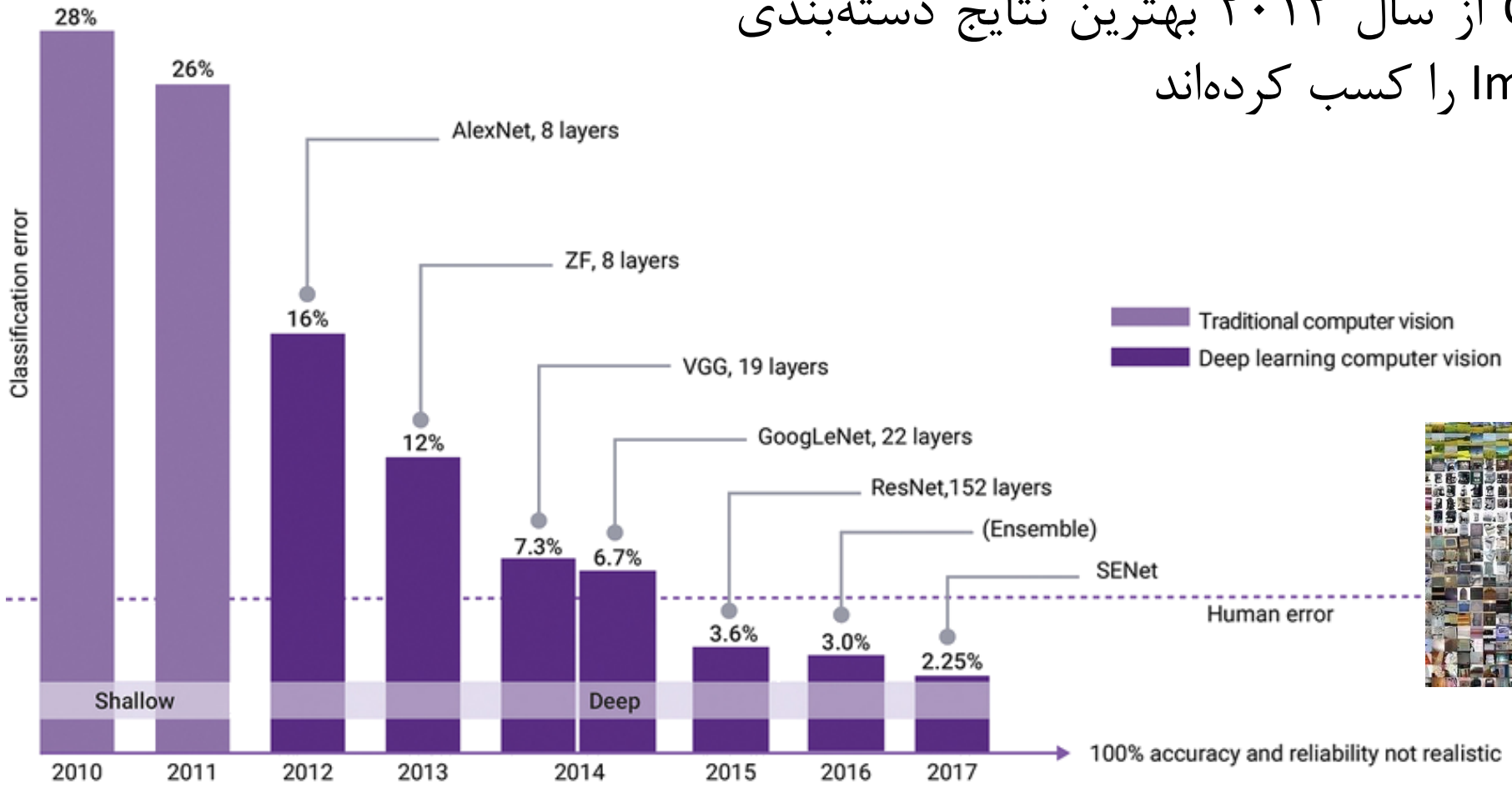

بسم الله الرحمن الرحيم

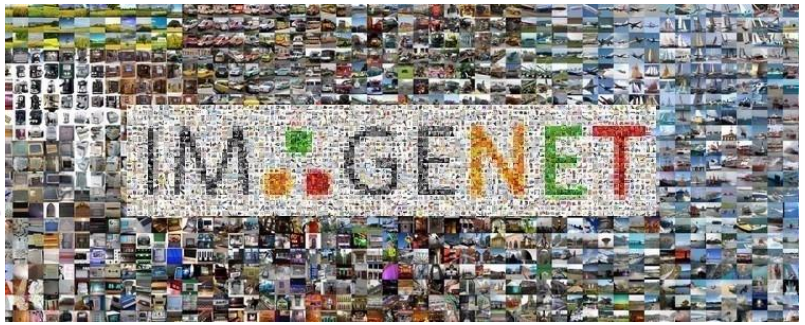
شبکه‌های عصبی کانولوشنی

Convolutional Neural Networks

نتایج ILSVRC

- معماری‌های مختلف CNN از سال ۲۰۱۲ بهترین نتایج دسته‌بندی تصویر در چالش ImageNet را کسب کرده‌اند

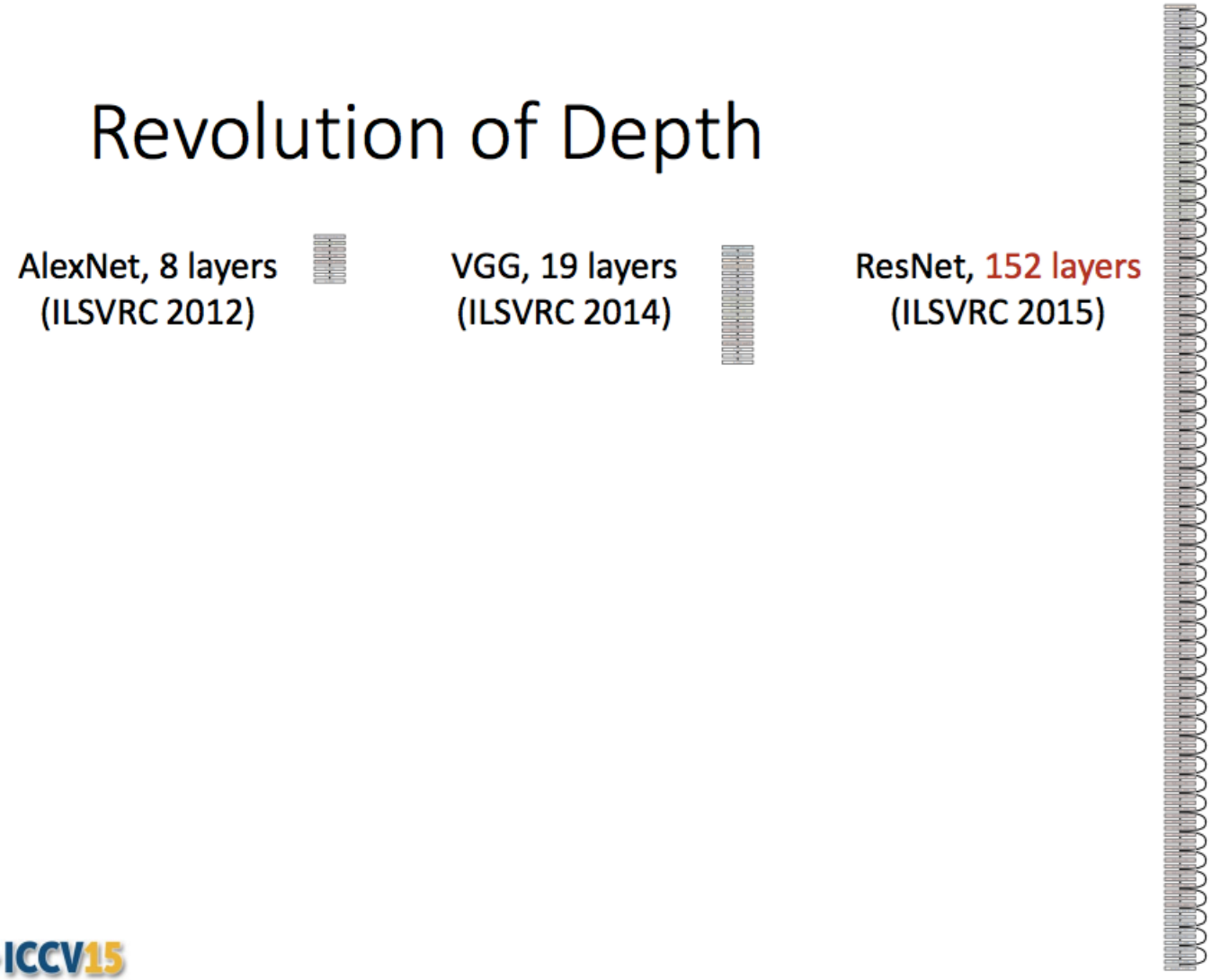




https://www.synopsys.com/designware-ip/technical-bulletin/computer-vision-lab-life.html

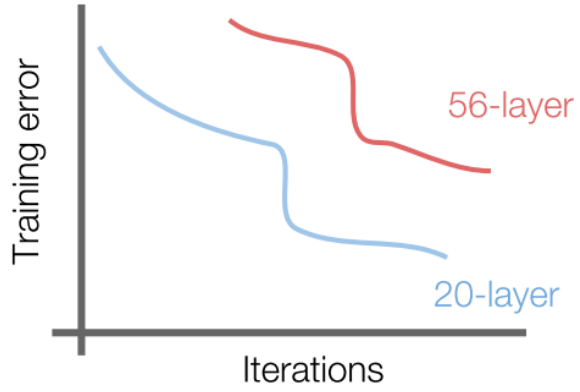
ResNet

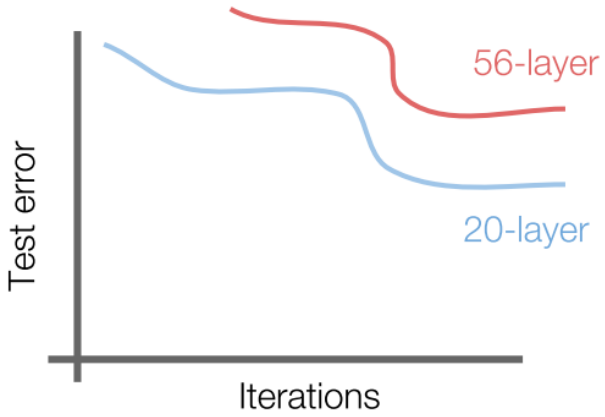
- شبکه ResNet برنده مسابقه ILSVRC’15 با خطای ۳.۵۷٪ شد
- با ۱۵۲ لایه، انقلابی در عمق شبکه‌های کانولوشنی به وجود آورد



He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2016).

ResNet

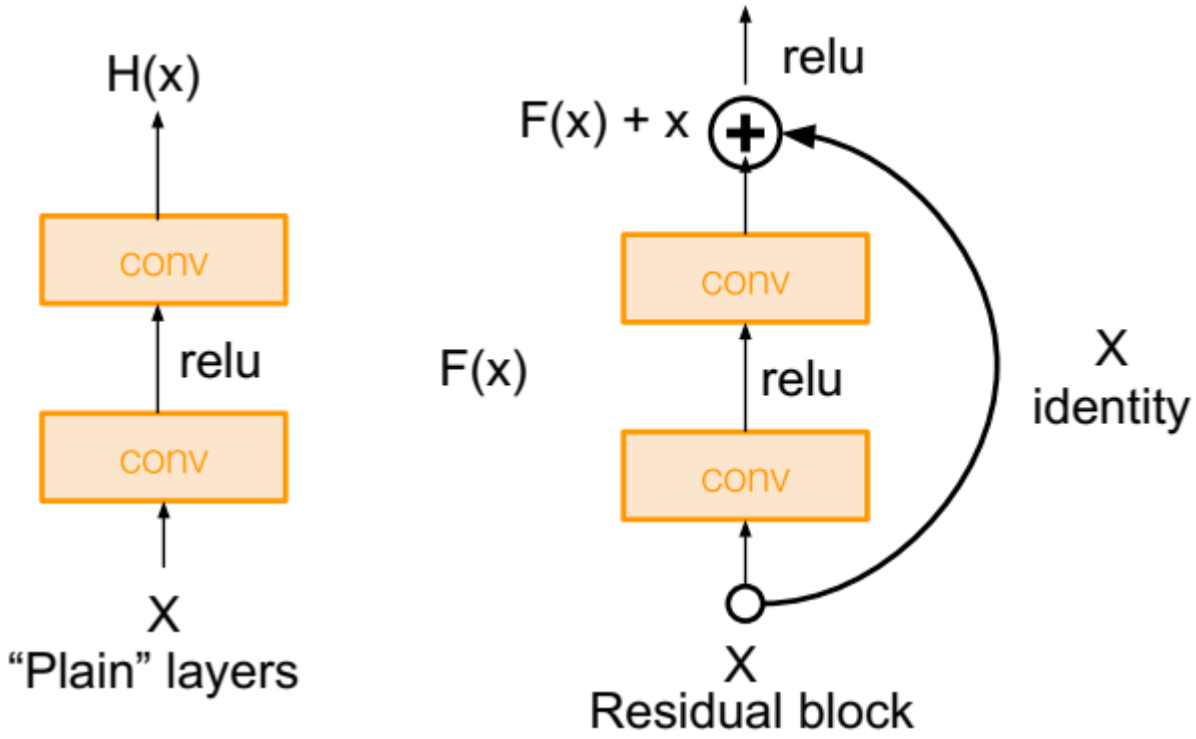


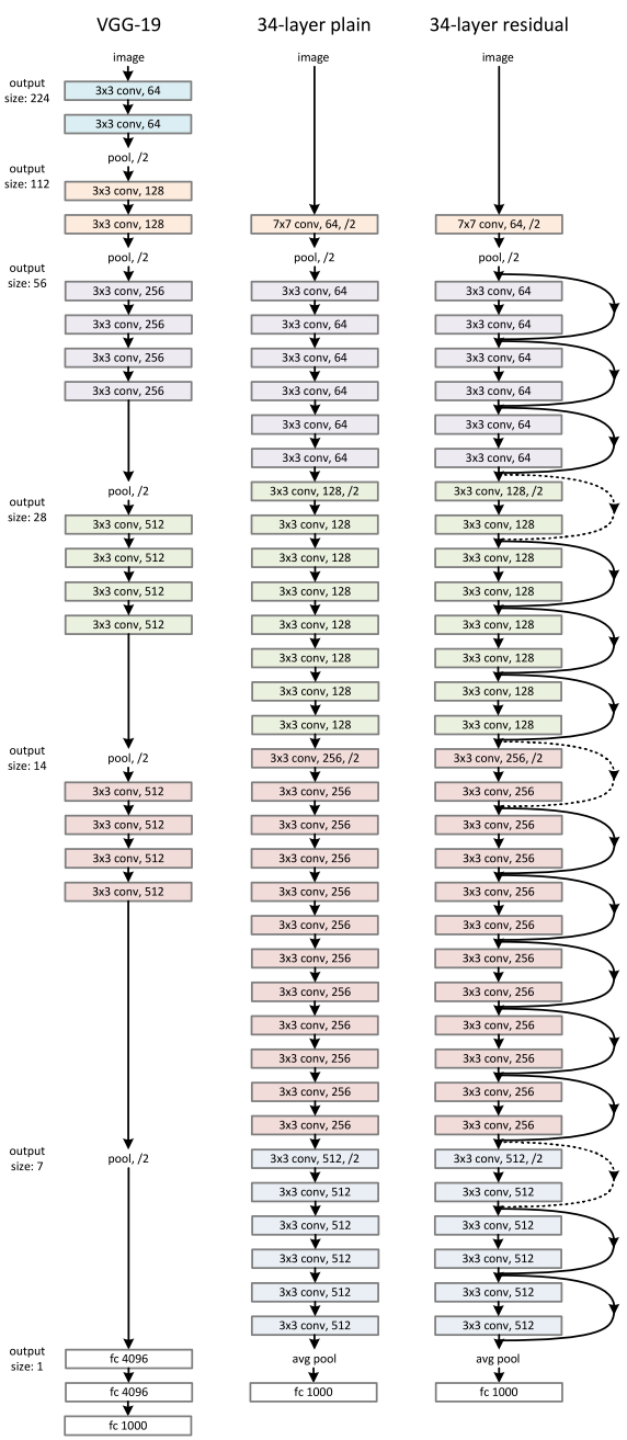


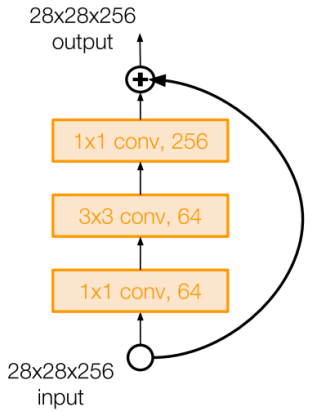
- اگر تعداد لایه‌های کانولوشنی ساده را بسیار زیاد کنیم چه اتفاقی می‌افتد؟
- چرا شبکه عمیق‌تر هم در آموزش و هم در آزمون عملکرد ضعیف‌تری دارد؟
  - البته مشکل از overfitting نیست!
- فرضیه: مشکل در مسئله بهینه‌سازی است
  - بهینه‌سازی مدل‌های عمیق‌تر دشوارتر است
- عملکرد مدل‌های عمیق‌تر باید حداقل به خوبی مدل‌های با عمق کمتر باشد
  - می‌توان وزن‌های مدل کم‌عمق را به لایه‌های نخست شبکه عمیق کپی کرد و لایه‌های اضافی را به گونه‌ای تنظیم کرد که نگاشت همانی را انجام دهند
- ایده ResNet آن است که لایه‌های شبکه بجای آموختن نگاشت مطلوب، باقی‌مانده آن را یاد بگیرند

ResNet

- لایه‌ها باید $F(x) = H(x) - x$ را بجای $H(x)$ بیاموزند







ResNet

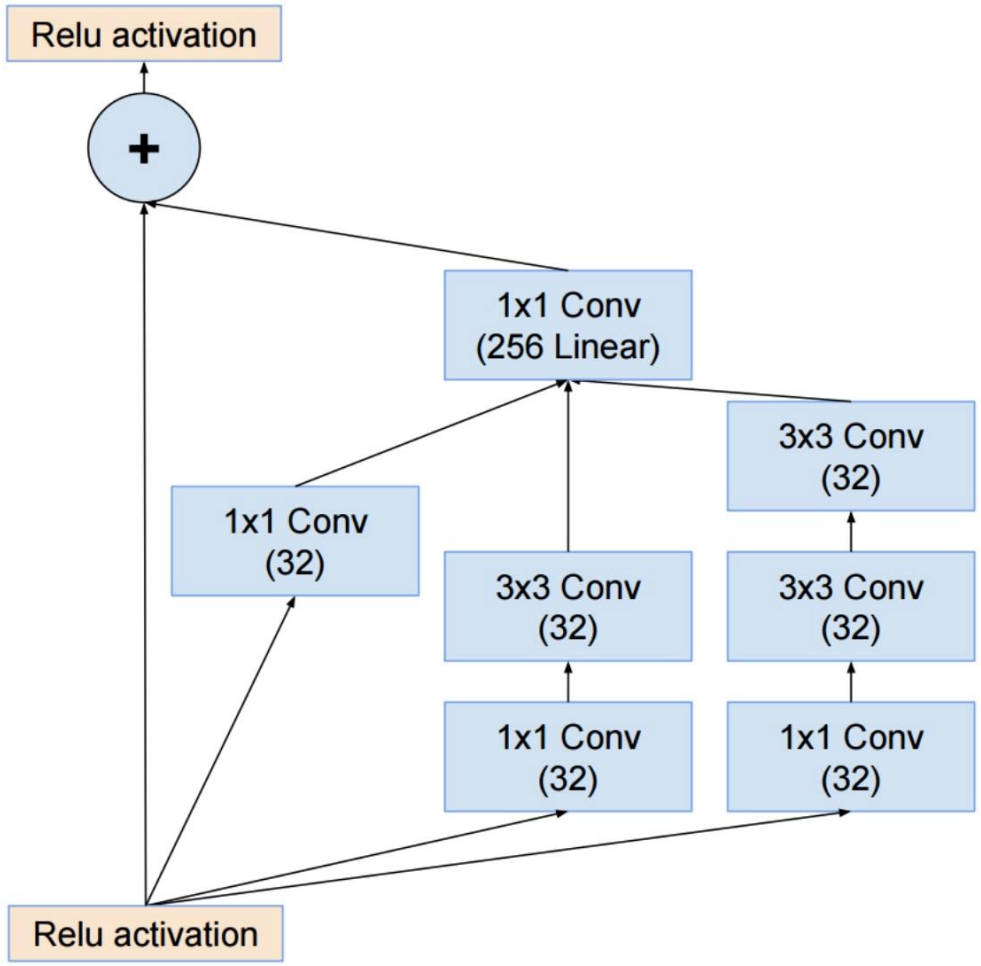
- از تعداد زیادی بلوک باقی‌مانده تشکیل شده است
- هر بلوک باقی‌مانده دارای ۲ لایه کانولوشنی ۳×۳ است
- به طور دوره‌ای، تعداد فیلترها ۲ برابر شده و رزولوشن مکانی نصف می‌شود
- در ابتدا دارای یک لایه کانولوشنی است
- پس از آخرین بلوک باقی‌مانده، ابعاد داده‌ها با استفاده از Average Pooling کاهش می‌یابد و یک لایه FC برای دسته‌بندی استفاده می‌شود
- برای مسئله ImageNet عمق‌های مختلف شبکه شامل ۳۴، ۵۰، ۱۰۱ و ۱۵۲ استفاده شده‌اند
- در شبکه‌های عمیق‌تر، از لایه کانولوشنی ۱×۱ برای بهبود بهره‌وری استفاده شده است

ResNet

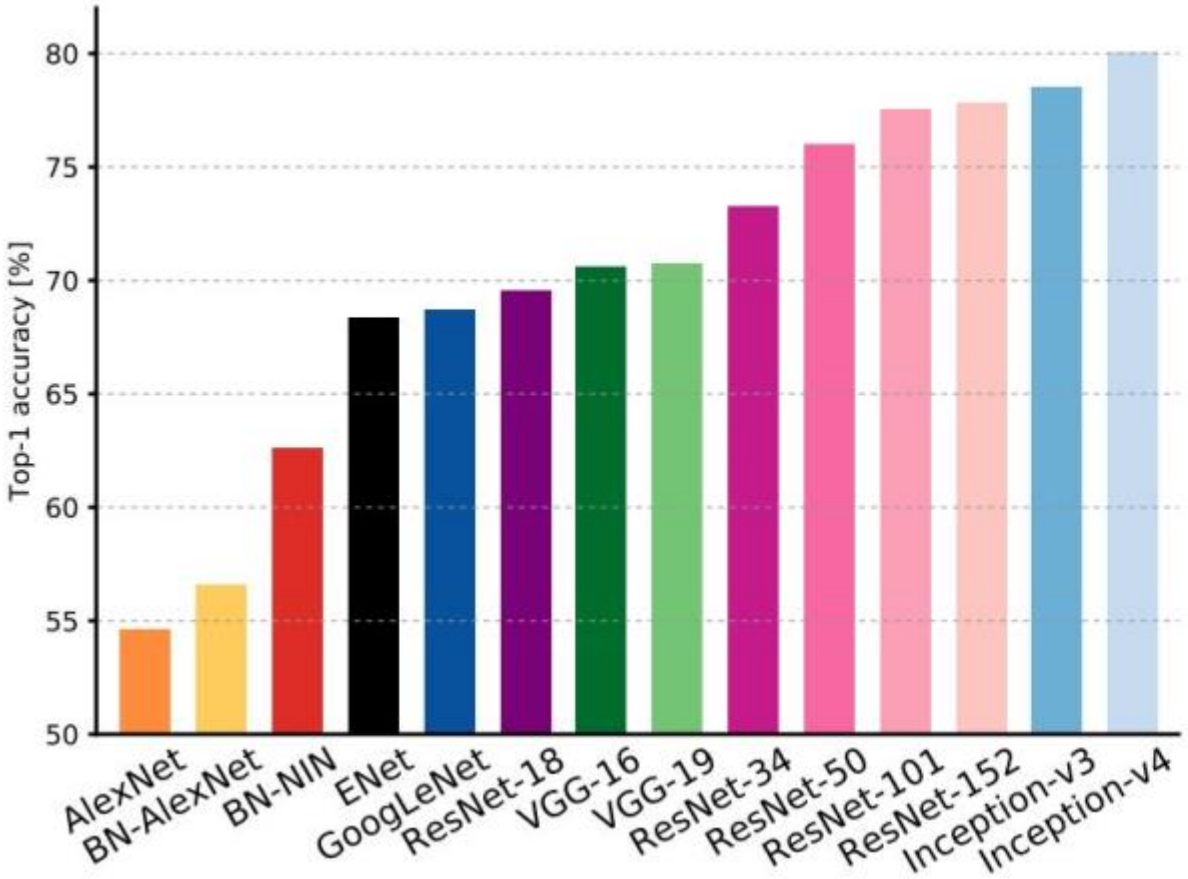
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

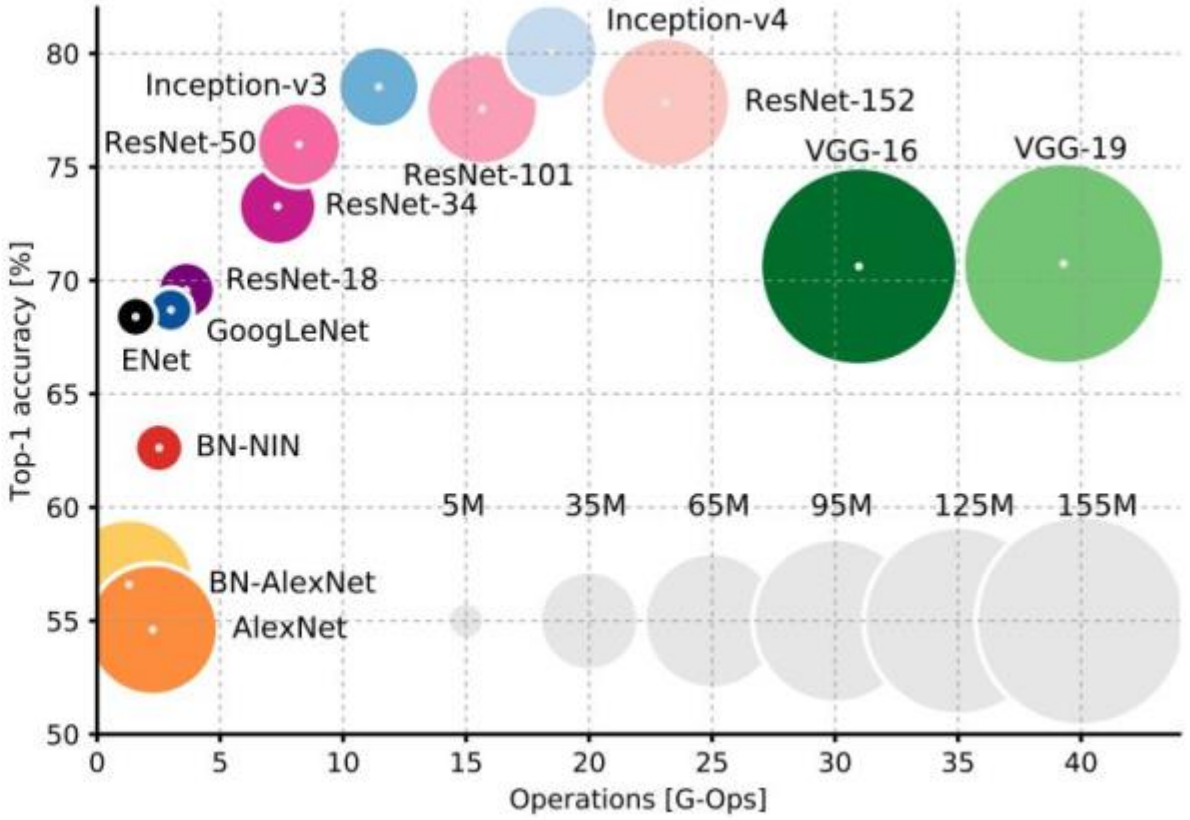
Inception V4

- ورودی همانی به ماژول Inception افزوده شده است



مقایسه


Top-1 accuracy [%]
80
75
70
65
60
55
50
AlexNet
BN-AlexNet
BN-NIN
ENet
GoogLeNet
ResNet-18
VGG-16
VGG-19
ResNet-34
ResNet-50
ResNet-101
ResNet-152
Inception-v3
Inception-v4


Top-1 accuracy [%]
80
75
70
65
60
55
50
0
5
10
15
20
25
30
35
40
Operations [G-Ops]
Inception-v4
Inception-v3
ResNet-152
ResNet-50
ResNet-101
VGG-16
VGG-19
ResNet-34
ResNet-18
GoogLeNet
ENet
BN-NIN
5M
35M
65M
95M
125M
155M
BN-AlexNet
AlexNet

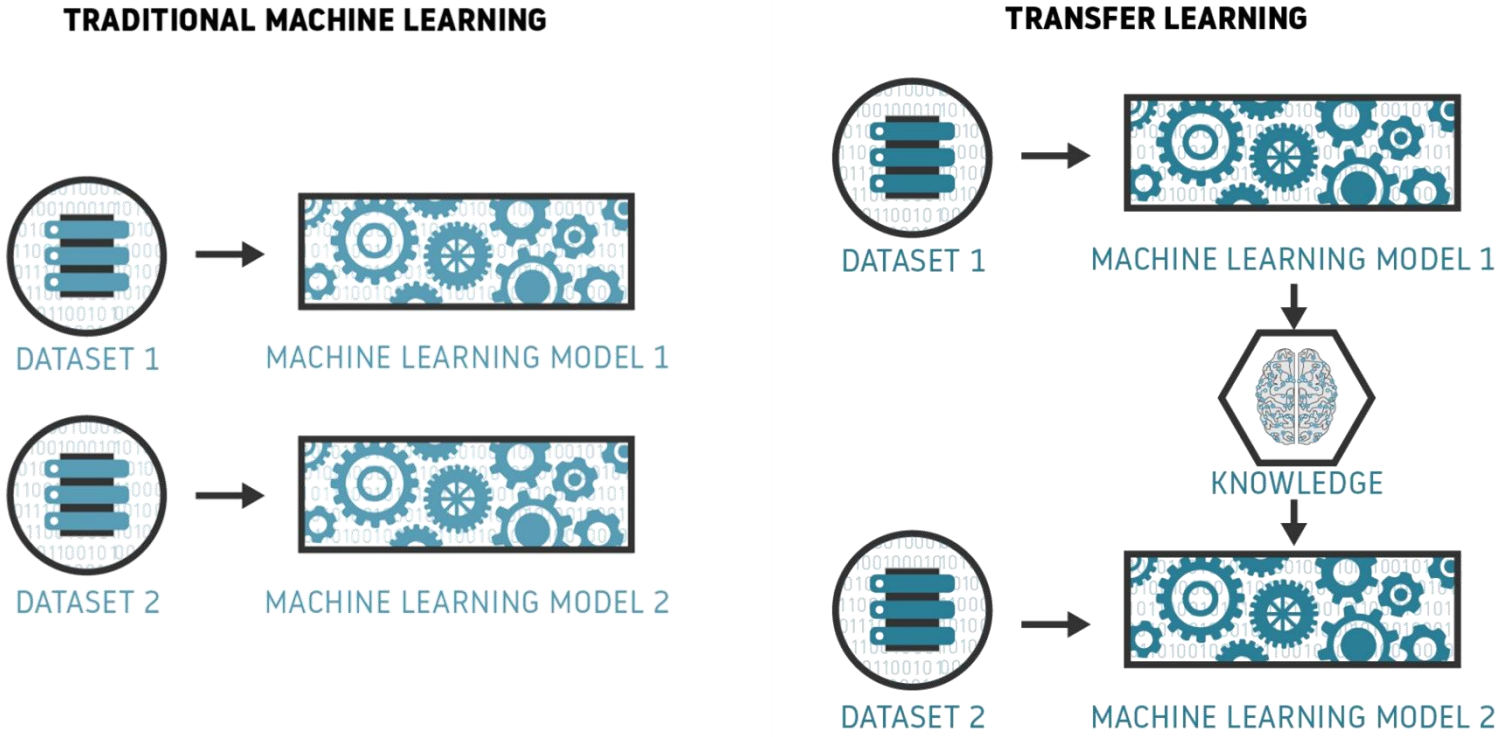
Metrics
Losses
Built-in small datasets
Keras Applications
Utilities
Code examples
Why choose Keras?
Community & governance
Contributing to Keras

Available models

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-

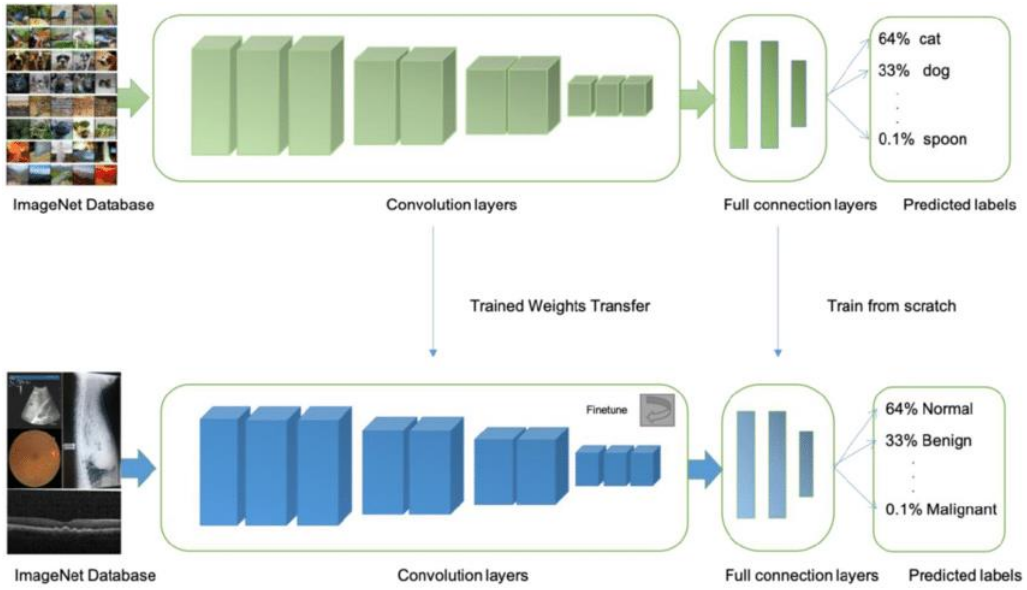
انتقال یادگیری

- چطور می‌شود از دانشی که توسط شبکه‌های عمیق برای حل مسائل پیچیده‌ای مانند ImageNet بدست آمده است برای حل یک مسئله دیگر استفاده کرد؟



شبکه‌های پیش‌آموخته (Pretrained)

- یک رویکرد رایج و بسیار مؤثر برای استفاده از یادگیری عمیق در مجموعه داده‌های تصویری کوچک، استفاده از یک شبکه پیش‌آموخته است
- یک شبکه پیش‌آموخته، شبکه‌ای است که وزن‌های آن قبلاً بر روی یک مجموعه داده بزرگ آموزش دیده و ذخیره شده است



شبکه‌های پیش‌آموخته (Pretrained)

- یک رویکرد رایج و بسیار مؤثر برای استفاده از یادگیری عمیق در مجموعه داده‌های تصویری کوچک، استفاده از یک شبکه پیش‌آموخته است
- یک شبکه پیش‌آموخته، شبکه‌ای است که وزن‌های آن قبلاً بر روی یک مجموعه داده بزرگ آموزش دیده و ذخیره شده است
- اگر مجموعه داده اولیه به اندازه کافی بزرگ و عمومی باشد، ویژگی‌های سلسله‌مراتبی آموخته شده توسط شبکه می‌تواند به طور موثر به عنوان یک مدل عمومی از دنیای بصری عمل کند
  - ویژگی‌های آن می‌تواند برای بسیاری از مسائل بینایی کامپیوتری مختلف مفید باشد
- این قابلیت جابجایی ویژگی‌های آموخته‌شده در مسائل مختلف، مزیت کلیدی یادگیری عمیق در مقایسه با بسیاری از رویکردهای قدیمی‌تر است

شبکه‌های پیش‌آموخته (Pretrained)

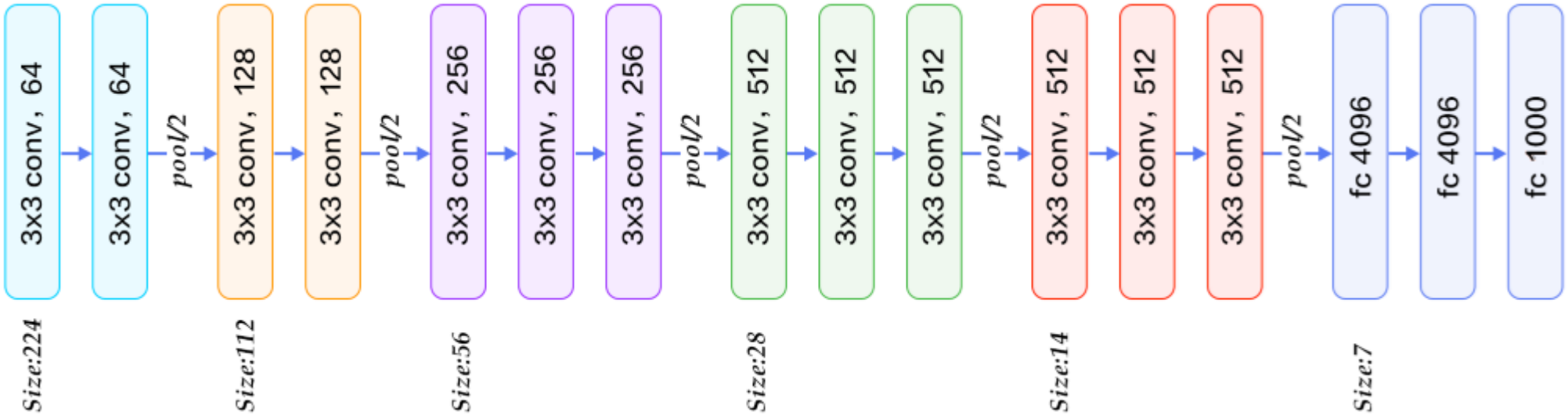
- یک شبکه کانولوشنی بزرگ را در نظر بگیرید که بر روی مجموعه داده ImageNet آموزش دیده است
  - ۱,۴ میلیون تصویر برچسب خورده از ۱,۰۰۰ کلاس مختلف (شامل چندین حیوان)
- انتظار داریم در مسئله دسته‌بندی سگ و گربه عملکرد بسیار خوبی داشته باشد
- در ادامه از معماری VGG16 استفاده می‌کنیم

شبکه‌های پیش‌آموخته (Pretrained)

- دو روش رایج برای استفاده از یک شبکه پیش‌آموخته وجود دارد:
  - استخراج ویژگی (feature extraction)
  - تنظیم دقیق (fine tuning)

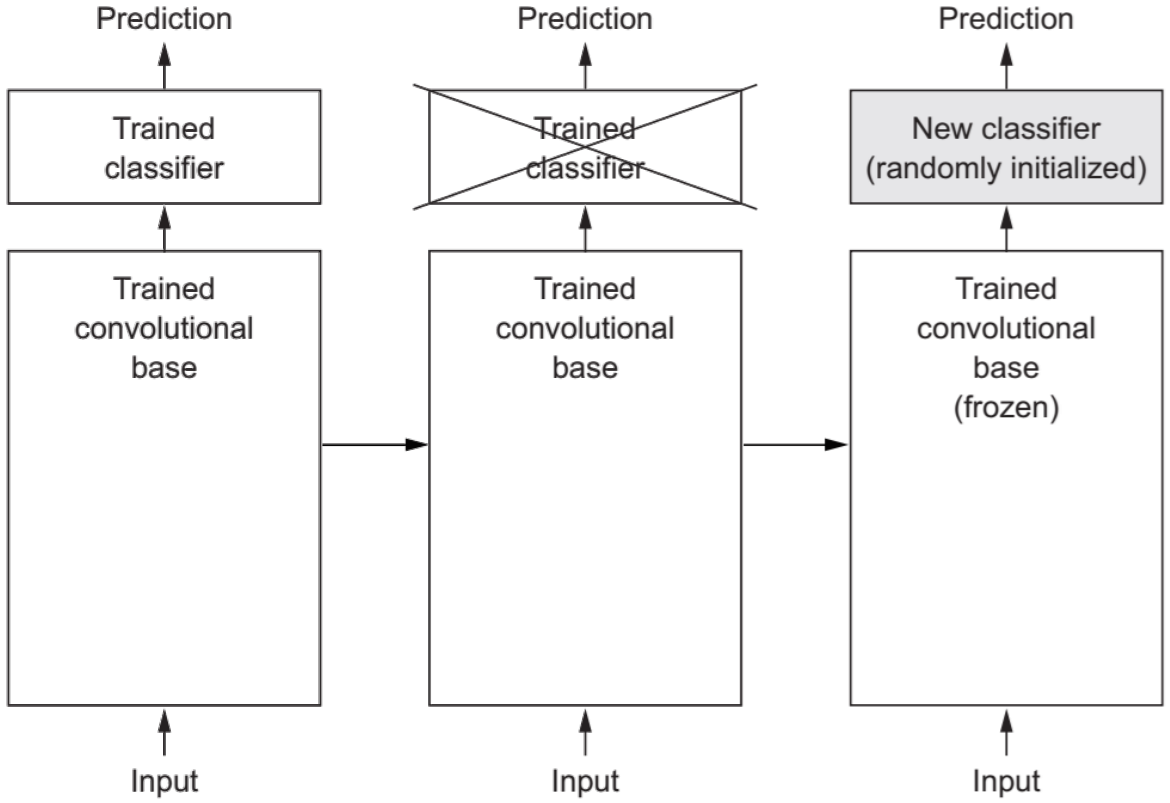
استخراج ویژگی

- استفاده از بازنمایی‌های آموخته شده توسط شبکه قبلی برای استخراج ویژگی از نمونه‌های جدید
  - از این ویژگی‌ها برای آموزش یک دسته‌بند جدید استفاده می‌شود



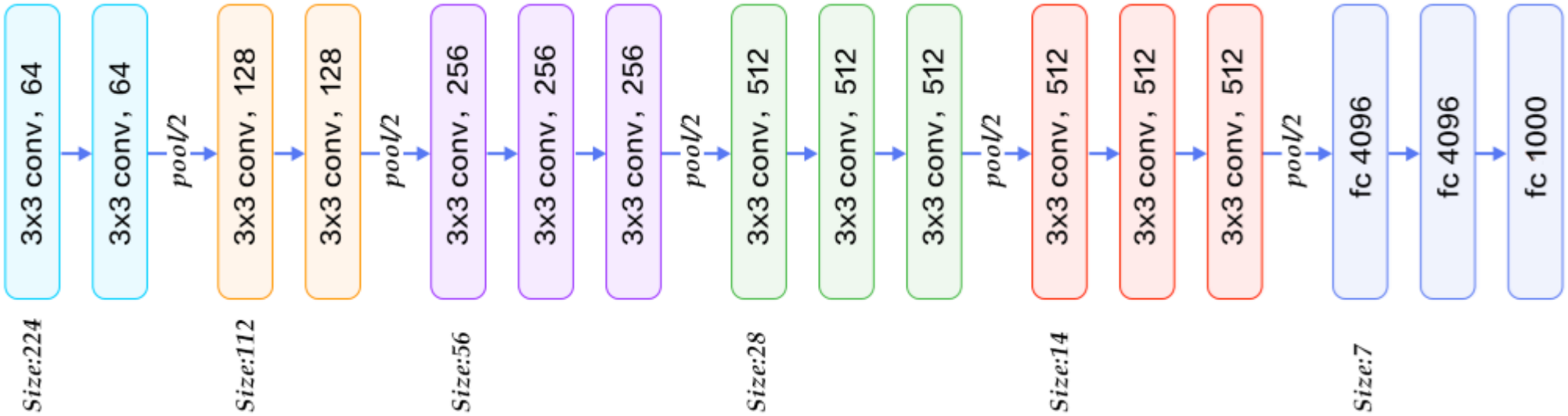
استخراج ویژگی

- استفاده از بازنمایی‌های آموخته شده توسط شبکه قبلی برای استخراج ویژگی از نمونه‌های جدید
  - از این ویژگی‌ها برای آموزش یک دسته‌بند جدید استفاده می‌شود



استخراج ویژگی

- نقشه‌های ویژگی یک شبکه کانولوشنی برابر با نقشه‌های حضور مفاهیم عمومی روی یک تصویر هستند
- بازنمایی‌هایی که توسط دسته‌بند کاملا متصل آموخته می‌شوند، مختص مجموعه کلاس‌هایی است که مدل بر روی آنها آموزش داده شده است



استخراج ویژگی

- سطح عمومی بودن (و بنابراین قابلیت استفاده مجدد) بازنمایی‌ها به عمق لایه در مدل بستگی دارد
  - لایه‌های ابتدایی ویژگی‌های محلّی و بسیار عمومی (مانند لبه‌های بصری، رنگ‌ها و بافت‌ها) را استخراج می‌کنند
  - به تدریج ویژگی‌ها انتزاعی‌تر و خاص‌تر می‌شوند (مانند "گوش گربه" یا "چشم سگ")
- اگر مجموعه داده جدید به طور اساسی متفاوت باشد، بهتر است فقط از چند لایه اول استفاده شود