224 \times 64$



$112 \times 112 \times 64$



224



224

downsampling

112

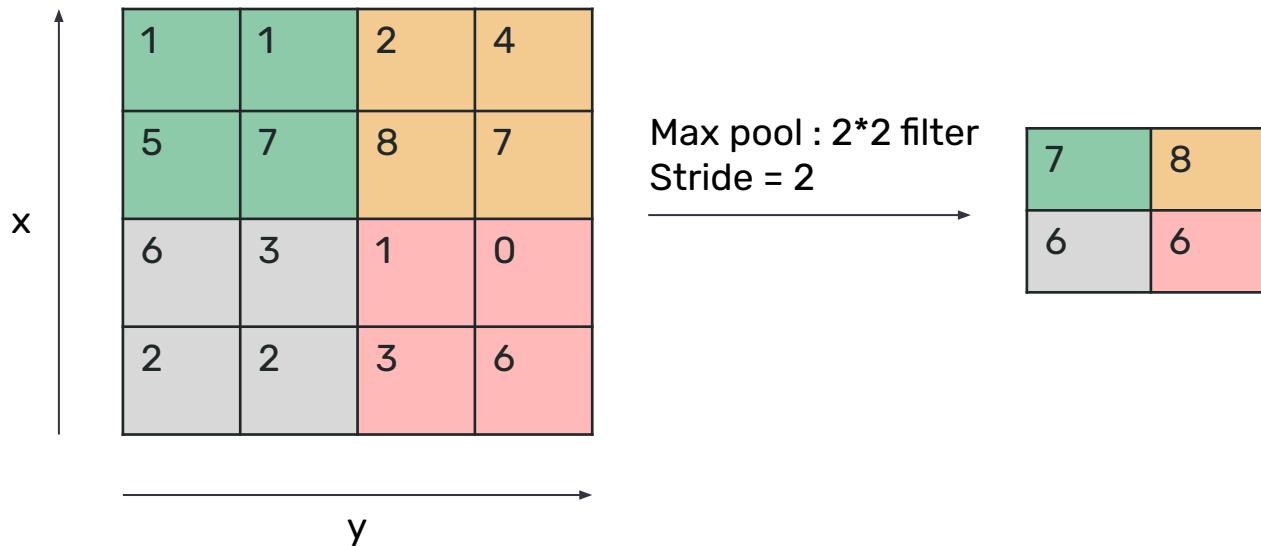
112

- بازنمایی کوچکتر

- اعمال جدالگانه روی هر کanal

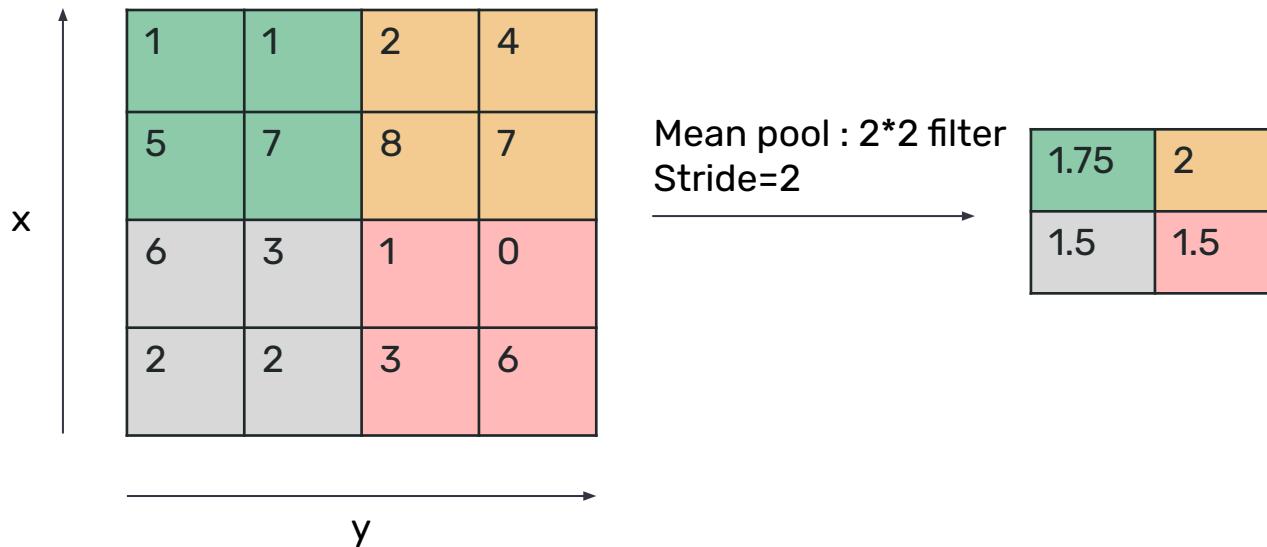
# Max Pool

یک برش از بعد کانال



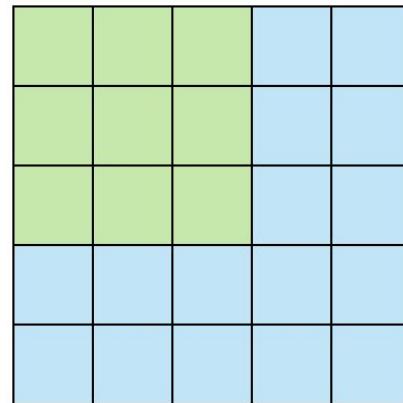
# Mean Pool

یک برش از بعد کانال

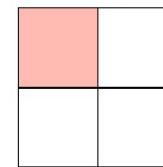


# Pooling layer

استفاده از کانولوشن با  $1 >$  stride منجر به کاهش ابعاد تصویر می‌شود.



Stride 2



Feature Map



# مثال از شمارش پارامترها

ورودی  $3*32*32$

10 فیلتر با سایز  $3*5*5$

Stride:1

Padding:2

تعداد پارامترها در این لایه:

$$76 = 1 + 5*5*3 \quad \bullet$$

$$760 = 76*10 \quad \bullet$$

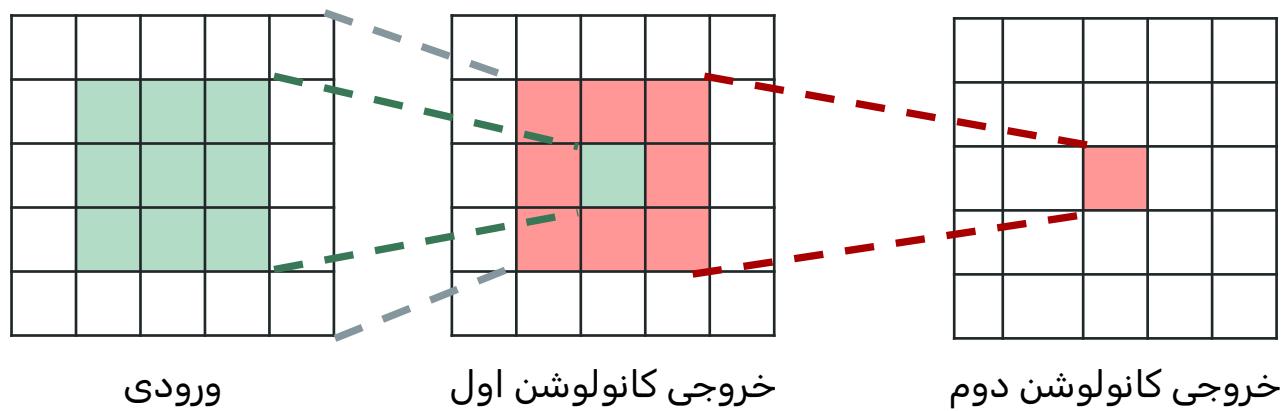
چه مزایایی نسبت به شبکه‌های عصبی چند لایه (MLP) دارد؟



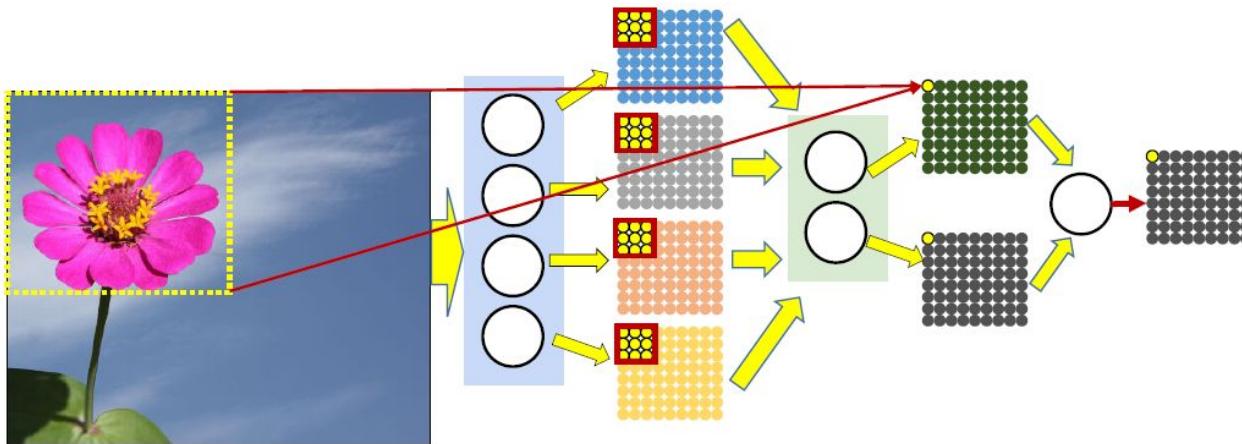
# Receptive Field

هر نورون چه محدوده‌ای را می‌تواند پوشش دهد؟

مثال: دو لایه فیلتر  $3 \times 3$  می‌تواند به اندازه یک فیلتر  $5 \times 5$  به لحاظ میدان پوشش عمل کند.



# انتزاعی شدن مفاهیم در لایه های عمیق



هر لایه بطور موثر تعداد پیکسل‌های لایه قبل را ارزیابی میکند که در نهایت کل تصویر ارزیابی میشود



# شبکه های عصبی پیچشی (CNN)

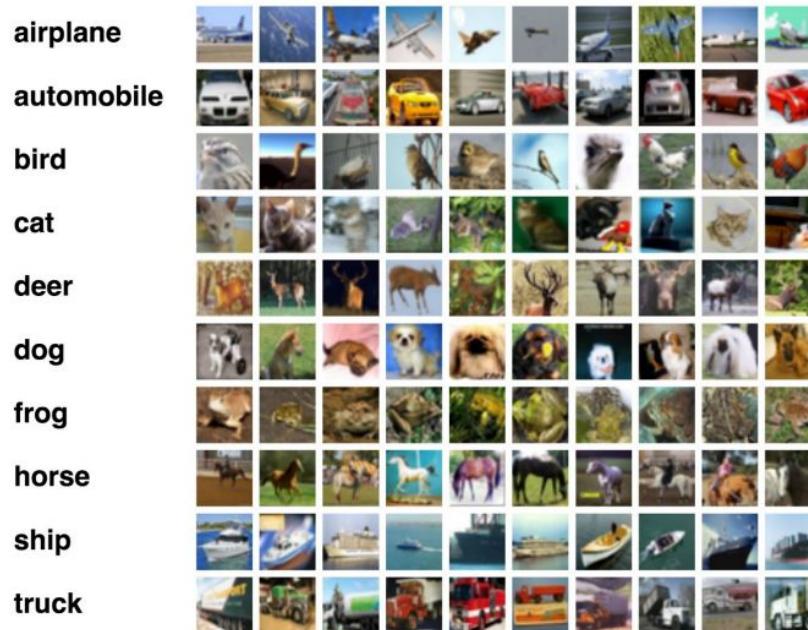
یادگیری الگوها در طی چند لایه چه فایده‌ای دارد؟

تعمیم پذیری بهتر

تعداد پارامتر های کمتر

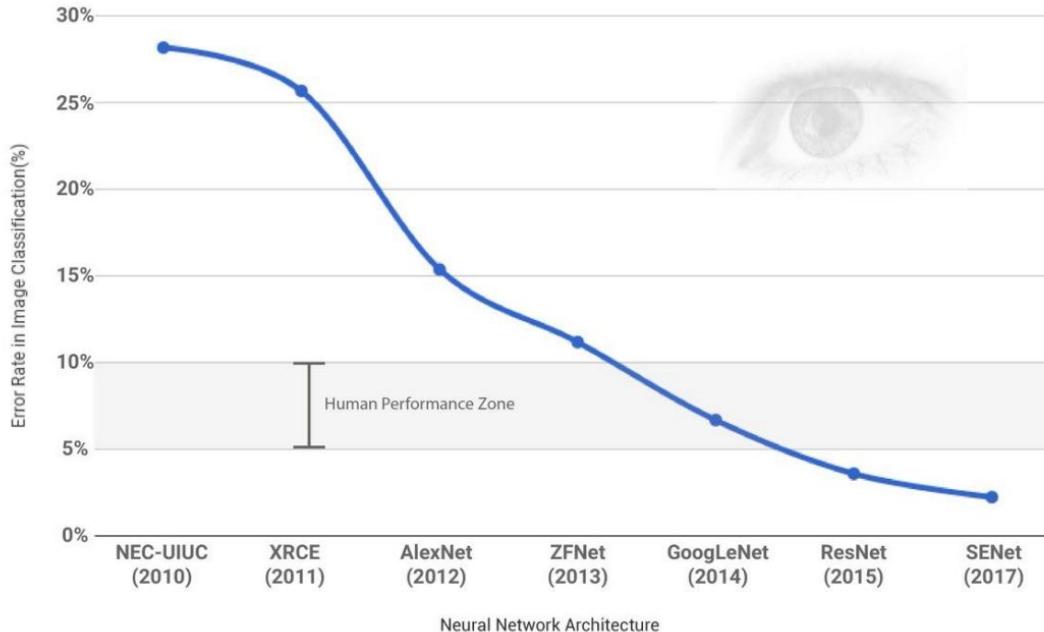


# Image net



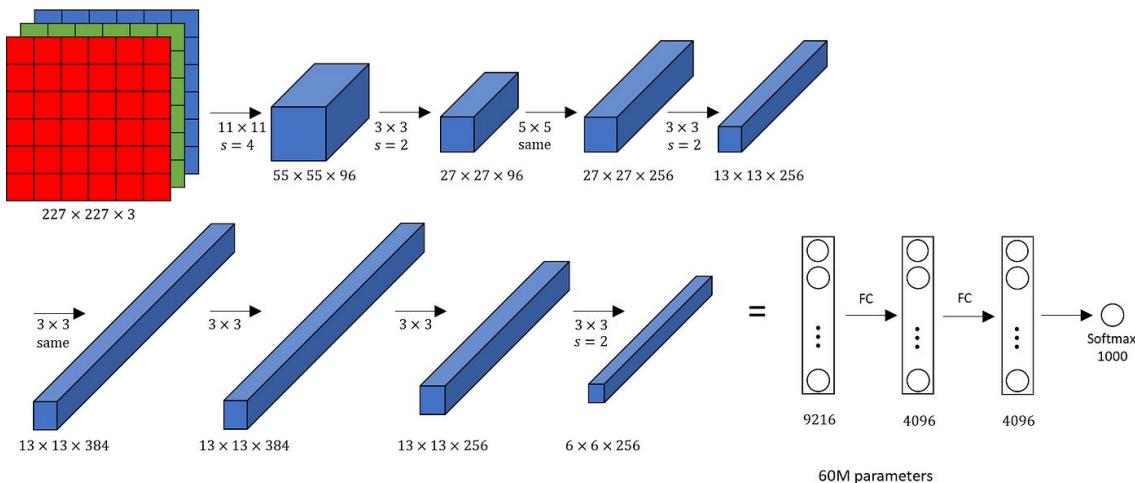
یک میلیون نمونه عکس هزار کتگوری

# مدل‌ها



- SOTA: ImageNet Benchmark

# AlexNet



استفاده از ReLU



استفاده از augmentation



Dropout 0.5



Batch size = 128



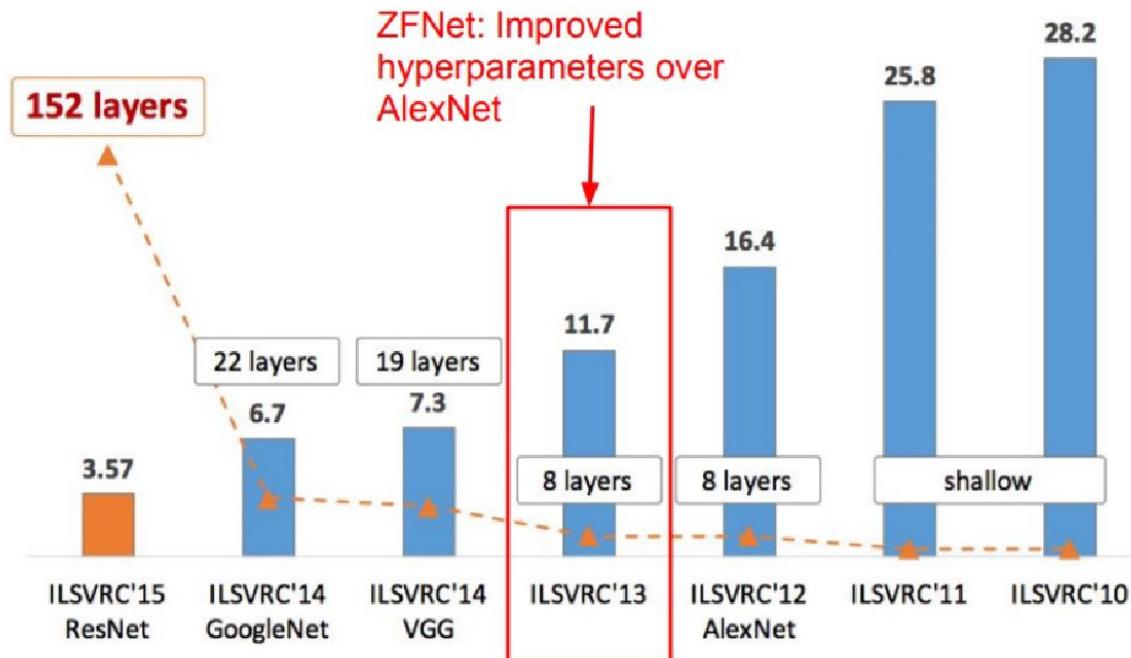
SGD Momentum



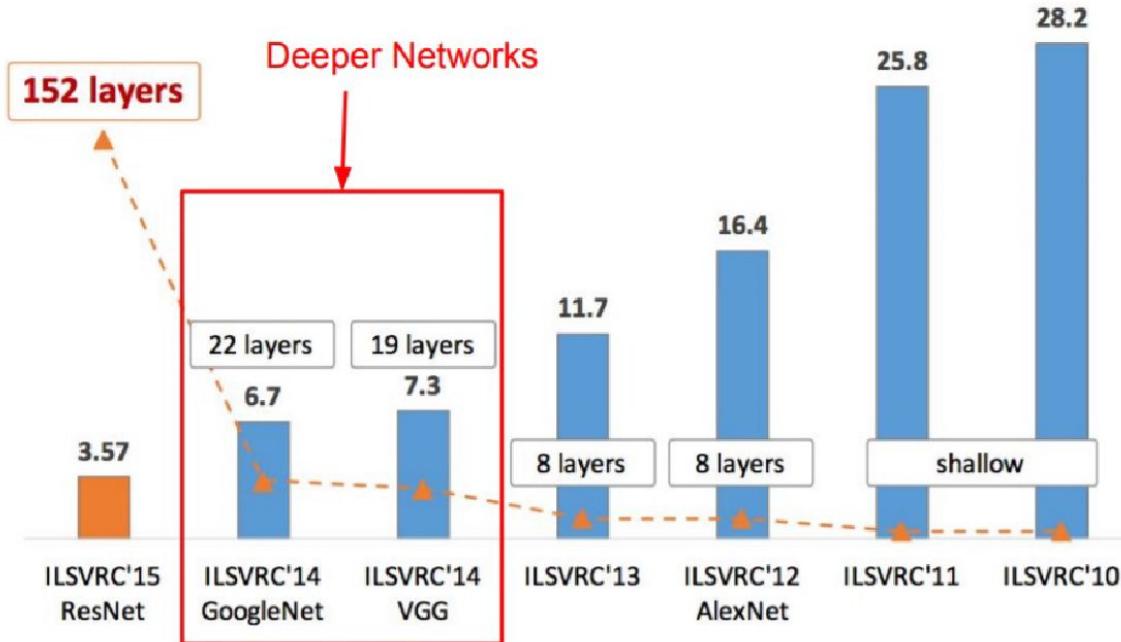
[1] Krizhevsky et al. ImageNet Classification with Deep Convolutional Neural Networks



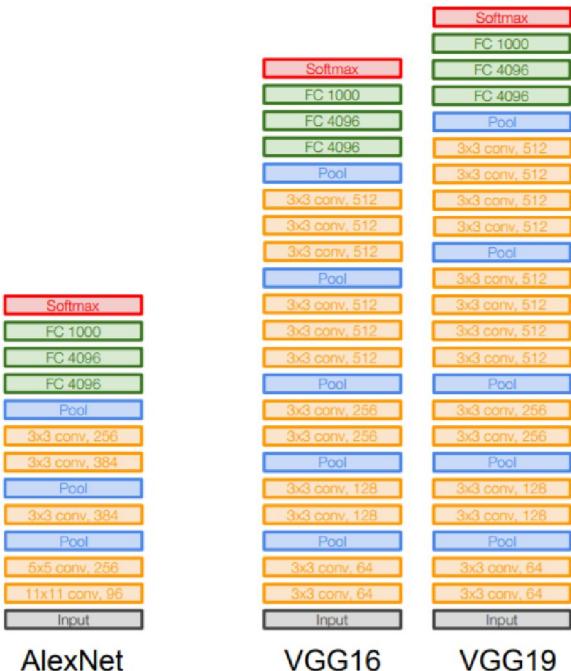
# معرفی مدل‌ها



# معرفی مدل‌ها



# VGGNet



## شبکه عمیق‌تر:

VGG16,VGG19 پڑھے 16-19 <- (AlexNet) پڑھے 8 ○

## استفاده از فیلتر ۳\*۳ فقط:

Stride 1, pad 1    ○

max pool stride 2 2\*2 ○

**پادآوری: چرا استفاده فیلتر ۳\*۳ فقط؟**

عميق، تر شدن، receptive فيلد يکسان

پارامتر کمتر و اکتیویشن بیشتر



# VGGNet

INPUT: [224x224x3] memory:  $224 \times 224 \times 3 = 150K$  params: 0 (not counting biases)

CONV3-64: [224x224x64] memory:  $224 \times 224 \times 64 = 3.2M$  params:  $(3 \times 3 \times 3) \times 64 = 1,728$

CONV3-64: [224x224x64] memory:  $224 \times 224 \times 64 = 3.2M$  params:  $(3 \times 3 \times 64) \times 64 = 36,864$

POOL2: [112x112x64] memory:  $112 \times 112 \times 64 = 800K$  params: 0

CONV3-128: [112x112x128] memory:  $112 \times 112 \times 128 = 1.6M$  params:  $(3 \times 3 \times 64) \times 128 = 73,728$

CONV3-128: [112x112x128] memory:  $112 \times 112 \times 128 = 1.6M$  params:  $(3 \times 3 \times 128) \times 128 = 147,456$

POOL2: [56x56x128] memory:  $56 \times 56 \times 128 = 400K$  params: 0

CONV3-256: [56x56x256] memory:  $56 \times 56 \times 256 = 800K$  params:  $(3 \times 3 \times 128) \times 256 = 294,912$

CONV3-256: [56x56x256] memory:  $56 \times 56 \times 256 = 800K$  params:  $(3 \times 3 \times 256) \times 256 = 589,824$

CONV3-256: [56x56x256] memory:  $56 \times 56 \times 256 = 800K$  params:  $(3 \times 3 \times 256) \times 256 = 589,824$

POOL2: [28x28x256] memory:  $28 \times 28 \times 256 = 200K$  params: 0

CONV3-512: [28x28x512] memory:  $28 \times 28 \times 512 = 400K$  params:  $(3 \times 3 \times 256) \times 512 = 1,179,648$

CONV3-512: [28x28x512] memory:  $28 \times 28 \times 512 = 400K$  params:  $(3 \times 3 \times 512) \times 512 = 2,359,296$

CONV3-512: [28x28x512] memory:  $28 \times 28 \times 512 = 400K$  params:  $(3 \times 3 \times 512) \times 512 = 2,359,296$

POOL2: [14x14x512] memory:  $14 \times 14 \times 512 = 100K$  params: 0

CONV3-512: [14x14x512] memory:  $14 \times 14 \times 512 = 100K$  params:  $(3 \times 3 \times 512) \times 512 = 2,359,296$

CONV3-512: [14x14x512] memory:  $14 \times 14 \times 512 = 100K$  params:  $(3 \times 3 \times 512) \times 512 = 2,359,296$

CONV3-512: [14x14x512] memory:  $14 \times 14 \times 512 = 100K$  params:  $(3 \times 3 \times 512) \times 512 = 2,359,296$

POOL2: [7x7x512] memory:  $7 \times 7 \times 512 = 25K$  params: 0

FC: [1x1x4096] memory:  $4096$  params:  $7 \times 7 \times 512 \times 4096 = 102,760,448$

FC: [1x1x4096] memory:  $4096$  params:  $4096 \times 4096 = 16,777,216$

FC: [1x1x1000] memory:  $1000$  params:  $4096 \times 1000 = 4,096,000$

Note:

Most memory is in early CONV

Most params are in late FC

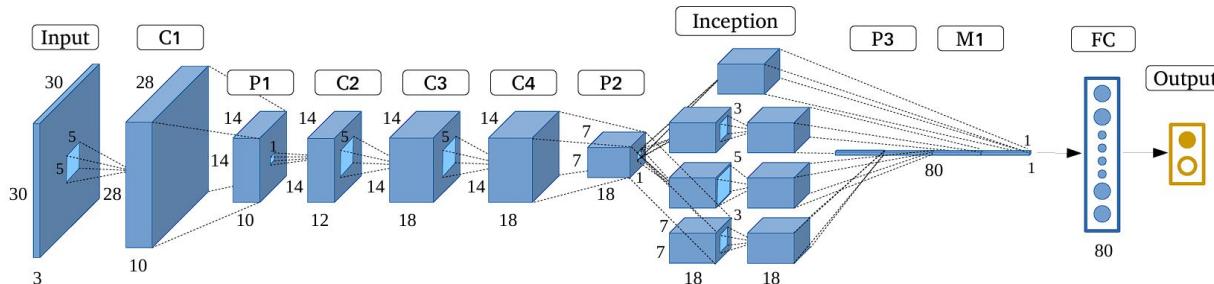
TOTAL memory:  $24M * 4$  bytes  $\approx 96MB / \text{image}$  (only forward!  $\sim 2$  for bwd)

TOTAL params: 138M parameters



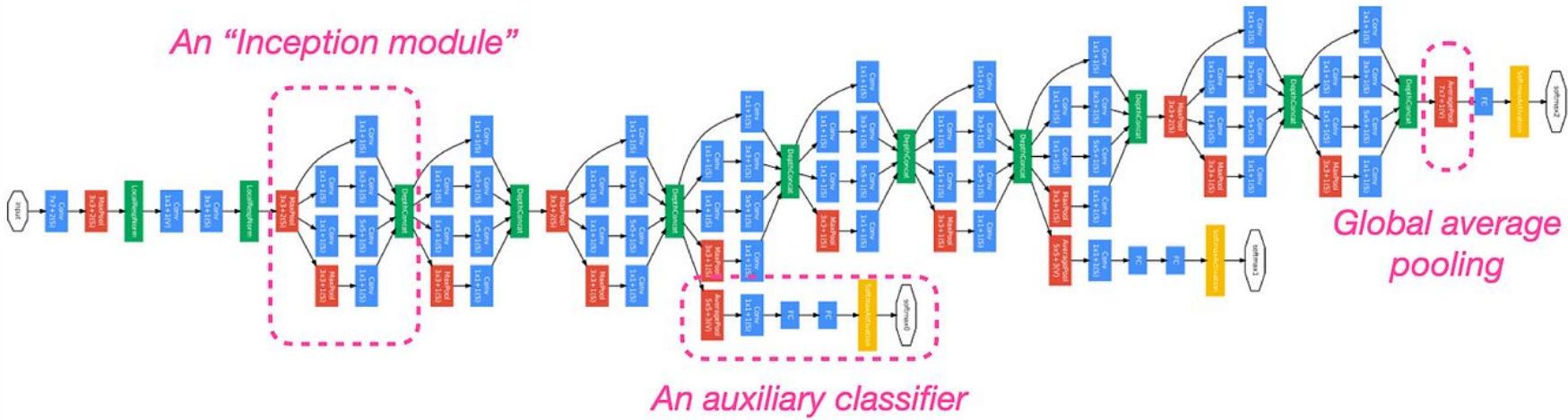
# GoogLeNet

- 22 لایه
- استفاده از مازول
- عدم استفاده لایه Fully connected
- 5 میلیون پارامتر (12 برابر کمتر از AlexNet)

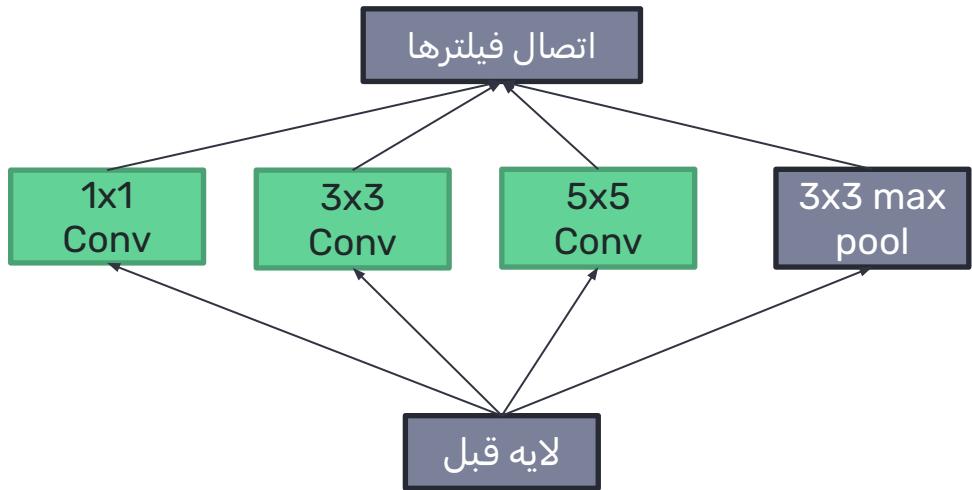


# GoogLeNet

## An “Inception module”



# Inception Module



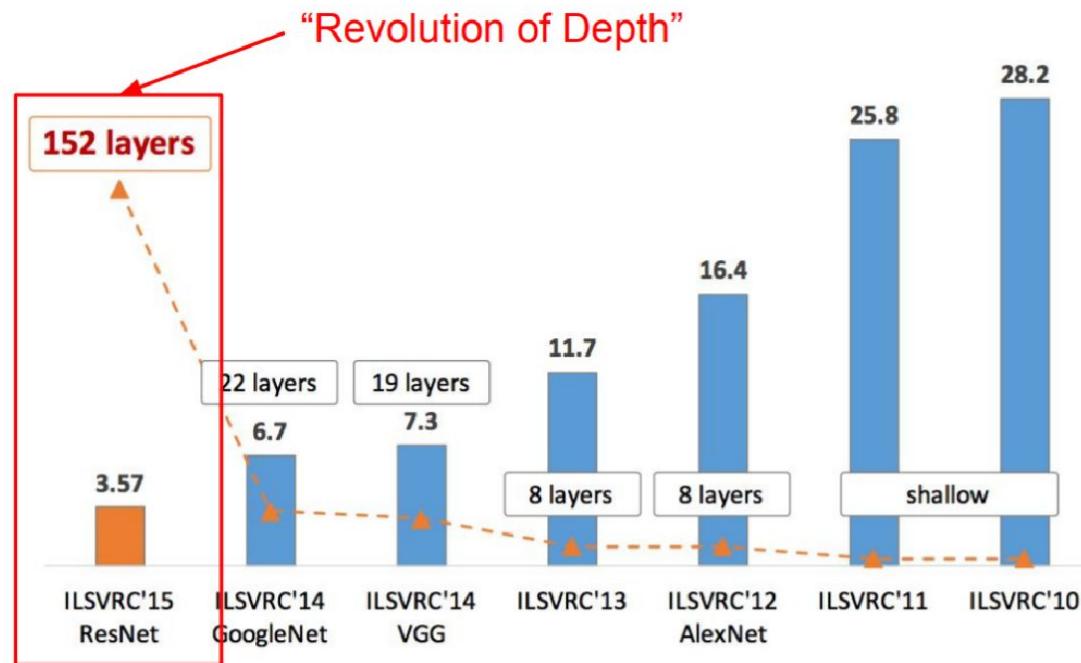
( $5 \times 5$ ,  $3 \times 3$ ,  $1 \times 1$ ) Receptive field

متصل کردن خروجی‌ها در راستای کانال

- 
- 

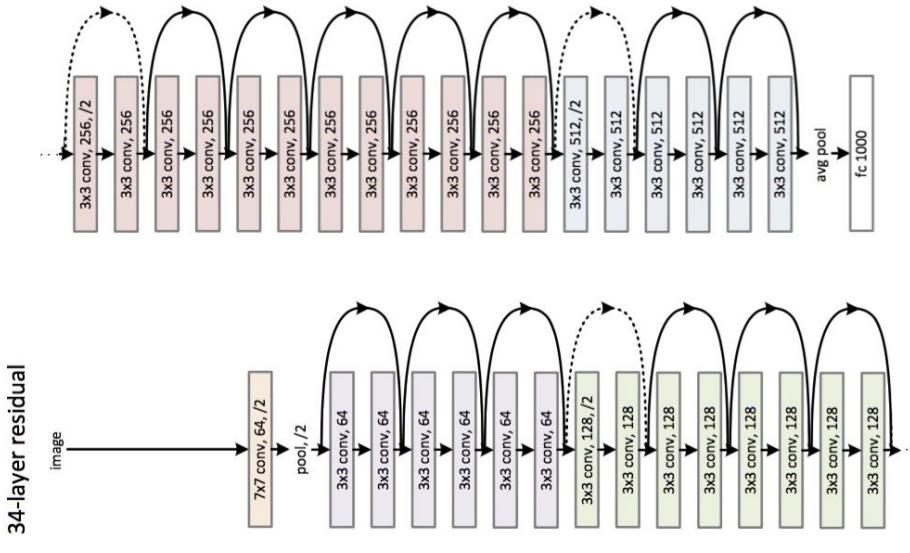


# معرفی مدل‌ها



# ResNet

شبکه‌های بسیار عمیق با یال موازی (residual connection)



ResNet-34

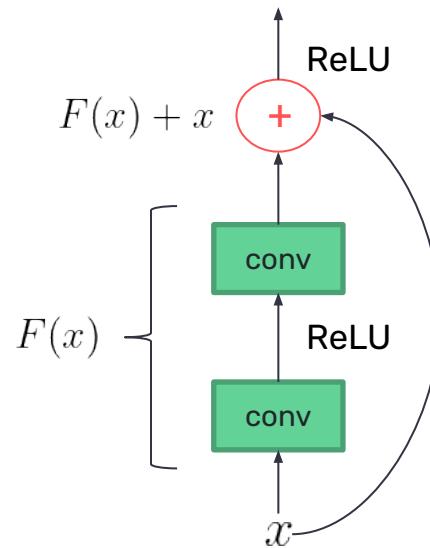
# ResNet

• جلوگیری از ناپدید شدن گرادیان (Vanishing Gradient)

$$\begin{aligned}x^{l+1} &= x^l + F(x^l) \\x^{l+2} &= x^{l+1} + F(x^{l+1}) \\x^{l+2} &= x^l + F(x^l) + F(x^{l+1}) \\x^L &= x^l + \sum_{i=l}^{L-1} F(x^i)\end{aligned}$$

$$\frac{\partial E}{\partial x^l} = \frac{\partial E}{\partial x^L} \frac{\partial x^L}{\partial x^l} = \frac{\partial E}{\partial x^L} \left( 1 + \frac{\partial}{\partial x^l} \sum_{i=l}^{L-1} F(x^i) \right)$$

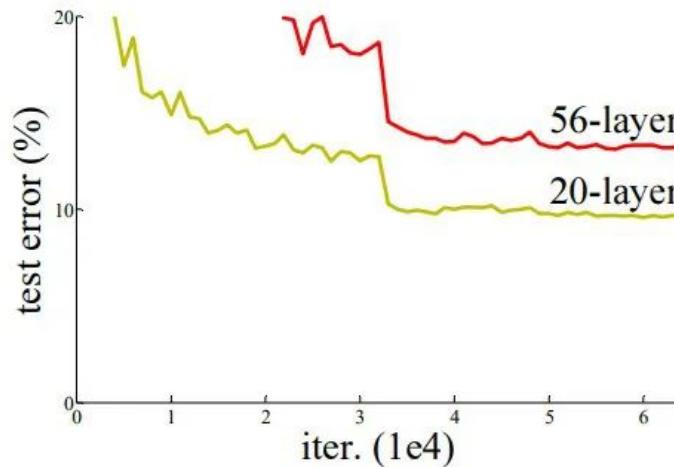
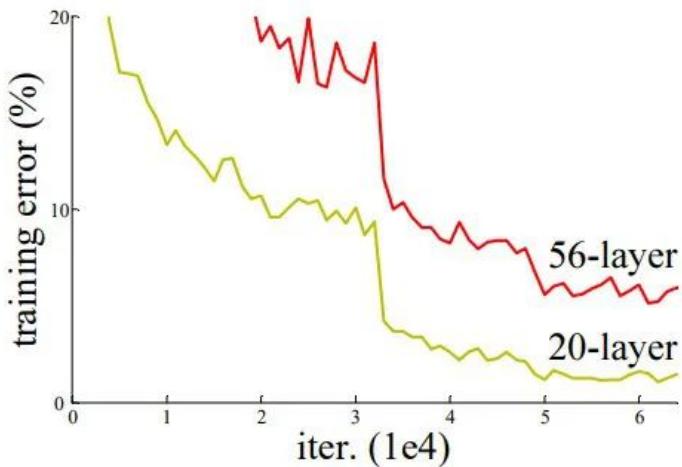
هر  $\frac{\partial E}{\partial x^l}$  افزایشی است و احتمال کم vanishing میشود.



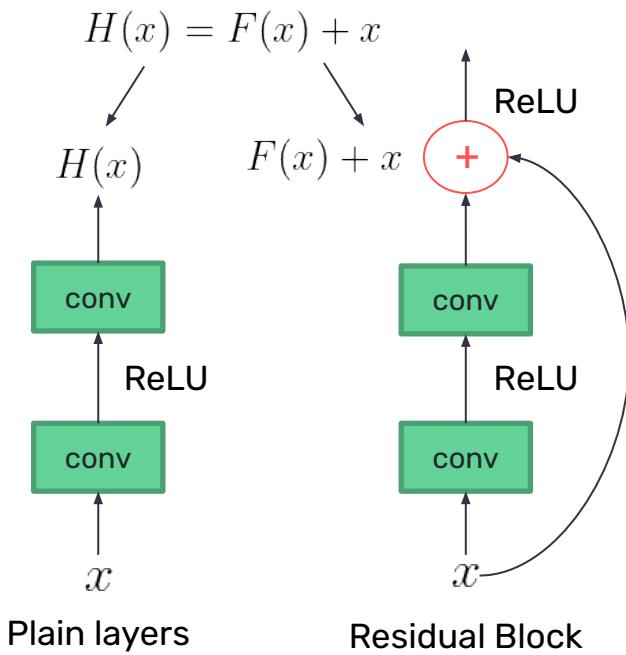
# ResNet

شبکه‌های عمیق اورفیت نمی‌شوند

**فرضیه:** بهینه‌سازی شبکه‌های عمیق سخت‌تر است

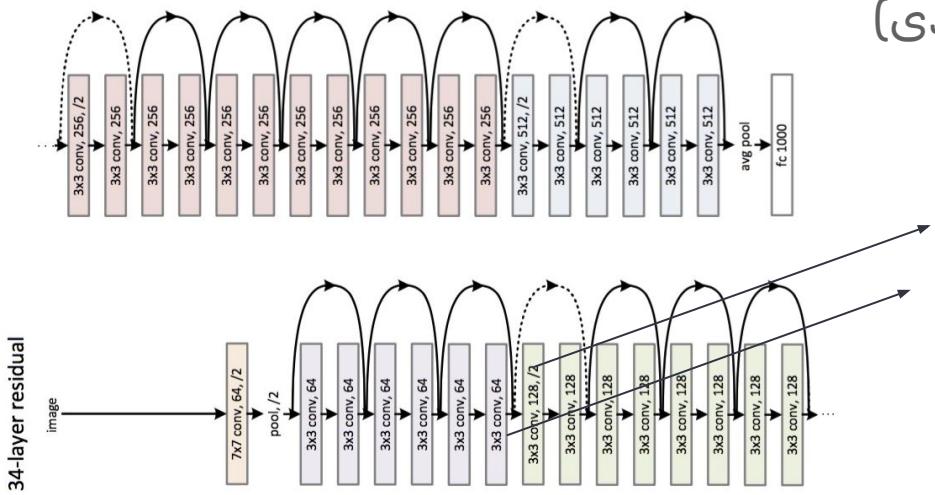


# ResNet



راه حل : کپی کردن لایه های  
آموخته شده از مدل کم عمق تر  
و تنظیم لایه های اضافی

# ResNet



- رو هم قرار گرفتن Residual Block
- بطور متنابه سایز فیلترها دوباره و سایز عکس نصف میشود با  $\text{stride} = 2$
- لایه FC ندارد (جز برای طبقه بندی)

stride 2 فیلتر با 128  
stride 1 فیلتر با 64

# ResNet

MSRA @ ILSVRC & COCO 2015 Competitions

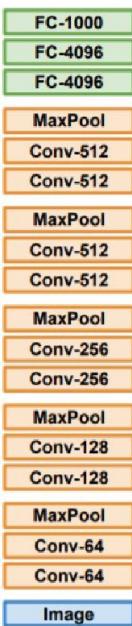
- **1st places** in all five main tracks
  - ImageNet Classification: “Ultra-deep” (quote Yann) **152-layer** nets
  - ImageNet Detection: **16%** better than 2nd
  - ImageNet Localization: **27%** better than 2nd
  - COCO Detection: **11%** better than 2nd
  - COCO Segmentation: **12%** better than 2nd

- انقلاب در عمق شبکه‌ها
- رتبه اول در تسک‌های مختلف



# Transfer Learning

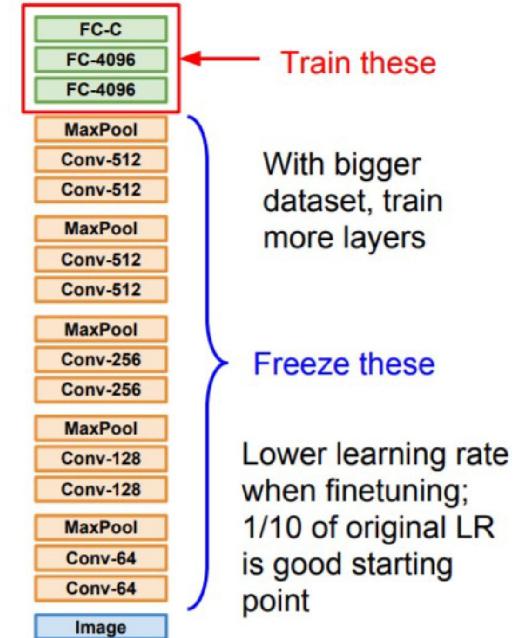
1. Train on Imagenet



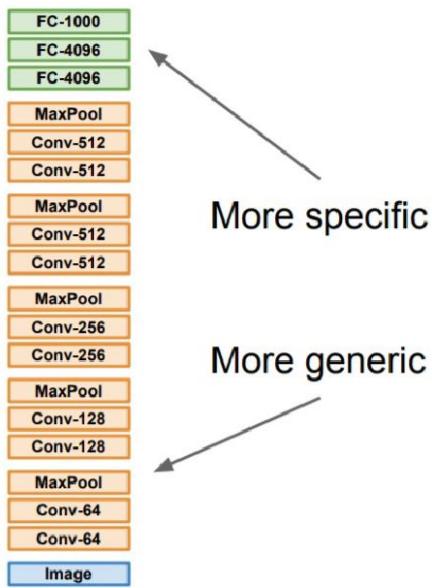
2. Small Dataset (C classes)



3. Bigger dataset



# Transfer Learning



	<b>very similar dataset</b>	<b>very different dataset</b>
<b>very little data</b>	Use Linear Classifier on top layer	You're in trouble... Try linear classifier from different stages
<b>quite a lot of data</b>	Finetune a few layers	Finetune a larger number of layers



# References

- Deep Learning Course, Dr. Soleimani 2023 - Sharif
- Understanding Deep Learning, 2023 - Book
- CS231n: Deep Learning for Computer Vision - Stanford

