PHY654

Machine learning (ML) in particle physics

Swagata Mukherjee • IIT Kanpur

23rd September 2024

October (Quiz 2 and Assignment 2)

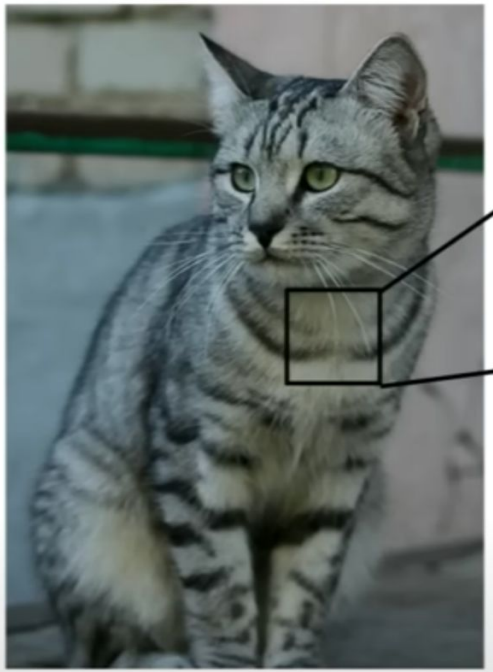
SUN	MON	TUE	WED	THU	FRI	SAT
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28 **Quiz 2**	29	30	31 **holiday**	1 Deadline for assignment 2 submission	2

Computer vision

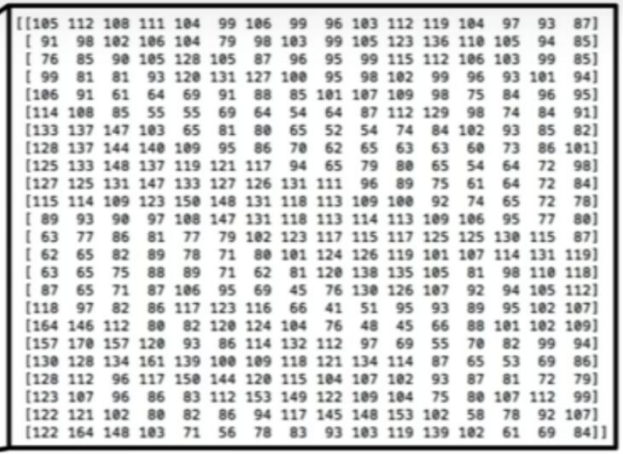
- Study of visual data (for example: image).
  - Driven by quest to train machines so that they can “see” like human beings.
- Interdisciplinary field of science.
- Challenging, because visual data is very complex.
- Computationally intensive. Needs lot of data to train.
- Some applications in society: self-driving car, face recognition system, diagnosis based on medical imaging

Image classification: a core task in computer vision. Very difficult task for a machine to do. Very easy for human beings.

The Problem: Semantic Gap



This image by Nikita is



```
[[105 112 108 111 104  99 106  99  96 103 112 119 104  97  93  87]
 [ 91  98 102 106 104  79  98 103  99 105 123 136 110 105  94  85]
 [ 76  85  90 105 128 105  87  96  95  99 115 112 106 103  99  85]
 [ 99  81  81  93 120 131 127 100  95  98 102  99  96  93 101  94]
 [106  91  61  64  69  91  88  85 101 107 109  98  75  84  96  95]
 [114 108  85  55  55  69  64  54  64  87 112 129  98  74  84  91]
 [133 137 147 103  65  81  80  65  52  54  74  84 102  93  85  82]
 [128 137 144 140 109  95  86  70  62  65  63  63  60  73  86 101]
 [125 133 148 137 119 121 117  94  65  79  80  65  54  64  72  98]
 [127 125 131 147 133 127 126 131 111  96  89  75  61  64  72  84]
 [115 114 109 123 150 148 131 118 113 109 100  92  74  65  72  78]
 [ 89  93  90  97 108 147 131 118 113 114 113 109 106  95  77  80]
 [ 63  77  86  81  77  79 102 123 117 115 117 125 125 130 115  87]
 [ 62  65  82  89  78  71  80 101 124 126 119 101 107 114 131 119]
 [ 63  65  75  88  89  71  62  81 120 138 135 105  81  98 110 118]
 [ 87  65  71  87 106  95  69  45  76 130 126 107  92  94 105 112]
 [118  97  82  86 117 123 116  66  41  51  95  93  89  95 102 107]
 [164 146 112  80  82 120 124 104  76  48  45  66  88 101 102 109]
 [157 170 157 120  93  86 114 132 112  97  69  55  70  82  99  94]
 [130 128 134 161 139 100 109 118 121 134 114  87  65  53  69  86]
 [128 112  96 117 150 144 120 115 104 107 102  93  87  81  72  79]
 [123 107  96  86  83 112 153 149 122 109 104  75  80 107 112  99]
 [122 121 102  80  82  86  94 117 145 148 153 102  58  78  92 107]
 [122 164 148 103  71  56  78  83  93 103 119 139 102  61  69  84]]
```

What the computer sees

Challenges:

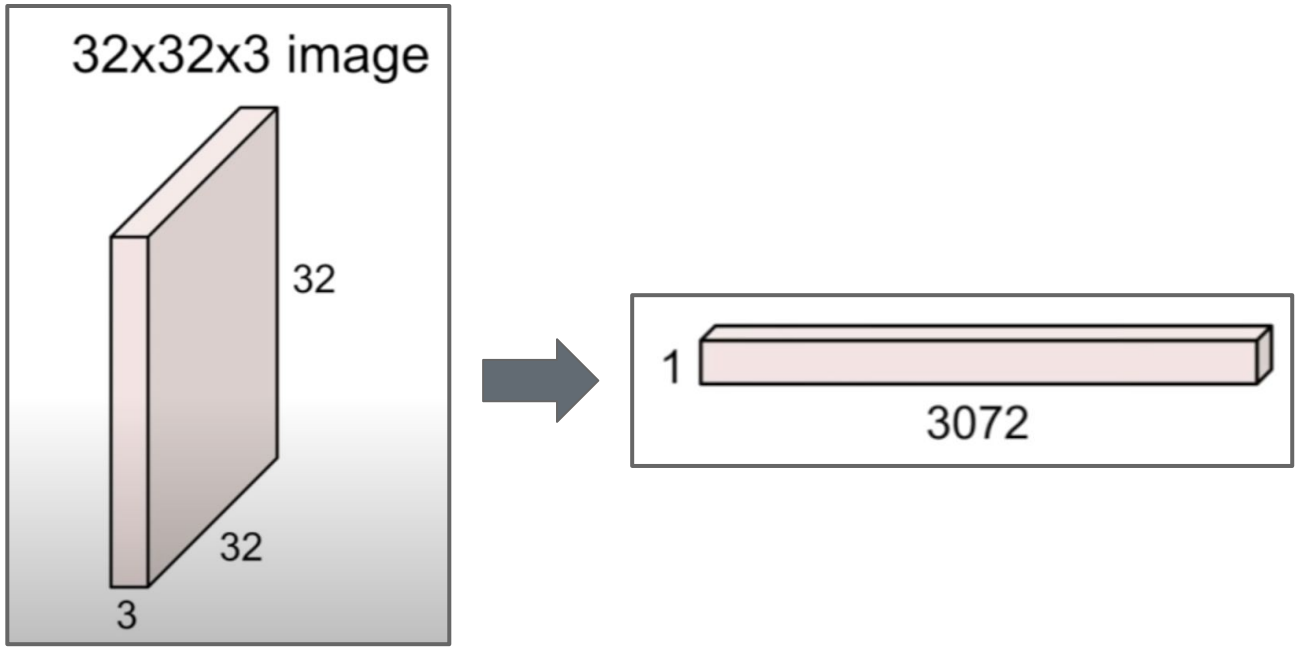
- Viewpoint / angle
- Pose / Deformation
- Illumination / light
- Obstruction
- Background clutter
- Size/Color variation

An image is just a big grid of numbers between [0, 255]:

e.g. 800 x 600 x 3
(3 channels RGB)

It is possible to train a DNN on an image (leads to too many parameters)

32x32x3 image -> stretch to 3072 x 1

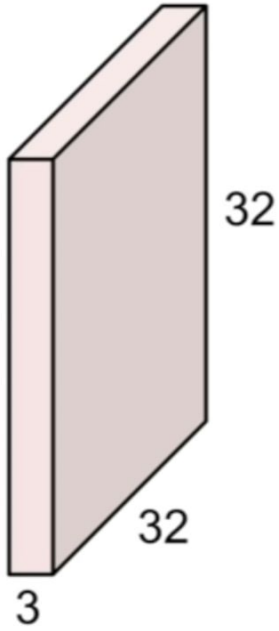


Spatial structure not preserved.

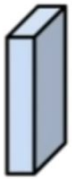
Convolution Layer

Filters always extend the full depth of the input volume

32x32x3 image

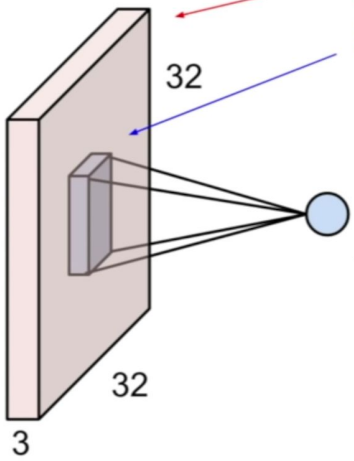


5x5x3 filter



Convolve the filter with the image i.e. “slide over the image spatially, computing dot products”

Spatial structure preserved.

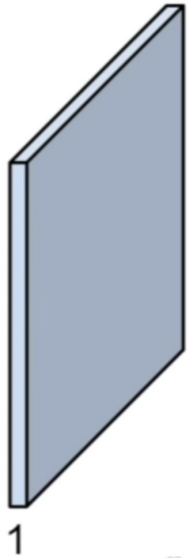


1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image (i.e. 5*5*3 = 75-dimensional dot product + bias)

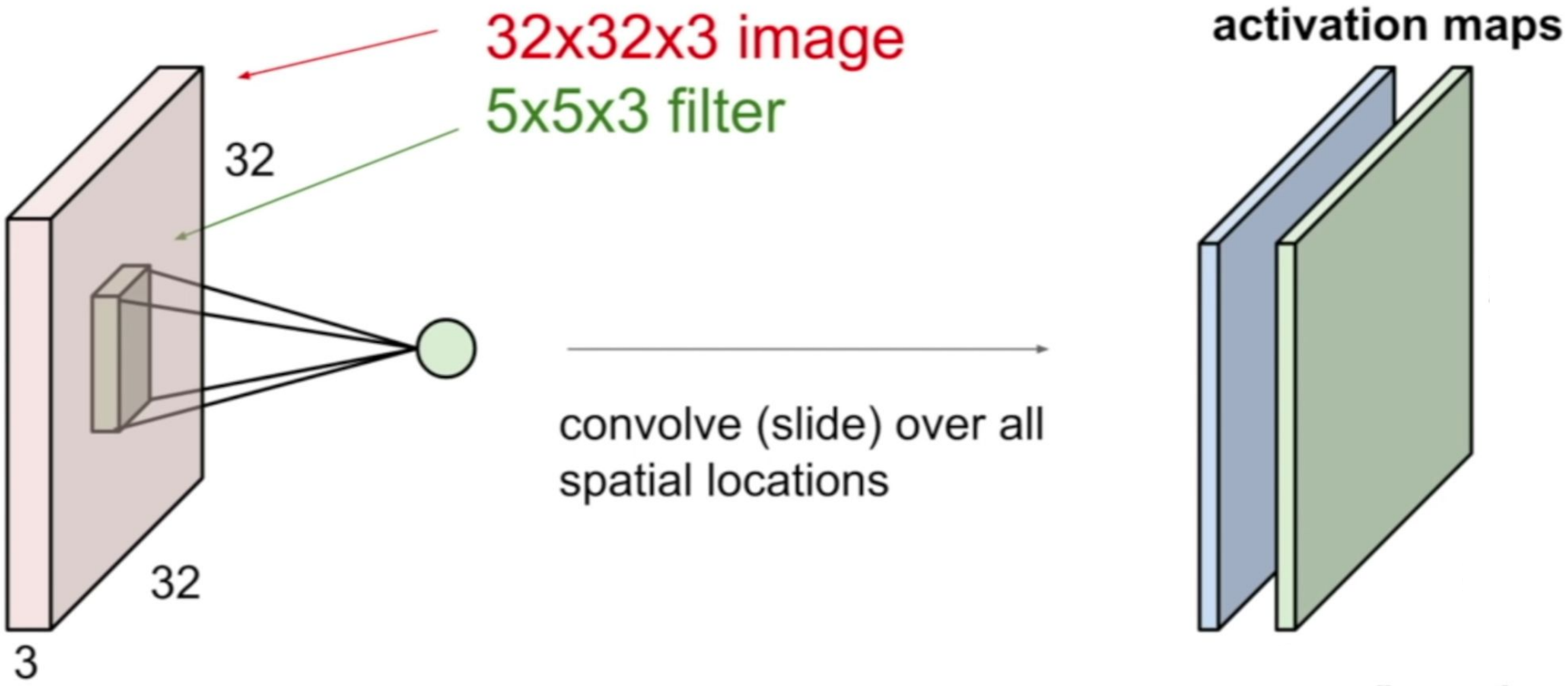
$$w^T x + b$$

convolve (slide) over all spatial locations

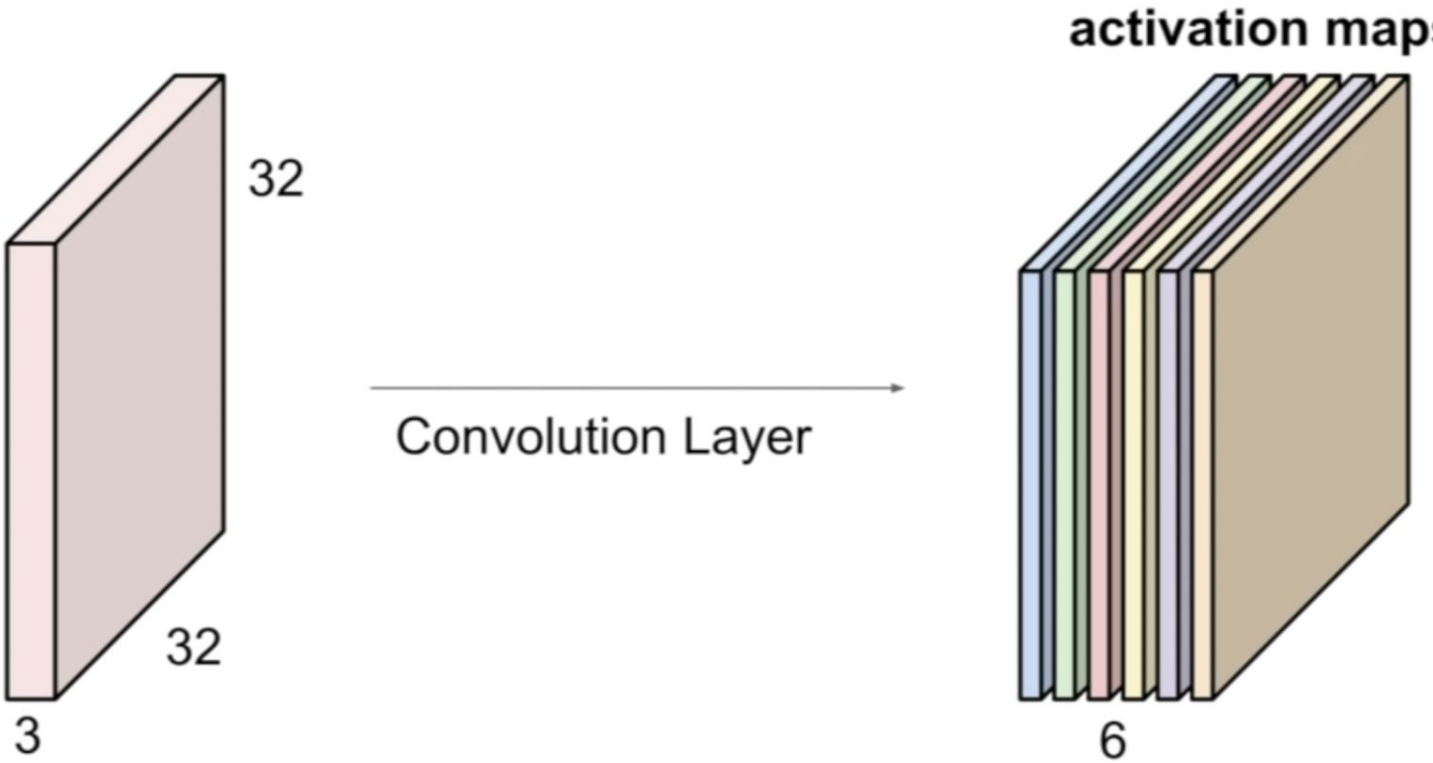
activation map

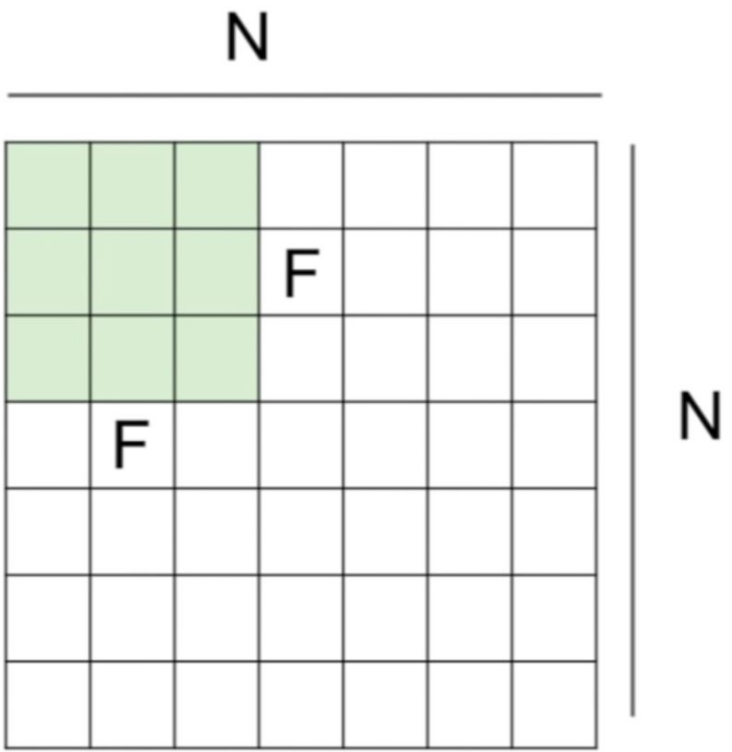


Consider a second filter (in green)



For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:





Output size:

(N - F) / stride + 1

e.g. N = 7, F = 3:

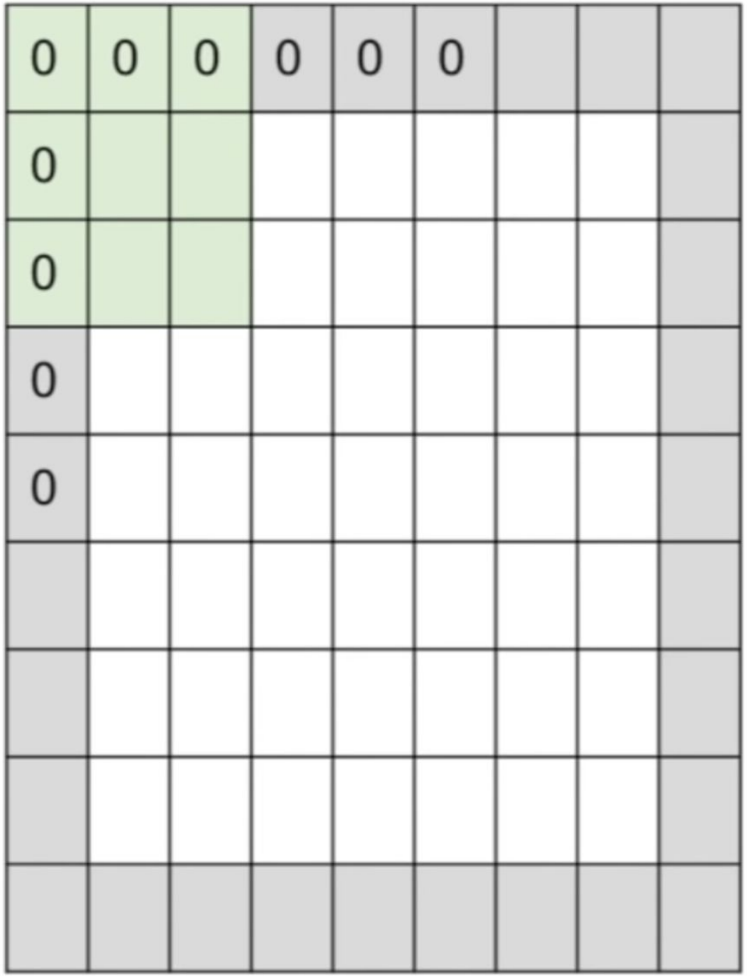
stride 1 => (7 - 3)/1 + 1 = 5

stride 2 => (7 - 3)/2 + 1 = 3

stride 3 => (7 - 3)/3 + 1 = 2.33

Avoid such cases.
Choose a different filter size.

Note:
Input image may also be non-square matrix.



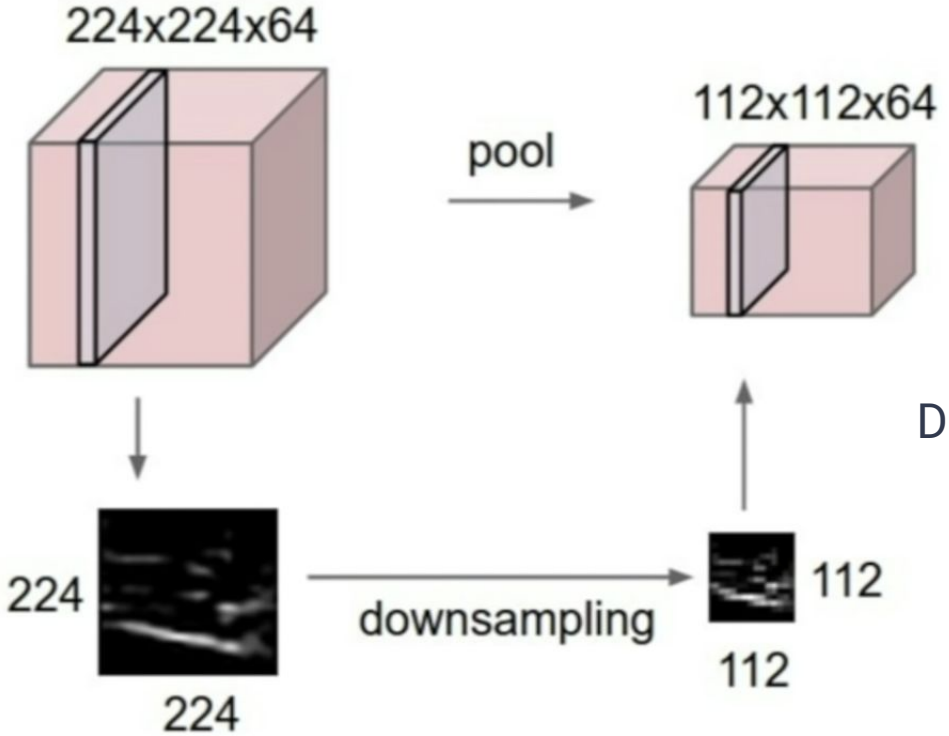
It is common to zero-pad the borders of an image.

Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Number of parameters in this layer?
each filter has 5*5*3 + 1 = 76 params (+1 for bias)
=> 76*10 = **760**

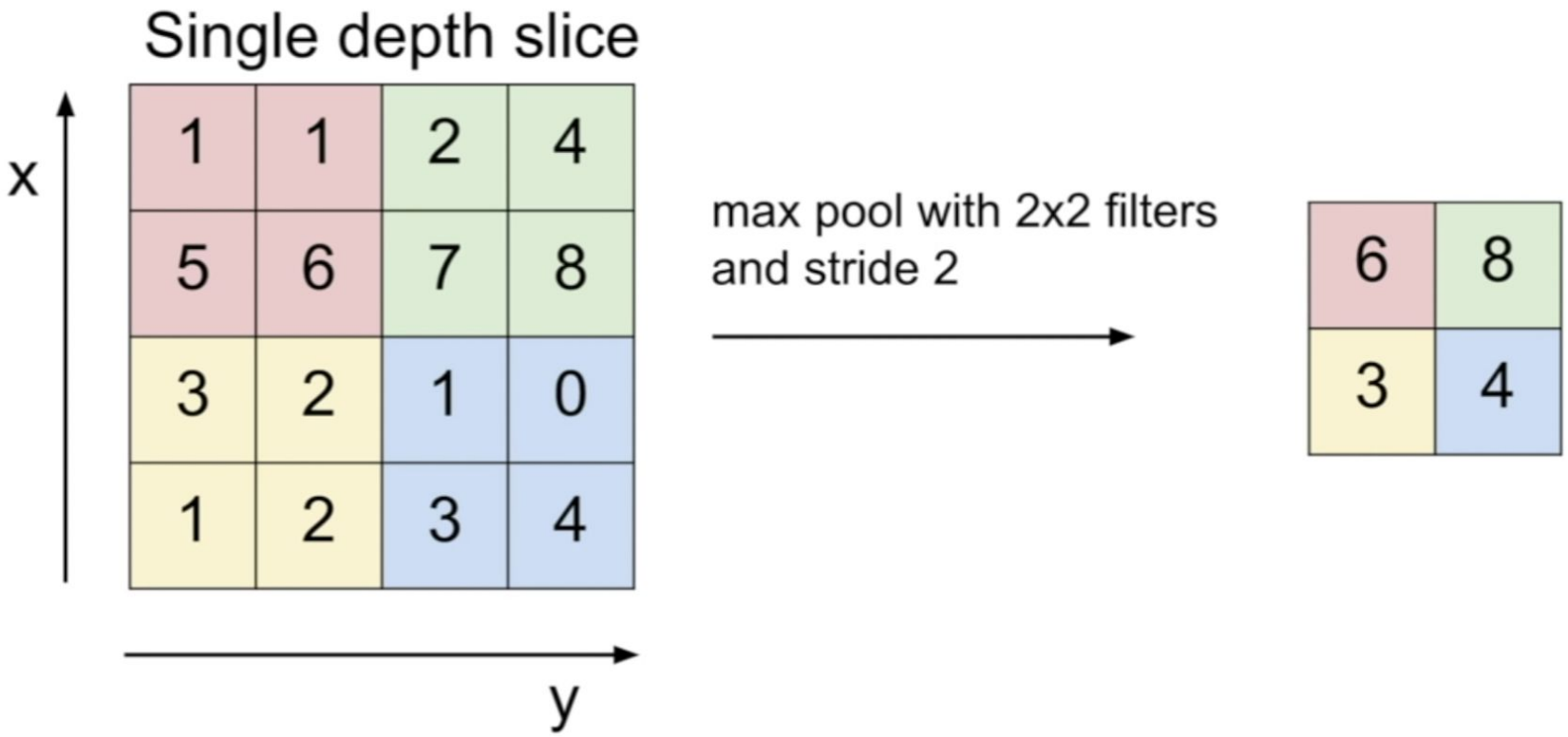
Pooling layer

- makes the representations smaller and more manageable
- operates over each activation map independently:

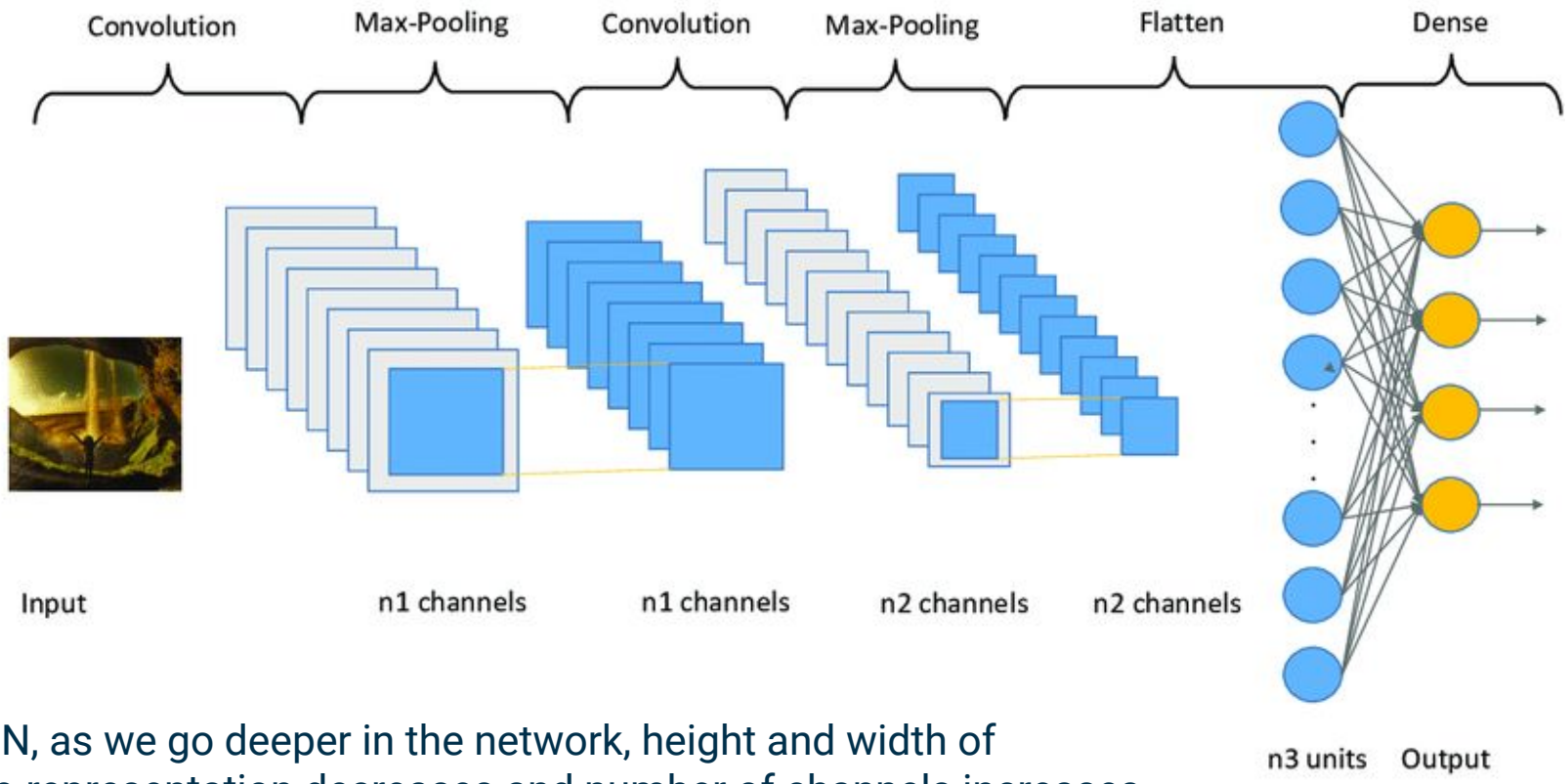


Depth remain same

MAX POOLING


Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4

A typical CNN example



In CNN, as we go deeper in the network, height and width of image-representation decreases and number of channels increases.

Classic CNN architectures

LeNet-5 http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf

~60 thousand parameters

AlexNet https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

~60 million parameters

VGG-16 https://arxiv.org/abs/1409.1556

~138 million parameters

Classic CNN architectures

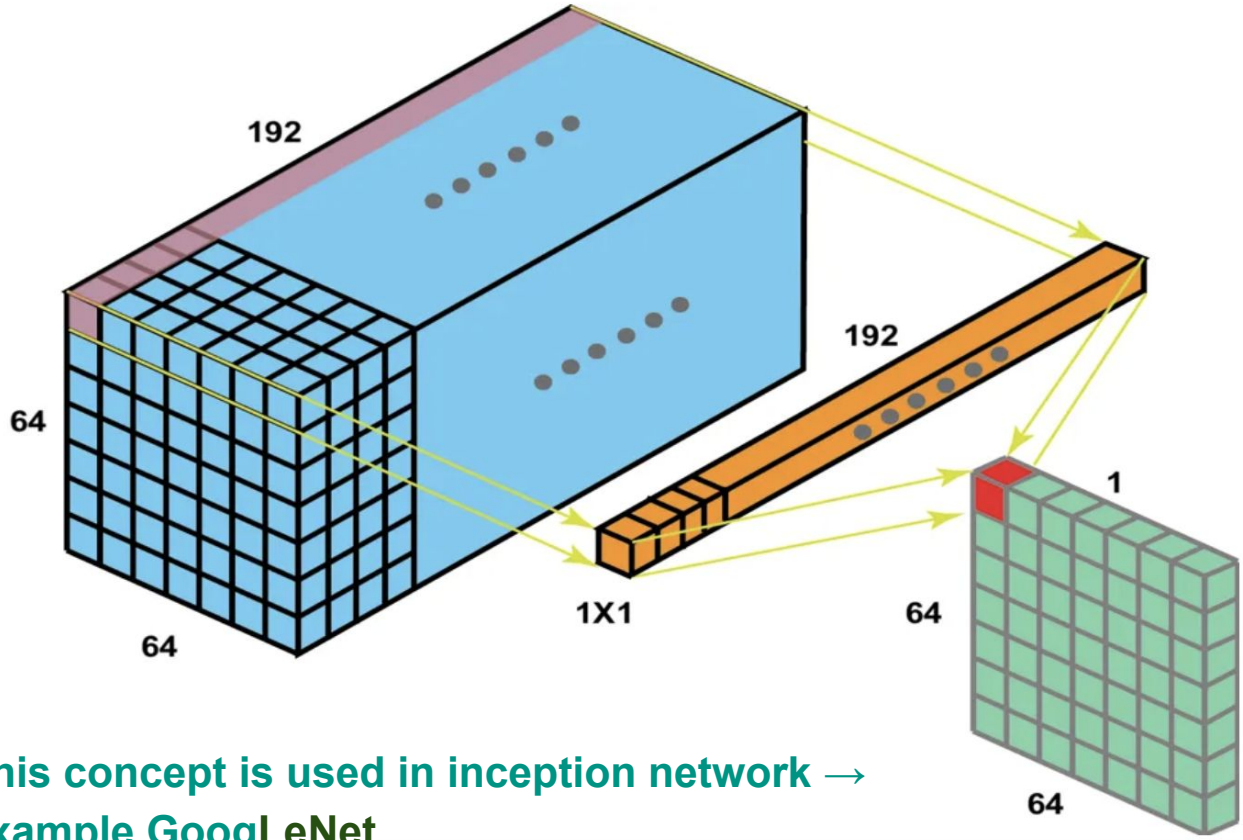
Available models

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
ResNet50V2	98	76.0%	93.0%	25.6M	103	45.6	4.4
ResNet101	171	76.4%	92.8%	44.7M	209	89.6	5.2
ResNet101V2	171	77.2%	93.8%	44.7M	205	72.7	5.4
ResNet152	232	76.6%	93.1%	60.4M	311	127.4	6.5
ResNet152V2	232	78.0%	94.2%	60.4M	307	107.5	6.6
InceptionV3	92	77.9%	93.7%	23.9M	189	42.2	6.9
InceptionResNetV2	215	80.3%	95.3%	55.9M	449	130.2	10.0
MobileNet	16	70.4%	89.5%	4.3M	55	22.6	3.4
MobileNetV2	14	71.3%	90.1%	3.5M	105	25.9	3.8
DenseNet121	33	75.0%	92.3%	8.1M	242	77.1	5.4
DenseNet169	57	76.2%	93.2%	14.3M	338	96.4	6.3
DenseNet201	80	77.3%	93.6%	20.2M	402	127.2	6.7
NASNetMobile	23	74.4%	91.9%	5.3M	389	27.0	6.7
NASNetLarge	343	82.5%	96.0%	88.9M	533	344.5	20.0
EfficientNetB0	29	77.1%	93.3%	5.3M	132	46.0	4.9

https://keras.io/api/applications/

Keras Applications are deep learning models that are made available alongside pre-trained weights.

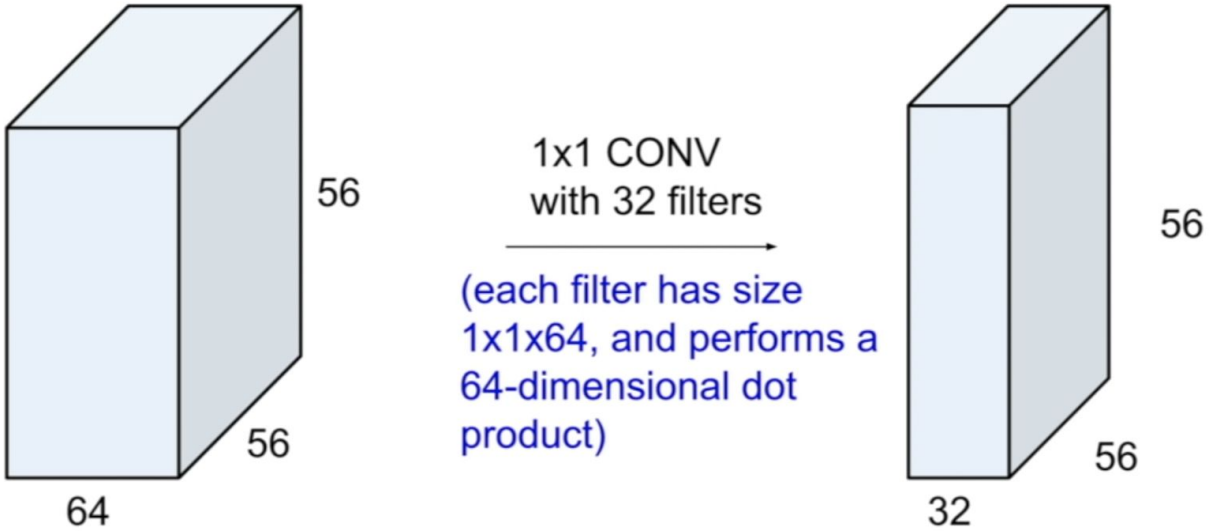
1 x 1 convolution (network in network)



It is useful to shrink the number of channels (or depth) when necessary.

This concept is used in inception network → example GoogLeNet

1 x 1 convolution (network in network)



It is useful to shrink the number of channels (or depth) when necessary.

This concept is used in inception network → example GoogLeNet

Fun fact: Where does the name inception come from?

The paper actually cites this meme. http://knowyourmeme.com/memes/we-need-to-go-deeper

More on parameters of CNN

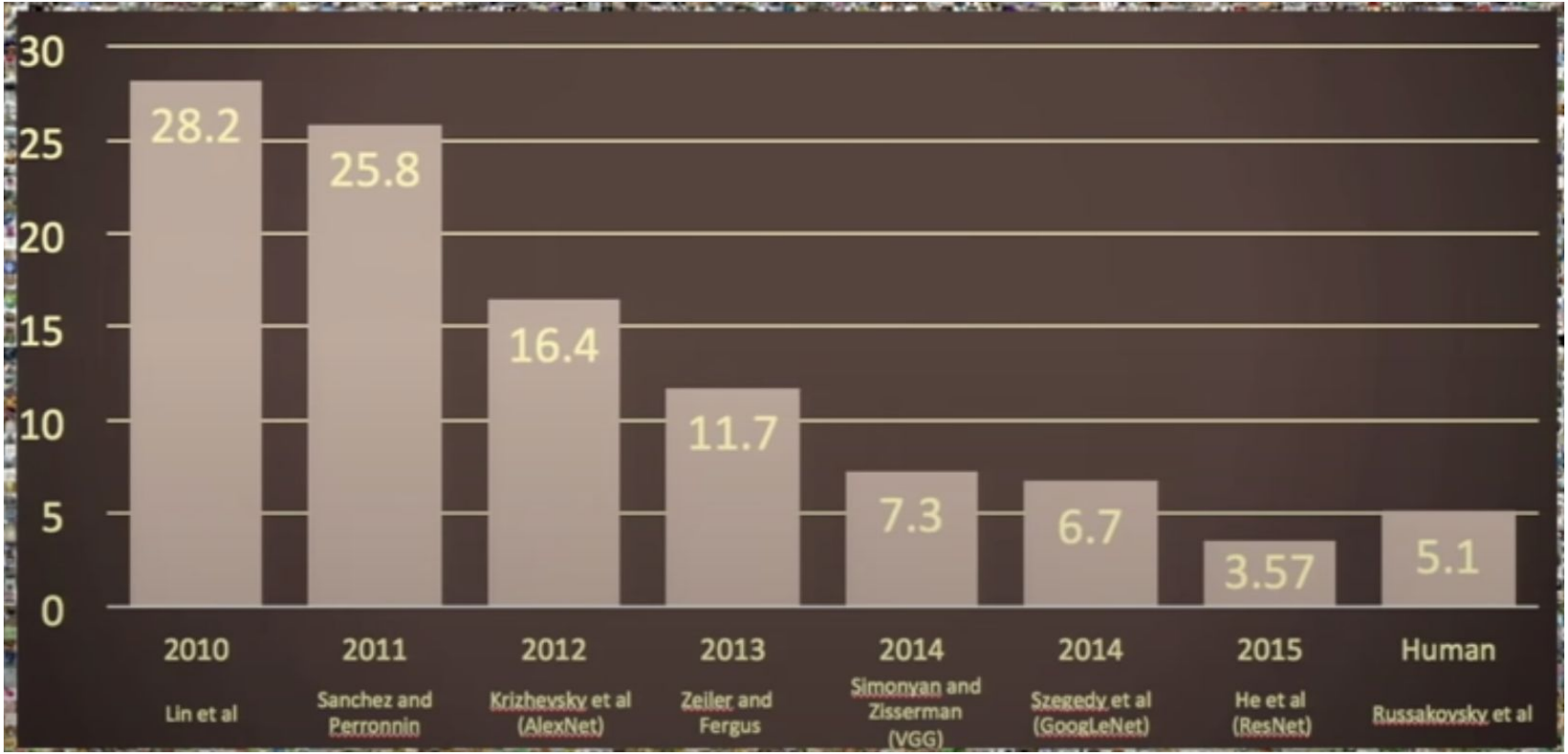
Pooling layers have no parameters.

Conv layers have relatively few parameters. Most parameters are in the fully connected layers.

Activation size goes down gradually as we go deeper in the network. If this drops very quickly, that is generally not good for performance.

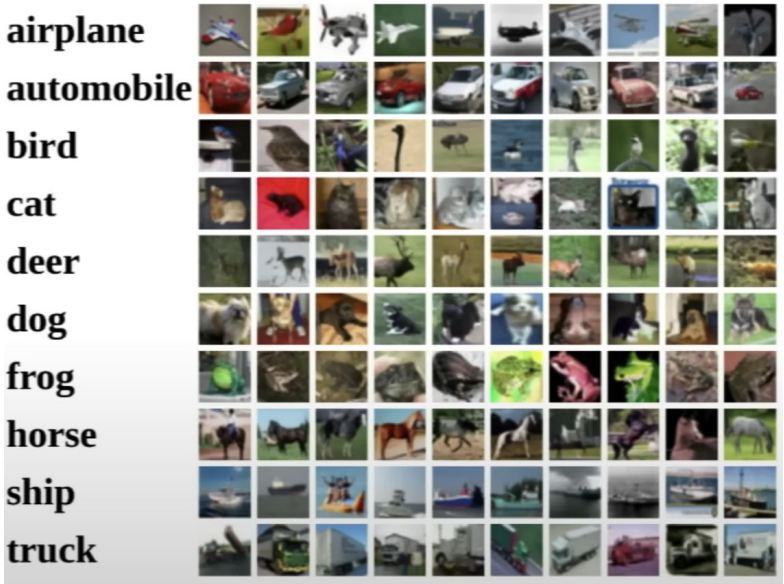
IMAGENET

ImageNet Large Scale Visual Recognition Challenge (Competition)



Example dataset: CIFAR 10

10 classes
50,000 training images
10,000 testing images



Widely used dataset in the world of computer vision.

Data augmentation

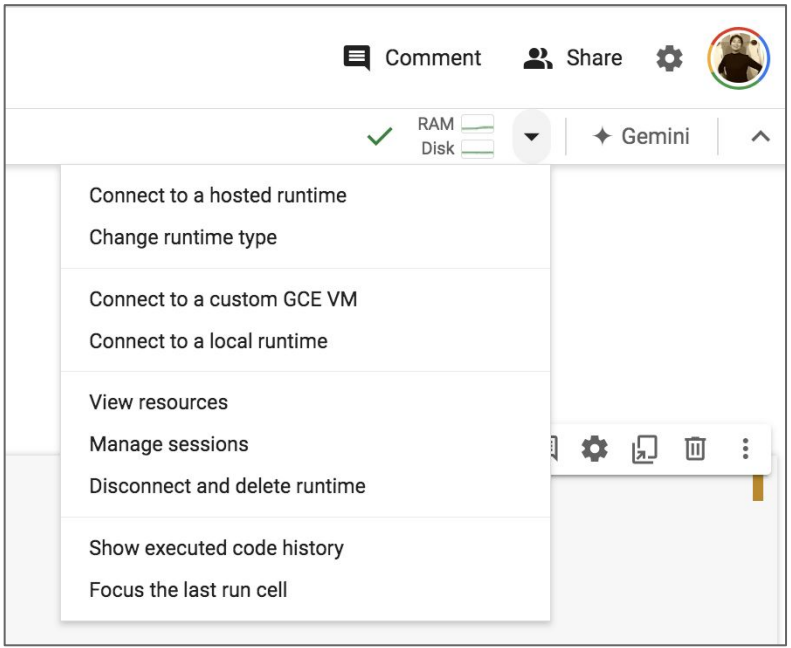
Original image

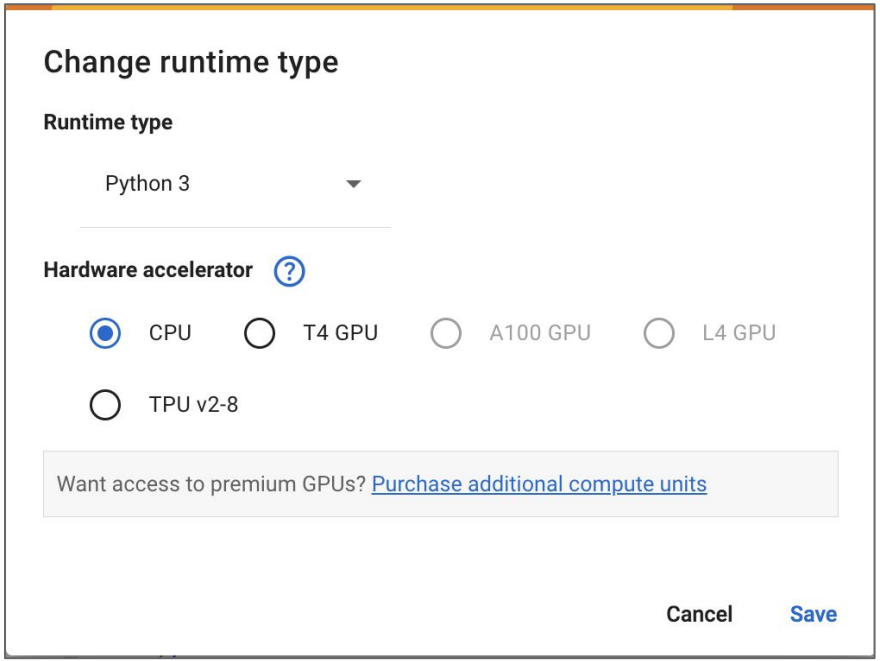
Flip

Rotation + zoom

More data helps to solve overfitting issue

CNN might need GPUs

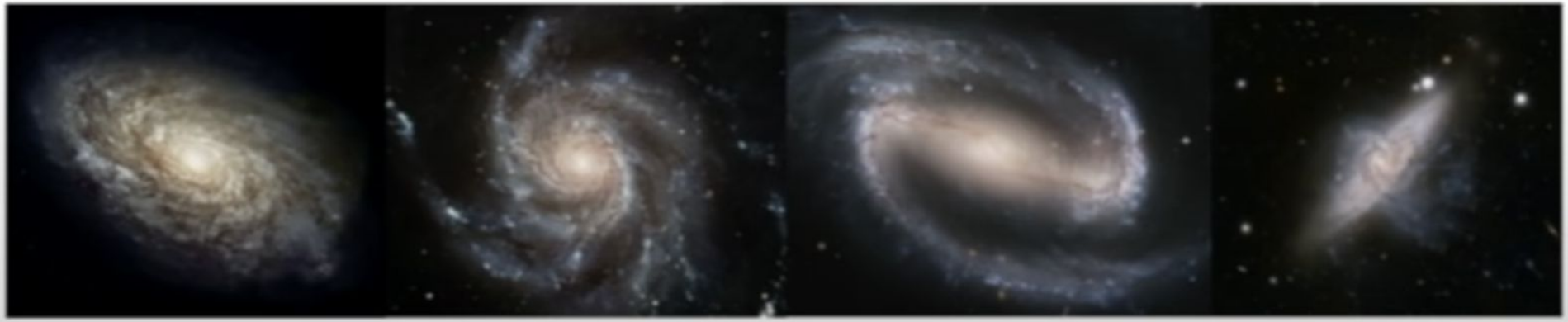




GPUs not guaranteed in colab.
Colab GPU may be fine to teaching / learning purpose. Not reliable for real research. You need your own GPU.

CNN: application in astrophysics

Galaxy classification



From left to right: public domain by NASA, usage permitted by ESA/Hubble, public domain by NASA, and public domain.

[Dieleman et al. 2014]