یادگیری عمیق

مدرس: محمدرضا محمدی

زمستان ۱۴۰۱

شبکه‌های عصبی کانولوشنی

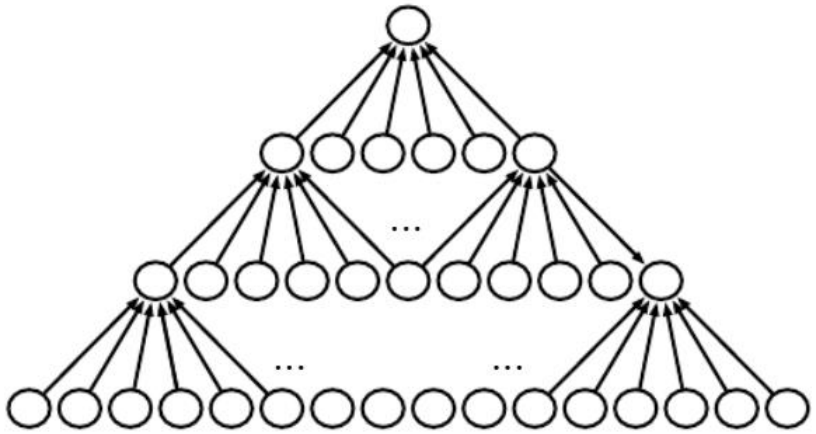
Convolutional Neural Networks

گسترش مرز (Padding)

- پس از هر لایه کانولوشنی، عرض بازنمایی یک واحد کمتر از عرض هسته در هر لایه کوچک می‌شود

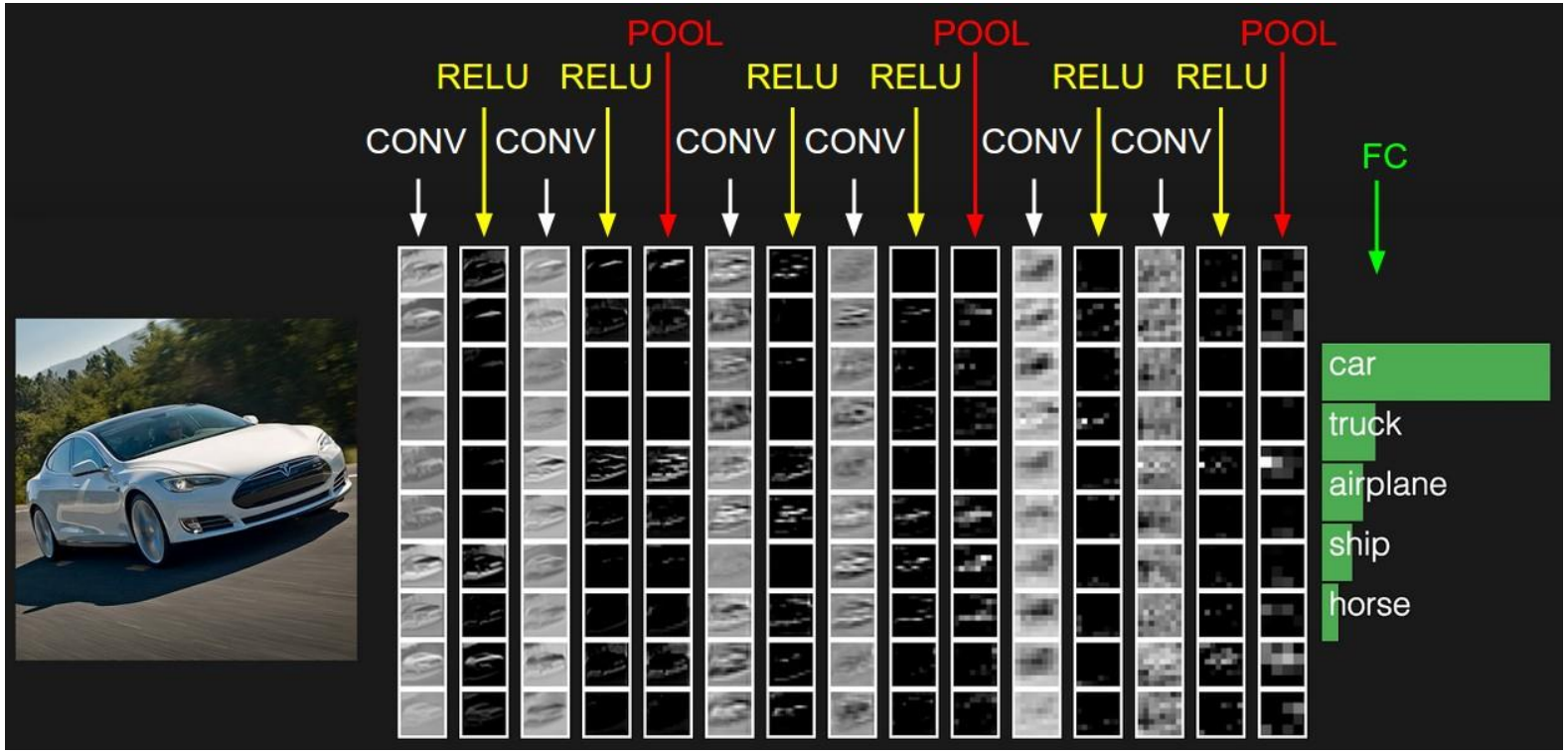
$$w_o = w_i - w_k + 1$$

- یک راه پرکاربرد افزودن صفر به اندازه مورد نیاز $(w_k - 1)$ به لایه ورودی است (zero-padding)

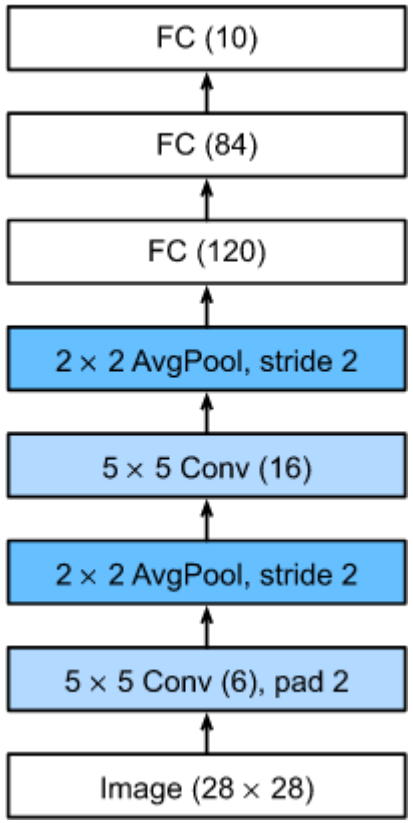


طراحی شبکه‌های کانولوشنی

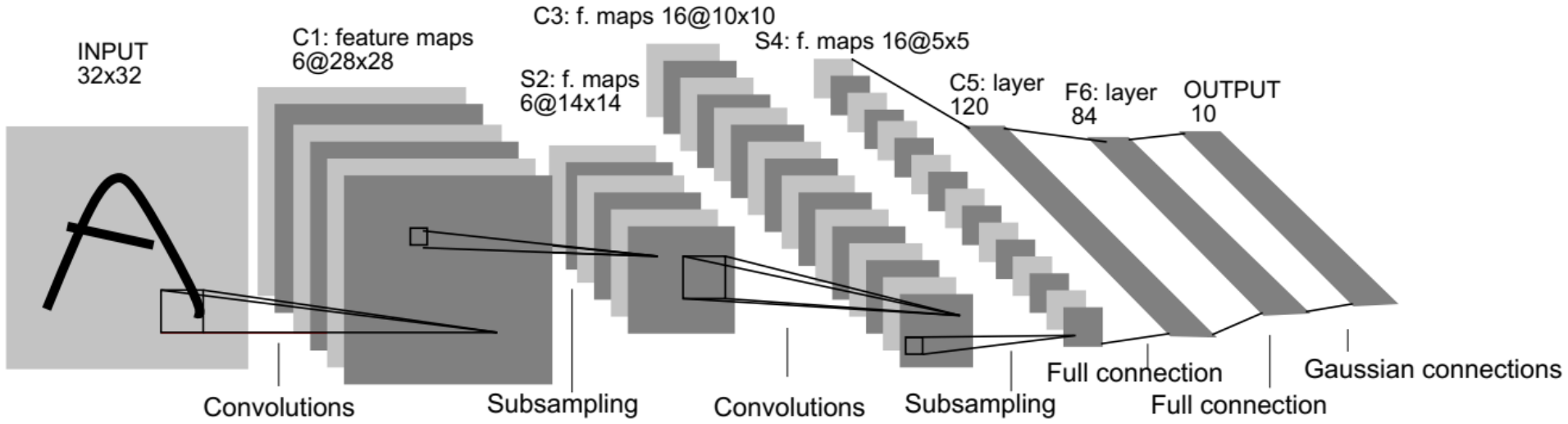
- تصاویر بزرگتر (از CIFAR) و مسائل پیچیده‌تر، باعث می‌شود از شبکه‌های بزرگتری استفاده کنیم
- ورودی بزرگتر: از گام‌های بیشتری استفاده می‌کنیم
- به طور معمول، عمق نقشه‌های ویژگی به تدریج در شبکه افزایش می‌یابد، در حالیکه اندازه نقشه‌های ویژگی کاهش می‌یابد



شبکه LeNet-5

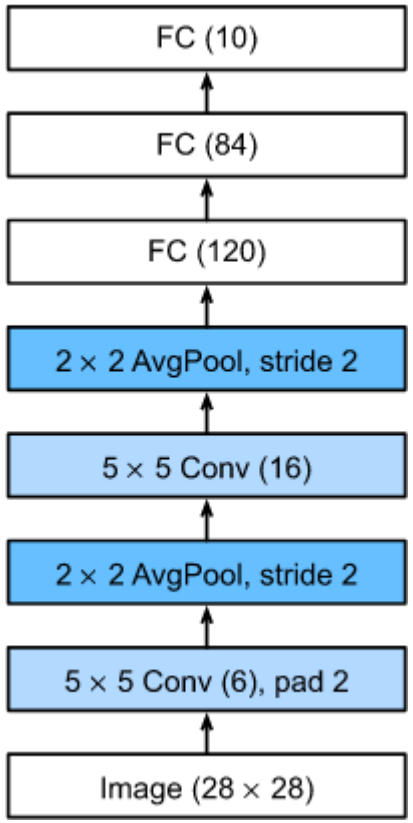


- شبکه LeNet-5 در سال ۱۹۹۸ برای شناسایی اعداد و حروف دستنویس پیشنهاد شد
- این شبکه تنها دارای ۵ لایه آموزشی است: ۲ لایه کانولوشنی و ۳ لایه کاملا متصل
  - در بخش کانولوشنی، از تابع Sigmoid و ادغام میانگین استفاده شده است
  - در لایه‌های FC میانی از تابع Sigmoid و در لایه انتهایی از تابع Gaussian استفاده شده است



LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 1998.

شبکه LeNet-5



- شبکه LeNet-5 در سال ۱۹۹۸ برای شناسایی اعداد و حروف دستنویس پیشنهاد شد
- این شبکه تنها دارای ۵ لایه آموزشی است: ۲ لایه کانولوشنی و ۳ لایه کاملا متصل
  - در بخش کانولوشنی، از تابع Sigmoid و ادغام میانگین استفاده شده است
  - در لایه‌های FC میانی از تابع Sigmoid و در لایه انتهایی از تابع Gaussian استفاده شده است

```
net = nn.Sequential(
    nn.Conv2d(1, 6, kernel_size=5, padding=2), nn.Sigmoid(),
    nn.AvgPool2d(kernel_size=2, stride=2),
    nn.Conv2d(6, 16, kernel_size=5), nn.Sigmoid(),
    nn.AvgPool2d(kernel_size=2, stride=2),
    nn.Flatten(),
    nn.Linear(16 * 5 * 5, 120), nn.Sigmoid(),
    nn.Linear(120, 84), nn.Sigmoid(),
    nn.Linear(84, 10))
```

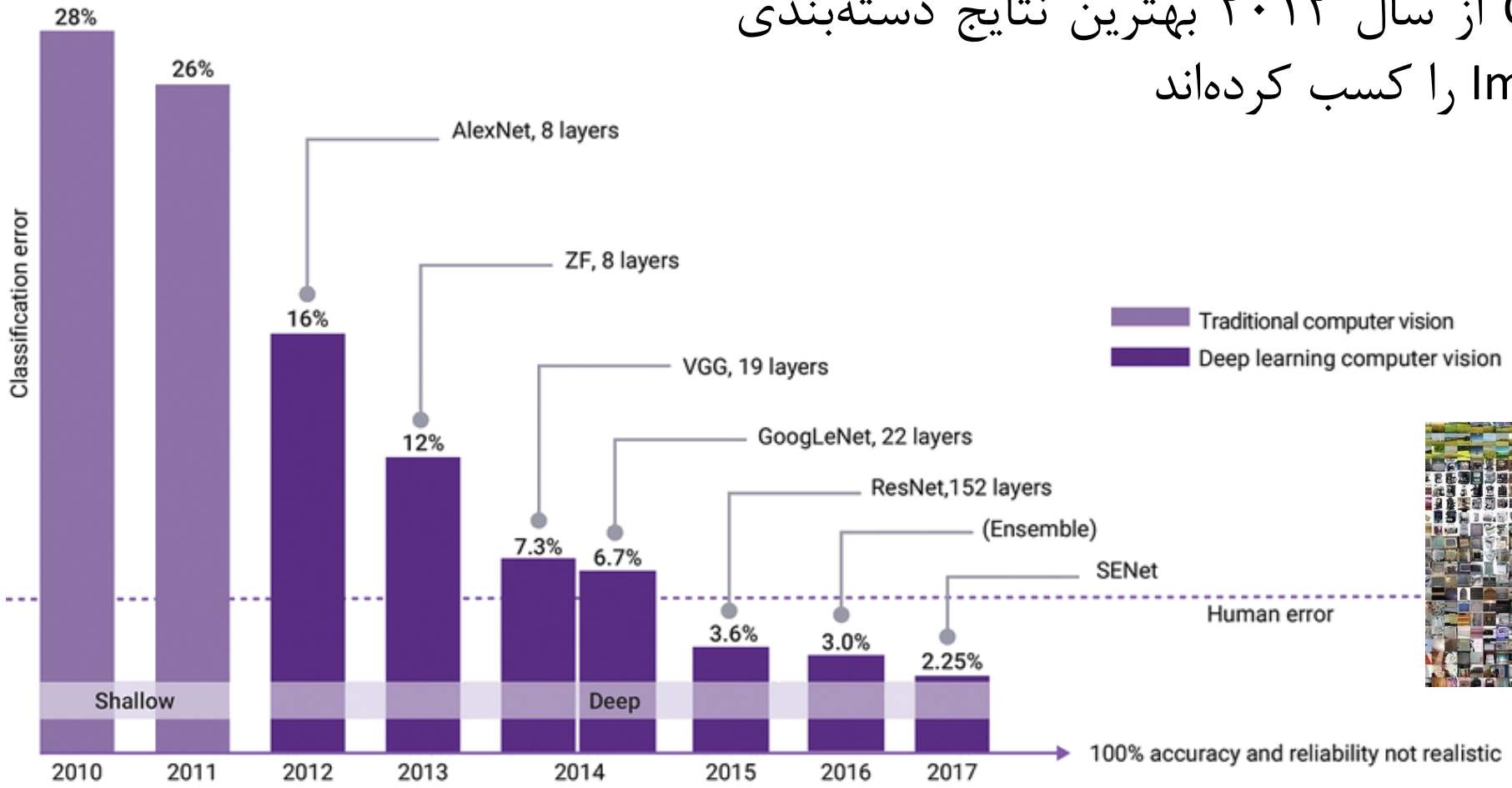

LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 1998.

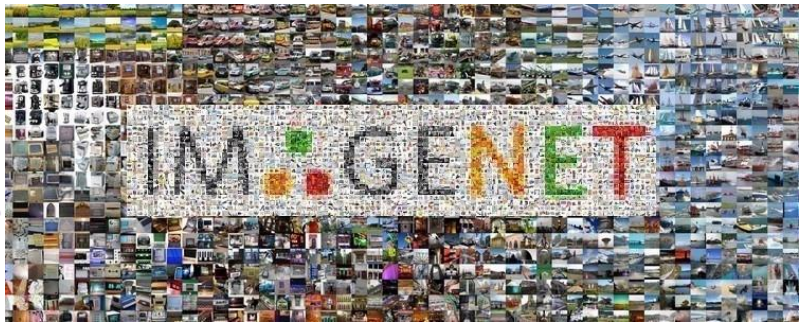
شبکه‌های عصبی کانولوشنی مدرن

Modern Convolutional Neural Networks

نتایج ILSVRC

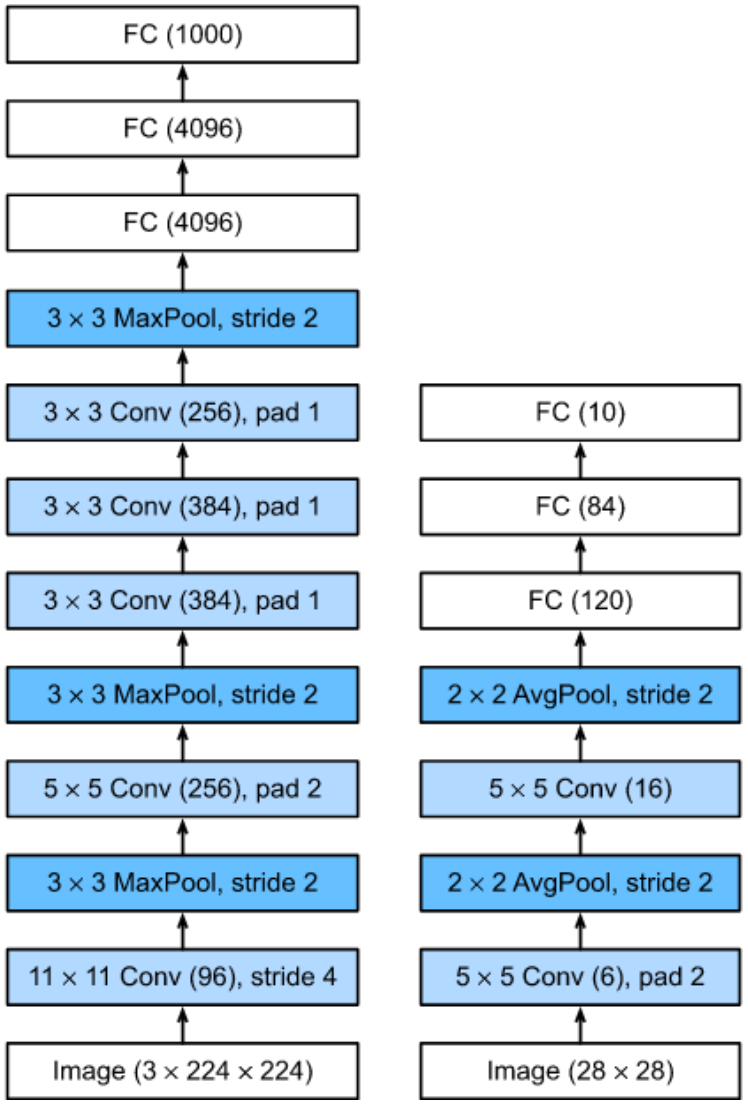
- معماری‌های مختلف CNN از سال ۲۰۱۲ بهترین نتایج دسته‌بندی تصویر در چالش ImageNet را کسب کرده‌اند





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

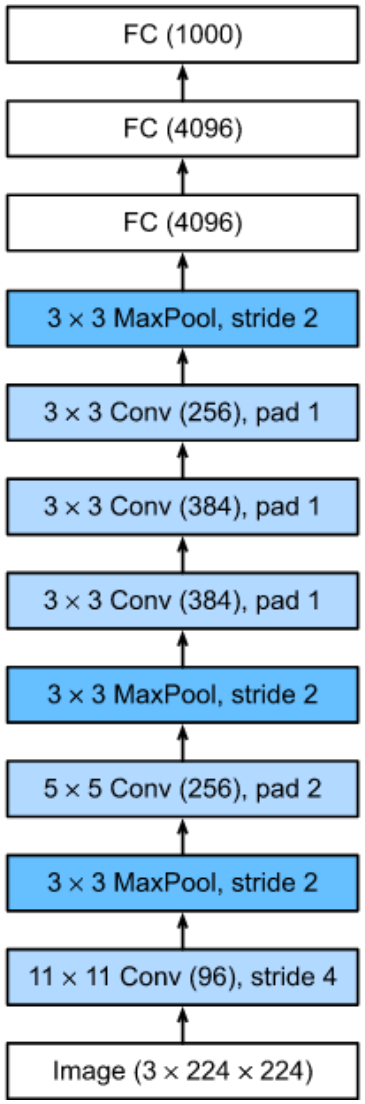
AlexNet



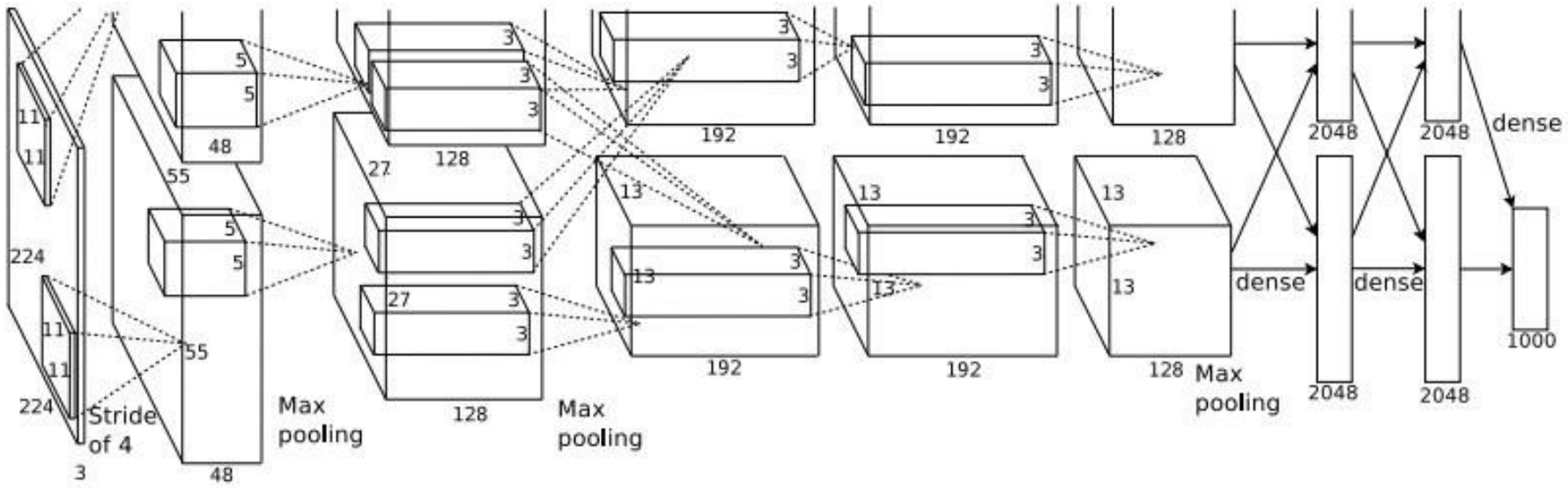
- اگرچه LeNet در مجموعه‌داده‌های کوچک اولیه به نتایج خوبی دست یافت، عملکرد CNN بر روی مجموعه‌داده‌های بزرگتر و واقعی‌تر هنوز مشخص نبود
- در الگوریتم‌های بینایی کامپیوتر رقیب، معمولا ابتدا از تصویر ویژگی‌های دست‌ساز استخراج می‌شوند
- با توسعه سخت‌افزارها و مجموعه‌داده‌های بزرگ، یادگیری ویژگی توسط شبکه‌های کانولوشنی عمیق نتایج بسیار خوبی بدست آوردند

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems,* 2012.

AlexNet

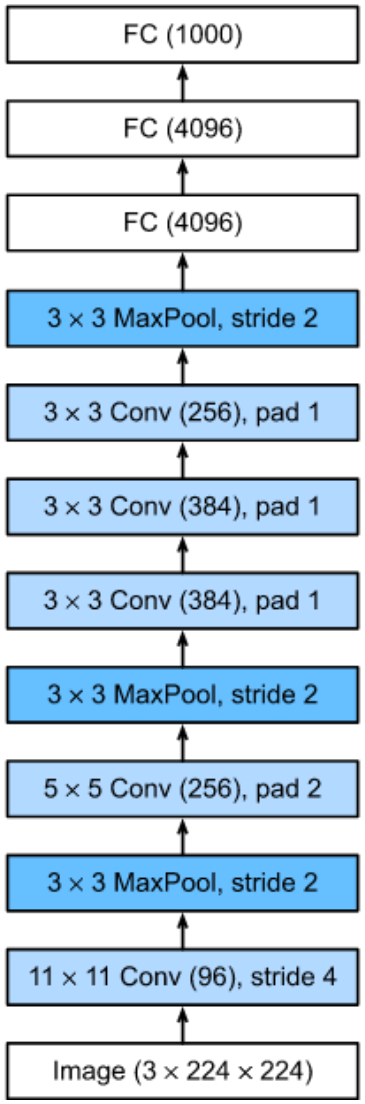


- شبکه AlexNet یک شبکه دارای ۸ لایه آموزشی است که در سال ۲۰۱۲ پیشنهاد شد و توانست خطای top-5 در چالش ILSVRC’12 را به ۱۵.۳٪ کاهش دهد
- استفاده از ReLU، Dropout و داده‌افزایی نیز در عملکرد AlexNet موثر بوده‌اند
- به دلیل محدودیت سخت‌افزار، به صورت موازی روی دو GPU پیاده‌سازی شده بود

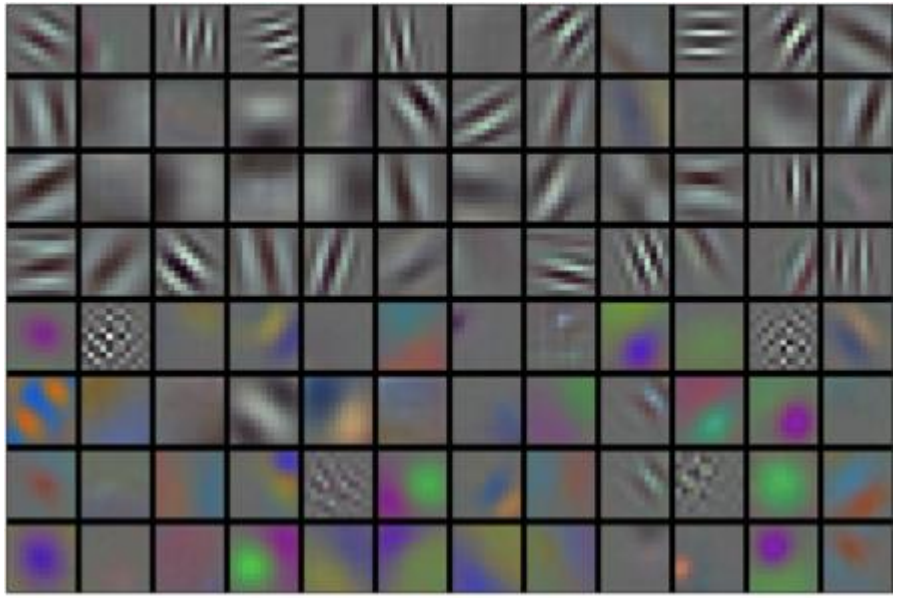


Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems,* 2012.

یادگیری بازنمایی



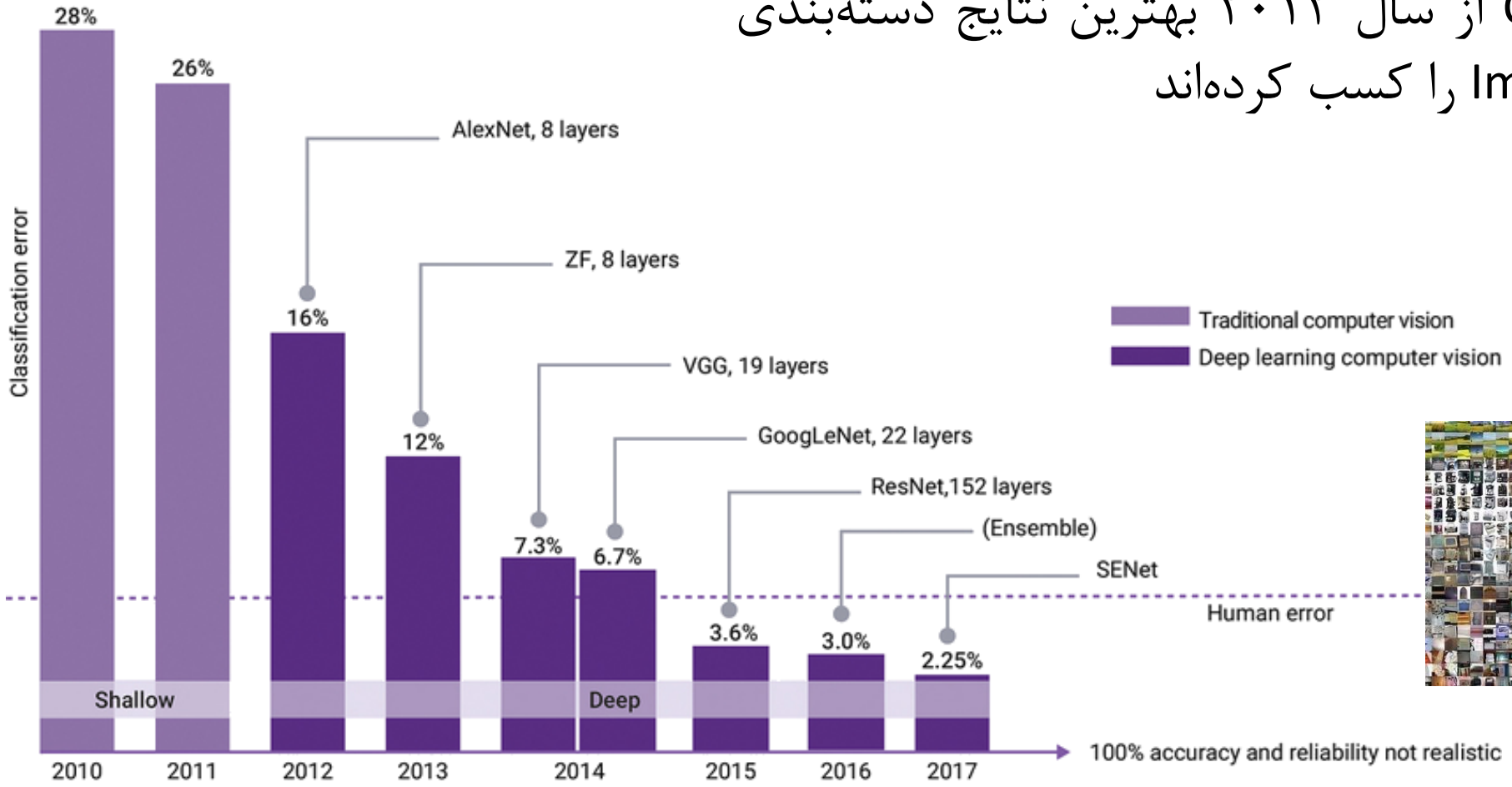
- تا سال ۲۰۱۲، بازنمایی‌های جدید از تصاویر معمولا به صورت دست‌ساز طراحی می‌شدند
- رویکرد دیگر طراحی مدل‌هایی است که پارامترهای آنها قابل آموزش است و می‌تواند بازنمایی مناسب برای حل مسئله را یاد بگیرد
- ۹۶ فیلتر ۳×۱۱×۱۱ لایه نخست:
  - توصیفگرهای سطح پائین تصویر
- در لایه‌های بالاتر، ساختارهای پیچیده‌تر و بزرگتری مانند چشم تشخیص داده می‌شوند

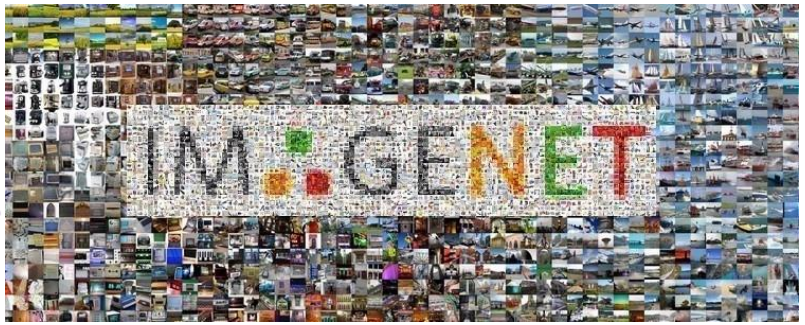


Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems,* 2012.

نتایج ILSVRC

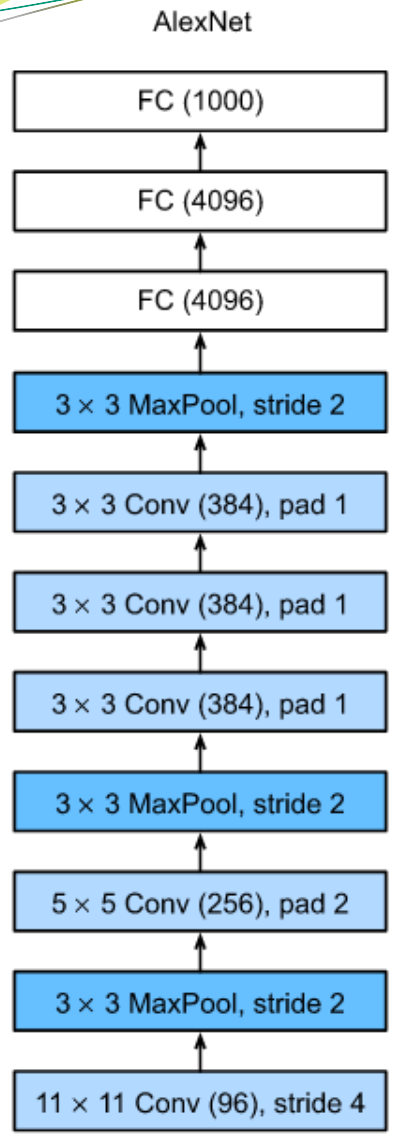
- معماری‌های مختلف CNN از سال ۲۰۱۲ بهترین نتایج دسته‌بندی تصویر در چالش ImageNet را کسب کرده‌اند

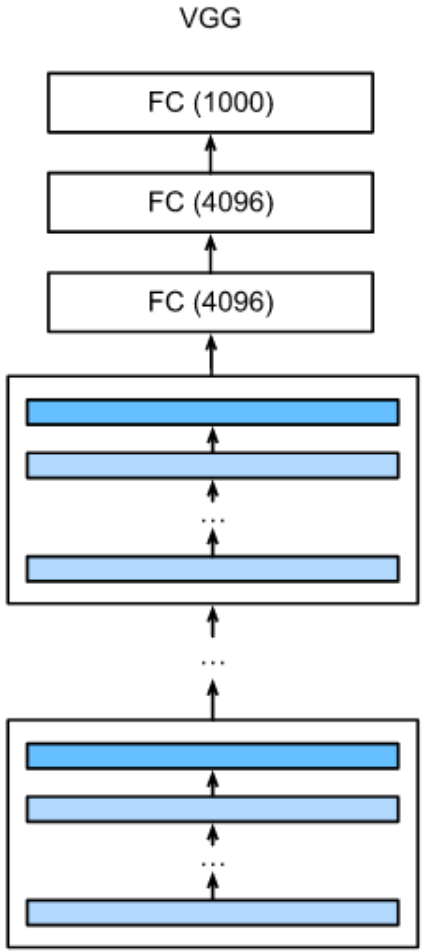


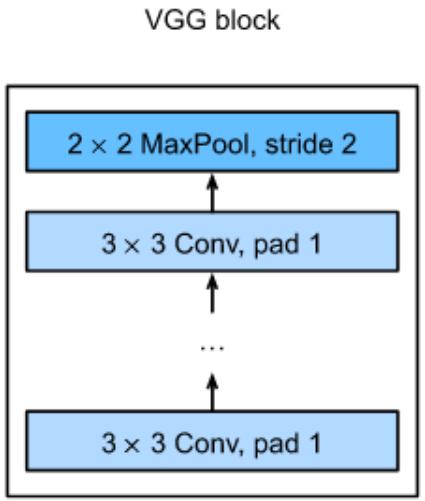


https://www.synopsys.com/designware-ip/technical-bulletin/computer-vision-lab-life.html

VGG





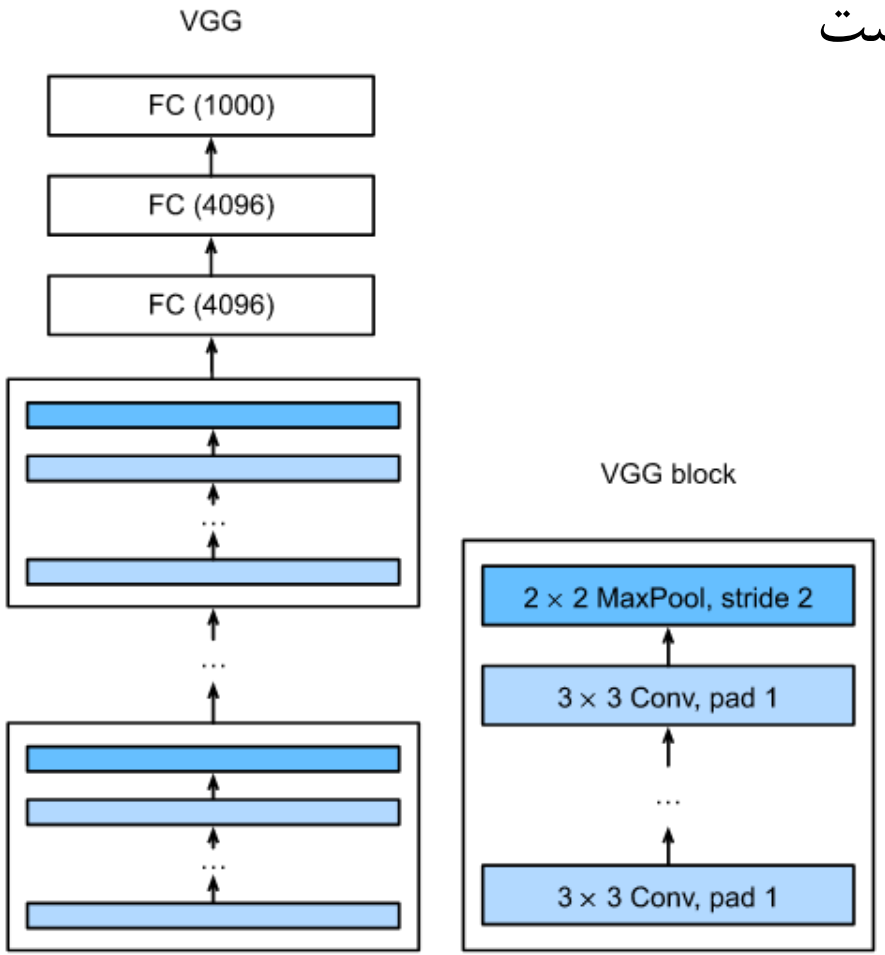


- شبکه VGG که در Visual Geometry Group توسعه داده شده است، مبتنی بر ایده استفاده از ساختار بلوکی است
- مشابه با AlexNet دو بخش کانولوشنی و کاملا متصل دارد
- بلوک سازنده پایه CNNهای کلاسیک دنباله‌ای از لایه‌های زیر است:
  - لایه‌های کانولوشنی
  - توابع غیرخطی
  - لایه ادغام

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint,* 2014.

VGG

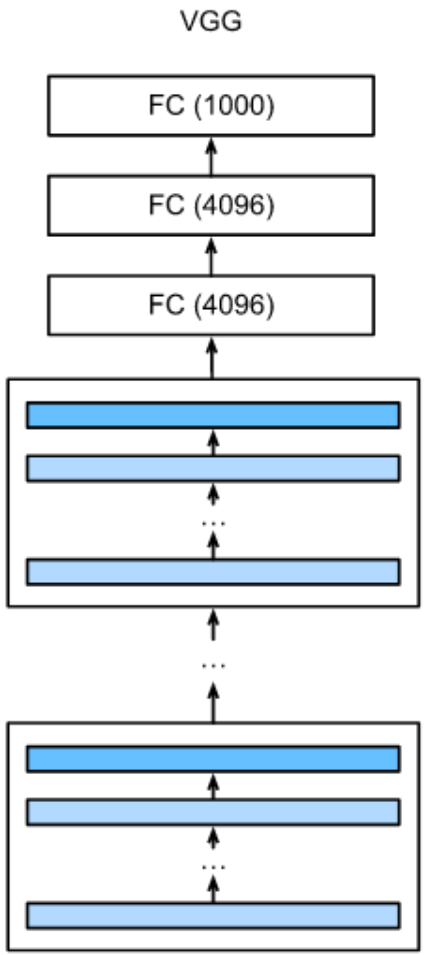
- در هر بلوک، تعداد فیلترها و تعداد کانال‌های خروجی قابل تنظیم است
- از پنج بلوک استفاده کرده است




```
def vgg_block(num_convs, in_channels, out_channels):
    layers = []
    for _ in range(num_convs):
        layers.append(nn.Conv2d(in_channels, out_channels,
                                kernel_size=3, padding=1))
        layers.append(nn.ReLU())
        in_channels = out_channels
    layers.append(nn.MaxPool2d(kernel_size=2,stride=2))
    return nn.Sequential(*layers)
```


Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint,* 2014.

VGG

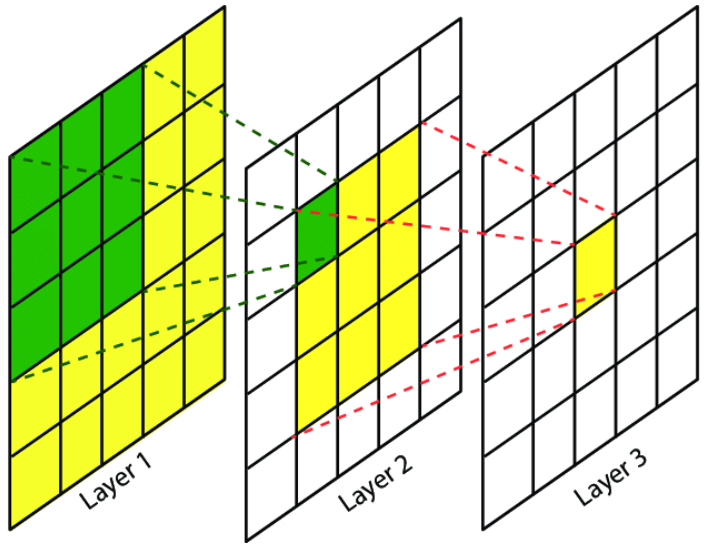



```
VGG11: conv_arch = ((1, 64), (1, 128), (2, 256), (2, 512), (2, 512))
VGG16: conv_arch = ((2, 64), (2, 128), (3, 256), (3, 512), (3, 512))
VGG19: conv_arch = ((2, 64), (2, 128), (4, 256), (4, 512), (4, 512))

def vgg(conv_arch):
    conv_blks = []
    in_channels = 1
    # The convolutional part
    for (num_convs, out_channels) in conv_arch:
        conv_blks.append(vgg_block(num_convs, in_channels, out_channels))
        in_channels = out_channels

    return nn.Sequential(
        *conv_blks, nn.Flatten(),
        # The fully-connected part
        nn.Linear(out_channels * 7 * 7, 4096), nn.ReLU(), nn.Dropout(0.5),
        nn.Linear(4096, 4096), nn.ReLU(), nn.Dropout(0.5),
        nn.Linear(4096, 10))

net = vgg(conv_arch)
```


Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint,* 2014.

VGG

- معماری VGG در سال ۲۰۱۴ تیم دوم مسابقه ILSVRC’14 شد
- فیلترهای کوچکتر و لایه‌های بیشتر
- در این معماری ابعاد تمام فیلترها ۳×۳ با گام ۱ است

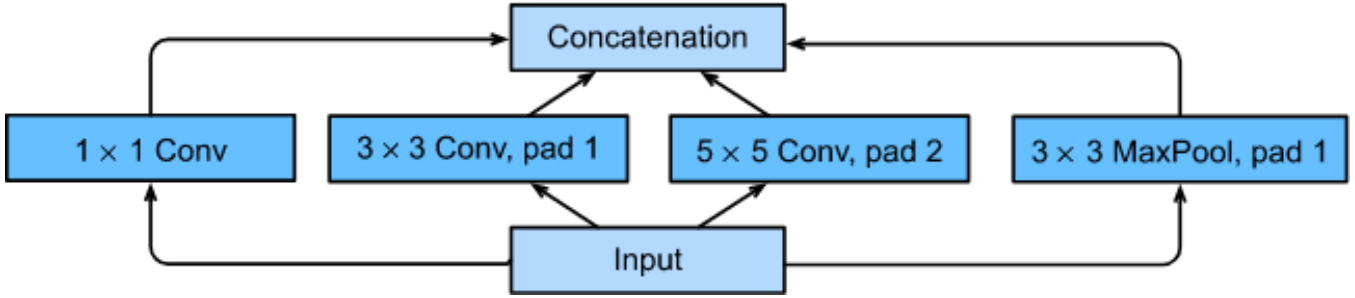


Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint,* 2014.

GoogLeNet

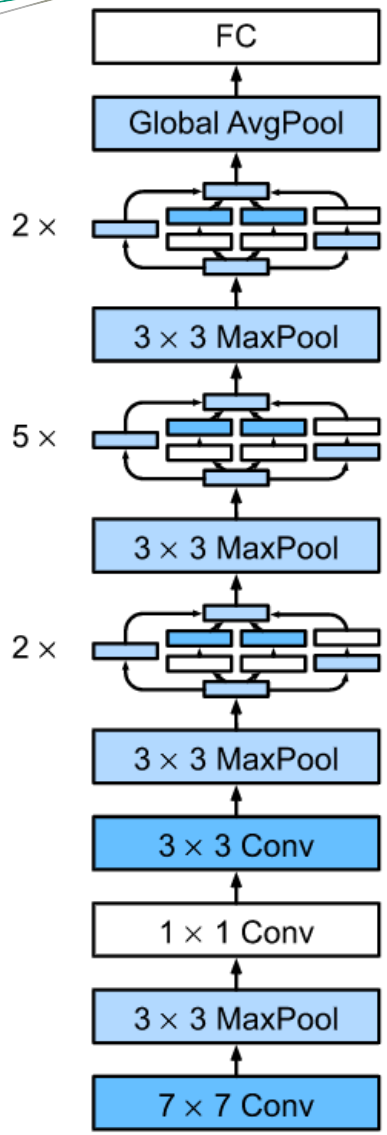
- شبکه GoogLeNet برنده مسابقه ILSVRC’14 با خطای ۶.۷٪ شد
- یکی از تمرکزهای مقاله پرداختن به این سوال بود که بهترین اندازه برای کرنل‌های کانولوشنی چند است؟

- فیلترهای هم‌عرض (موازی) تحت عنوان Inception Module معرفی شدند
  - کانولوشن‌های دارای ابعاد مختلف و ادغام
  - سپس، خروجی تمام فیلترها به هم الحاق می‌شوند (در عمق)
- در این ساختار عمق نقشه‌ها حتما زیاد می‌شود

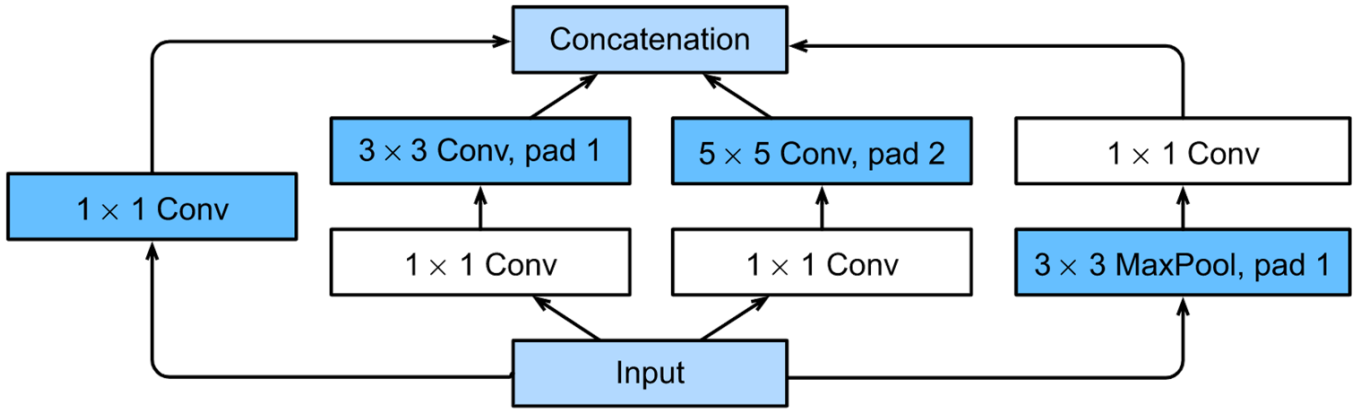


Szegedy, Christian, et al. "Going deeper with convolutions." *CVPR*, 2015.

GoogLeNet



- برای کاهش عمق و ترکیب آنها می‌توان از فیلترهای ۱×۱ استفاده کرد
- در معماری GoogLeNet از چندین بلوک Inception استفاده شده است
- برای بهبود جریان گرادیان، دو دسته‌بند کمکی در لایه‌های میانی شبکه قرار داده شده بود که امروزه با توجه به الگوریتم‌های یادگیری جدیدتر چندان ضروری نیست
- به منظور کاهش پارامترها، از GAP استفاده کرده است
  - در این مثال ادغام ۷×۷



Szegedy, Christian, et al. "Going deeper with convolutions." *CVPR*, 2015.

Global Average Pooling

- بر خلاف لایه Flatten، تعداد پارامترهای مدل را بسیار کاهش می‌دهد
- شبکه GoogLeNet با ۲۲ لایه حدود ۶.۸ میلیون پارامتر دارد در حالیکه AlexNet با ۸ لایه دارای ۶۰ میلیون و VGG با ۱۹ لایه دارای نزدیک به ۱۴۰ میلیون پارامتر است