

CONVOLUTIONAL NEURAL NETWORKS

Discrete Convolution

1D:
[Spectra]

$$f(x) * g(x) = \sum_{k=-\infty}^{k=+\infty} f(k).g(k - x)$$

2D:
[Images]

$$f(x, y) * g(x, y) = \sum_{k=-\infty}^{k=+\infty} \sum_{l=-\infty}^{l=+\infty} f(k, l).g(x - k, y - l)$$

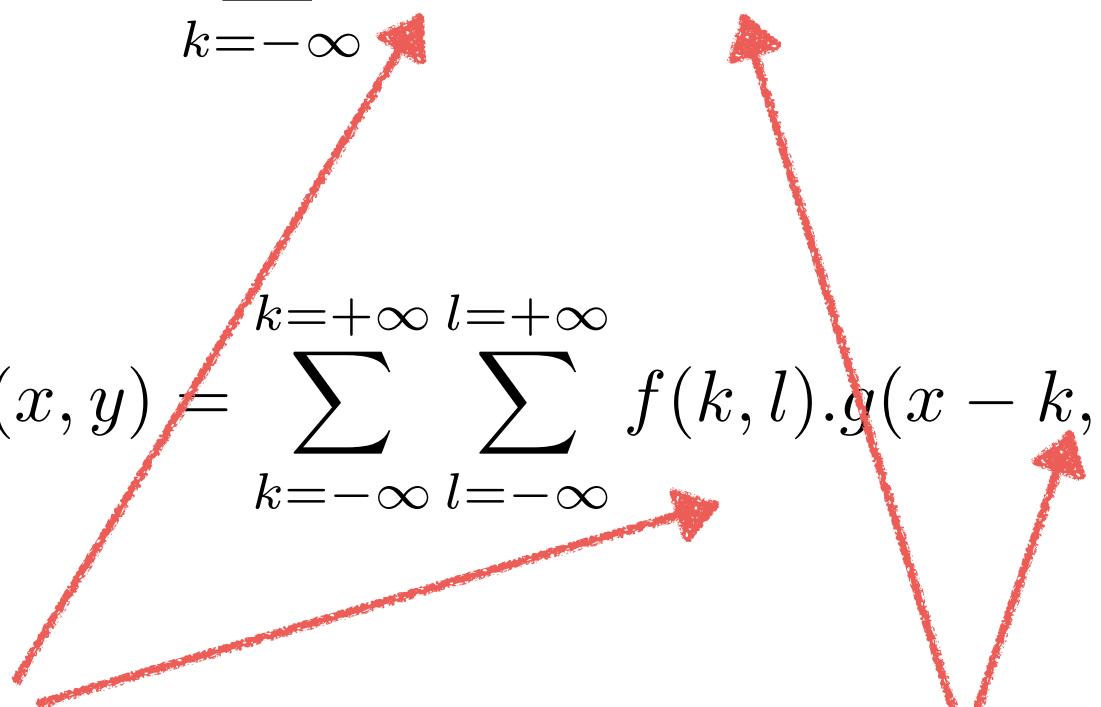
DISCRETE CONVOLUTION

1D:
[Spectra]

$$f(x) * g(x) = \sum_{k=-\infty}^{k=+\infty} f(k).g(k - x)$$

2D:
[Images]

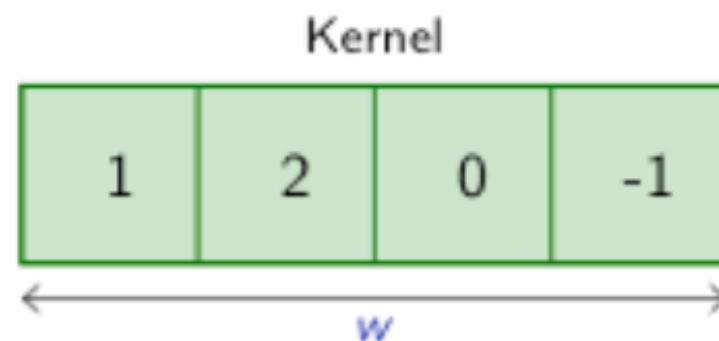
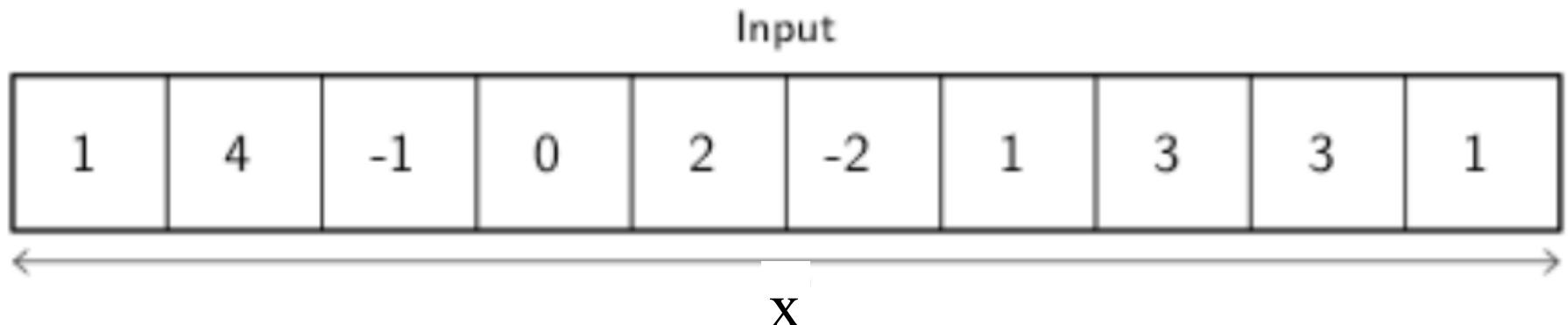
$$f(x, y) * g(x, y) = \sum_{k=-\infty}^{k=+\infty} \sum_{l=-\infty}^{l=+\infty} f(k, l).g(x - k, y - l)$$



CONVOLUTION KERNEL

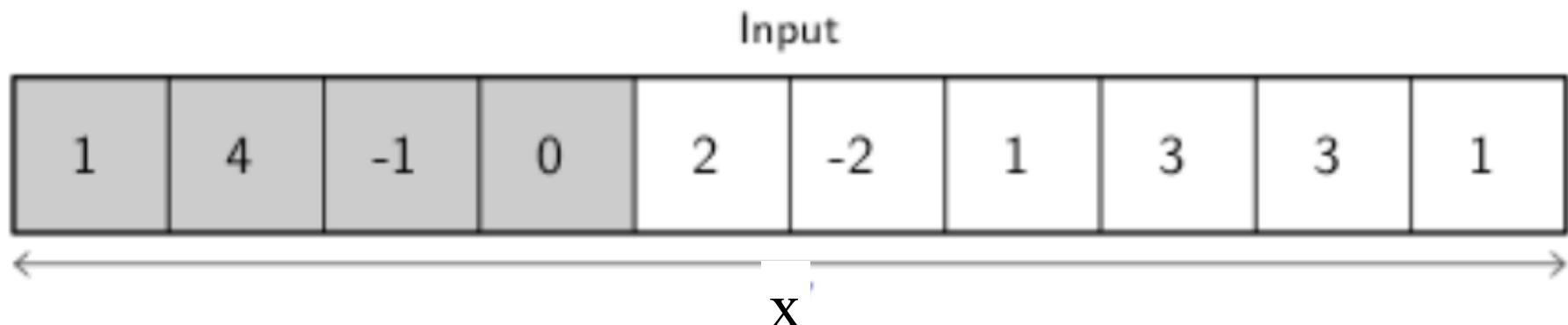
INPUT DATA

1-D CONVOLUTION



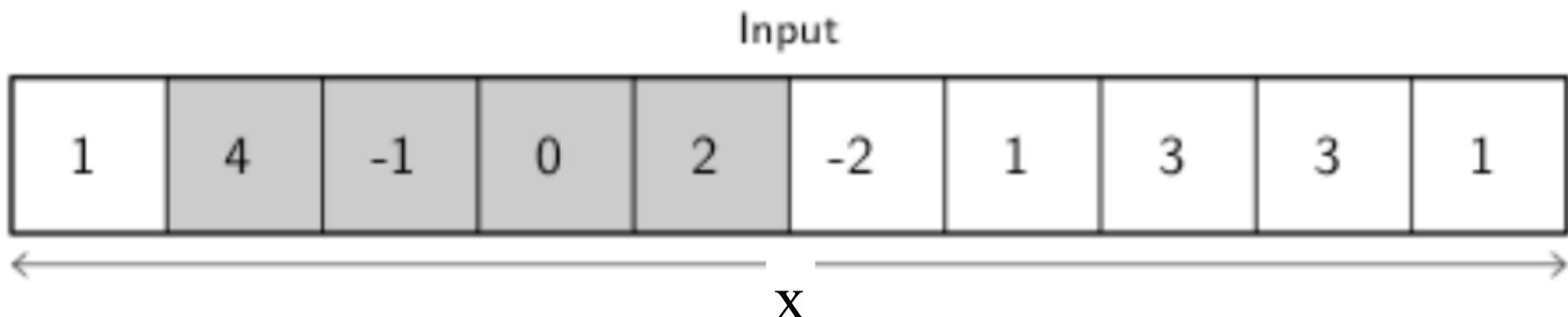
credit

1-D CONVOLUTION



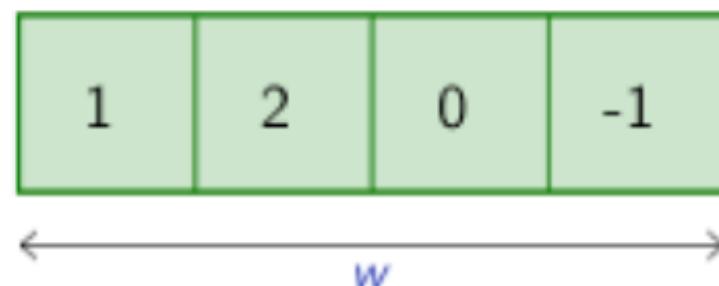
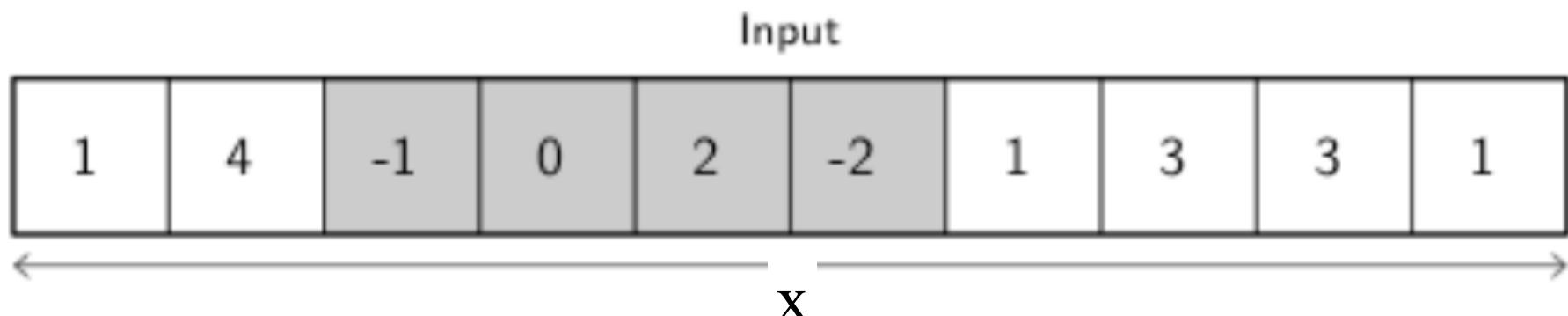
credit

1-D CONVOLUTION



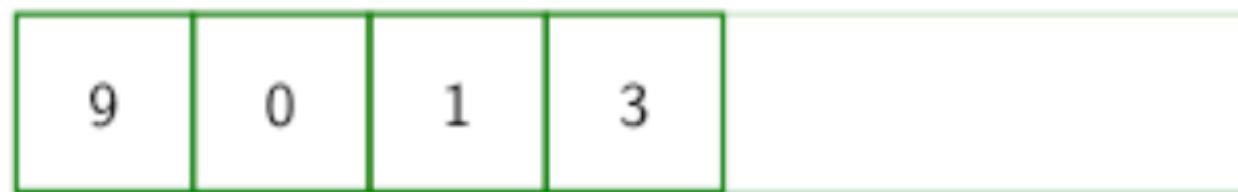
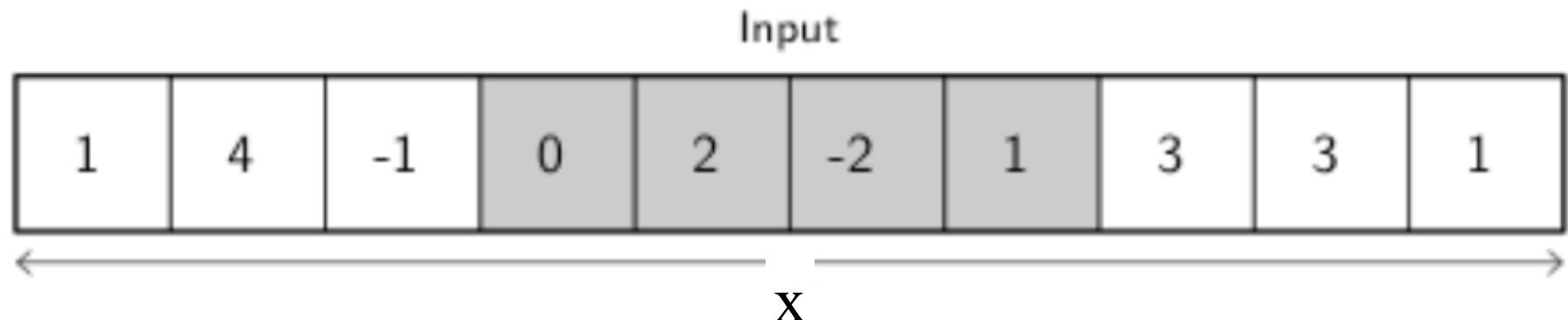
credit

1-D CONVOLUTION



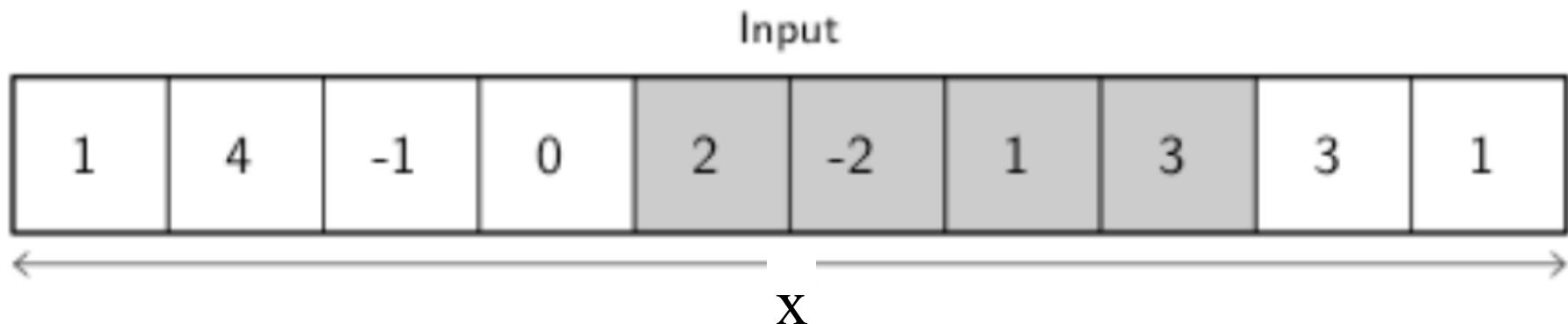
credit

1-D CONVOLUTION



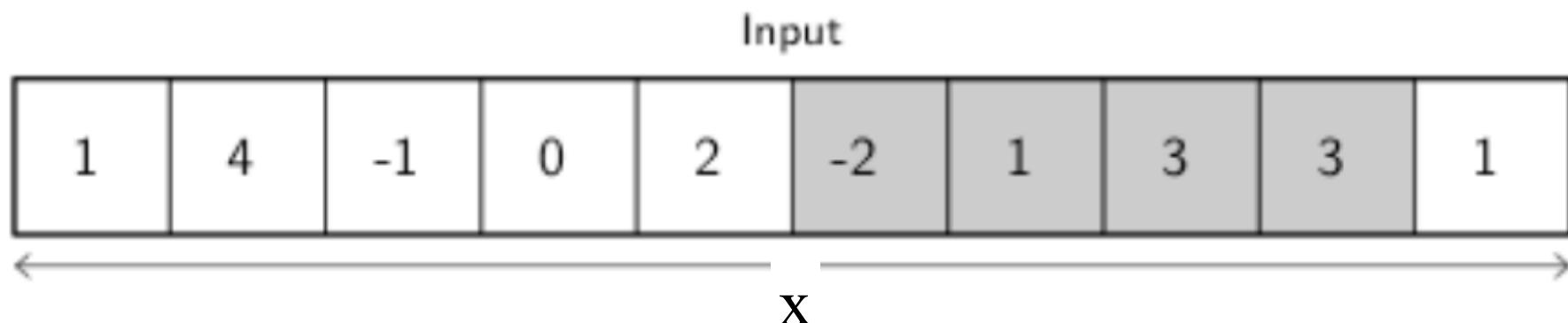
credit

1-D CONVOLUTION



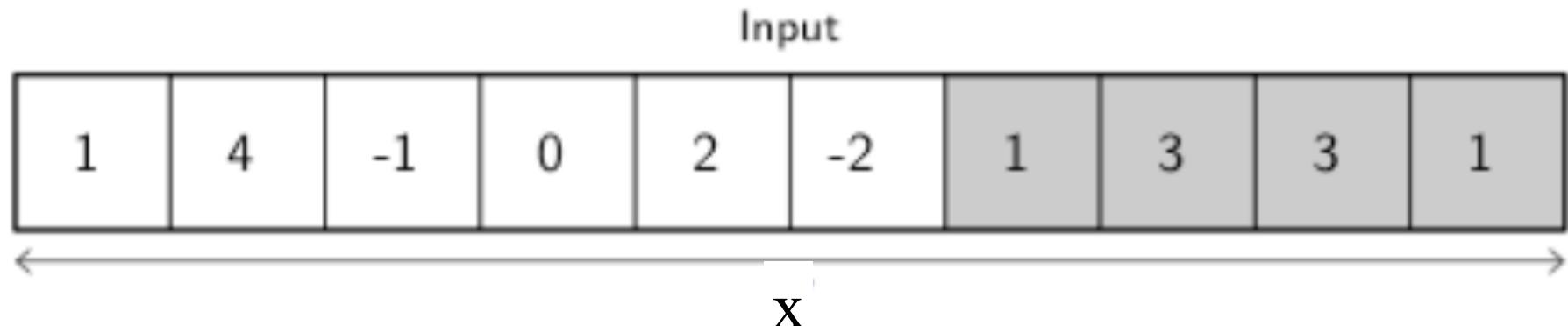
credit

1-D CONVOLUTION



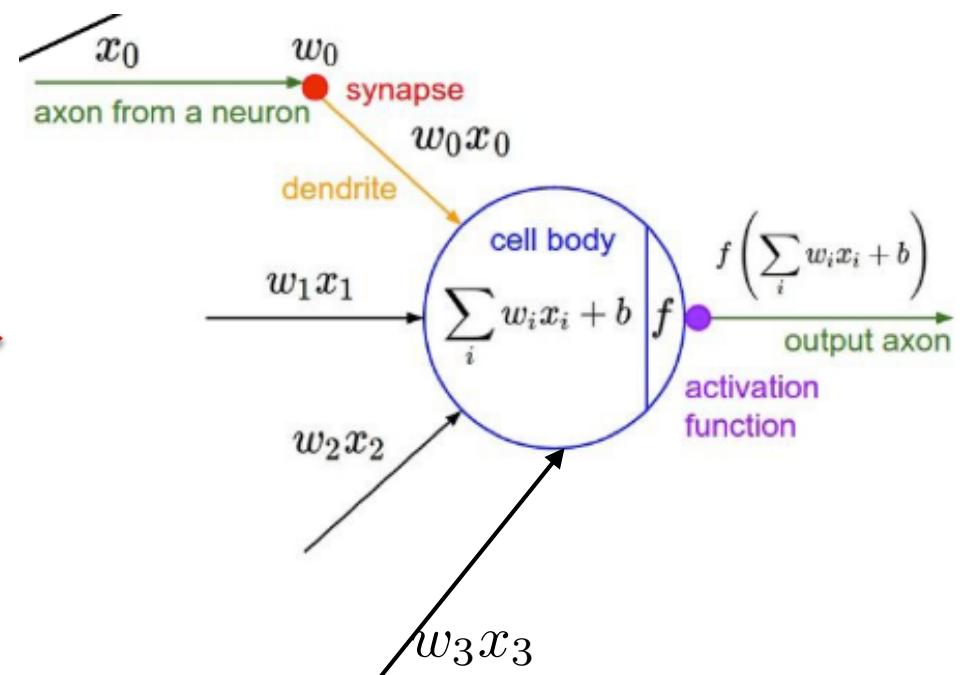
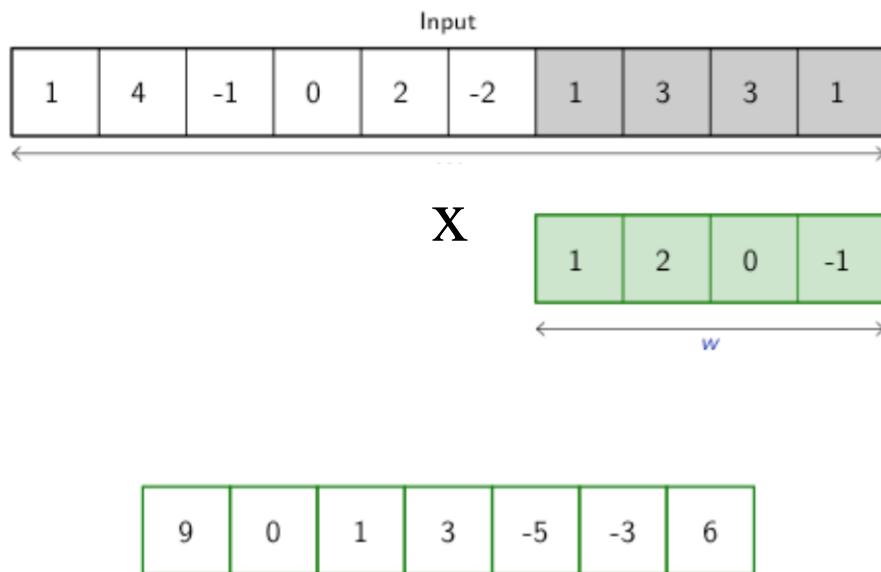
credit

1-D CONVOLUTION

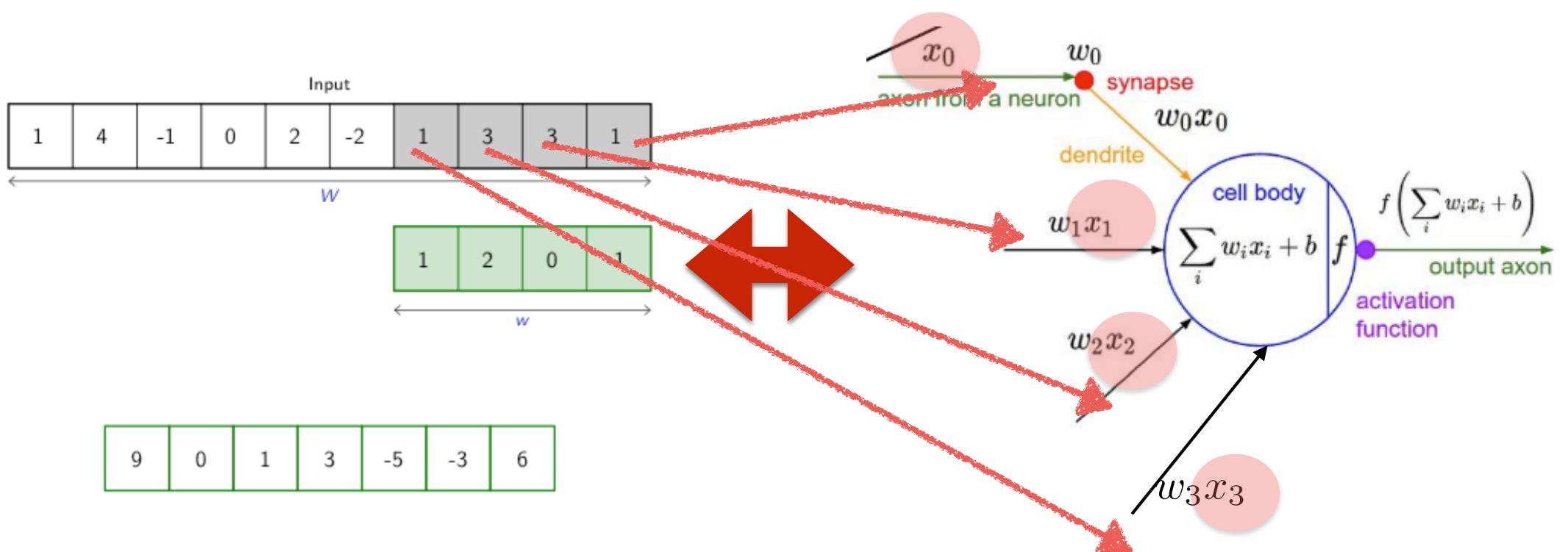


credit

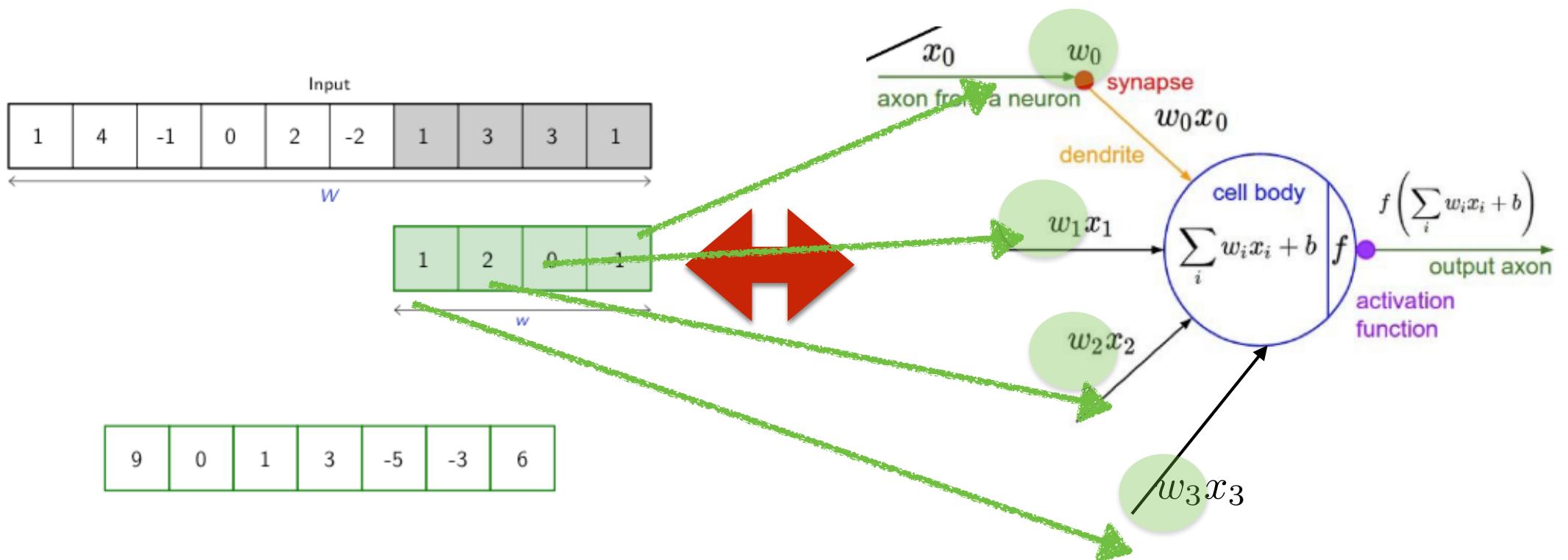
THE CONVOLUTION BUILDING BLOCK OPERATION (BEFORE ACTIVATION) IS EQUIVALENT TO A NEURON WITH AS MANY INPUTS AS KERNEL ELEMENTS AND WEIGHTS EQUAL TO THE KERNEL



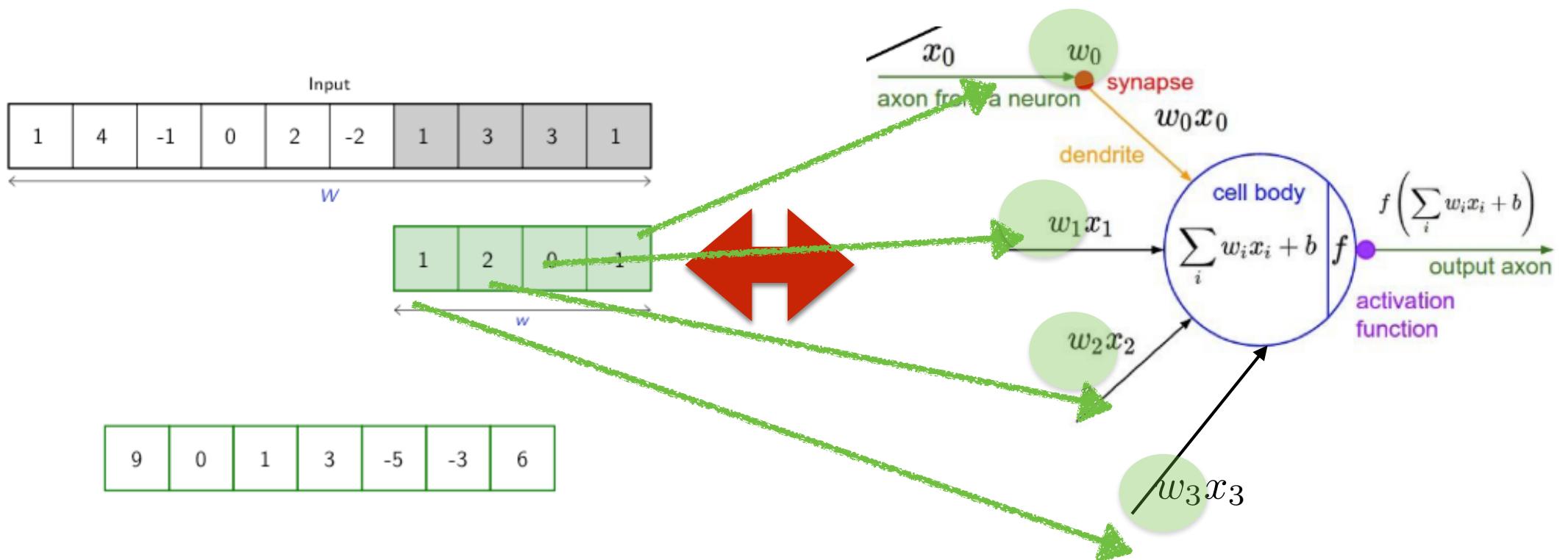
THE CONVOLUTION BUILDING BLOCK OPERATION (BEFORE ACTIVATION) IS EQUIVALENT TO A NEURON WITH AS MANY INPUTS AS KERNEL ELEMENTS AND WEIGHTS EQUAL TO THE KERNEL



THE CONVOLUTION BUILDING BLOCK OPERATION (BEFORE ACTIVATION) IS EQUIVALENT TO A NEURON WITH AS MANY INPUTS AS KERNEL ELEMENTS AND WEIGHTS EQUAL TO THE KERNEL



THE CONVOLUTION BUILDING BLOCK OPERATION (BEFORE ACTIVATION) IS EQUIVALENT TO A NEURON WITH AS MANY INPUTS AS KERNEL ELEMENTS AND WEIGHTS EQUAL TO THE KERNEL



WITH THE ADVANTAGE THAT THE SAME WEIGHTS ARE APPLIED TO ALL THE SIGNAL: TRANSLATION INVARIANCE

2-D CONVOLUTION

SAME IDEA, BUT THE KERNEL IS NOW 2D

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

KERNEL

INPUT (IMAGE)

OUTPUT

Credit: animations from https://github.com/vdumoulin/conv_arithmetic

2-D CONVOLUTION

SAME IDEA, BUT THE KERNEL IS NOW 2D

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

IN THE EXAMPLE: EACH 3x3 REGION GENERATES AN OUTPUT

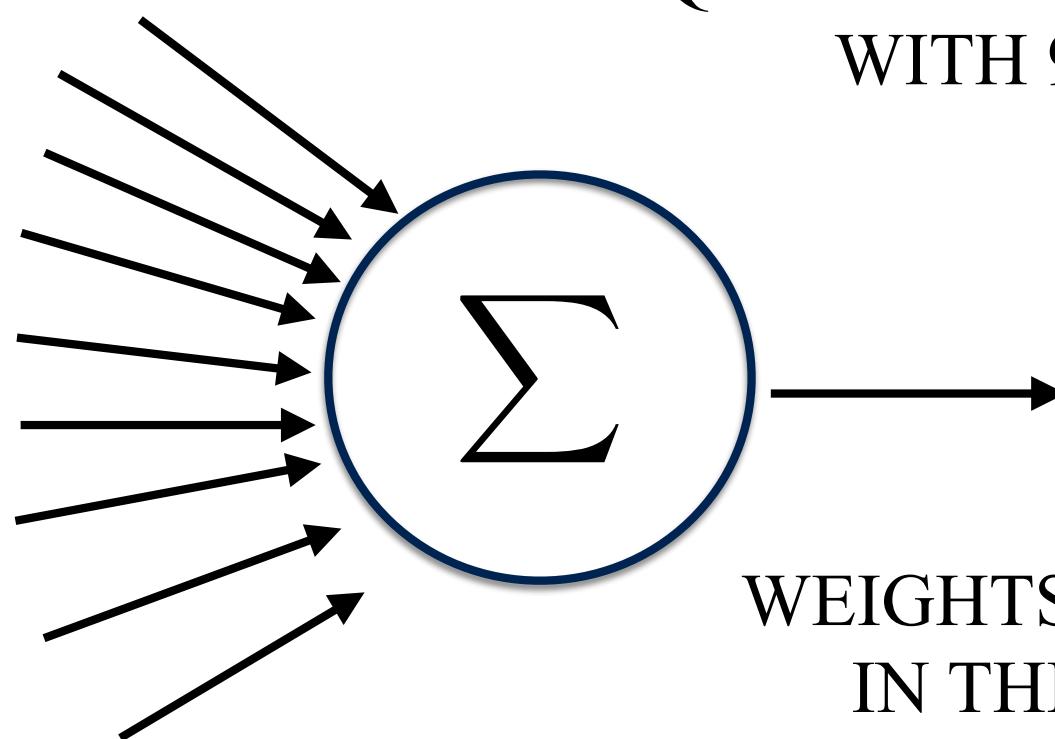
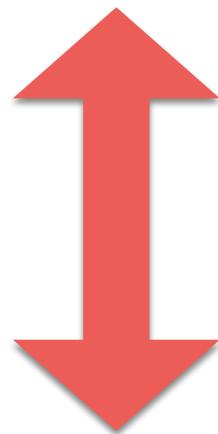
$$Size_{output} = Size_{input} - Size_{kernel} + 1$$

Credit: animations from https://github.com/vdumoulin/conv_arithmetic

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3



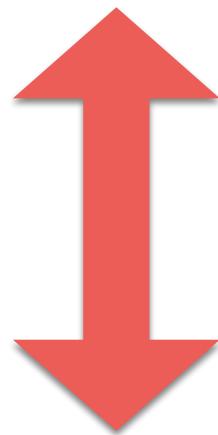
EQUIVALENT TO A NEURON
WITH 9 INPUTS

WEIGHTS ARE CODED
IN THE KERNEL

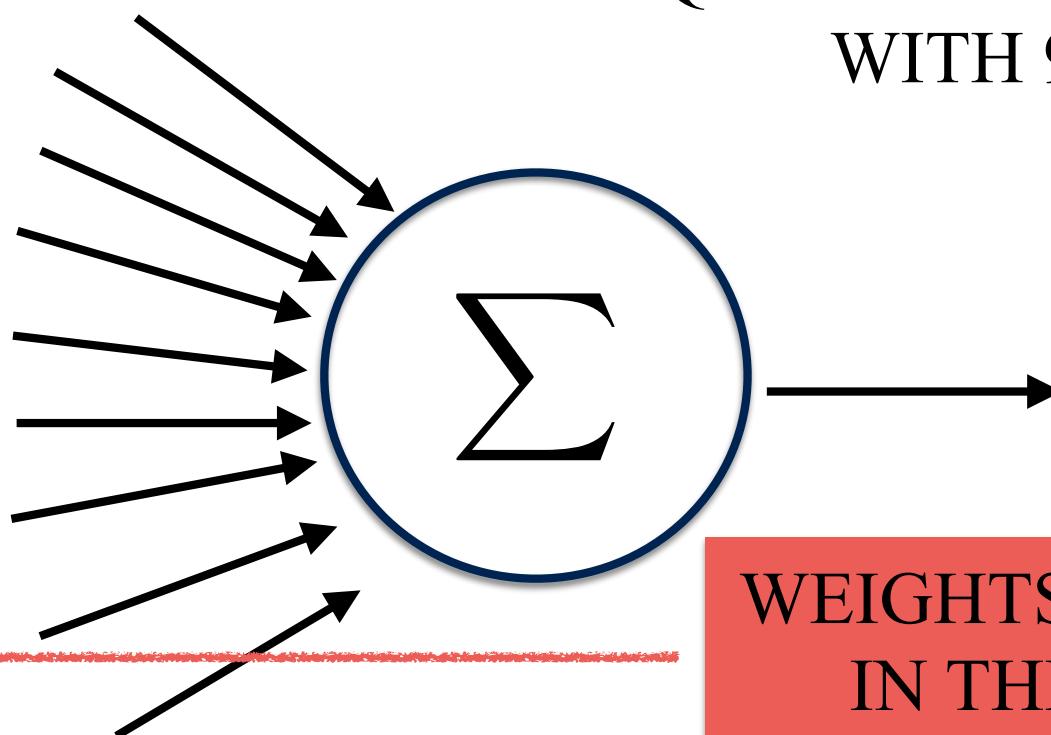
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3



EQUIVALENT TO A NEURON
WITH 9 INPUTS



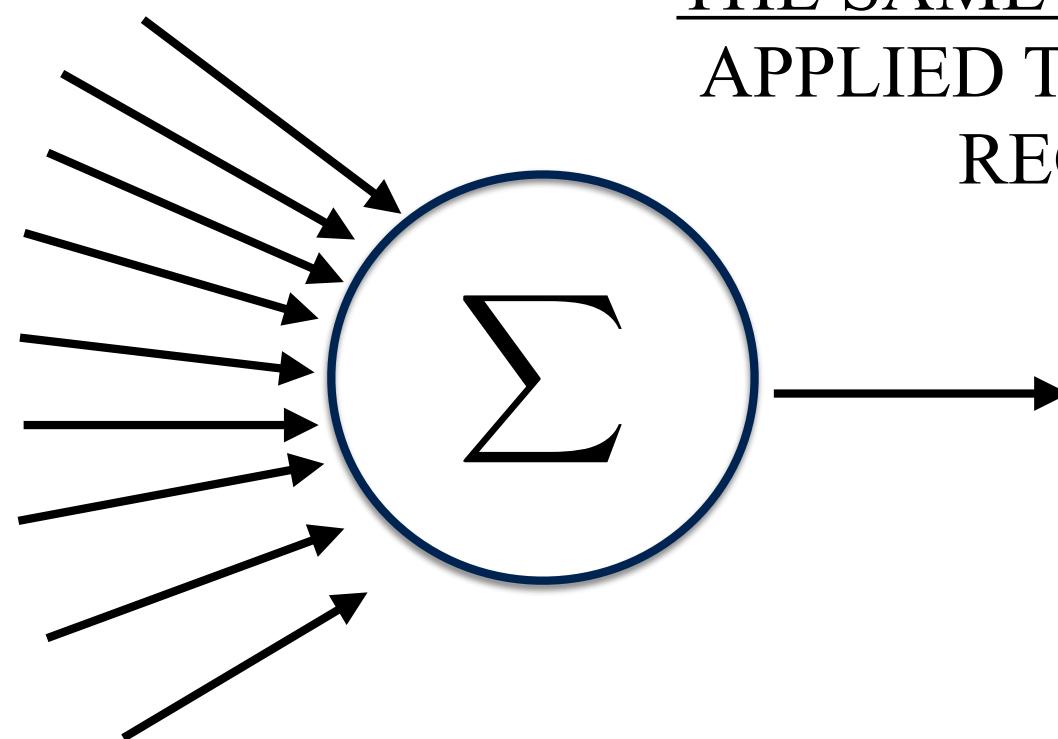
THIS IS WHAT
THE
NETWORK
LEARNS!

WEIGHTS ARE CODED
IN THE KERNEL

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

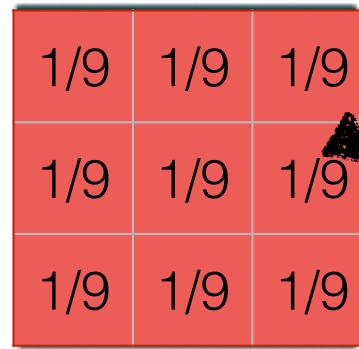
3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

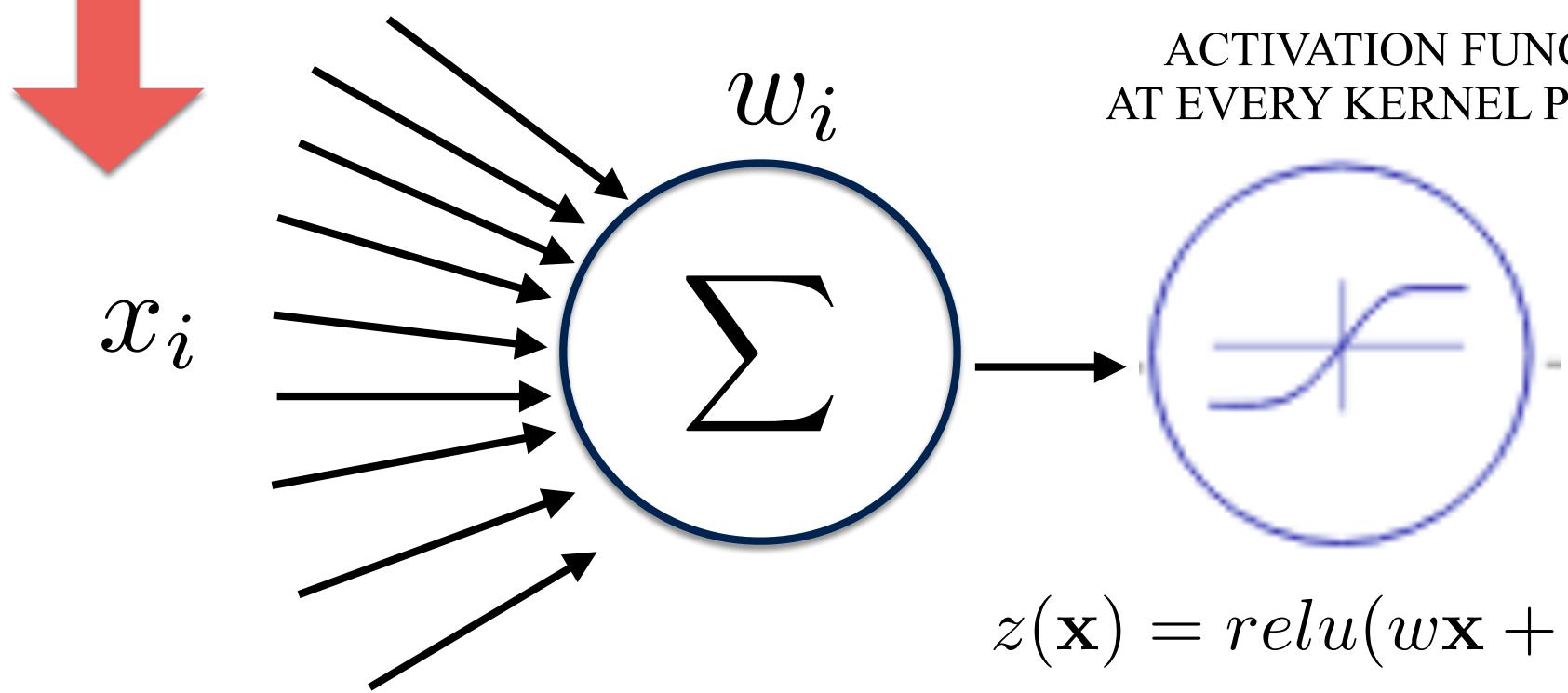
1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3



THE KEY IS AGAIN THAT
THE SAME WEIGHTS ARE
APPLIED TO ALL IMAGE
REGIONS

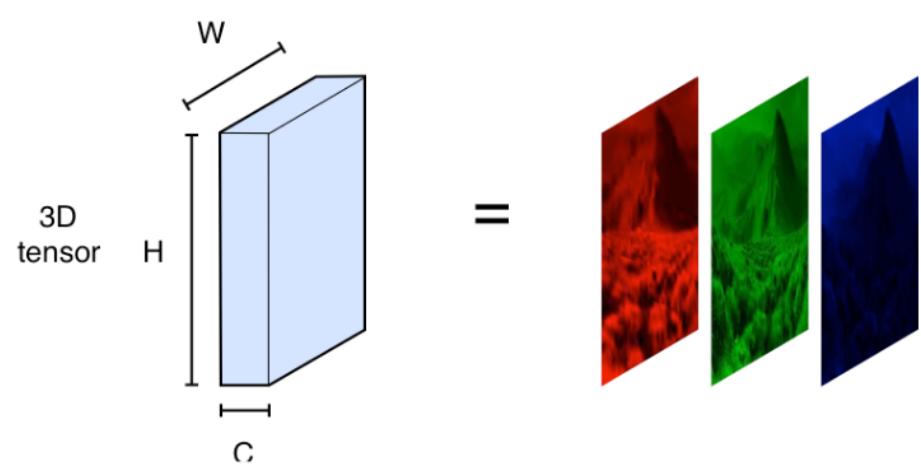
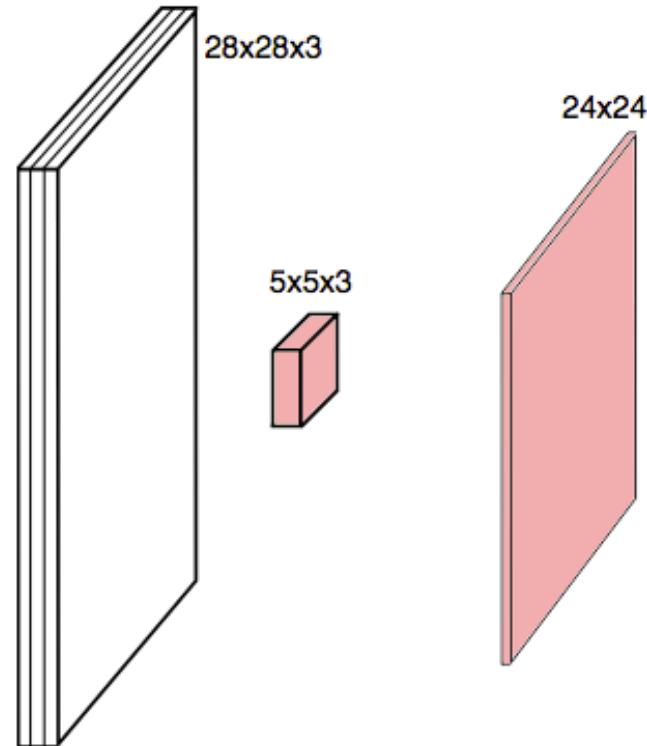
x_i		
		

w_i	
[weights]	



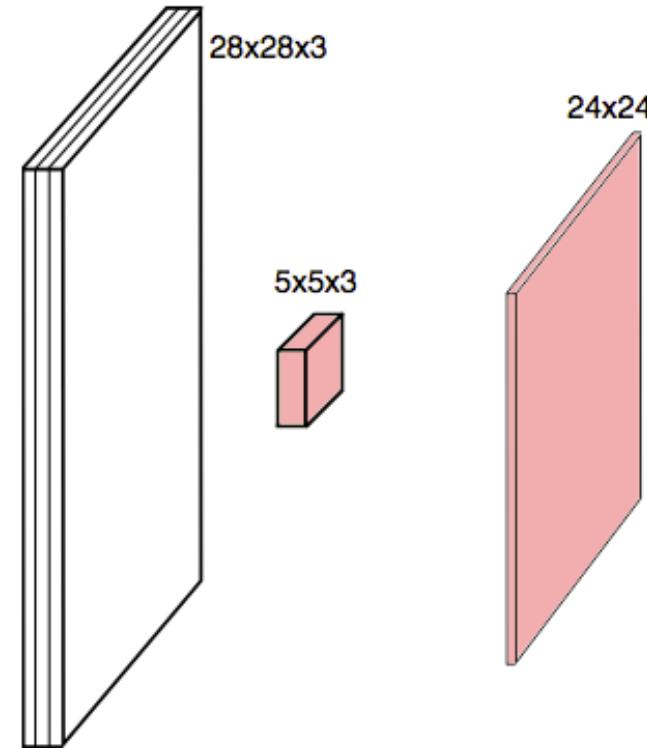
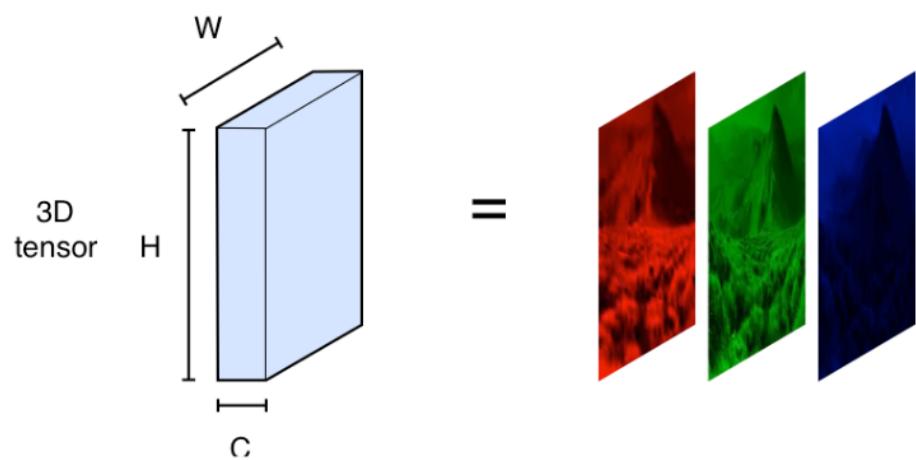
CONVOLUTIONS CAN ALSO BE COMPUTED ACROSS CHANNELS (OR COLORS)

A COLOR IMAGE IS A
TENSOR
OF SIZE height x width x
channels



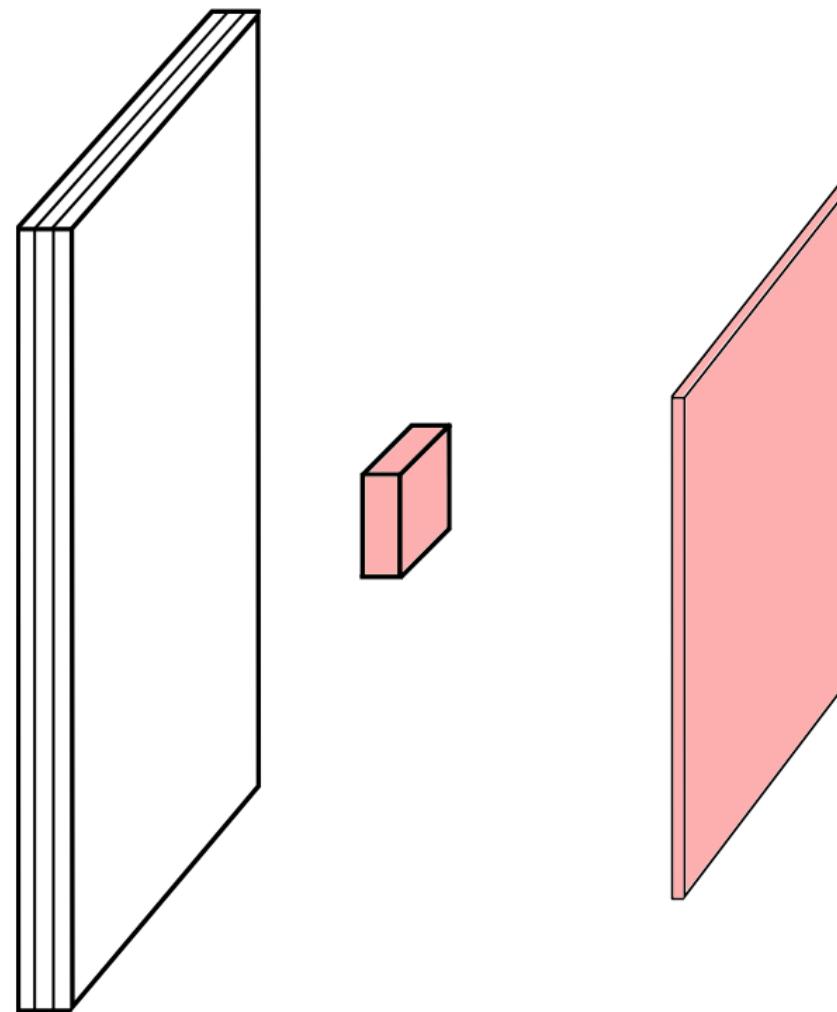
CONVOLUTIONS CAN ALSO BE COMPUTED ACROSS CHANNELS (OR COLORS)

A COLOR IMAGE IS A
TENSOR
OF SIZE height x width x
channels



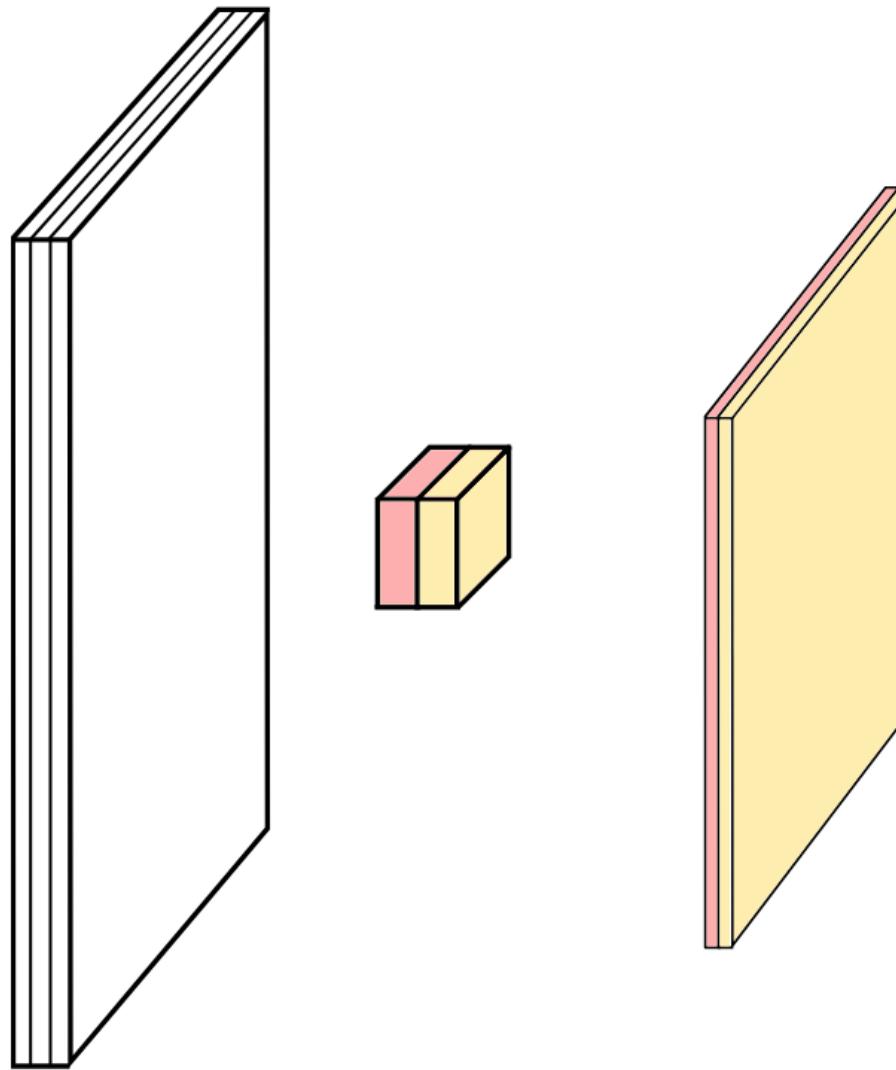
THEN THE KERNEL
HAS ALSO 3
CHANNELS

MULTIPLE CONVOLUTIONS WITH DIFFERENT KERNELS
CAN BE PERFORMED



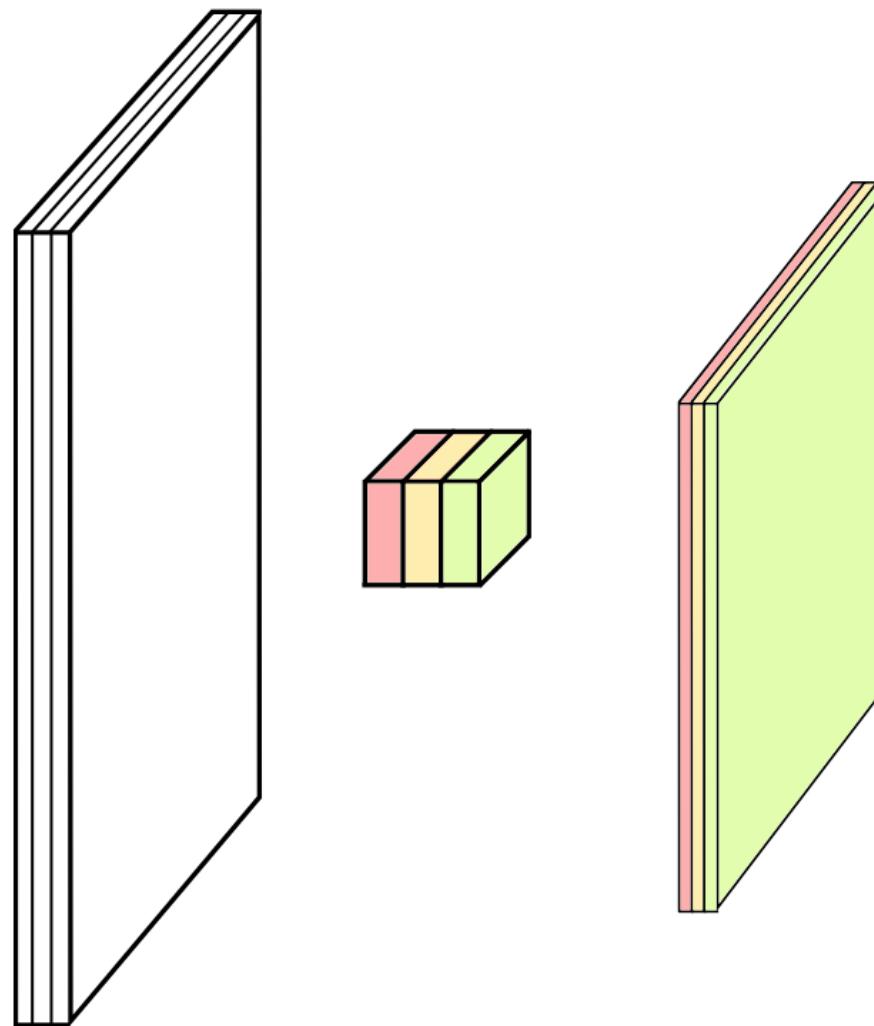
credit

MULTIPLE CONVOLUTIONS WITH DIFFERENT KERNELS CAN BE PERFORMED



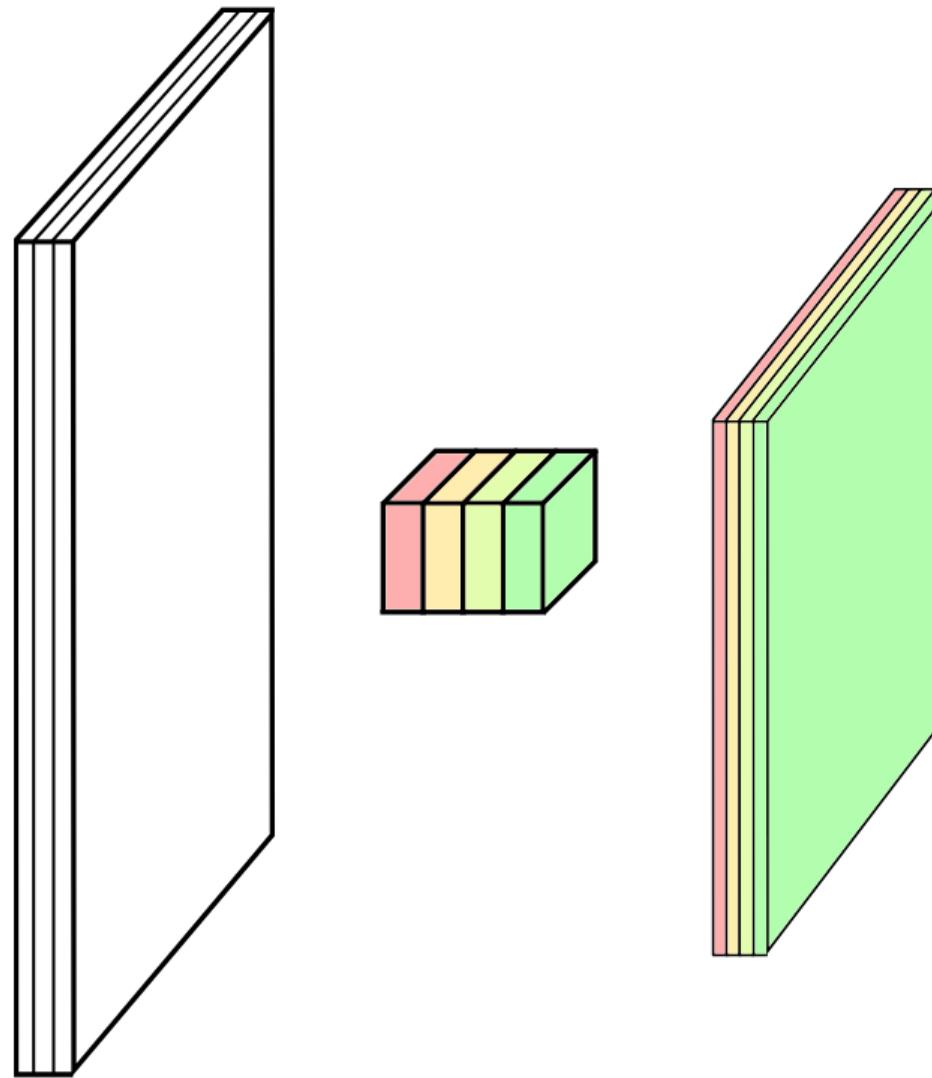
credit

MULTIPLE CONVOLUTIONS WITH DIFFERENT KERNELS CAN BE PERFORMED



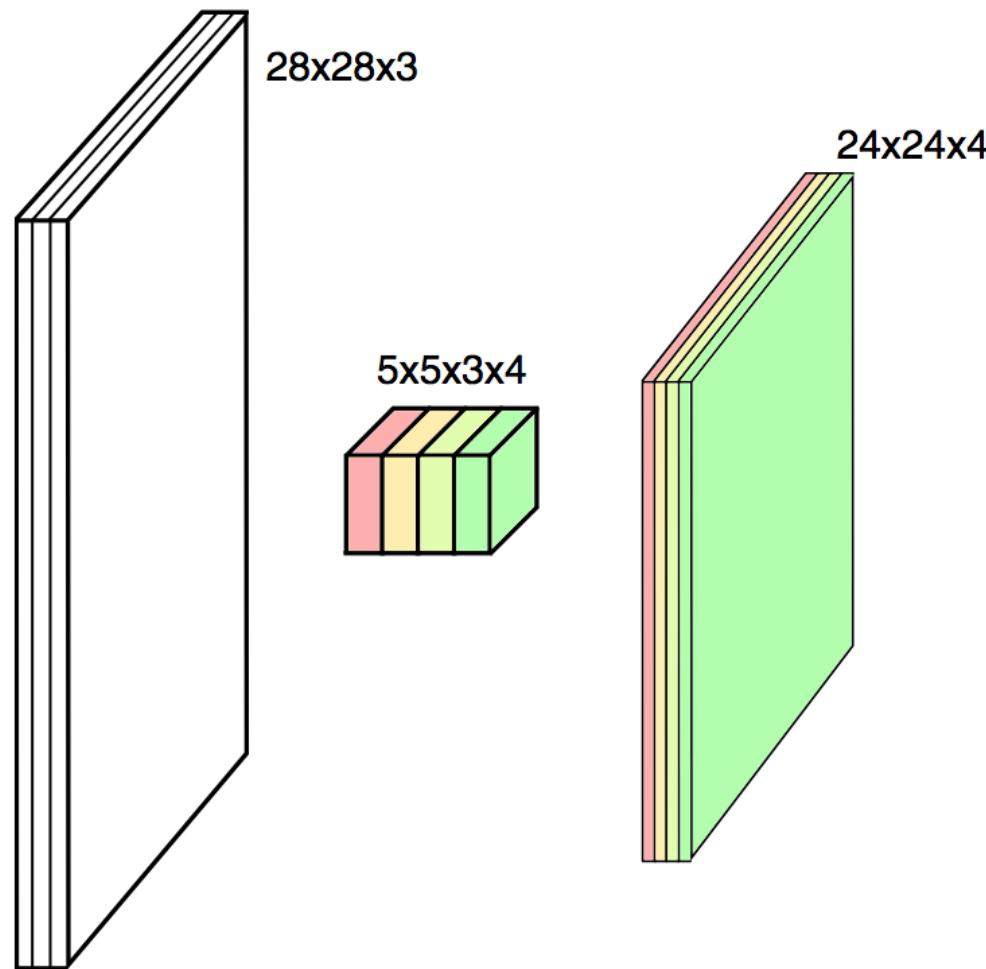
credit

MULTIPLE CONVOLUTIONS WITH DIFFERENT KERNELS CAN BE PERFORMED



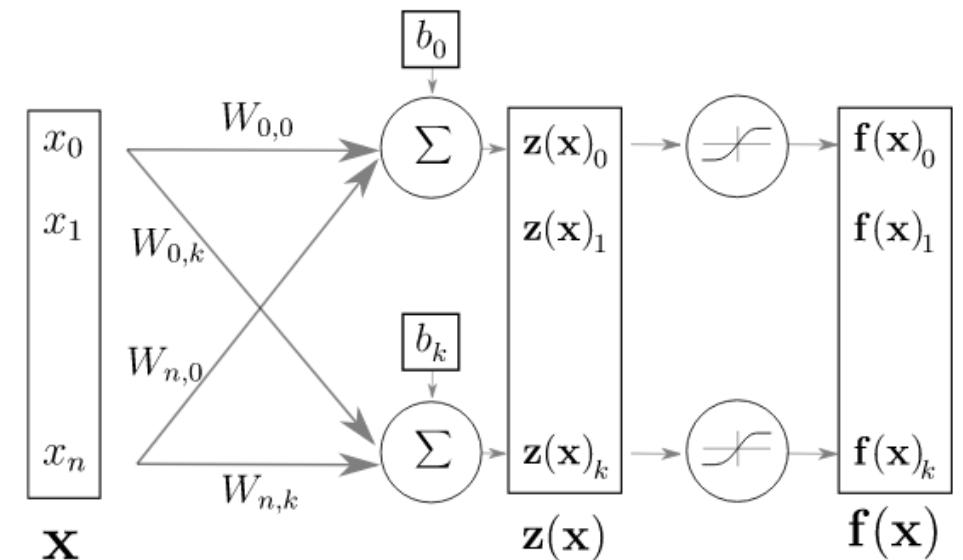
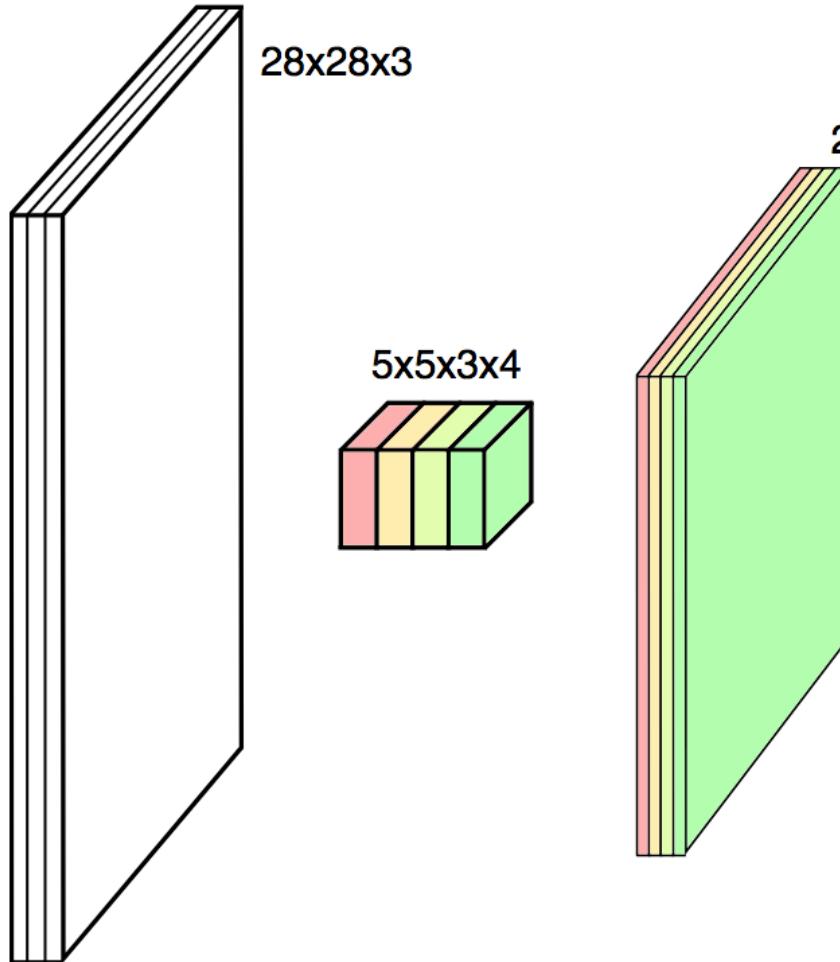
credit

MULTIPLE CONVOLUTIONS WITH DIFFERENT KERNELS CAN BE PERFORMED



credit

MULTIPLE CONVOLUTIONS WITH DIFFERENT KERNELS CAN BE PERFORMED

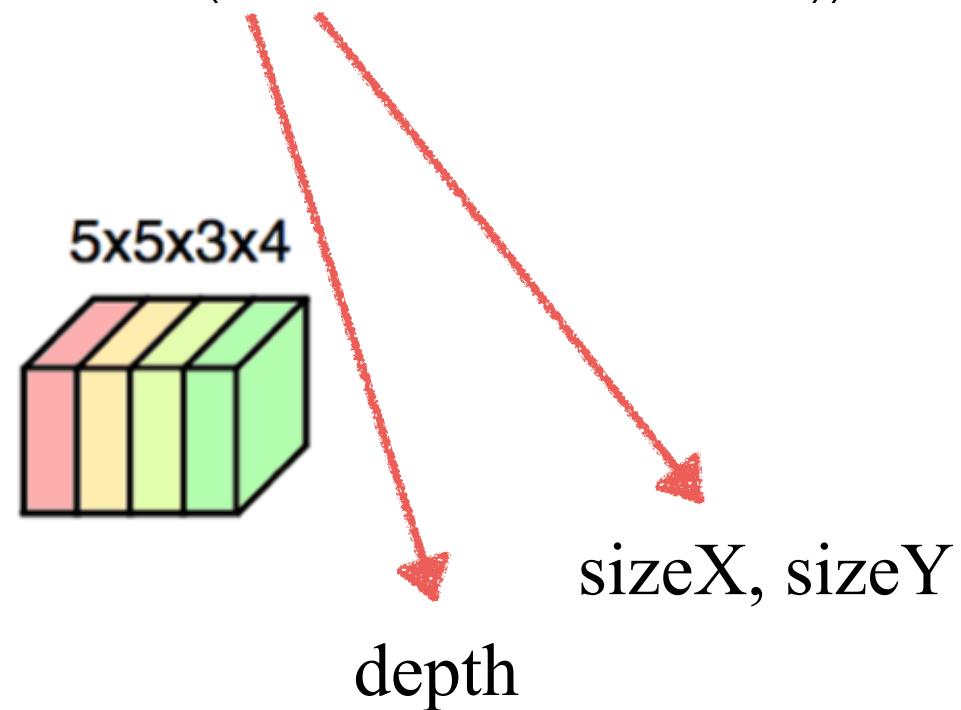


credit

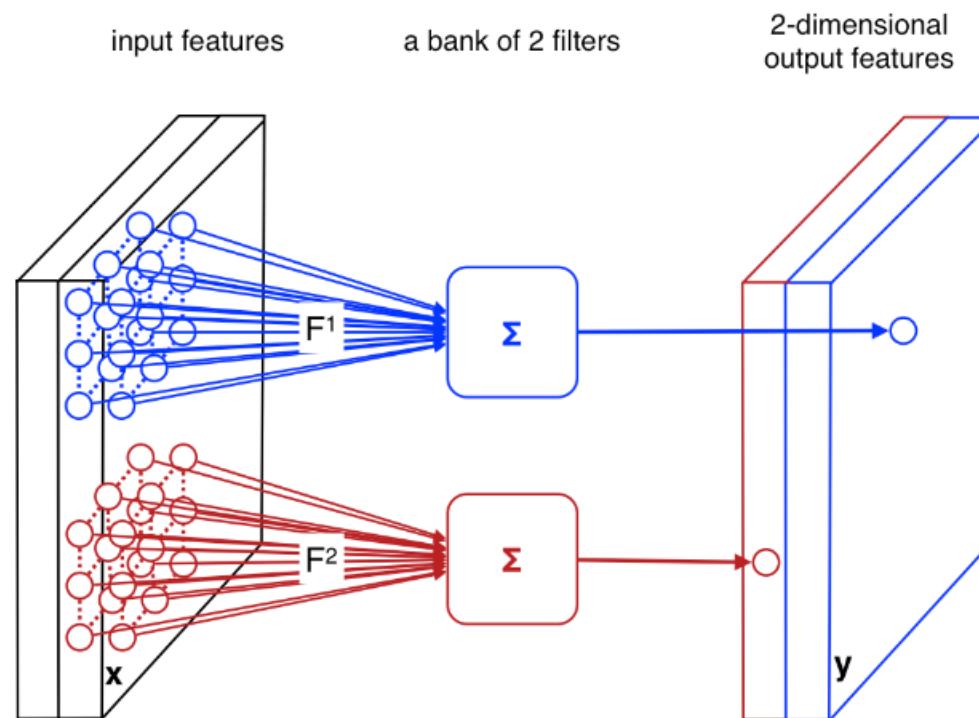
IN KERAS...

```
model = Sequential()
```

```
model.add(Convolution2D(4,5,5, activation="relu"))
```



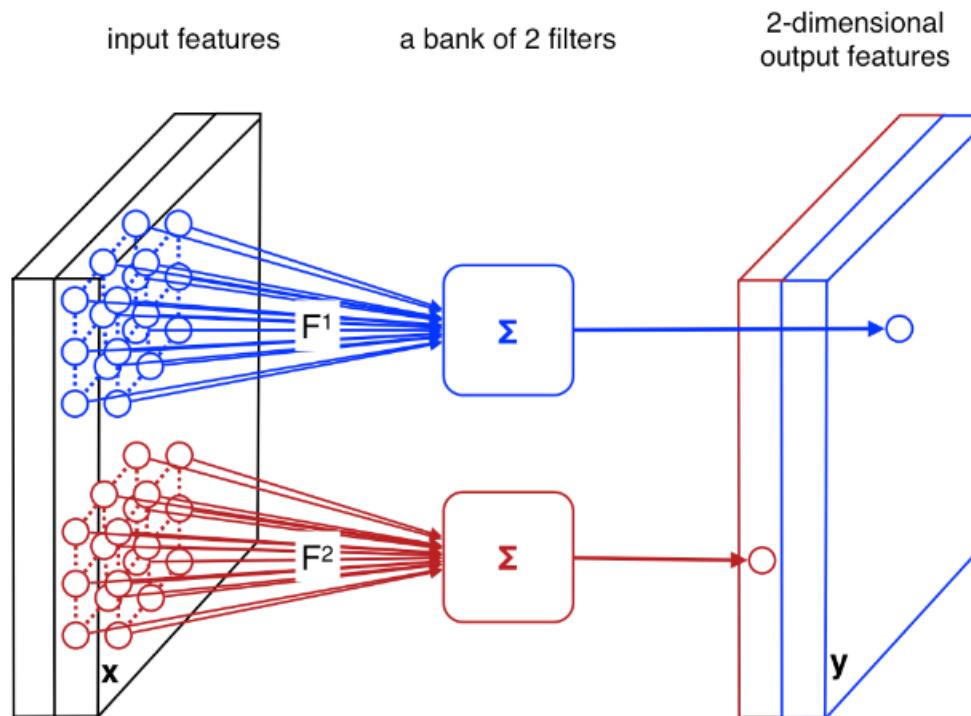
SINCE CONVOLUTIONS OUTPUT ONE SCALAR, THEY CAN BE SEEN AS AN INDIVIDUAL NEURON WITH A RECEPTIVE FIELD LIMITED TO THE KERNEL DIMENSIONS



Credit

SINCE CONVOLUTIONS OUTPUT ONE SCALAR< THEY CAN BE SEEN AS AN INDIVIDUAL NEURON WITH A RECEPTIVE FIELD LIMITED TO THE KERNEL DIMENSIONS

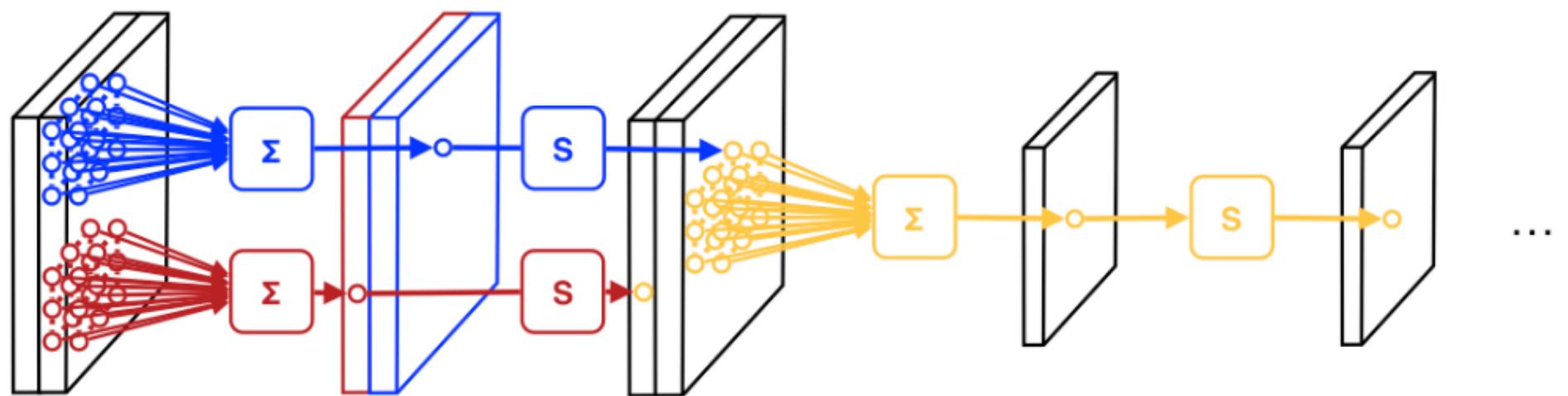
THE SAME NEURON IS FIRED WITH DIFFERENT AREAS FROM THE INPUT



Credit

DOWNSAMPLING

DOWNSAMPLING IS APPLIED TO REDUCE THE OVERALL SIZE OF TENSORS



POOLING

CONVOLUTIONS ARE OFTEN FOLLOWED BY AN OPERATION OF DOWNSAMPLING [POOLING]

VERY SIMPLE OPERATION - ONLY ONE OUT OF EVERY N PIXELS ARE KEPT

OFTEN MATCHED WITH AN INCREASE OF THE FEATURE CHANNELS

TYPES OF POOLING

SUM POOLING

$$y = \sum x_{uv}$$

SQUARE SUM POOLING

$$y = \sqrt{\sum x_{uv}^2}$$

MAX POOLING

$$y = \max(x_{uv})$$

TYPES OF POOLING

SUM POOLING

$$y = \sum x_{uv}$$

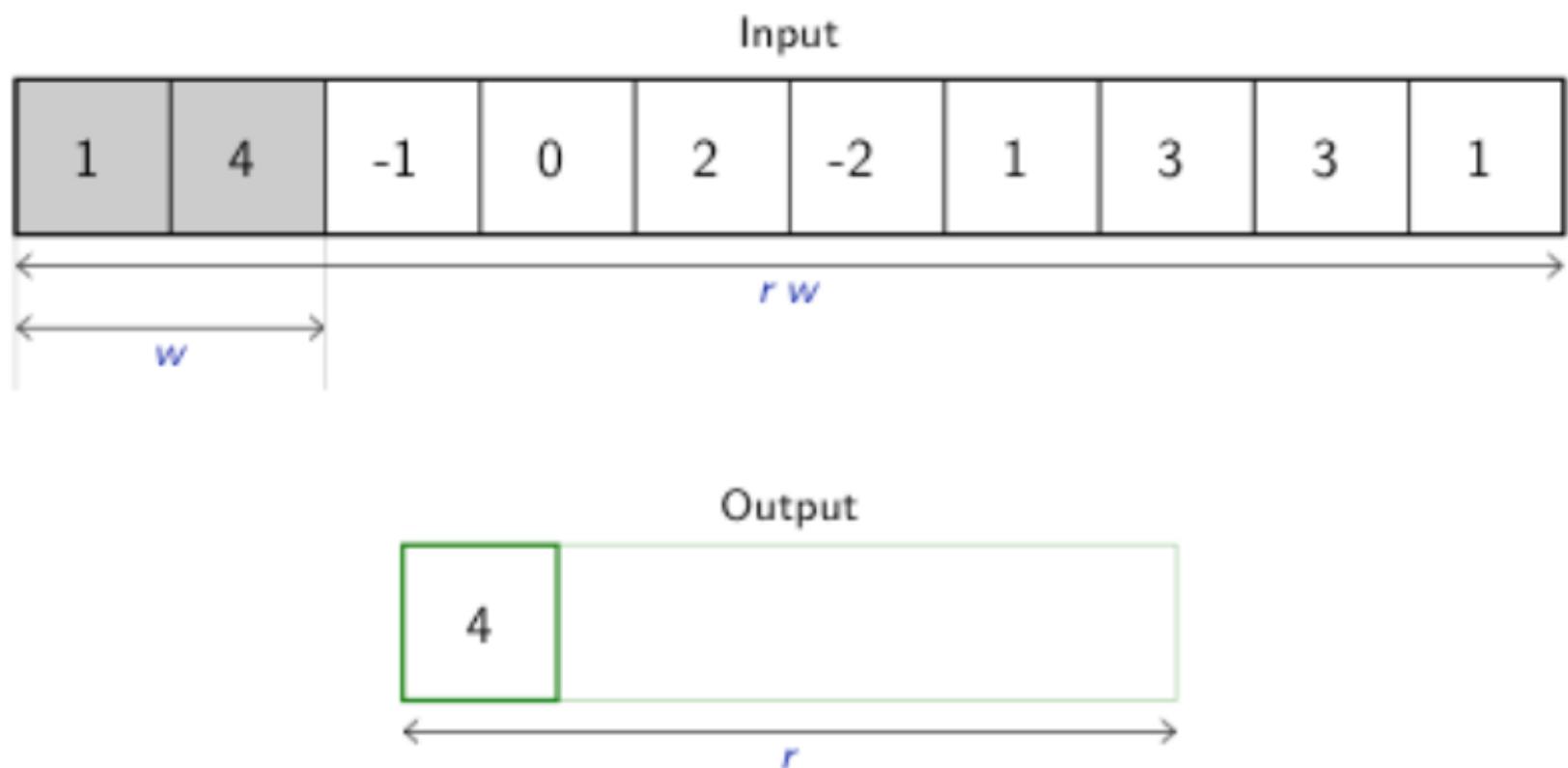
SQUARE SUM POOLING

$$y = \sqrt{\sum x_{uv}^2}$$

MAX POOLING

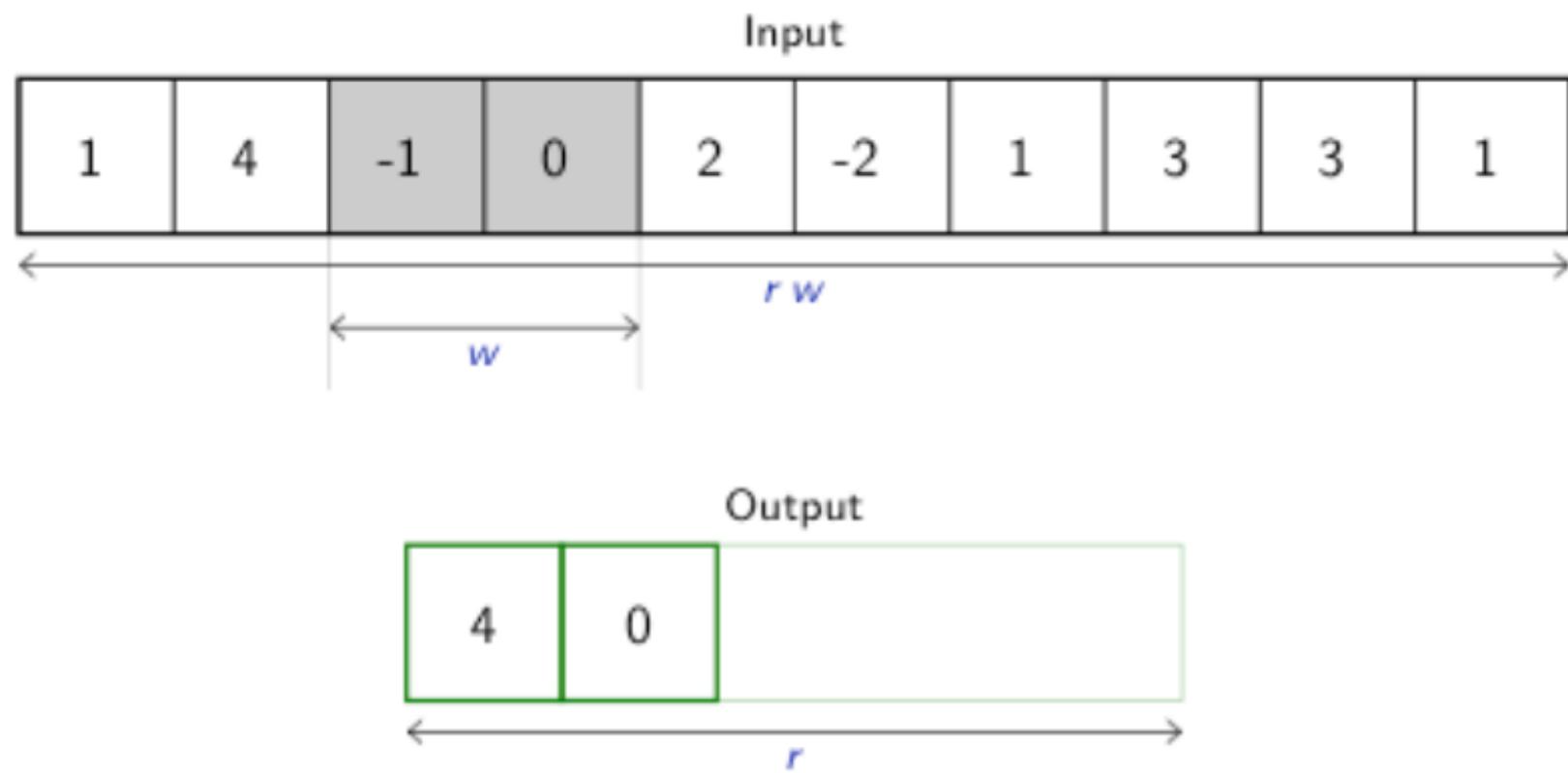
$$y = \max(x_{uv})$$

MAX POOLING 1D



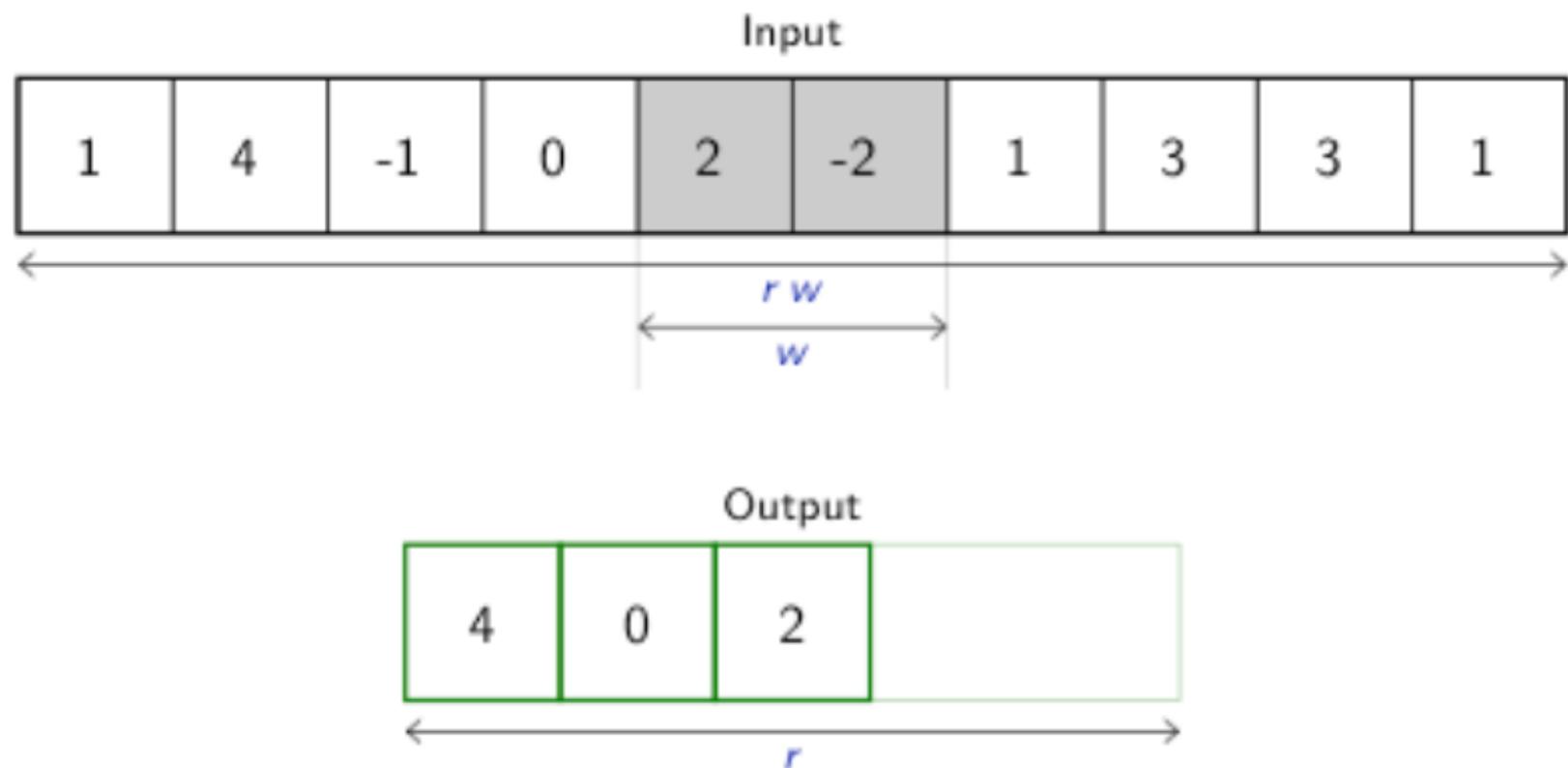
Credit: F. Fleuret

MAX POOLING 1D



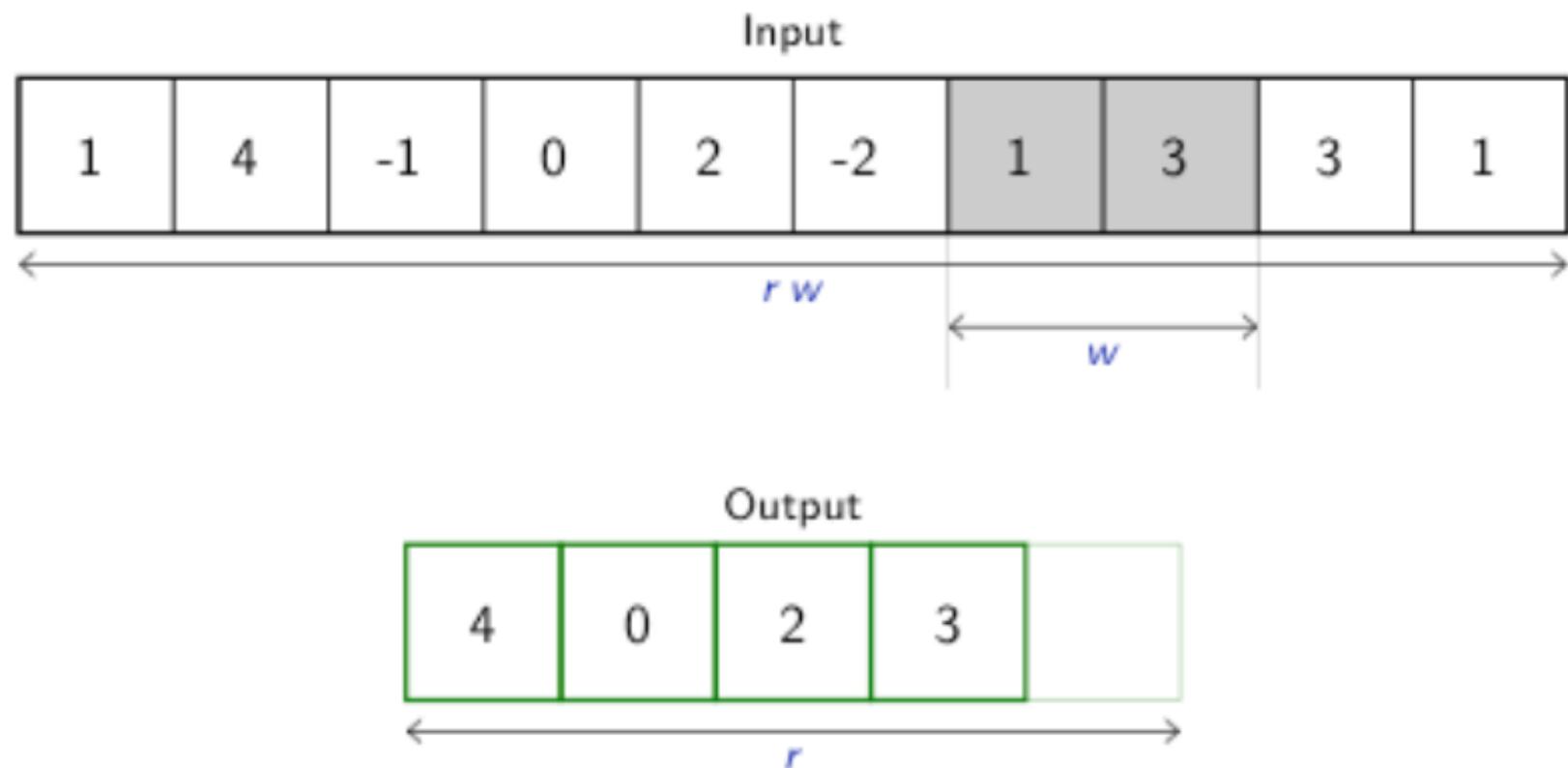
Credit: F. Fleuret

MAX POOLING 1D



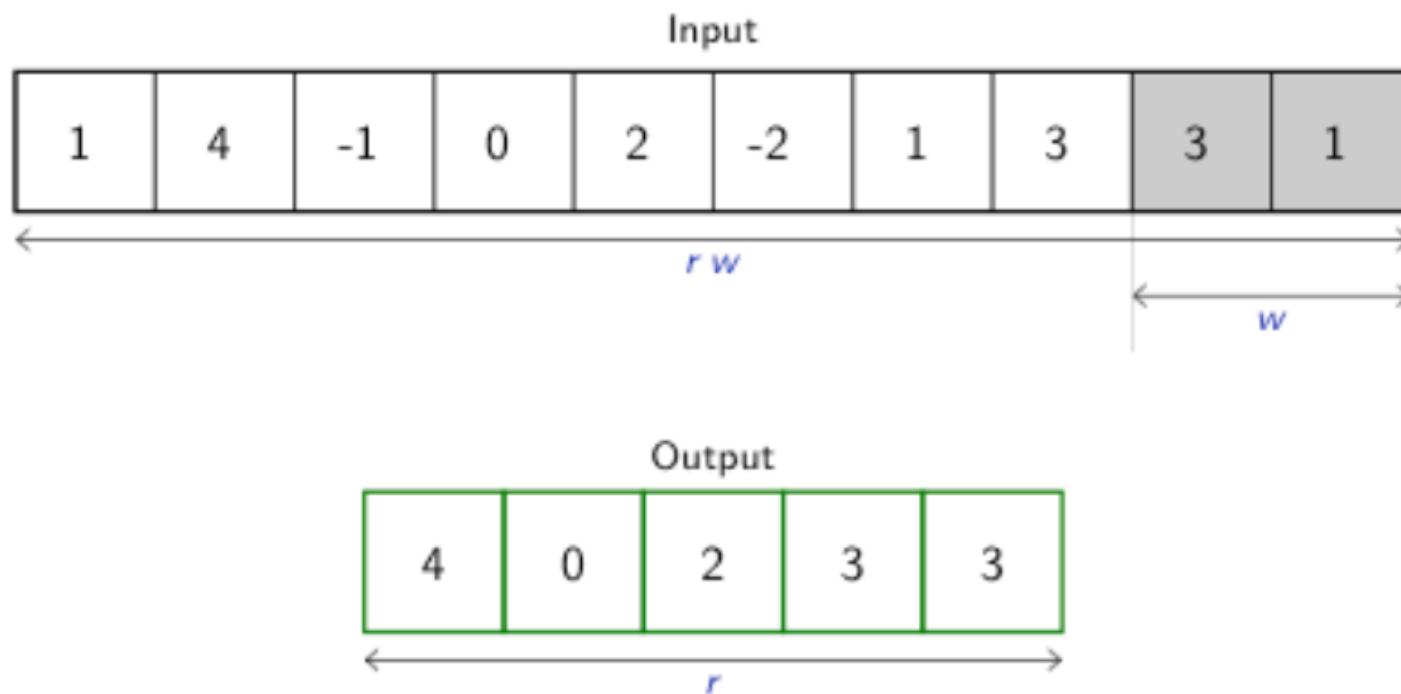
Credit: F. Fleuret

MAX POOLING 1D



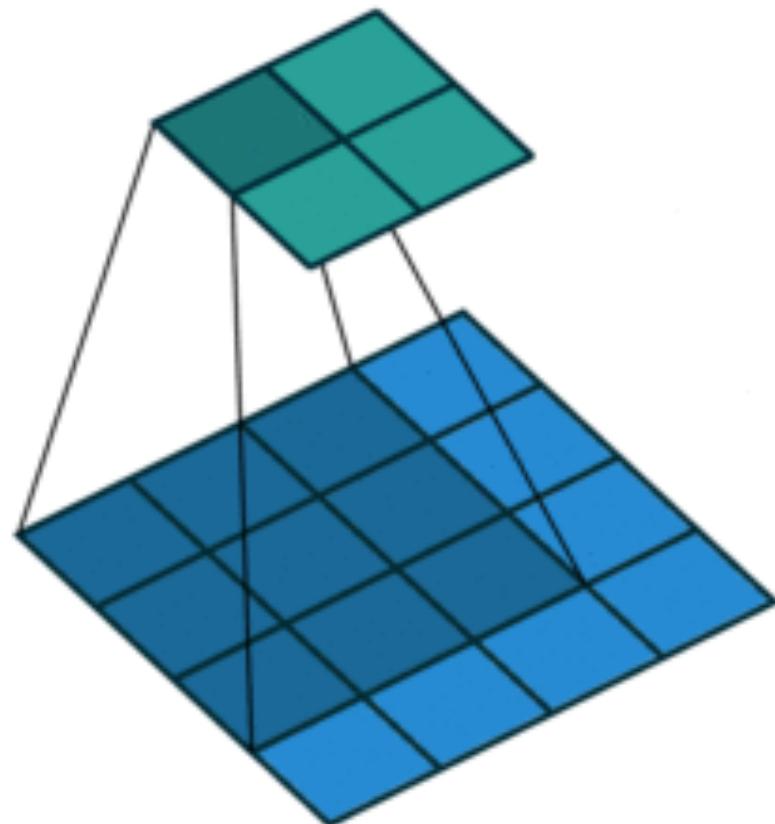
Credit: F. Fleuret

MAX POOLING 1D

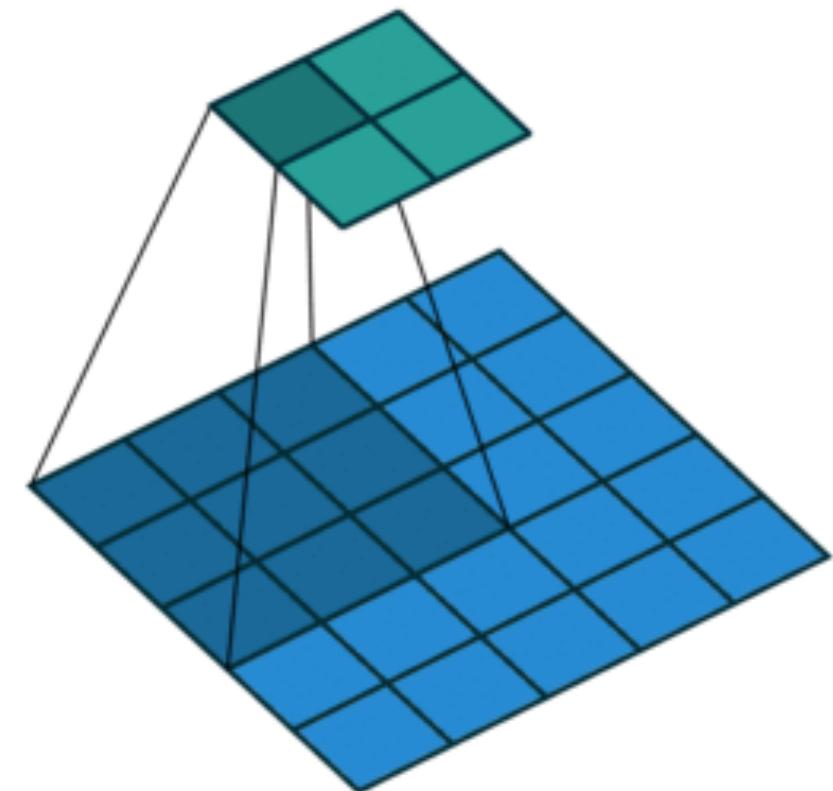


Credit: F. Fleuret

OPTIONS: STRIDES

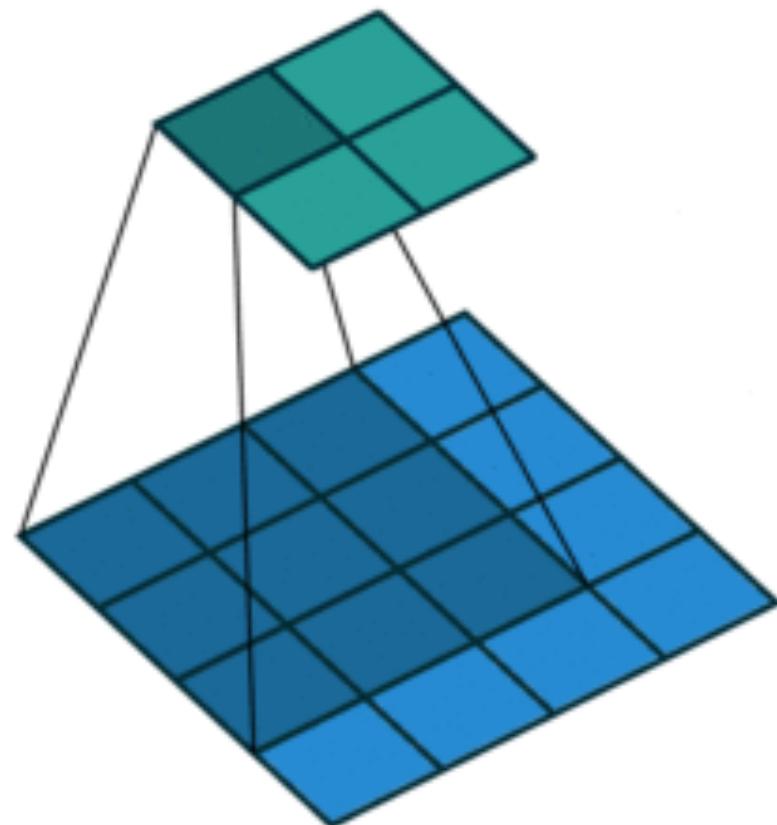


NO STRIDES

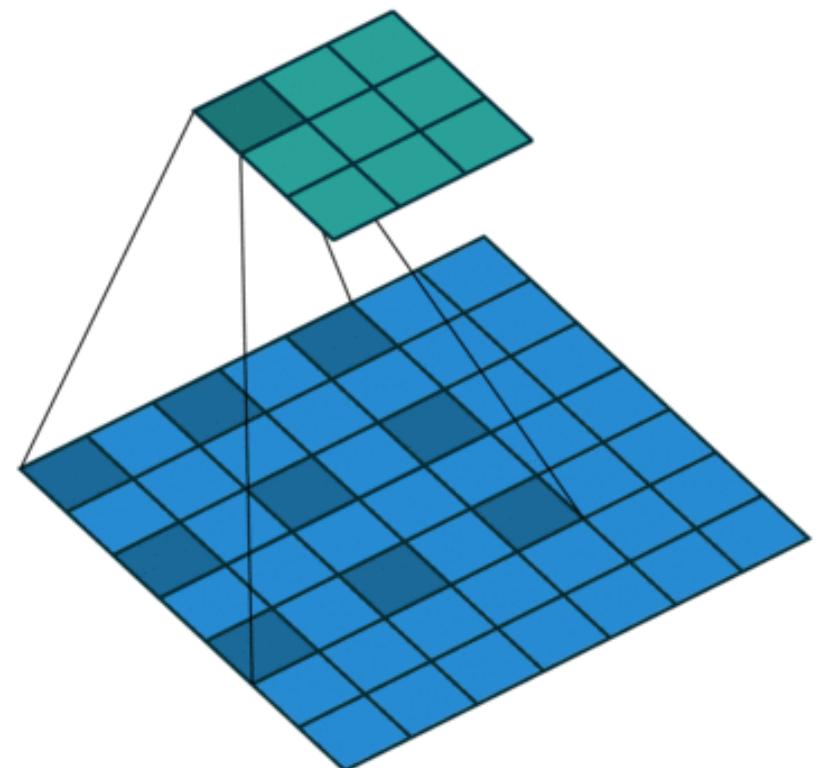


STRIDES

OPTIONS: DILATION

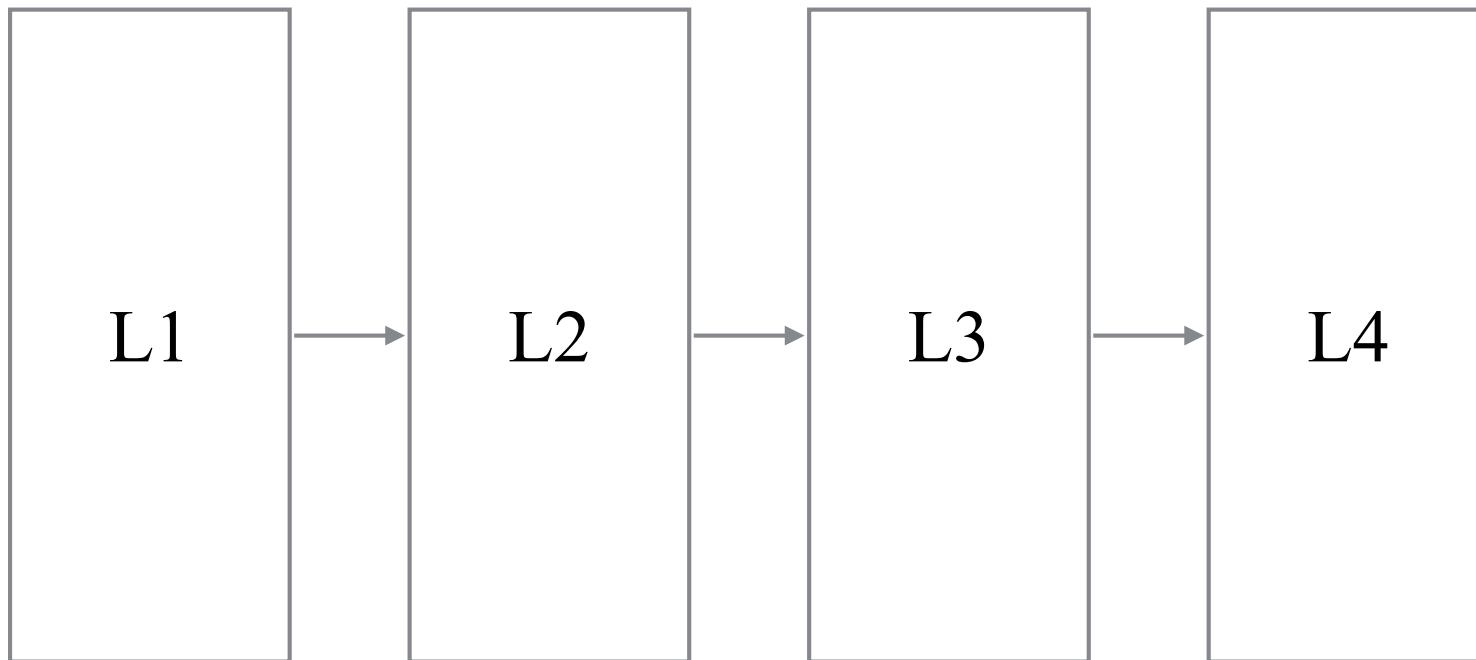


NO STRIDES



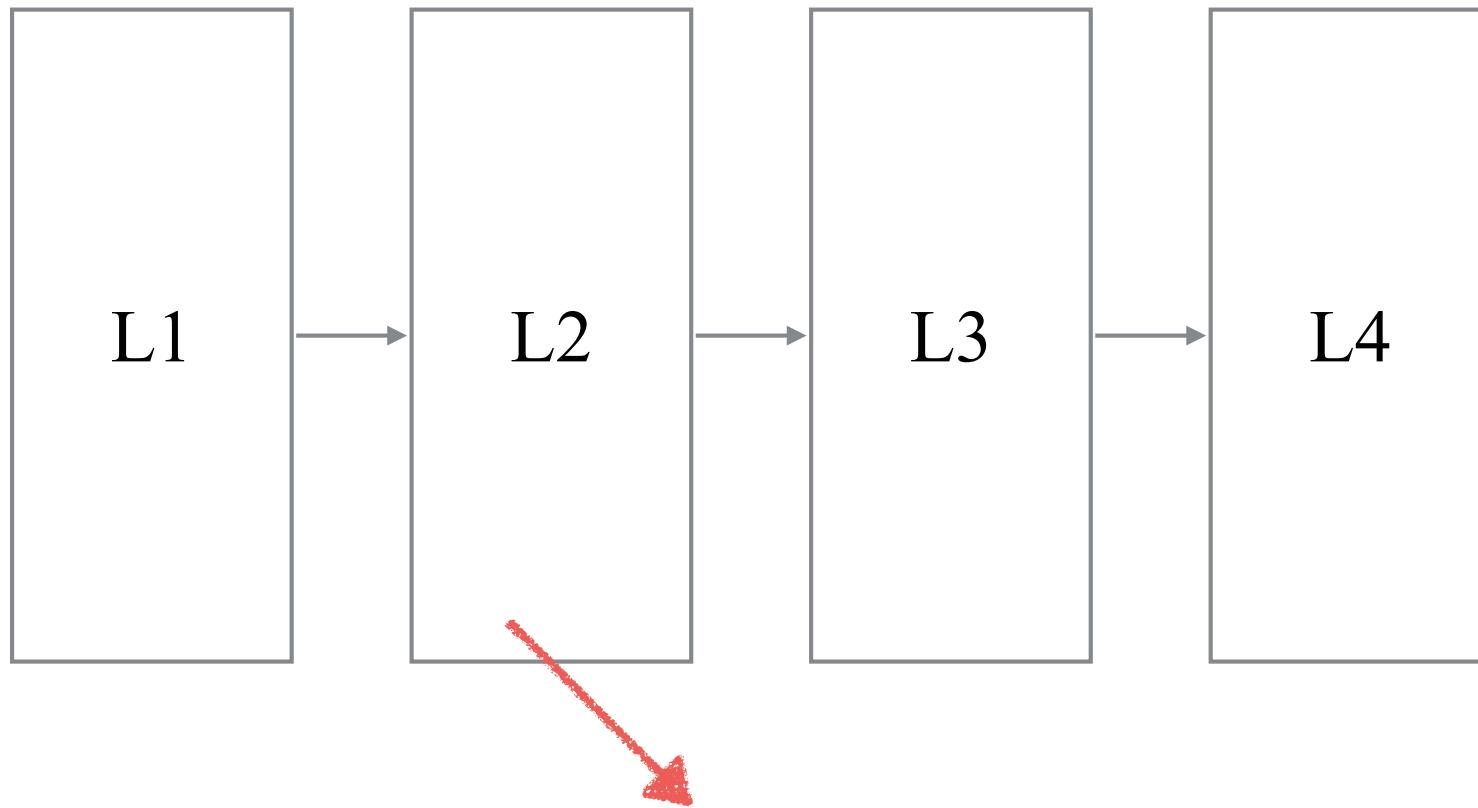
DILATION

CONVNET OR CNN



A CONCATENATION OF MULTIPLE
CONVOLUTIONAL BLOCKS

CONVNET OR CNN



EACH BLOCK TYPICALLY MADE OF:

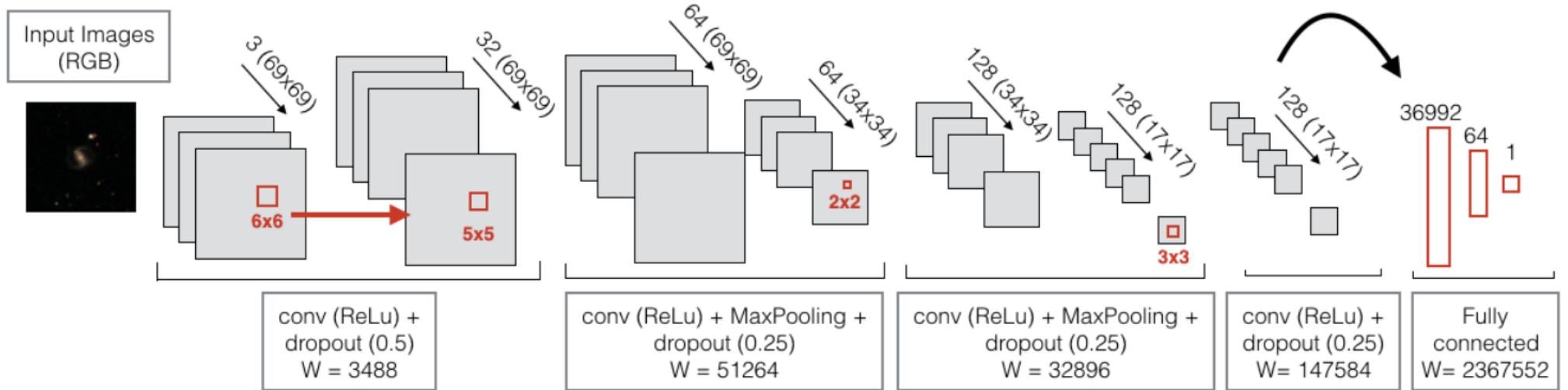
CONV

ACTIVATION

POOLING

(+dropout
for training)

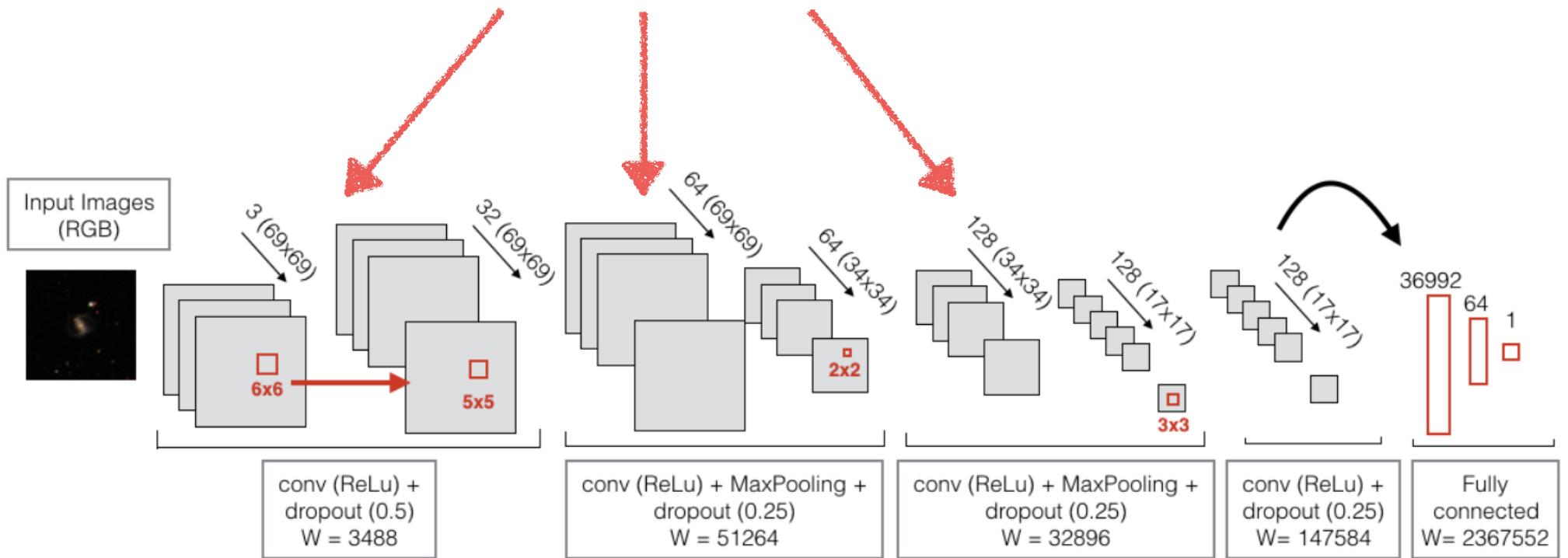
EXAMPLE OF VERY SIMPLE CNN



Dominguez-Sanchez+18

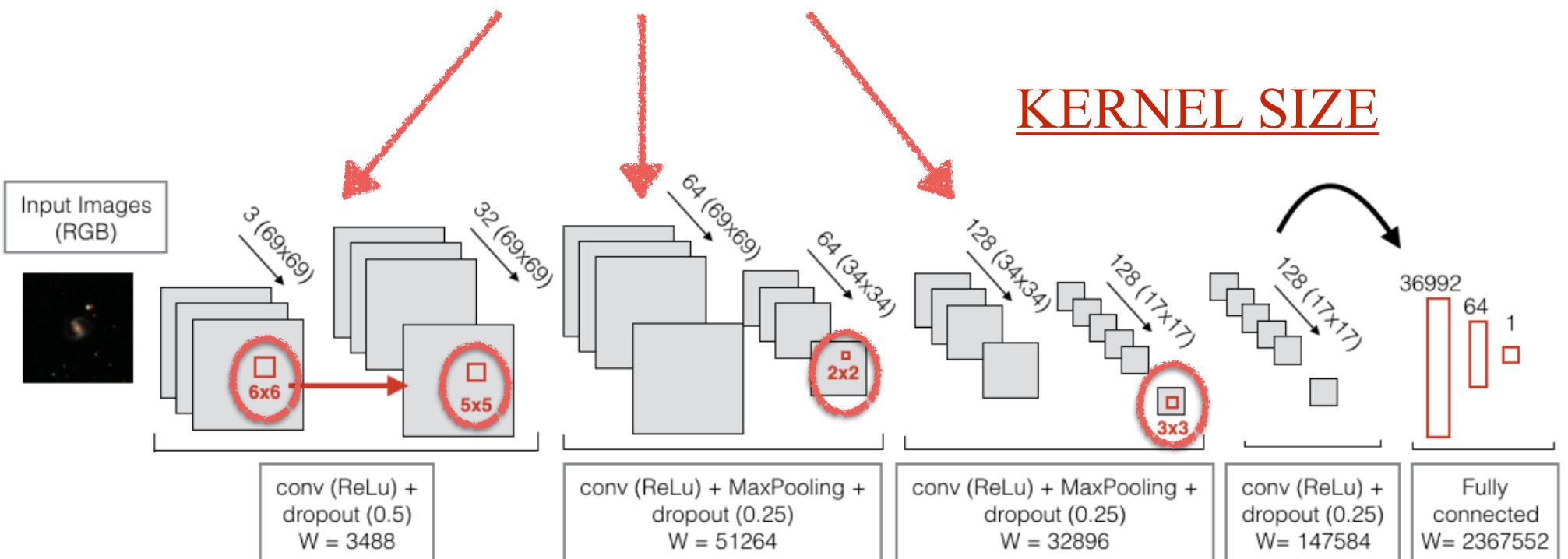
EXAMPLE OF VERY SIMPLE CNN

3 convolutional layers



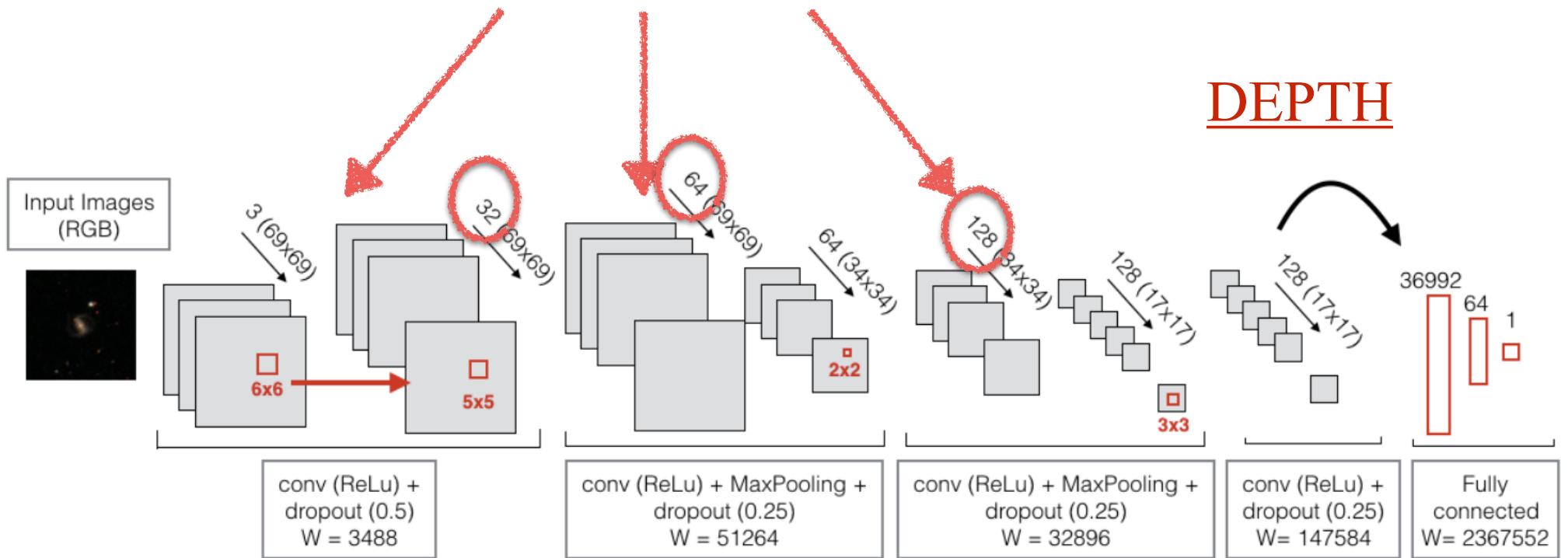
EXAMPLE OF VERY SIMPLE CNN

3 convolutional layers



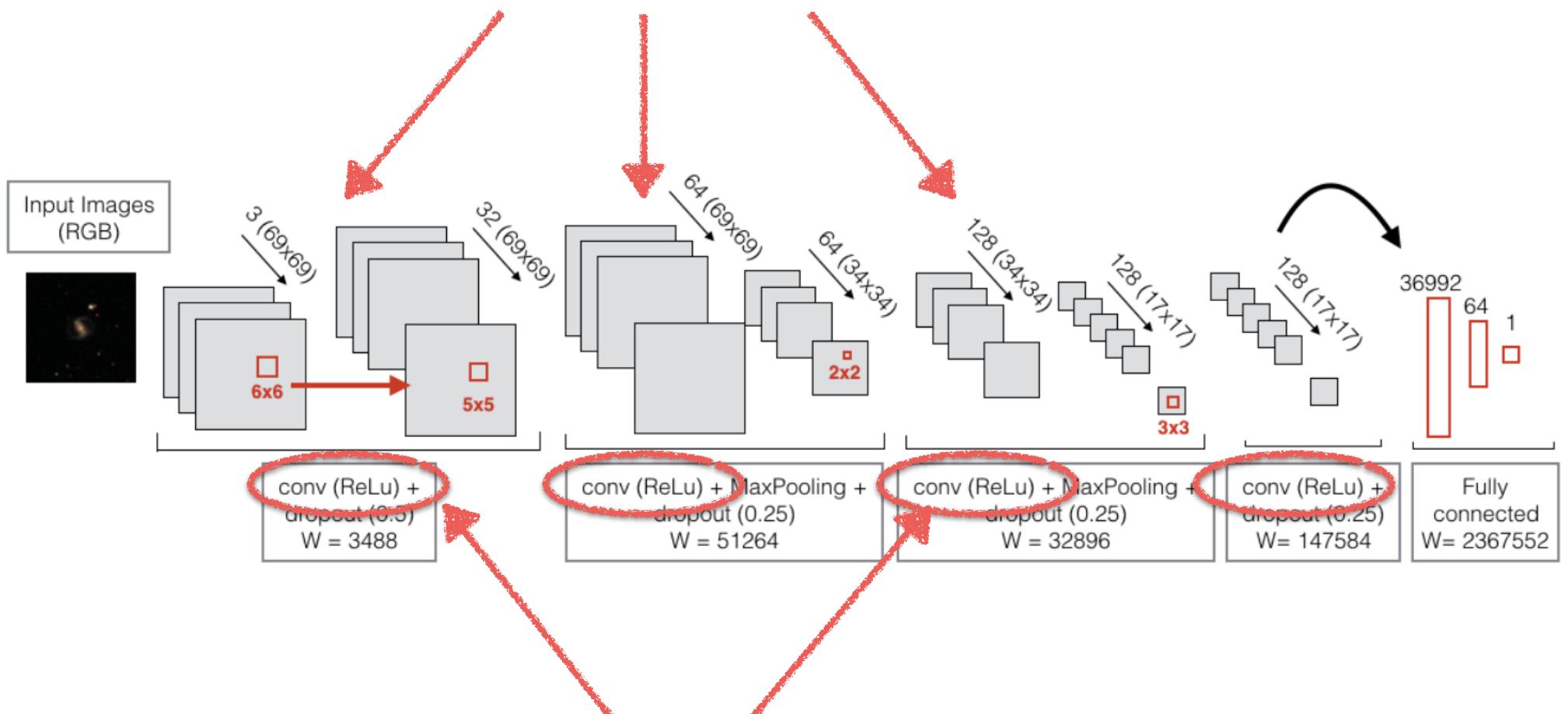
EXAMPLE OF VERY SIMPLE CNN

3 convolutional layers



EXAMPLE OF VERY SIMPLE CNN

3 convolutional layers

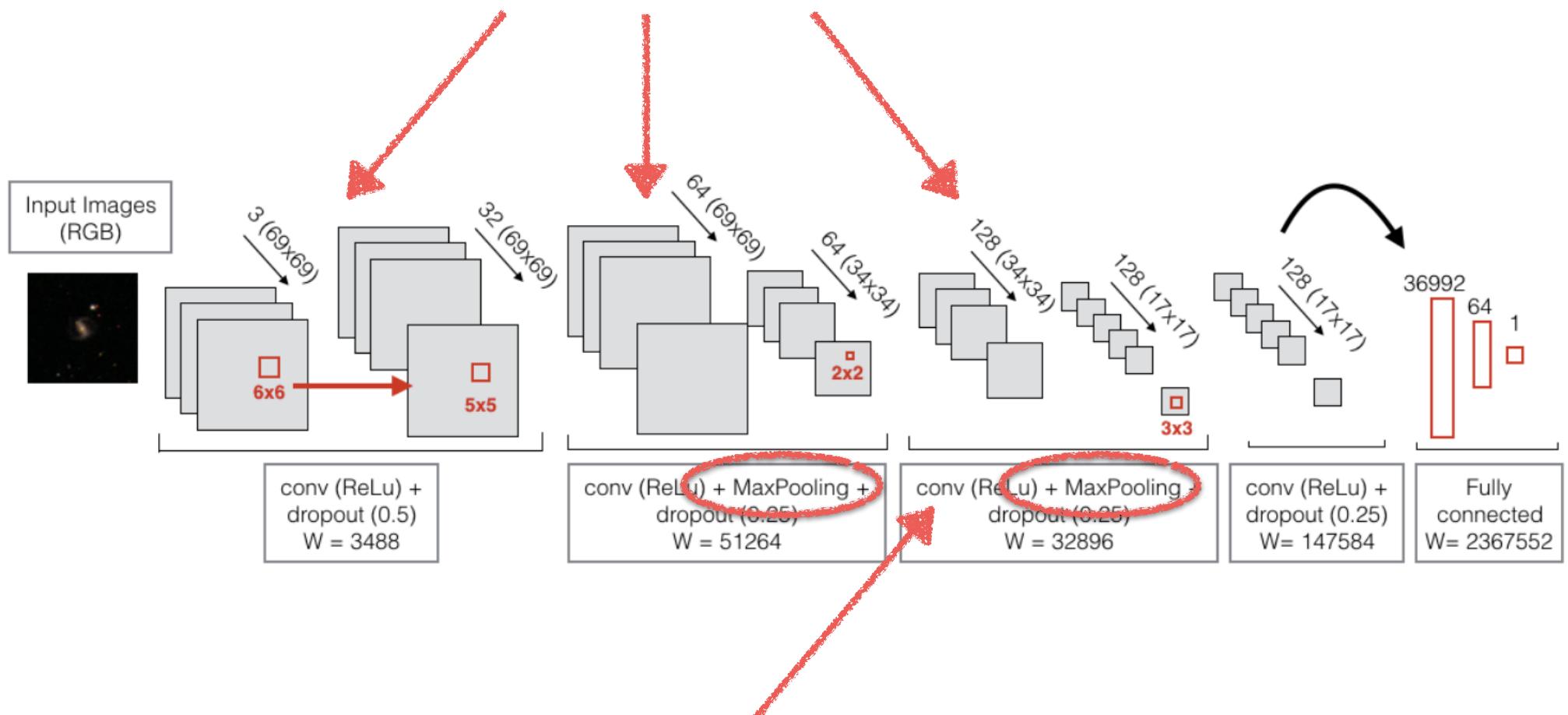


ReLU activation

Dominguez-Sanchez+18

EXAMPLE OF VERY SIMPLE CNN

3 convolutional layers



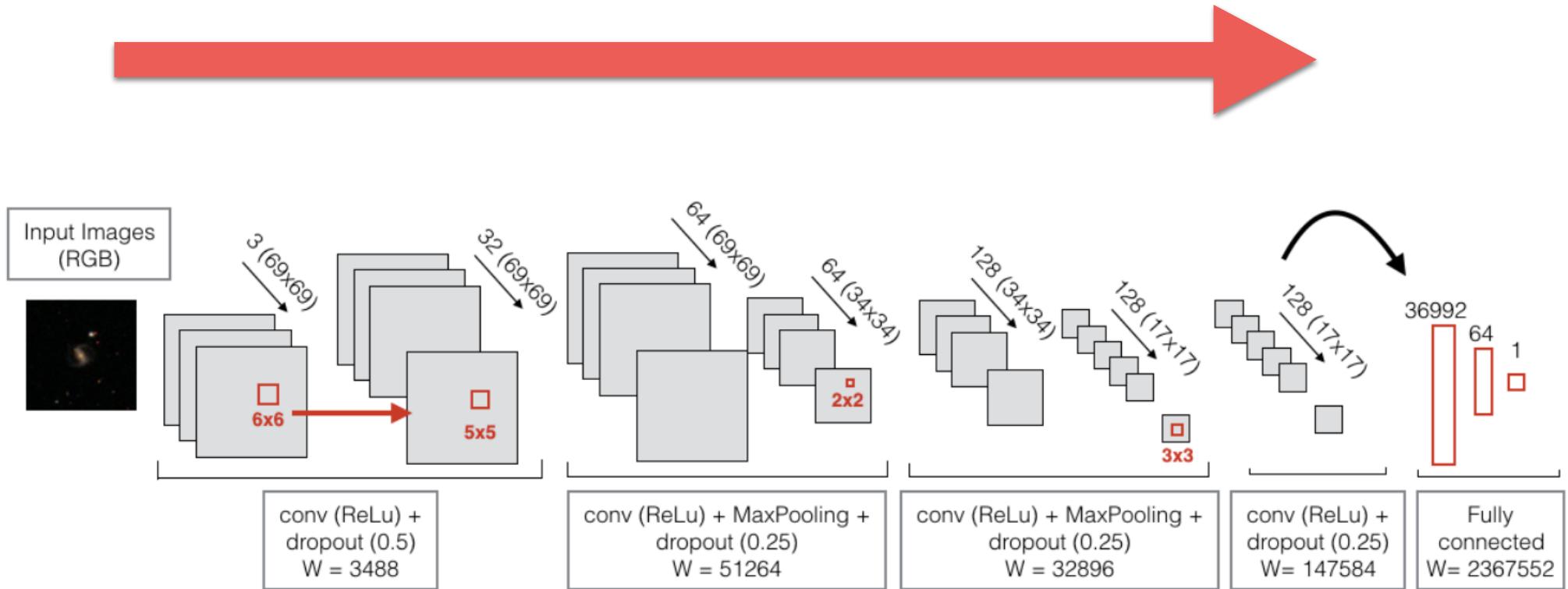
Pooling

Dominguez-Sanchez+18

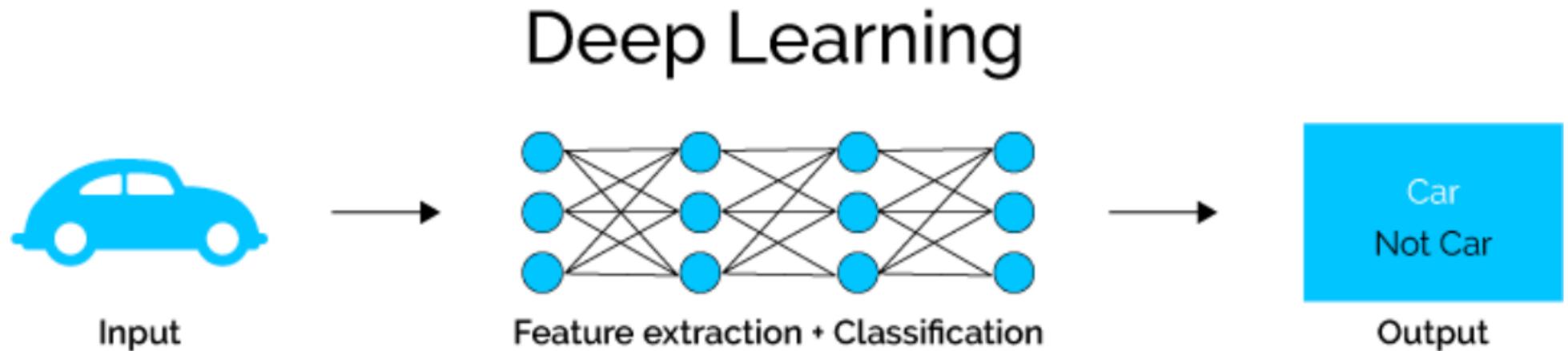
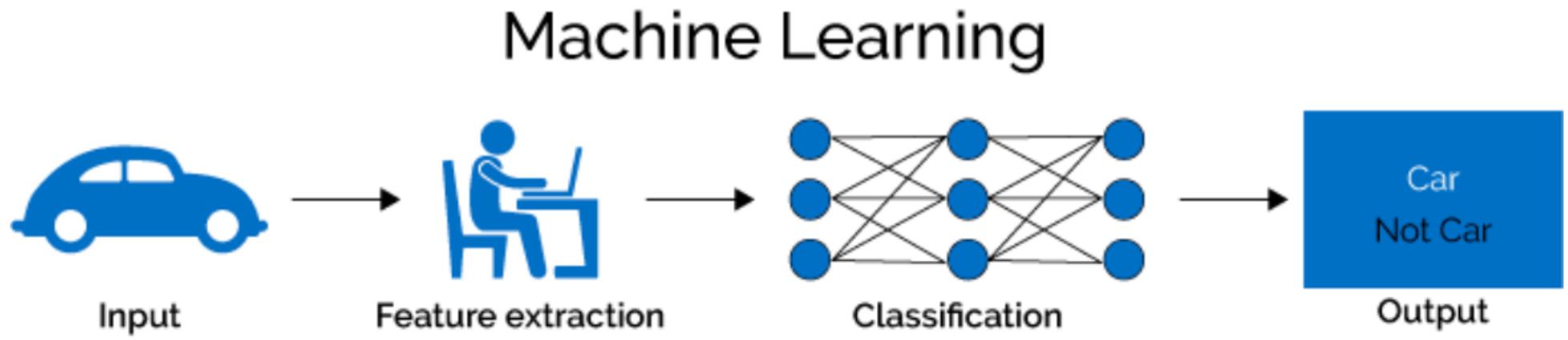
EXAMPLE OF VERY SIMPLE CNN

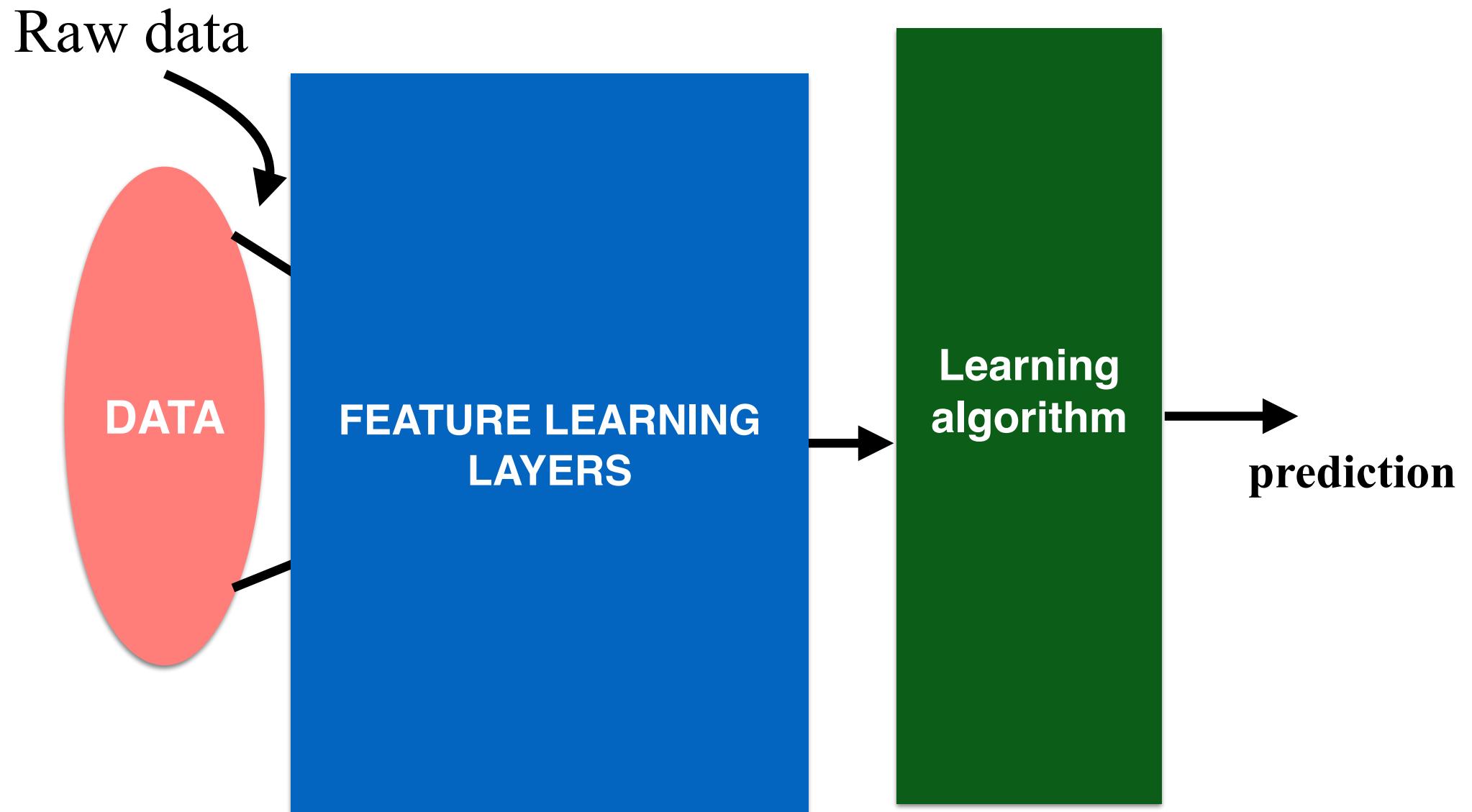
OVERALL:

- decrease of tensor size
- increase of depth

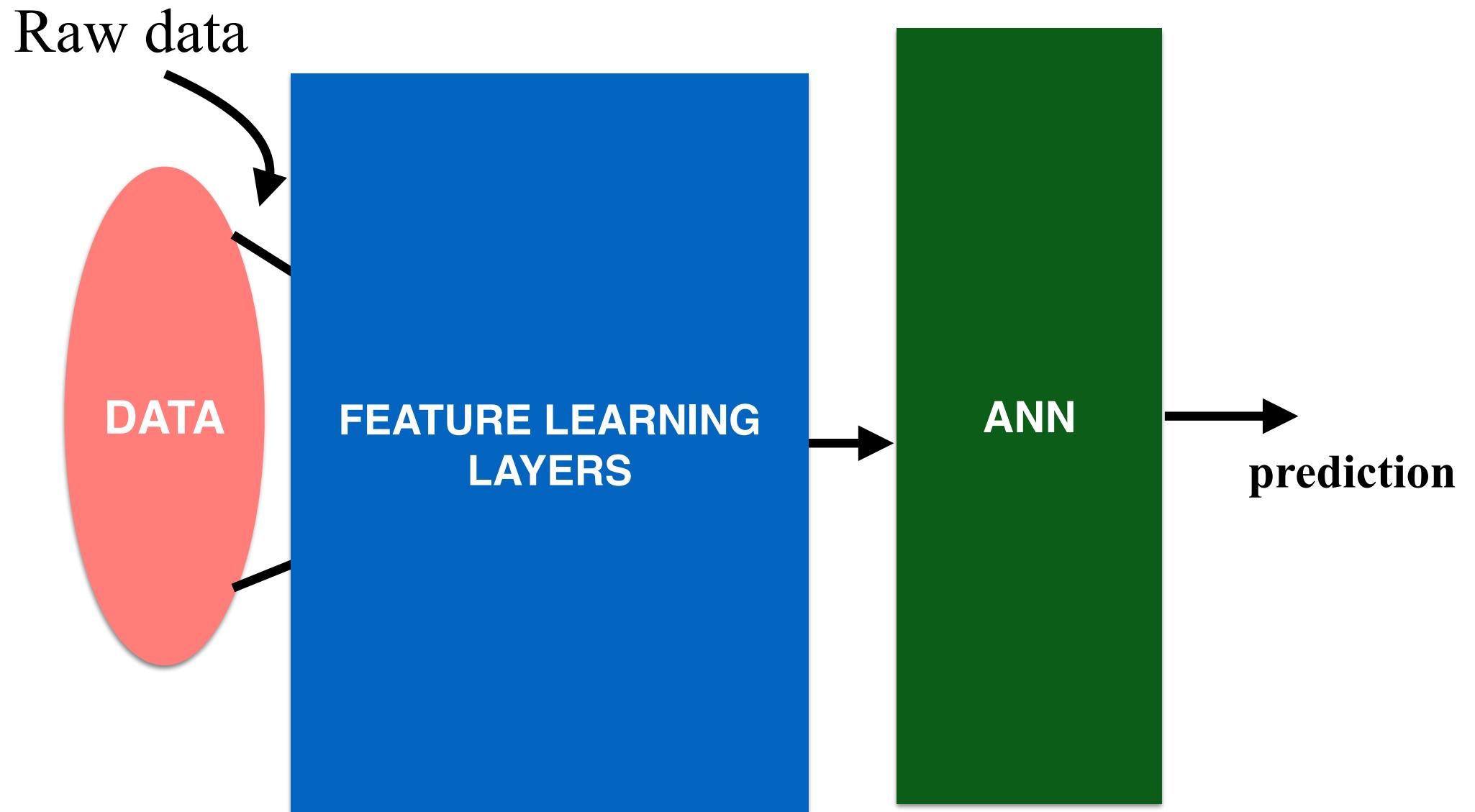


THIS IS A CHANGE OF PARADIGM!

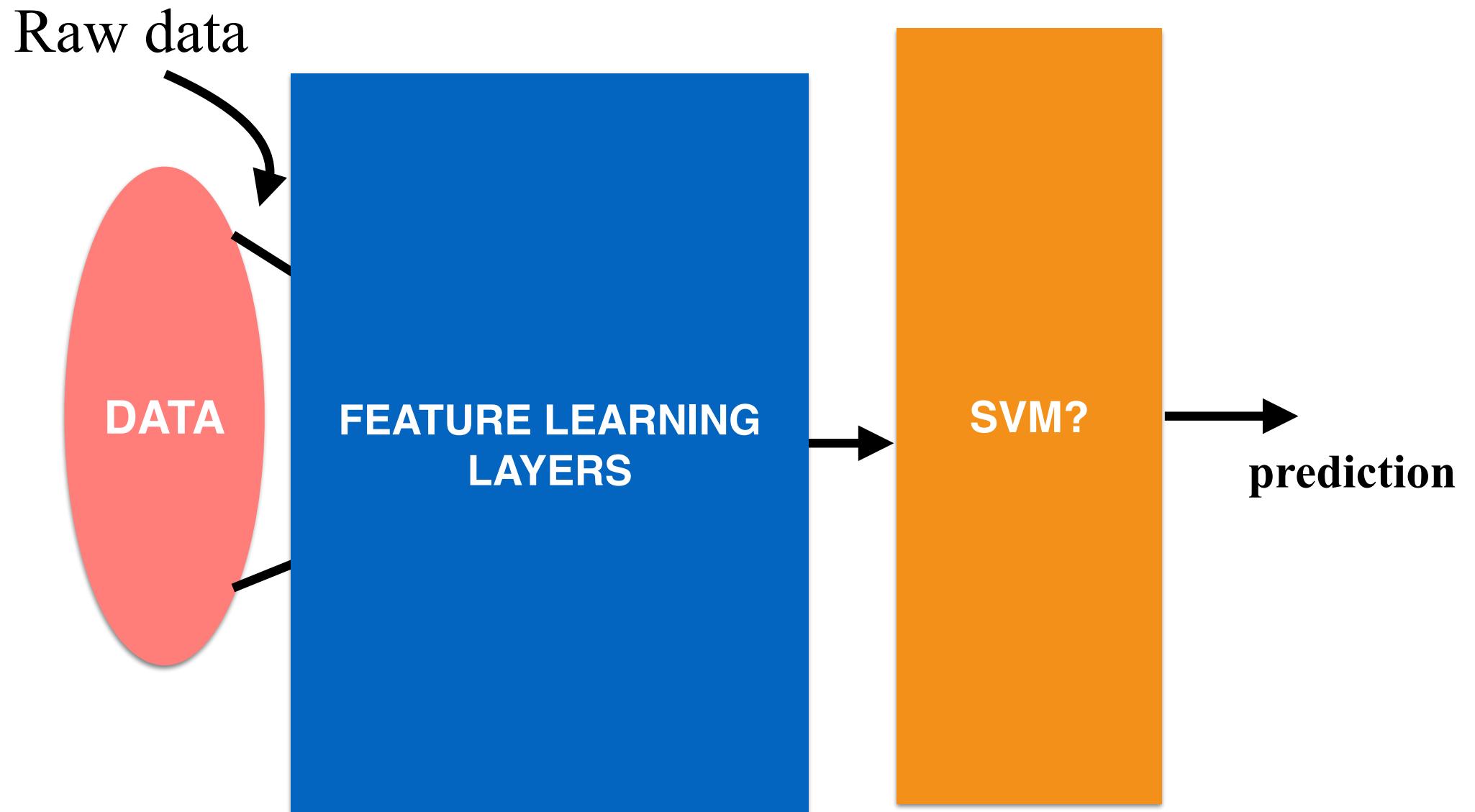




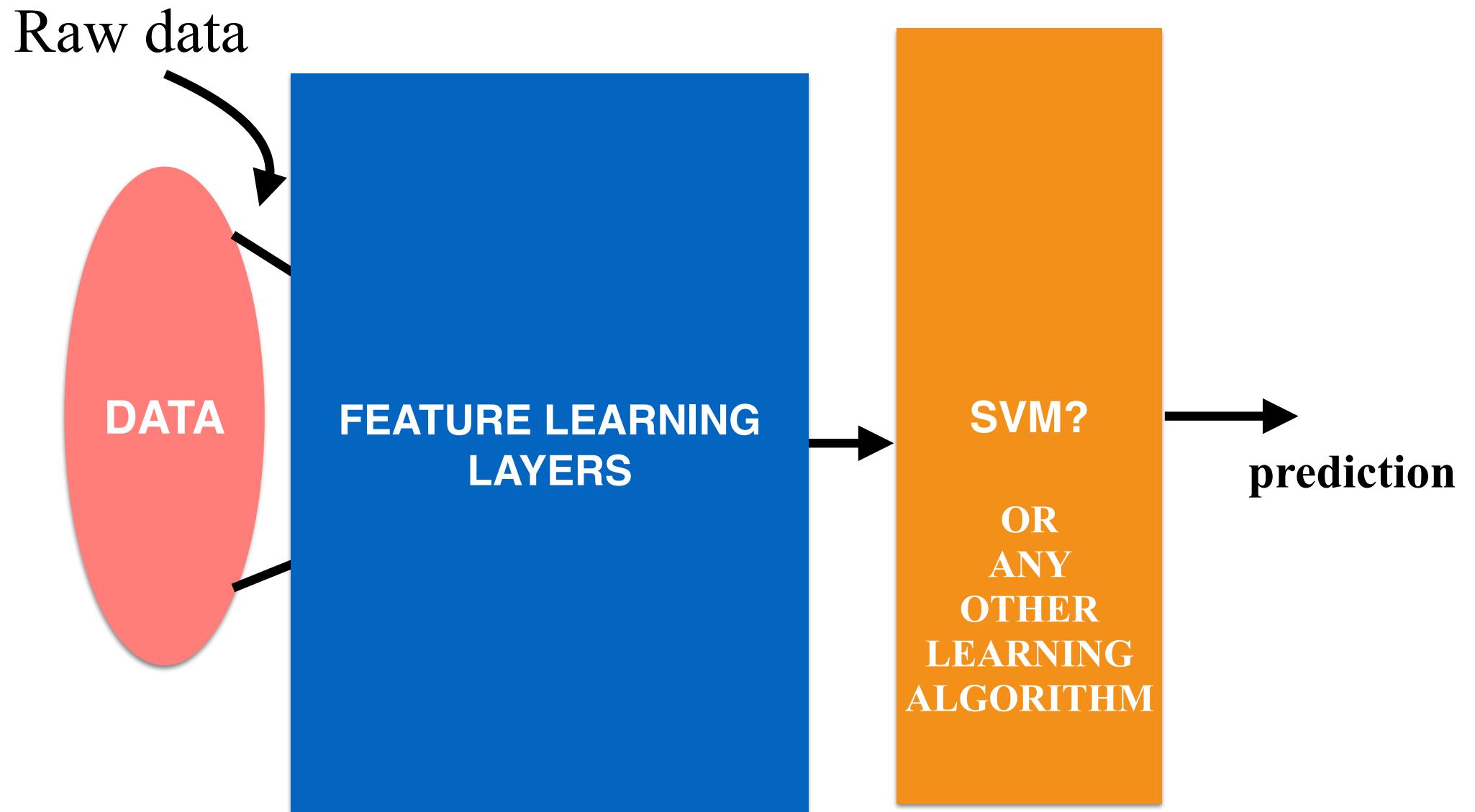
THE LEARNING ALGORITHM CAN BE CHANGED



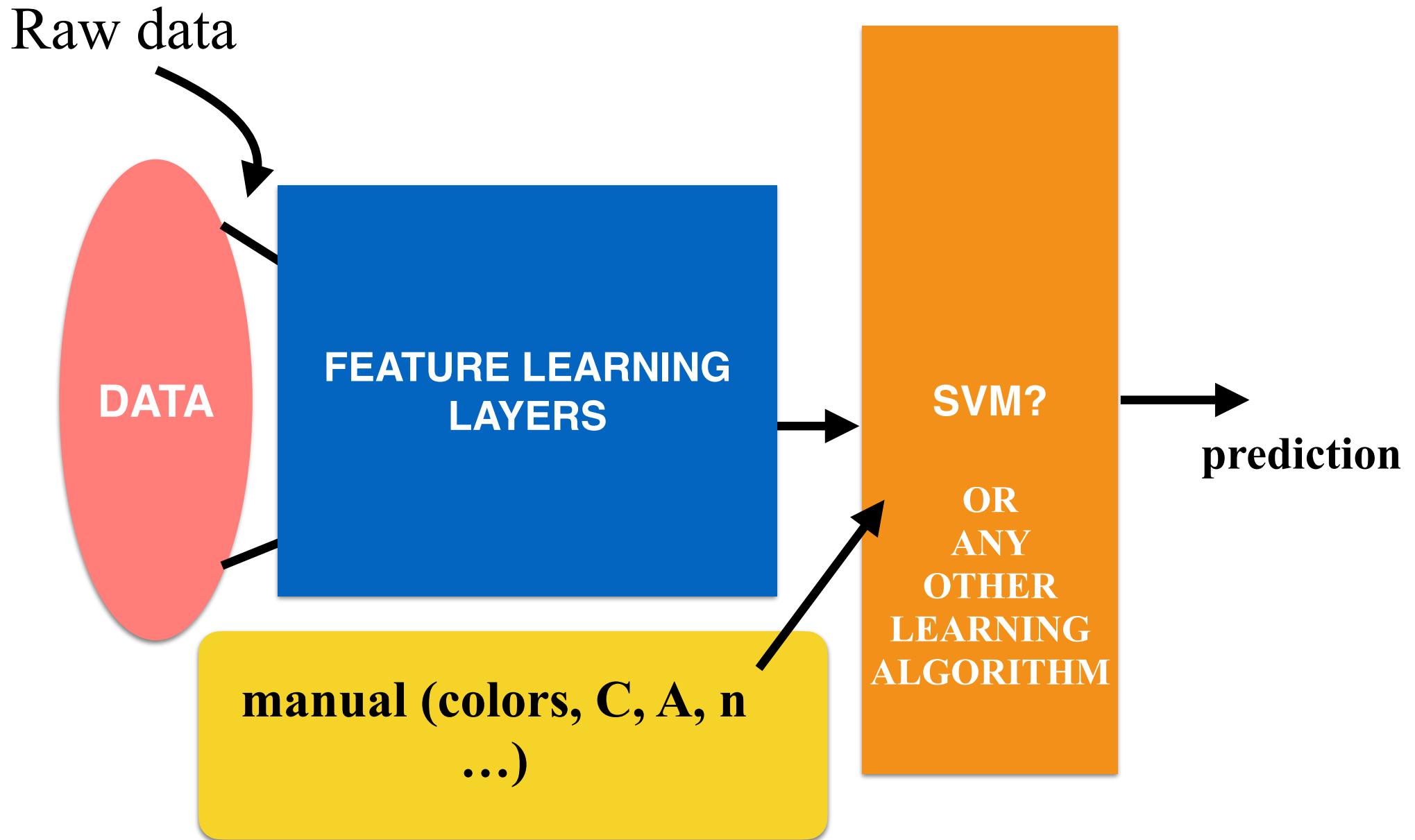
THE LEARNING ALGORITHM CAN BE CHANGED



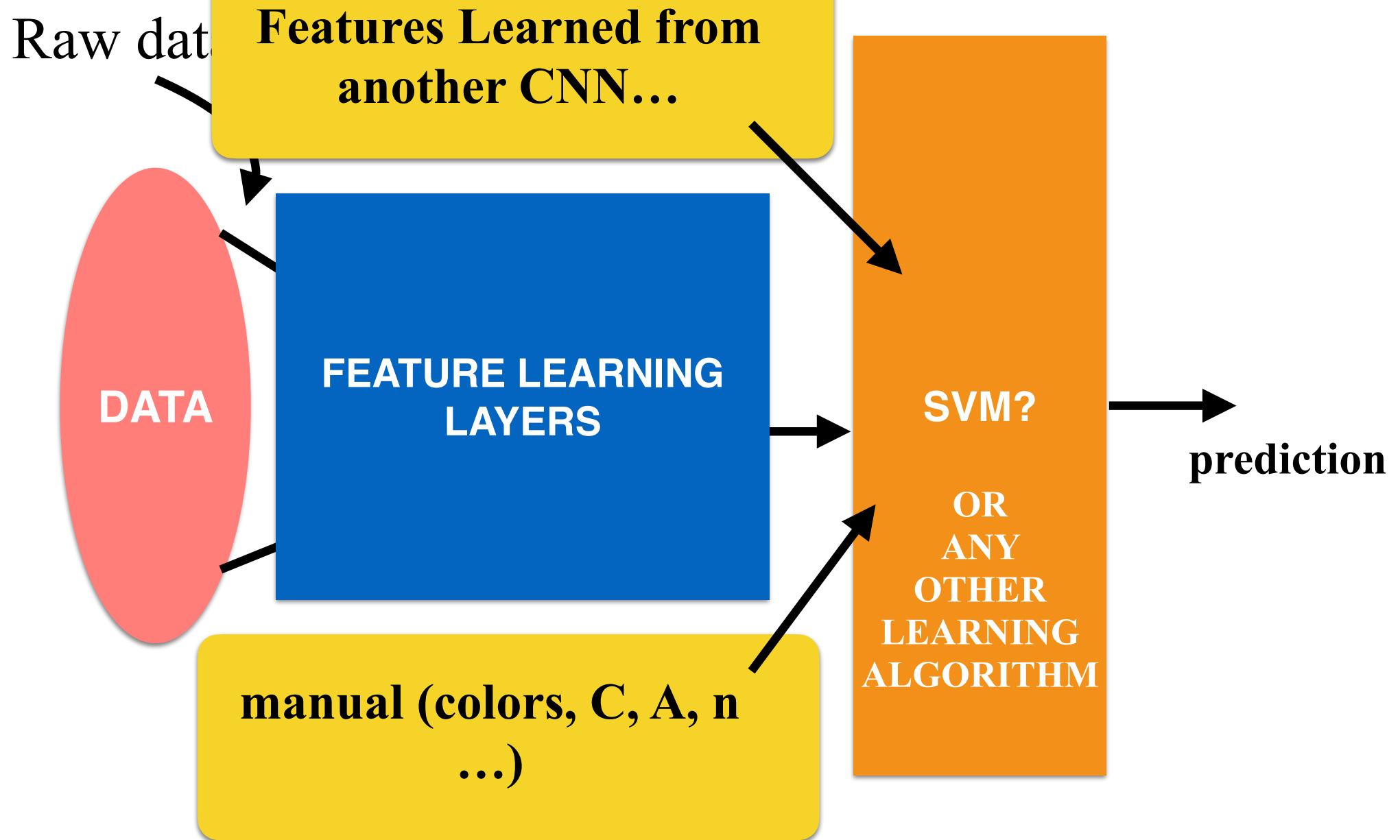
THE LEARNING ALGORITHM CAN BE CHANGED



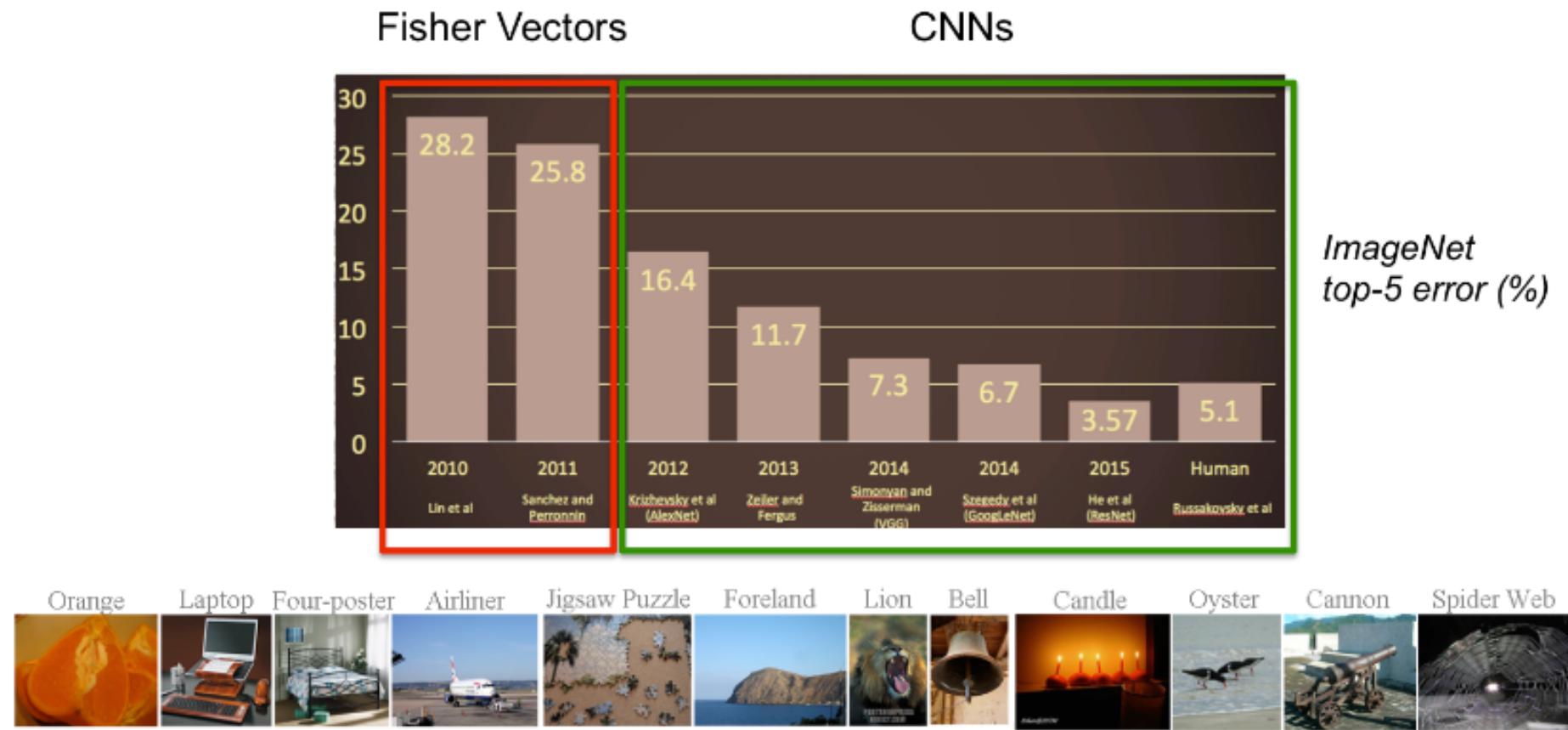
THE FEATURES CAN
BE MANIPULATED OR COMBINED



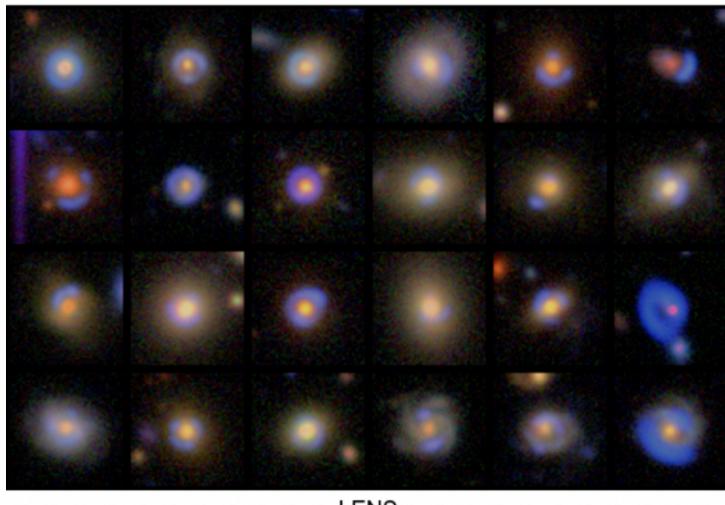
THE FEATURES CAN
BE MANIPULATED OR COMBINED



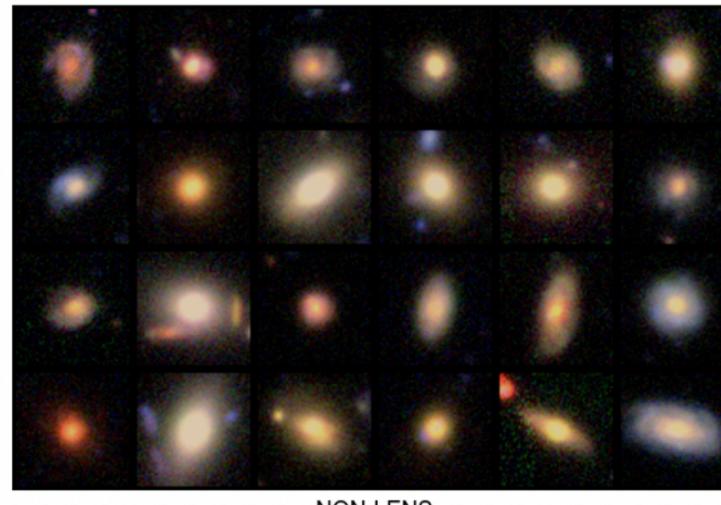
THIS IS A CHANGE OF PARADIGM!



1. Classification



LENS

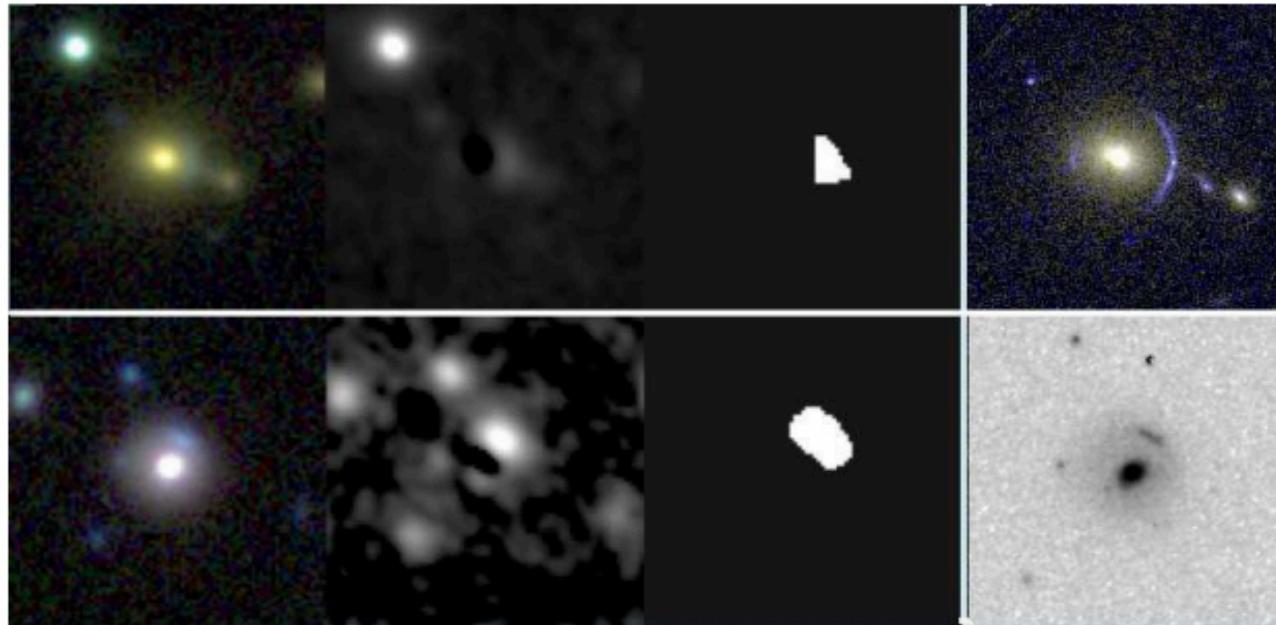


NON-LENS

**Detection of Strong Lenses
Valuable information of
Dark Matter properties**

**Future surveys will
increase the samples by
orders of magnitude.**

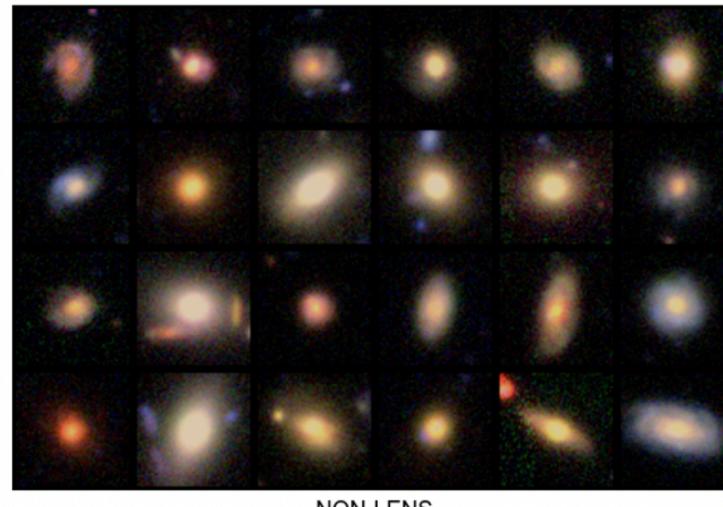
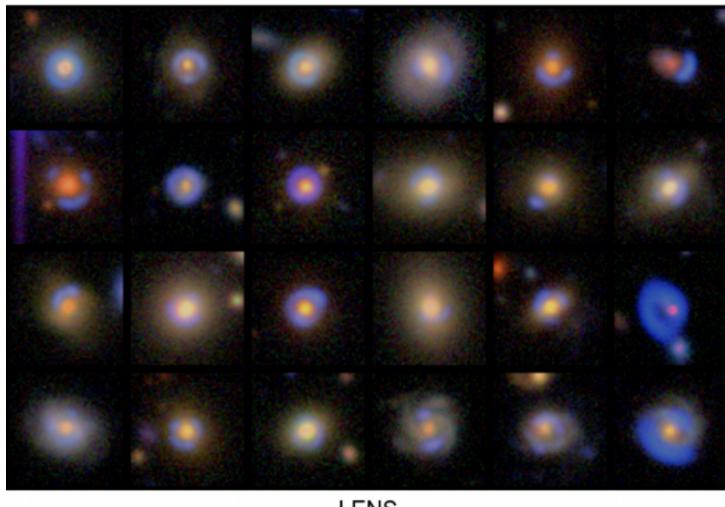
Jacobs+17



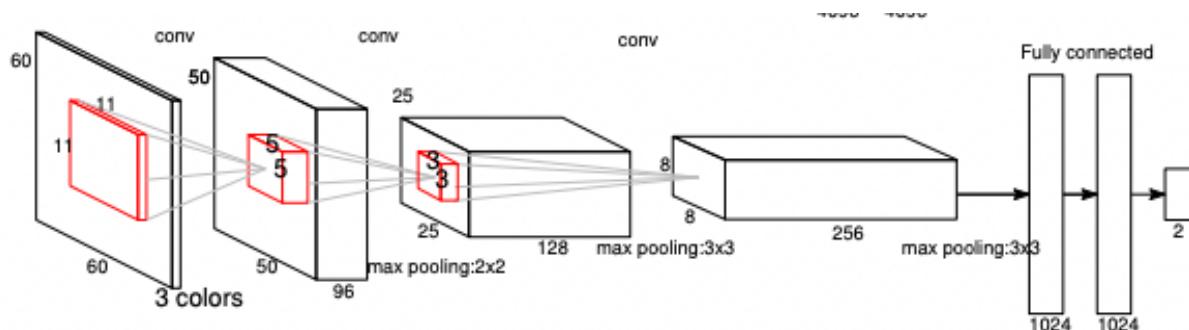
"Pre Deep Learning"
Approach

Gavazzi+17

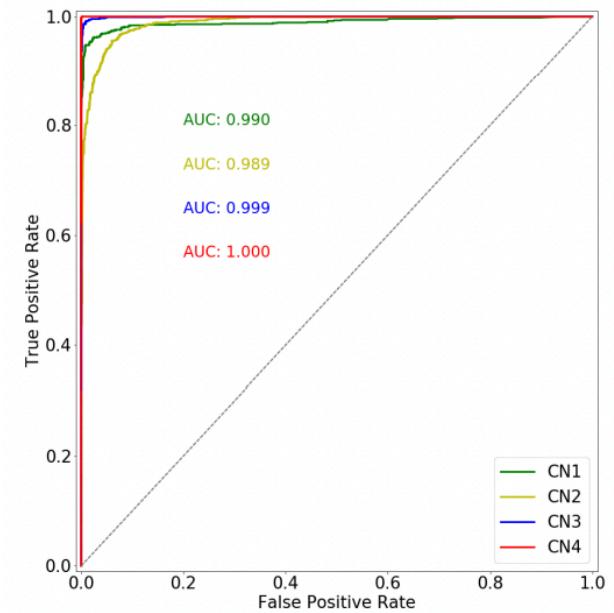
1. Classification



Jacobs+17



It illustrates the change of paradigm from an algorithmic centric focus to a purely data driven approach to data



1. Classification

CNN based classifications reach unprecedented accuracy

Name	type	AUROC	TPR ₀	TPR ₁₀	short description
CMU-DeepLens-Resnet-ground3	Ground-Based	0.98	0.09	0.45	CNN
CMU-DeepLens-Resnet-Voting	Ground-Based	0.98	0.02	0.10	CNN
LASTRO EPFL	Ground-Based	0.97	0.07	0.11	CNN
CAS Swinburne Melb	Ground-Based	0.96	0.02	0.08	CNN
AstrOmatic	Ground-Based	0.96	0.00	0.01	CNN
Manchester SVM	Ground-Based	0.93	0.22	0.35	SVM / Gabor
Manchester2	Ground-Based	0.89	0.00	0.01	Human Inspection
ALL-star	Ground-Based	0.84	0.01	0.02	edges/gradiants and Logistic Reg.
CAST	Ground-Based	0.83	0.00	0.00	CNN / SVM
YattaLensLite	Ground-Based	0.82	0.00	0.00	SExtractor
LASTRO EPFL	Space-Based	0.93	0.00	0.08	CNN
CMU-DeepLens-Resnet	Space-Based	0.92	0.22	0.29	CNN
GAMOCLASS	Space-Based	0.92	0.07	0.36	CNN
CMU-DeepLens-Resnet-Voting	Space-Based	0.91	0.00	0.01	CNN
AstrOmatic	Space-Based	0.91	0.00	0.01	CNN
CMU-DeepLens-Resnet-aug	Space-Based	0.91	0.00	0.00	CNN
Kapteyn Resnet	Space-Based	0.82	0.00	0.00	CNN
CAST	Space-Based	0.81	0.07	0.12	CNN
Manchester1	Space-Based	0.81	0.01	0.17	Human Inspection
Manchester SVM	Space-Based	0.81	0.03	0.08	SVM / Gabor
NeuralNet2	Space-Based	0.76	0.00	0.00	CNN / wavelets
YattaLensLite	Space-Based	0.76	0.00	0.00	Arcs / SExtractor
All-now	Space-Based	0.73	0.05	0.07	edges/gradiants and Logistic Reg.
GAHEC IRAP	Space-Based	0.66	0.00	0.01	arc finder

Metcalf+19

ALSO FOR GALAXY MORPHOLOGY

SVMs

CNNs

[HUERTAS-COMPANY+14]

AUTOMATIC

Late-Type

13

Early-Type

87

75

25

Early-Type



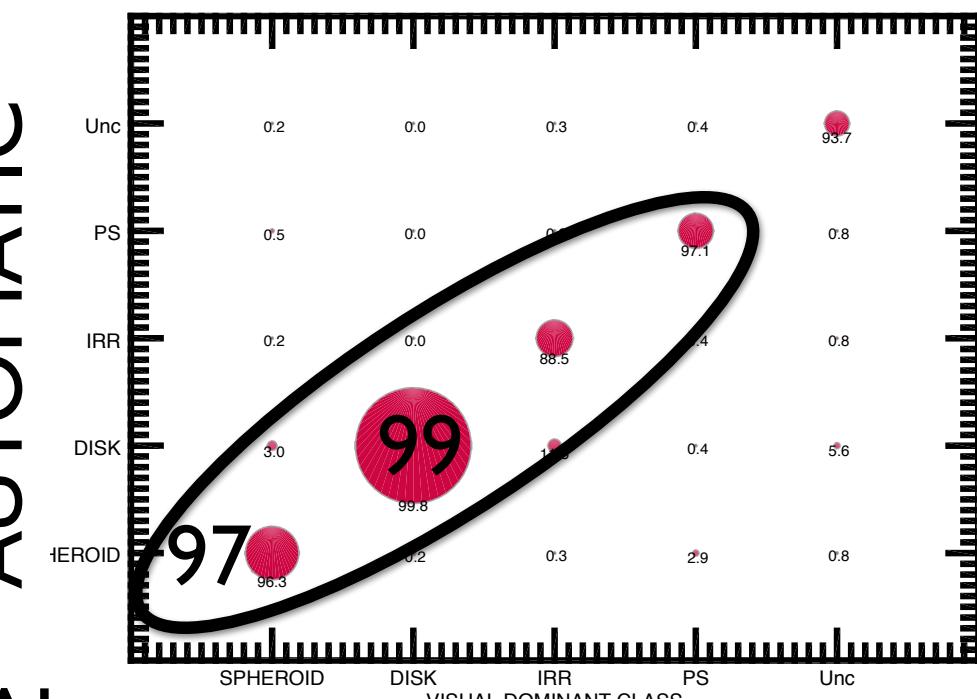
Late-Type



AUTOMATIC

VISUAL

[HUERTAS-COMPANY+15b]

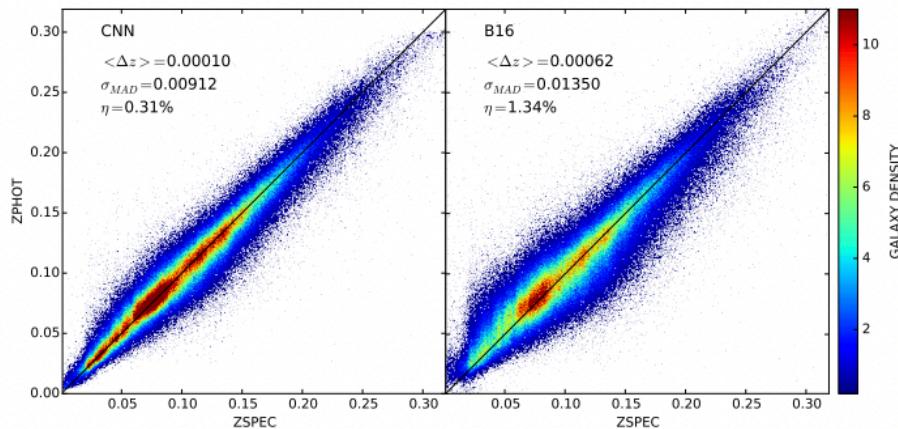


VISUAL

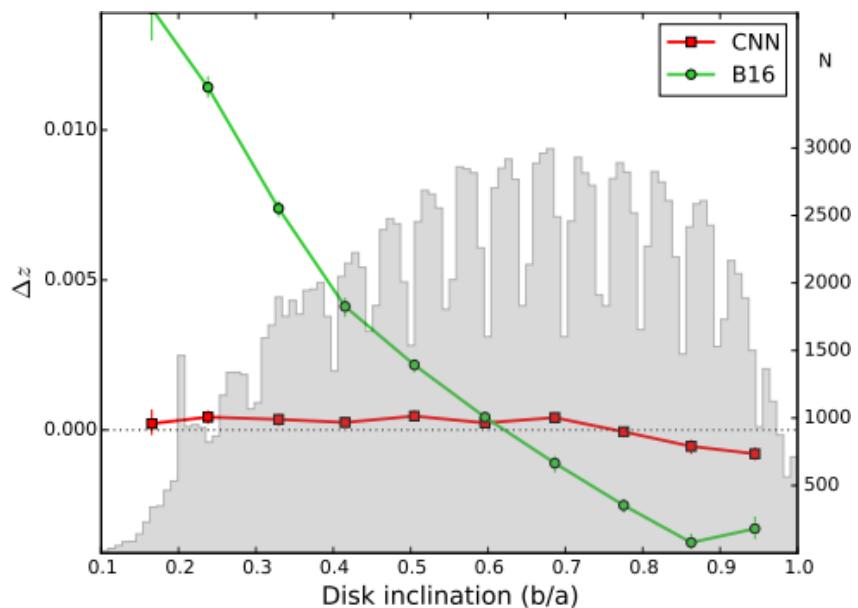
VISUAL

Photometric Redshifts

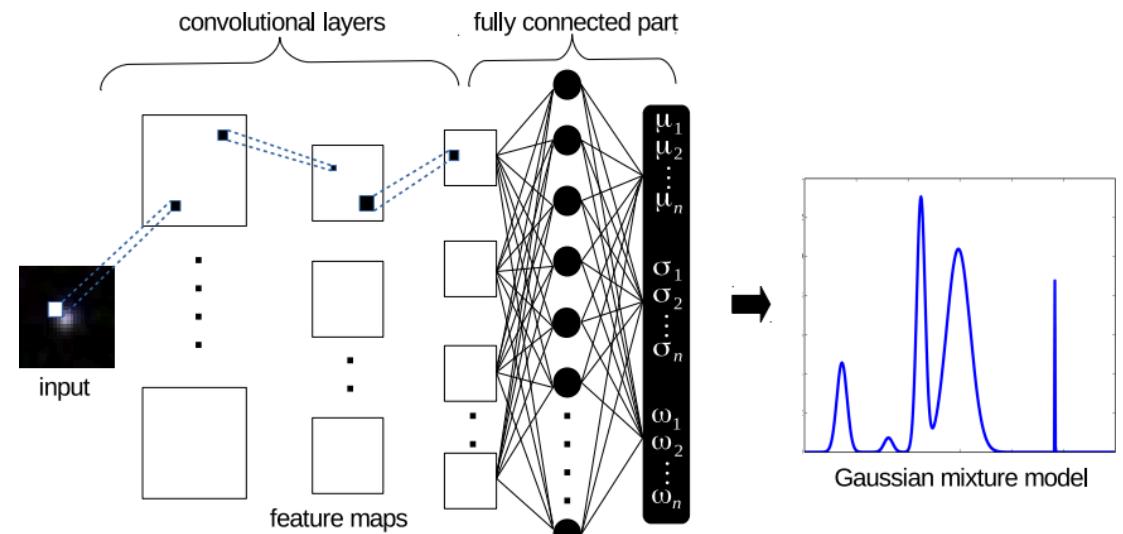
Deep Learning Classical approach



Geometric Effects are automatically considered
(beyond photometry)



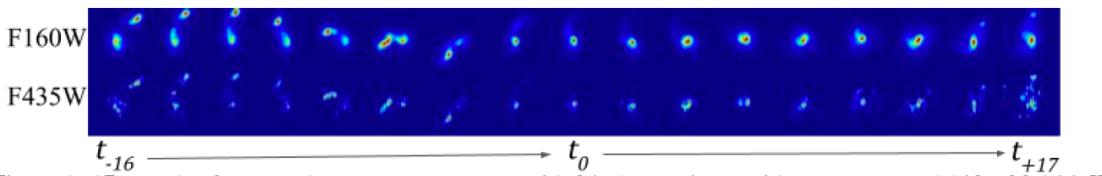
Pasquet+18



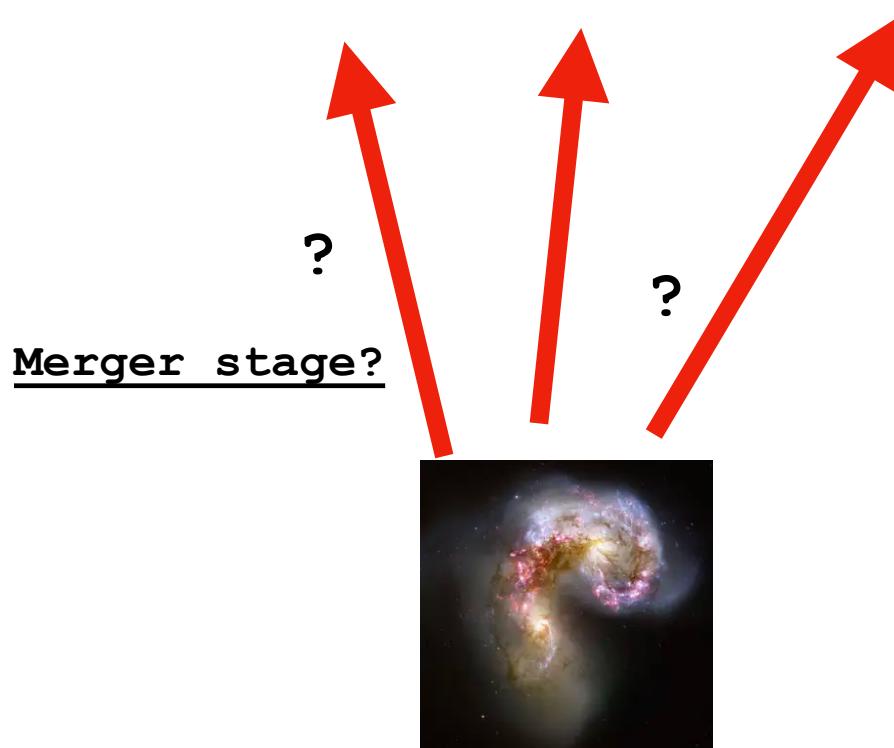
Disanto+18

Uncertainty quantification through Mixture Density Networks

Mergers of Galaxies

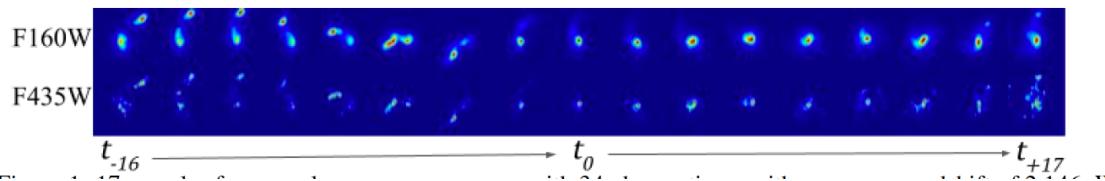


Merger of galaxies sequence
from cosmological simulations

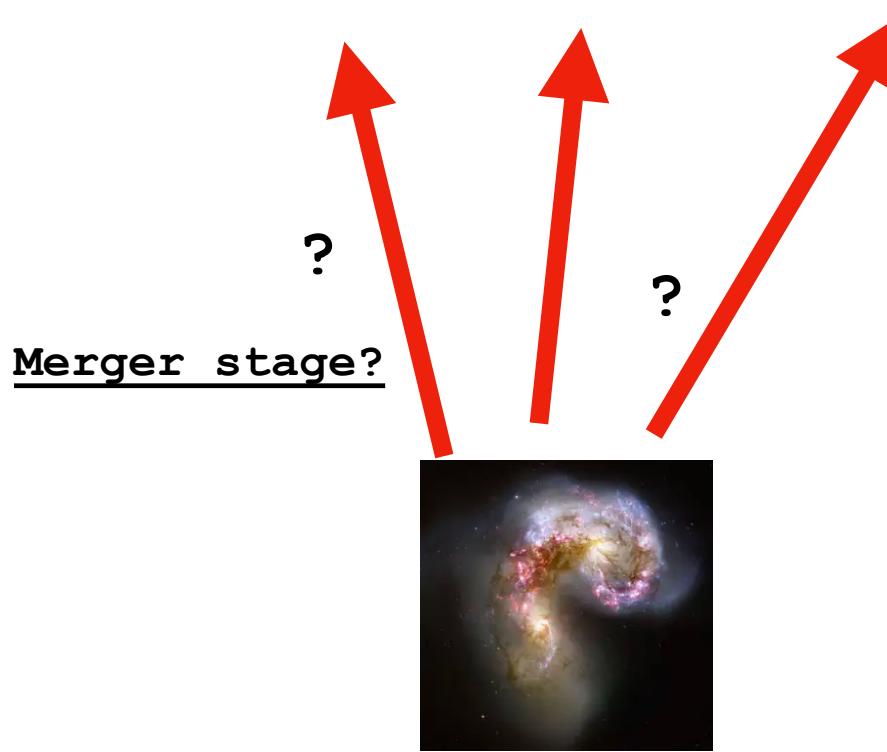


**Neural Networks to find
relations between observables
and physical processes**

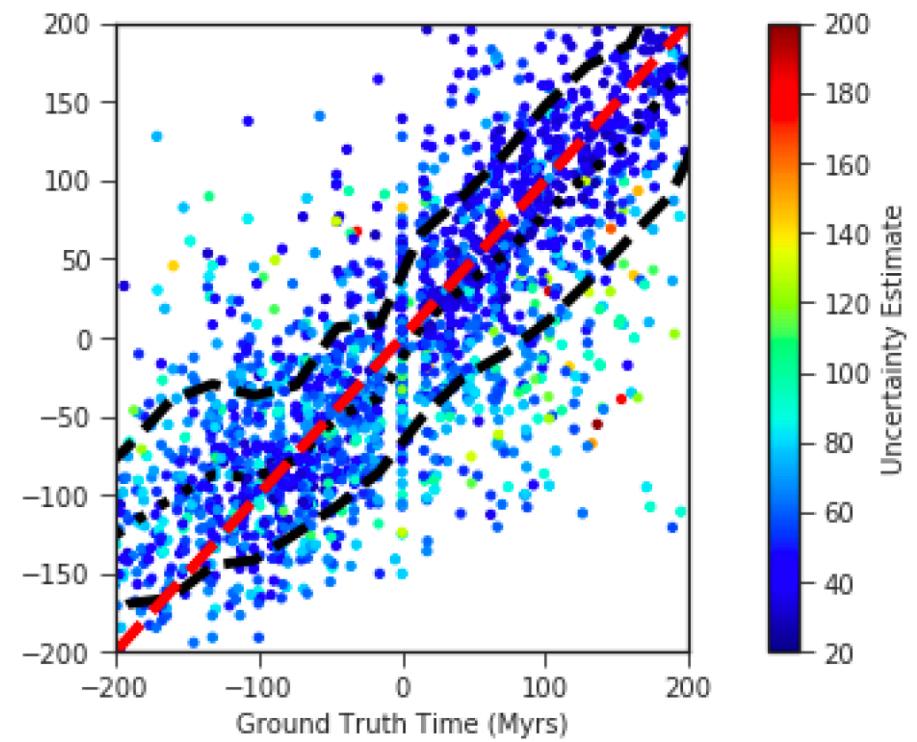
Mergers of Galaxies



Merger of galaxies sequence from cosmological simulations

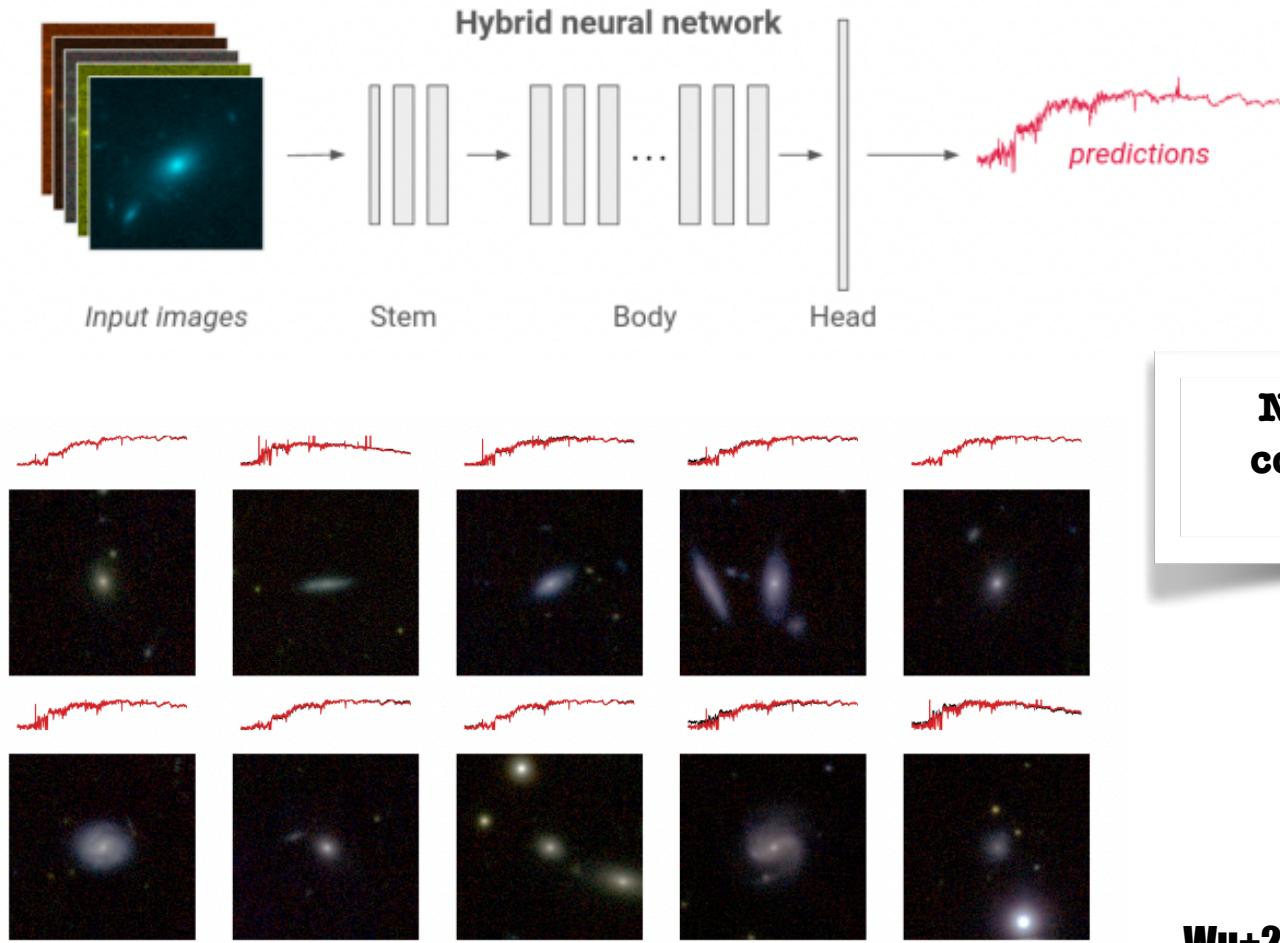


Neural Networks to find relations between observables and physical processes



Koppula+21

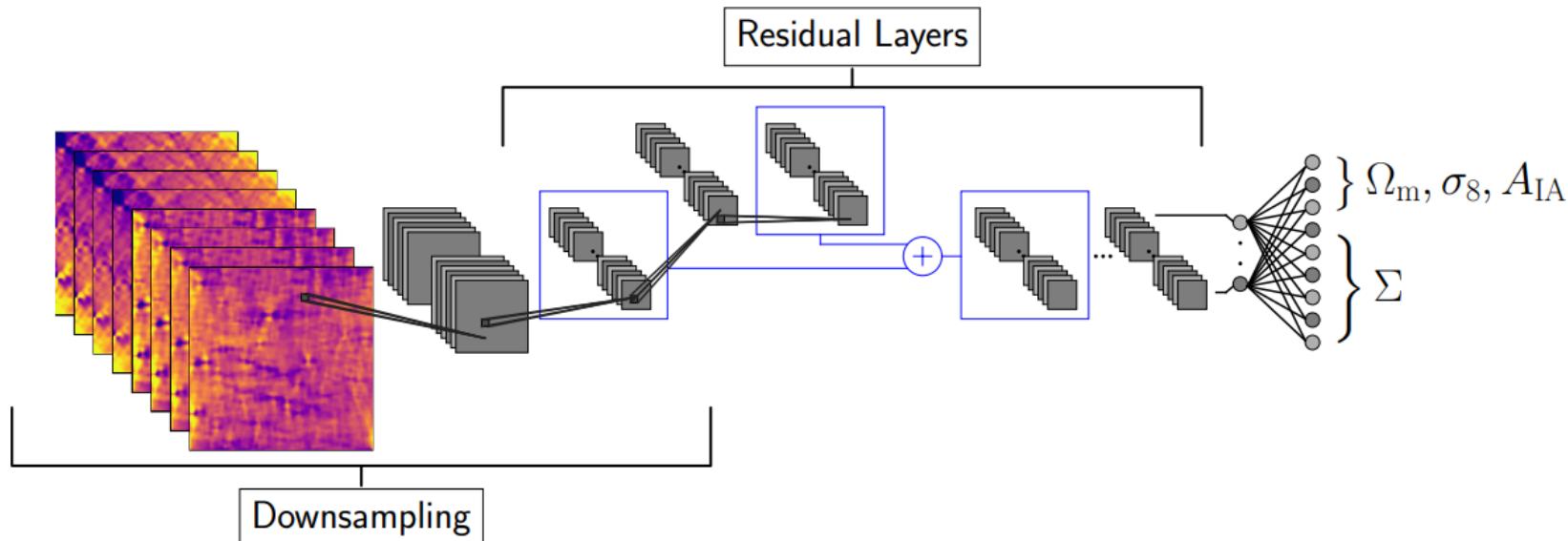
Virtual Observatory



Neural Networks to learn complex mapping between observables

Wu+21

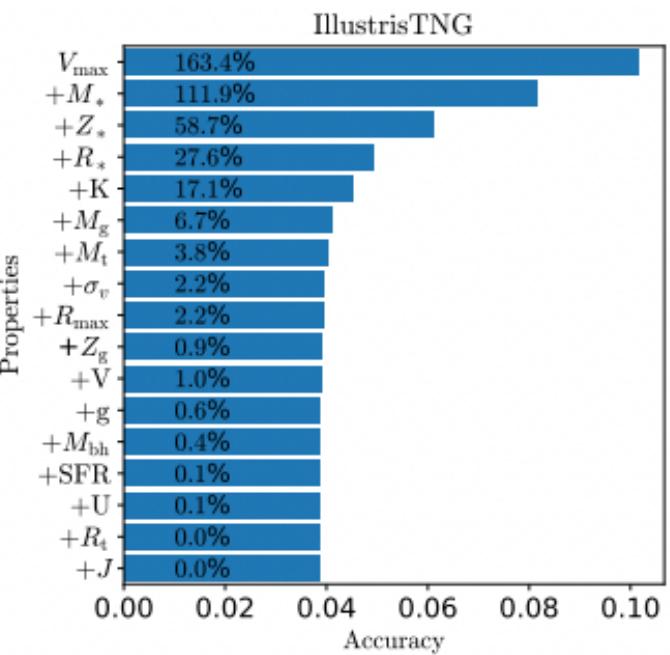
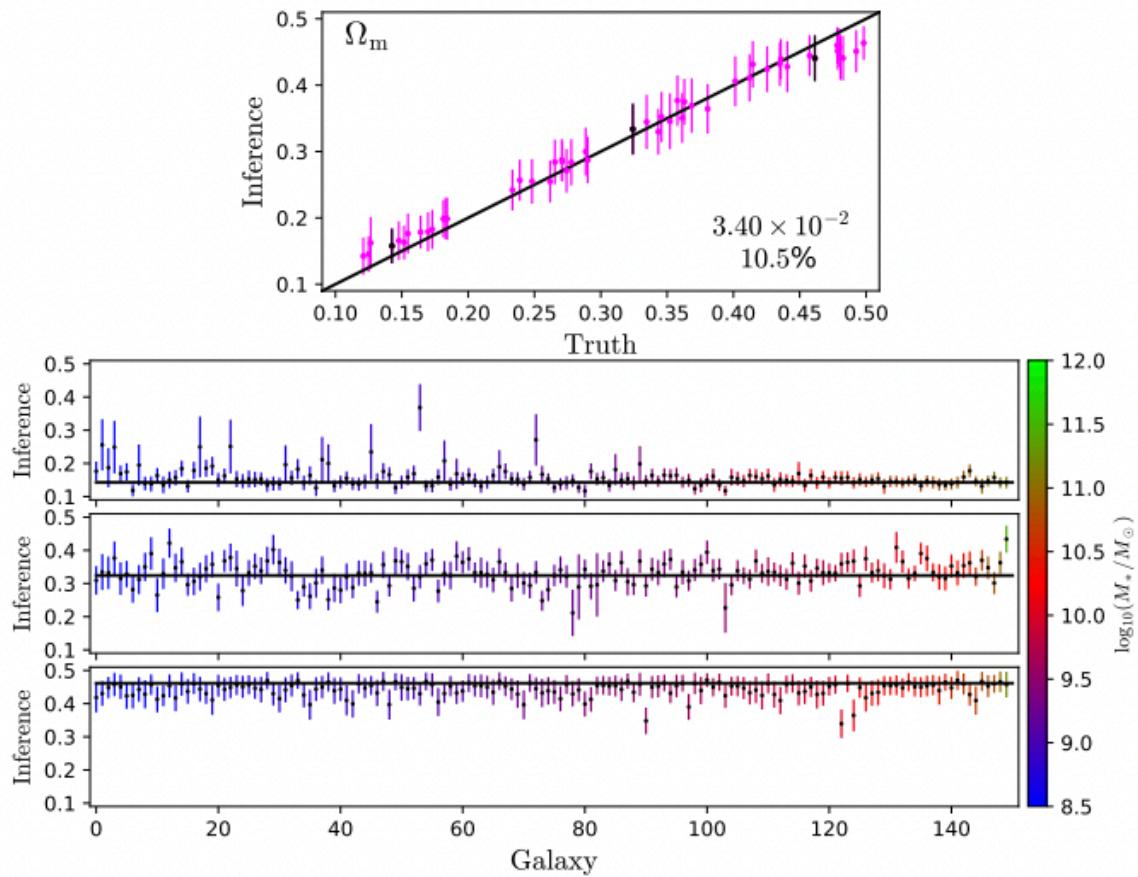
Deep Learning for Cosmological Inference



Motivation: Generalize comparison of observations with theory, beyond basic summary statistics

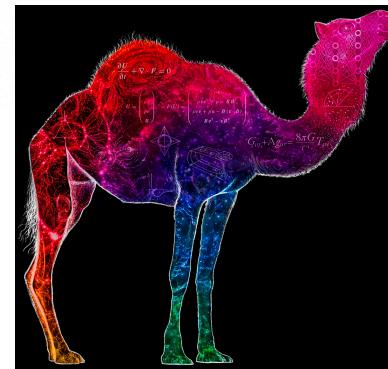
Neural Networks are used as efficient feature extractors

Cosmological Inference



Villaescusa-Navarro+22

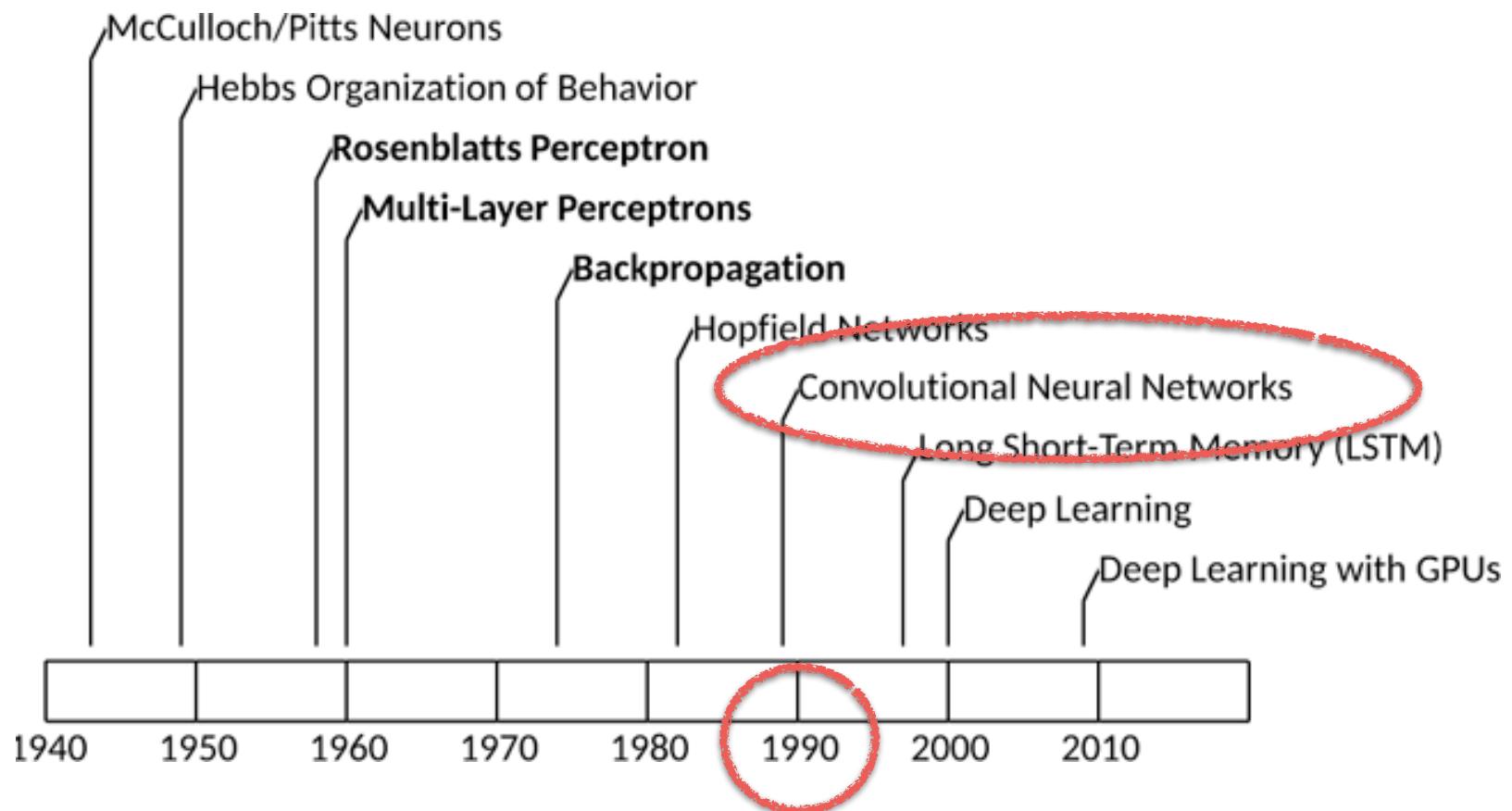
CAMELS



Neural Networks to learn new relations between models and observables

Ravanbakhsh+17, Brehmer+19, Ribli+19, Pan+19, Ntampaka+19, Alexander+20, Arjona+20, Coogan+20, Escamilla-Rivera+20, Hortua+20, Vama+20, Vernardos+20, Wang+20, Mao+20, Arico+20, Villaescusa_navarro+20, Singh+20, Park+21, Modi+21, Villaescusa-Navarro+21ab, Moriwaki+21, DeRose+21, Makinen+21, Villaescusa-Navarro+22

WELL, BUT THIS IS AN “OLD” IDEA - WHY NOW?



WELL, BUT THIS IS AN “OLD” IDEA - WHY NOW?

1 - MORE DATA TO TRAIN! DEEP NETWORKS HAVE A
LARGE NUMBER OF PARAMETERS - THX TO SOCIAL
MEDIA ...

WELL, BUT THIS IS AN “OLD” IDEA - WHY NOW?

2 - GPUs - TRAINING OF THESE DEEP NETWORKS
HAS REMAINED PROHIBITIVELY TIME CONSUMING
WITH CPUs - THX TO VIDEO GAMES...

GPUs



NVIDIA TITANX GPU

GPUs vs. CPUs

CPUs

FEWER CORES (~10x)

EACH CORE IS FASTER

USEFUL FOR
SEQUENTIAL TASKS

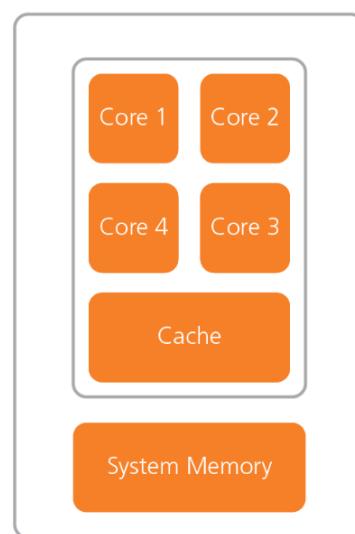
GPUs

MORE CORES (100x)

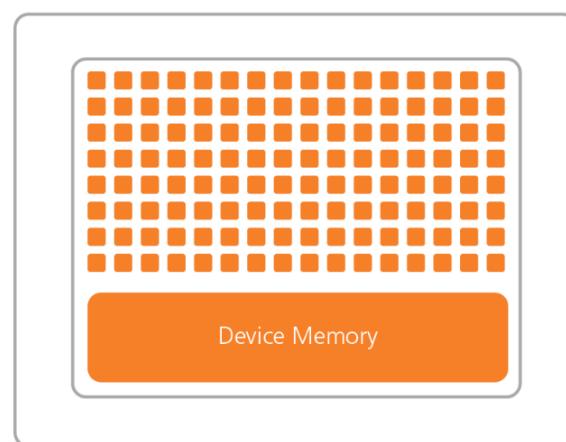
EACH CORE IS SLOWER

USEFUL FOR PARALLEL
TASKS

CPU (Multiple Cores)



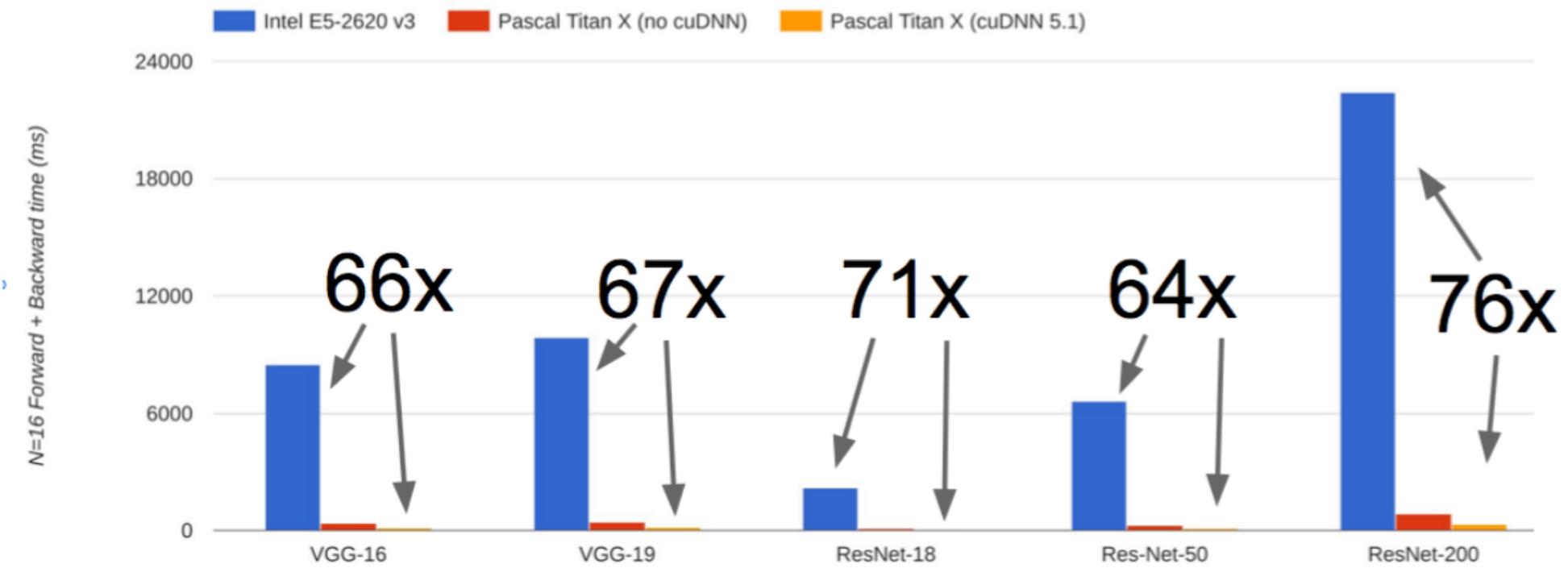
GPU (Hundreds of Cores)



Slide Credit:

GPUs vs. CPUs

More benchmarks available [here](#).



THE PRICE TO PAY?

1. LARGE NUMBER OF PARAMETERS IMPLIES LARGE DATASETS TO TRAIN
2. LOOSE EVEN MORE DEGREE OF CONTROL OF WHAT THE ALGORITHM IS DOING SINCE THE FEATURE EXTRACTION PROCESS BECOMES UNSUPERVISED

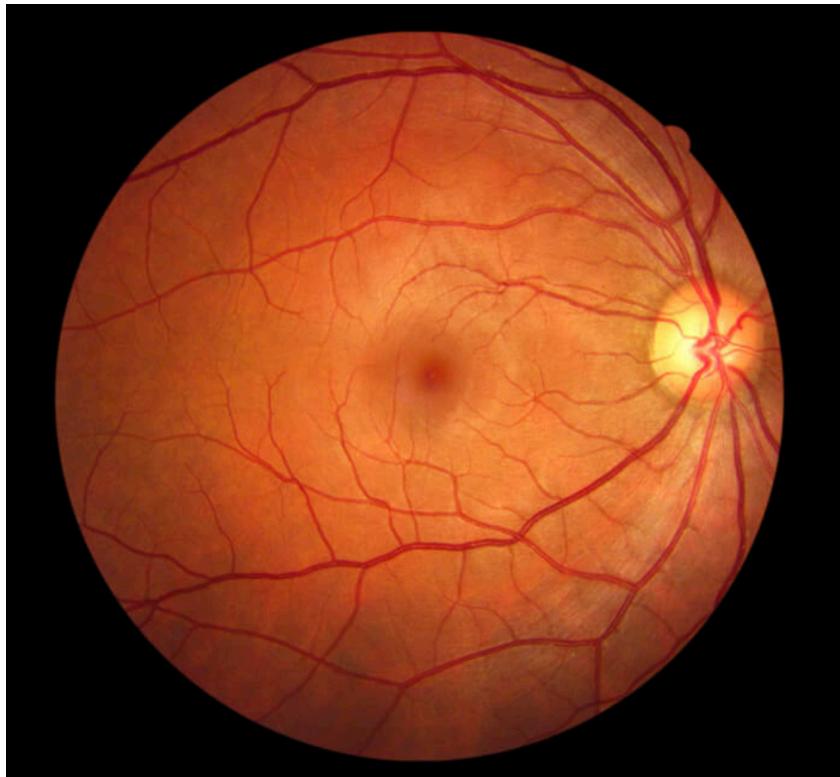
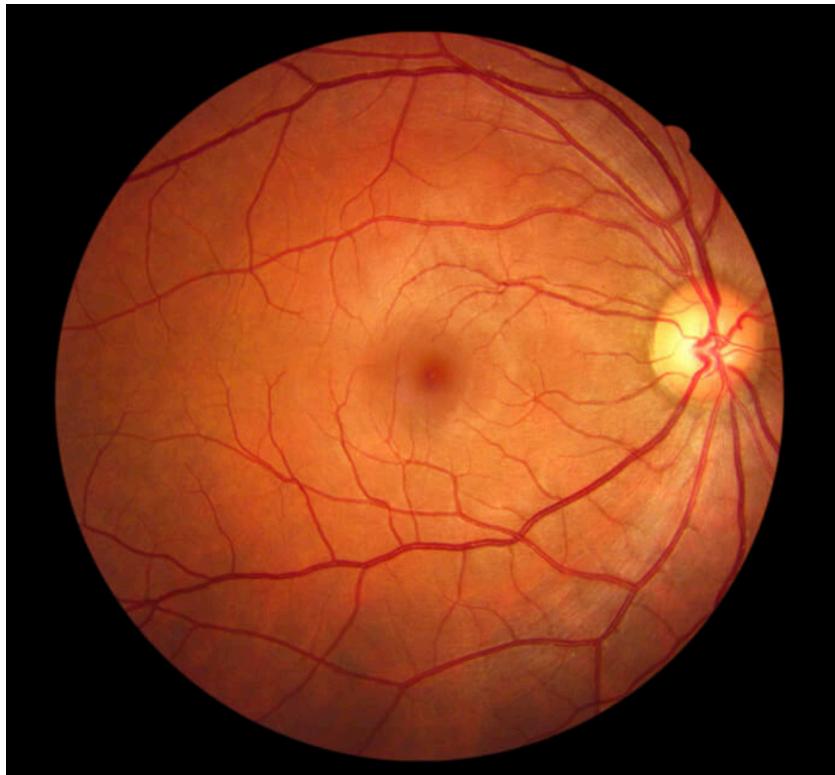


IMAGE OF THE BACK OF THE EYE





**DEEP LEARNING CAN
IDENTIFY
THE PATIENT'S
GENDER WITH 95%
ACCURACY**

IMAGE OF THE BACK OF THE EYE

