

Resnet50:

Layer name↵	Output size↵	50-layers↵
Conv1↵	112*112↵	7*7,64,stride 2↵ 3*3 max pool,stride 2↵
Conv2_x↵	56*56↵	1*1,64↵ 3*3,64↵ 1*1,256↵ } *3
Conv3_x↵	28*28↵	1*1,128↵ 3*3,128↵ 1*1,512↵ } *4
Conv4_x↵	14*14↵	1*1,256↵ 3*3,256↵ 1*1,1024↵ } *6
Conv5_x↵	7*7↵	1*1,512↵ 3*3,512↵ 1*1,2048↵ } *3
↵	1*1↵	Average pool,6-d fc,softmax↵

VGG16:

Layer name↵
Conv3-64↵
Conv3-64↵
maxpool↵
Conv3_128↵
Conv3_128↵
maxpool↵
Conv3_256↵
Conv3_256↵
Conv3_256↵
maxpool↵
Conv3_512↵
Conv3_512↵
Conv3_512↵
maxpool↵
Conv3_512↵
Conv3_512↵
Conv3_512↵
maxpool↵
fc-4096↵
fc-4096↵
fc-6↵
softmax↵

MobileNet-v1:

	Type/Stride↵	Filter shape↵
	Conv / s2↵	3*3*3*32↵
	Conv dw / s1↵	3*3*32 dw↵
	Conv / s1↵	1*1*32*64↵
	Conv dw / s2↵	3*3*64 dw↵
	Conv / s1↵	1*1*64*128↵
	Conv dw / s1↵	3*3*128 dw↵
	Conv / s1↵	1*1*128*128↵
	Conv dw / s2↵	3*3*128 dw↵
	Conv / s1↵	1*1*128*256↵
	Conv dw / s1↵	3*3*256 dw↵
	Conv / s1↵	1*1*256*256↵
	Conv dw / s2↵	3*3*256 dw↵
	Conv / s1↵	1*1*256*512↵
5*	Conv dw / s1↵	3*3*512 dw↵
	Conv / s1↵	1*1*512*512↵
	Conv dw / s2↵	3*3*512 dw↵
	Conv / s1↵	1*1*512*1024↵
	Conv dw / s2↵	3*3*1024 dw↵
	Conv / s1↵	1*1*1024*1024↵
	Ave Pool / s1↵	Pool 7*7↵
	Fc / s1↵	1024*6↵
	Softmax / s1↵	classifier↵

Alexnet:

Layer name↵	Kernel size↵	stride↵
Conv1↵	11↵	4↵
Maxpool1↵	3↵	2↵
Conv2↵	5↵	1↵
Maxpool2↵	3↵	2↵
Conv3↵	3↵	1↵
Conv4↵	3↵	1↵
Conv5↵	3↵	1↵
Maxpool3↵	3↵	2↵
Fc1↵	4096↵	None↵
Fc1↵	4096↵	None↵
Fc1↵	6↵	None↵

GoogLeNet-Incepation-v1:

Type↵	Patch size/stride↵	Output size↵
Conv ↵	7*7/2↵	112*112*64↵
maxpool↵	3*3/2↵	56*56*64↵
Conv↵	3*3/1↵	56*56*192↵
maxpool↵	3*3/2↵	28*28*192↵
Inception(3a)↵	↵	28*28*256↵
Inception(3b)↵	↵	28*28*480↵
maxpool↵	3*3/2↵	14*14*480↵
Inception(4a)↵	↵	14*14*512↵
Inception(4b)↵	↵	14*14*512↵
Inception(4c)↵	↵	14*14*512↵
Inception(4d)↵	↵	14*14*528↵
Inception(4e)↵	↵	14*14*832↵
maxpool↵	3*3/2↵	7*7*832↵
Inception(5a)↵	↵	7*7*832↵
Inception(5b)↵	↵	7*7*1024↵
Ave pool↵	7*7/1↵	1*1*1024↵
Linear↵	↵	1*1*6↵
softmax↵	↵	1*1*6↵