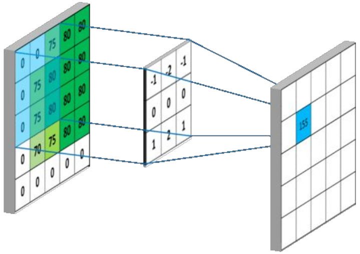
CEFET/RJ

Convolutional Neural Networks



Machine Learning Course - PPCIC

Rafaela Castro

October/2020

About me

Master’s degree at CEFET-RJ | Data Architect at TRF2

Admission: May 2018 Conclusion: July 2020

Advisor: Eduardo Bezerra

Project: Apply convolutional neural networks to model spatiotemporal data

rafaela.nascimento@eic.cefet-rj.br

/in/rafaela00castro

/rafaela00castro

Task: image classification

PS: images are numbers for computers!

Why **not use only** fully connected (FC) layers for this task?

Fully connected layers

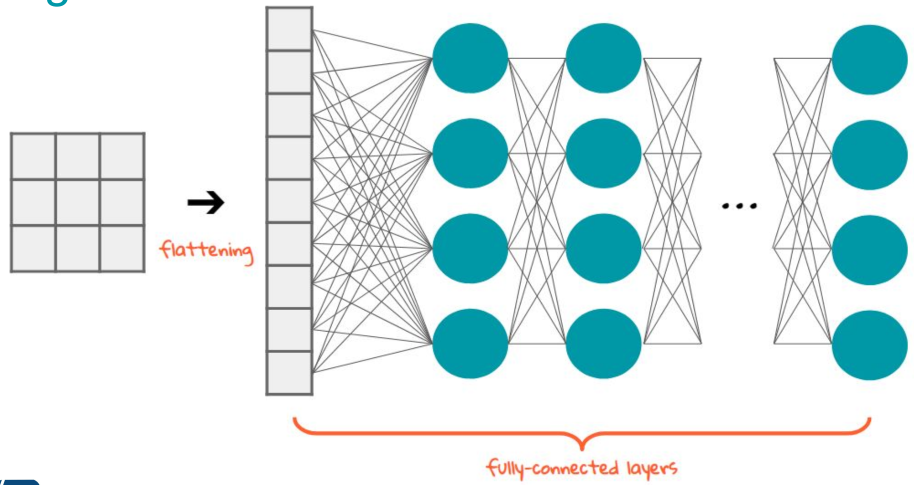
- Require a lot of connections >> many weights
- Example: classify grayscale image of size 200x200
- Architecture: suppose only one hidden layer with 500 units
- Neural Net with 20 million weights 😲
  - Input layer = 40000 units (200 x 200)
  - hidden layer = 500 units

https://www.cs.toronto.edu/~lczhang/360/lec/w04/convnet.html

Fully connected layers (2)

Problems of using just FC for image classification:

- Many weights
  - long processing time
  - risk of overfitting
- Spatial structure is ignored



https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480

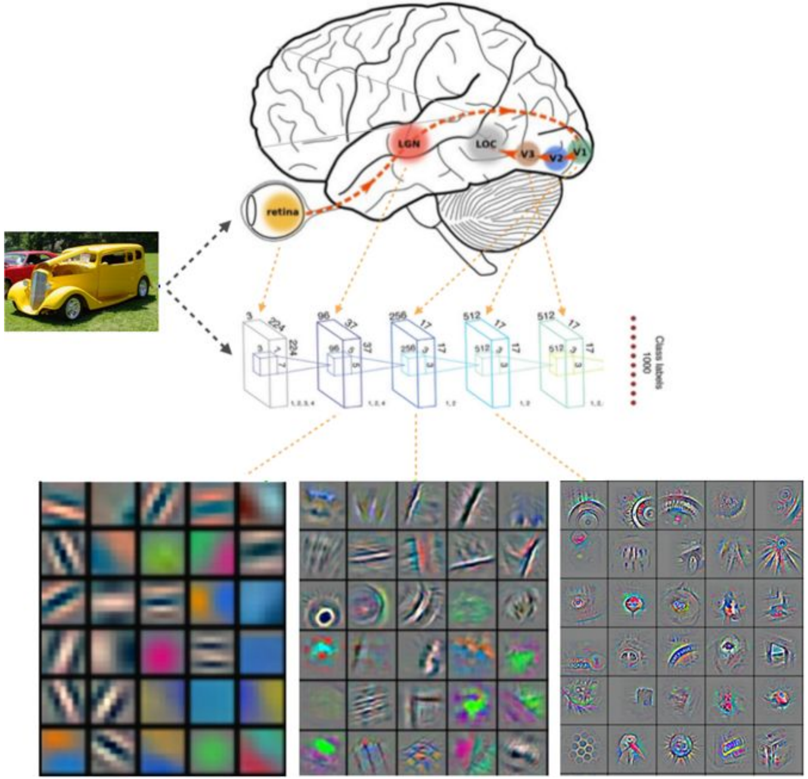
Convolutional Neural Networks (CNN) are the solution...

CNN (or ConvNet) introduction

- Introduced by Yann LeCun et al. (1989)
- Inspired in the visual cortex
- Sort images into categories even better than humans in some cases (milestone for deep learning - AlexNet paper, 2012)
- Advantages of using convolutional layers over fully connected layers
  - Local connectivity
  - Weight sharing

Inspired in the visual cortex

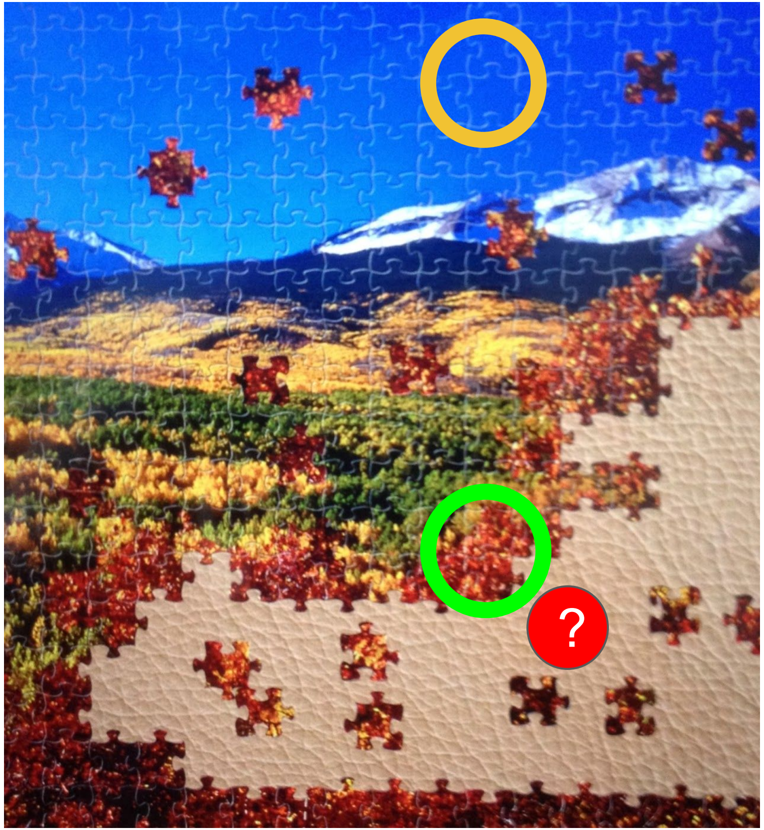
- Lower layers detect simple aspects of the image (edges and spots).
- Higher layers detect more complicated patterns (shapes)



https://medium.com/@vlomonaco/what-i-learned-at-the-deep-learning-summer-school-2017-in-bilbao-c6eae2963554

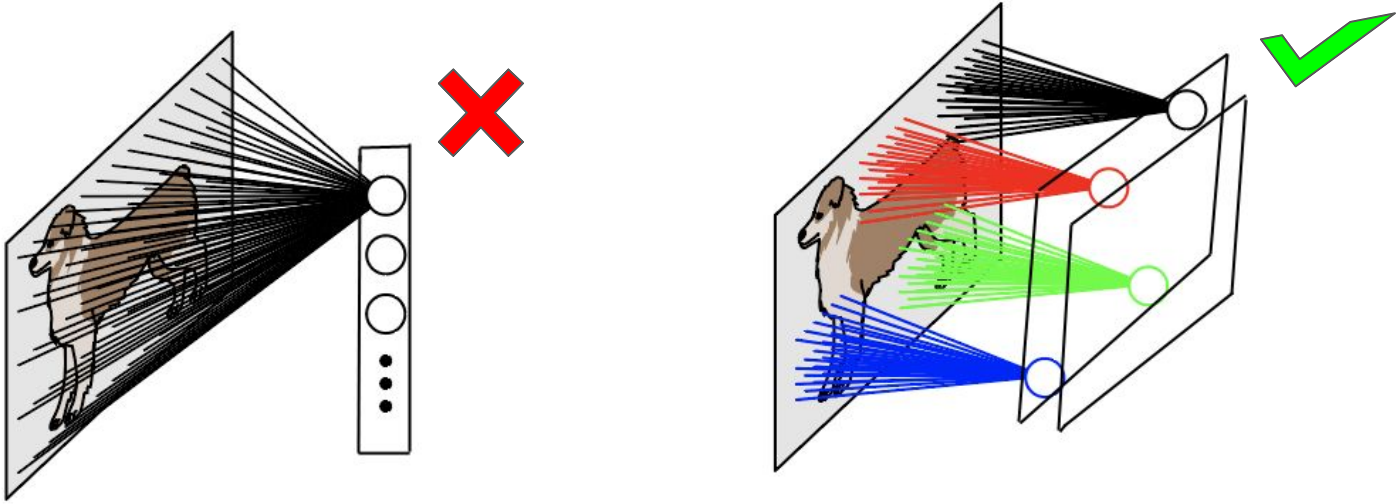
Local connectivity (intuition)

- For a particular piece ( ), features from the nearby pieces ( ) give more information than the pieces further away ( ).



Local connectivity

CNN are not entirely connected. The neurons in a layer will only be connected to a small region (receptive field) of the previous layer.



https://freecontent.manning.com/deep-learning-for-image-like-data/

Weight sharing (intuition)

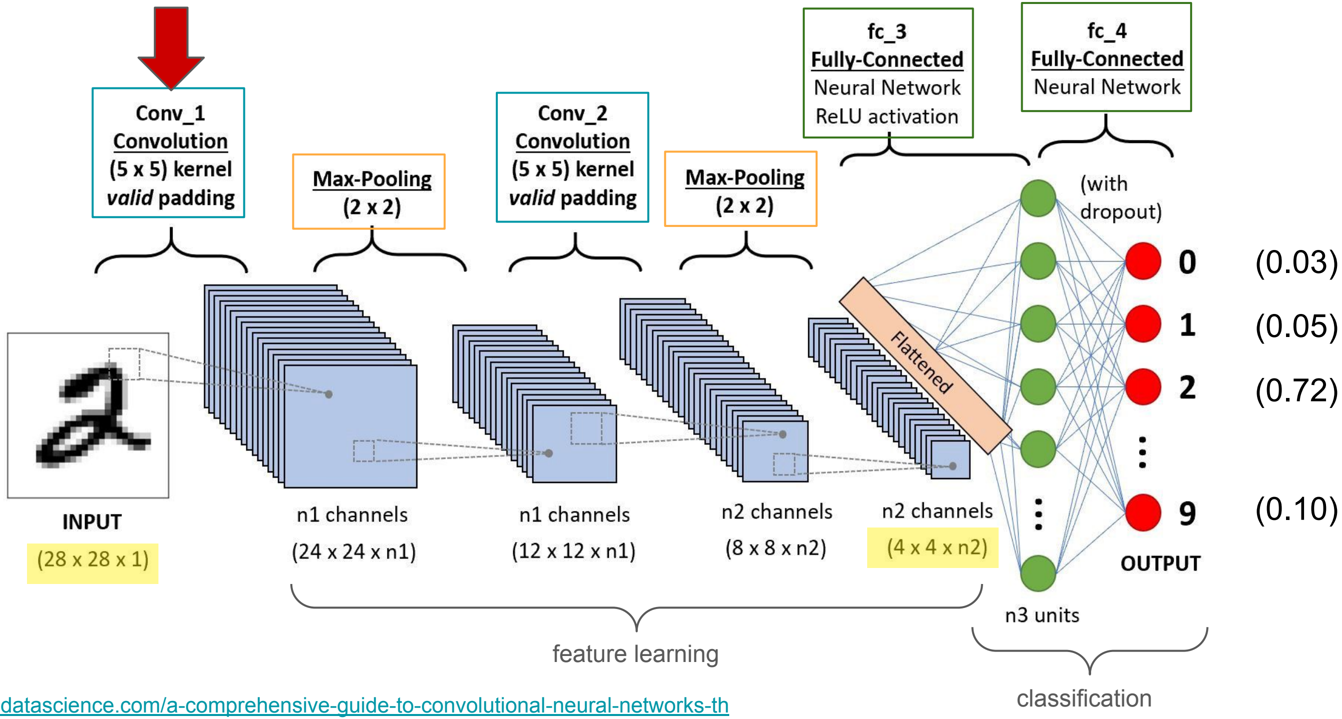
If a feature detector knows how to detect a local feature in one region of the image, then the same detector is useful in all other regions of the image.

https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

Weight sharing

- In CNN terminology, detector = filter (or kernel).
- The filter consists of a matrix that is learned during training.
- CNN shares the same parameters (weights and biases) across different locations in an image (at the same depth slice).

Typical architecture for image classification



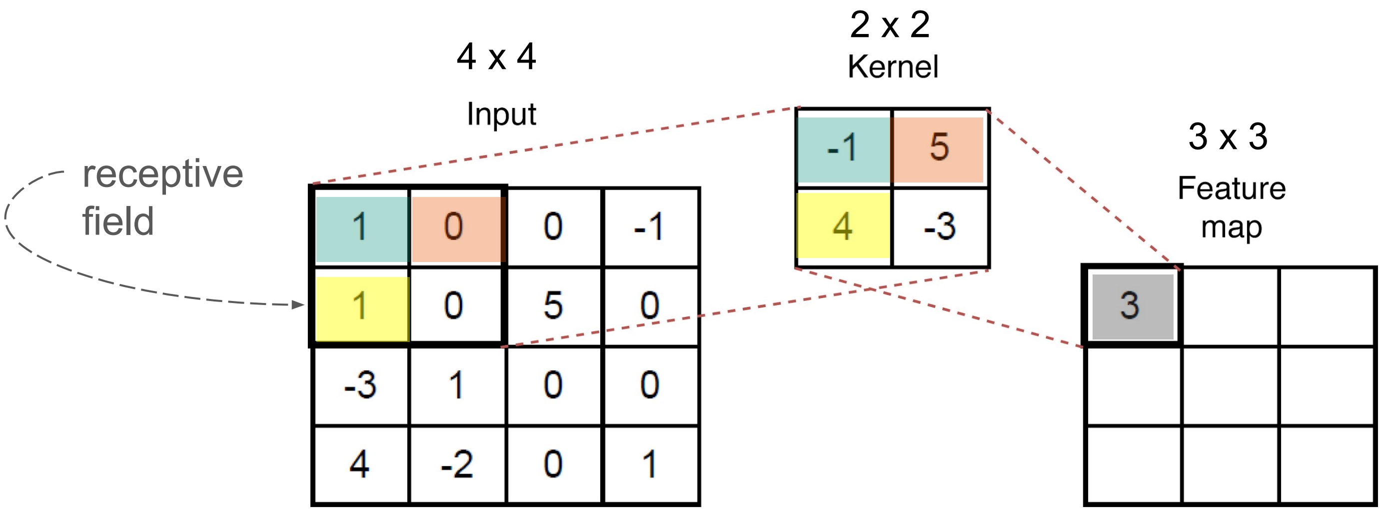
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Convolution operation

During the forward pass:

- The convolutional layer convolves the filter (or kernel) across the receptive field of the input volume.
  - i.e., computes dot products between the entries of the filter (weights) and the input at any position.
- The result of applying convolution for each filter is a 2-dimensional feature map (or activation map).

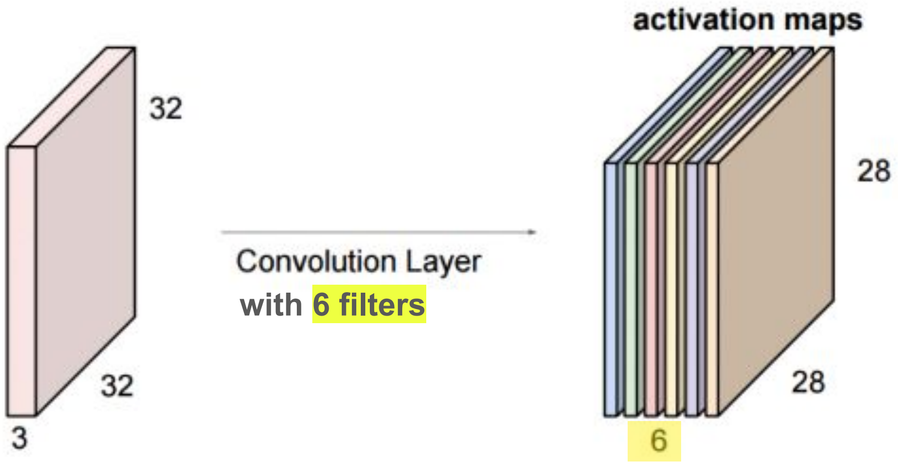
Convolution operation (2)



$$(1 \times -1) + (0 \times 5) + (1 \times 4) + (0 \times -3) = 3$$

Convolution operation (3)

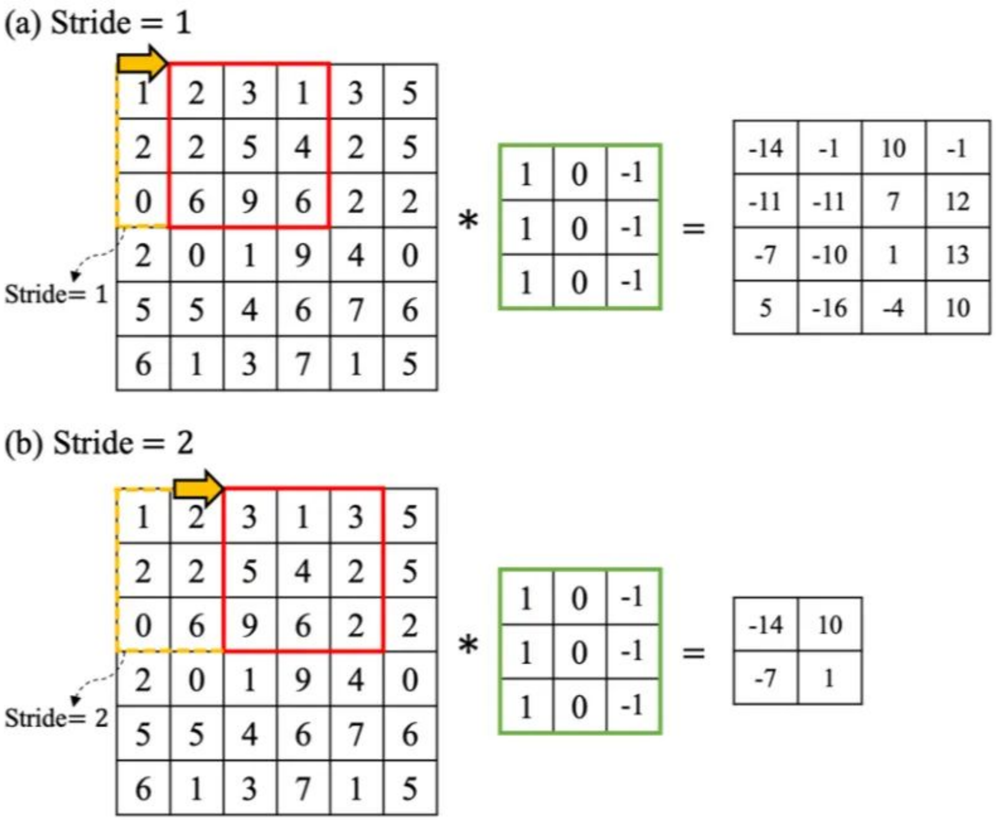
The convolutional layer stacks these feature maps (or activation maps) along the depth dimension and produces the output volume.



https://cs231n.github.io/convolutional-networks/

Stride

Hyperparameter that controls how the kernel (or filter) slides through the input, i.e., the position at which convolution operation must begin.



https://www.brilliantcode.net/1584/convolutional-neural-networks-1-convolution-layer-stride-padding-kernel/

Padding

Hyperparameter that defines an additional border in the input. But why?

- Convolution operation shrinks the size of the input >> padding helps generate output equal in size to the input (*same padding*)
- Pixels in the corner are convolved only a few number of times as compared to the central pixels (information loss) >> padding can be used to overcome this issue.

PS: *valid padding* = convolution operation without padding

Padding (2)

2D CNN

stride = 1
padding = 1

zero-padding

5 x 5
Input size

0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

3 x 3
Kernel

0	-1	0
-1	5	-1
0	-1	0

5 x 5
Feature map

114				

https://www.pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/

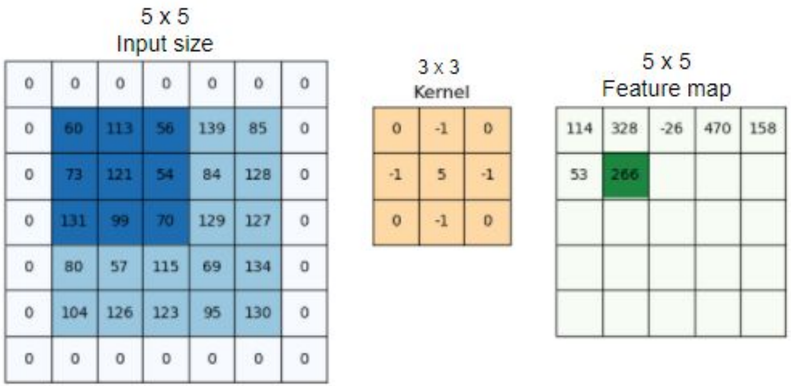
Output size calculation

The dimension (height x width) of the output feature map depends on the input size (I), kernel size (K), stride (S) and padding (P).

$$O = \frac{I - K + 2P}{S} + 1$$

$$O = \frac{5 - 3 + 2 \times 1}{1} + 1$$

$$O = 5$$

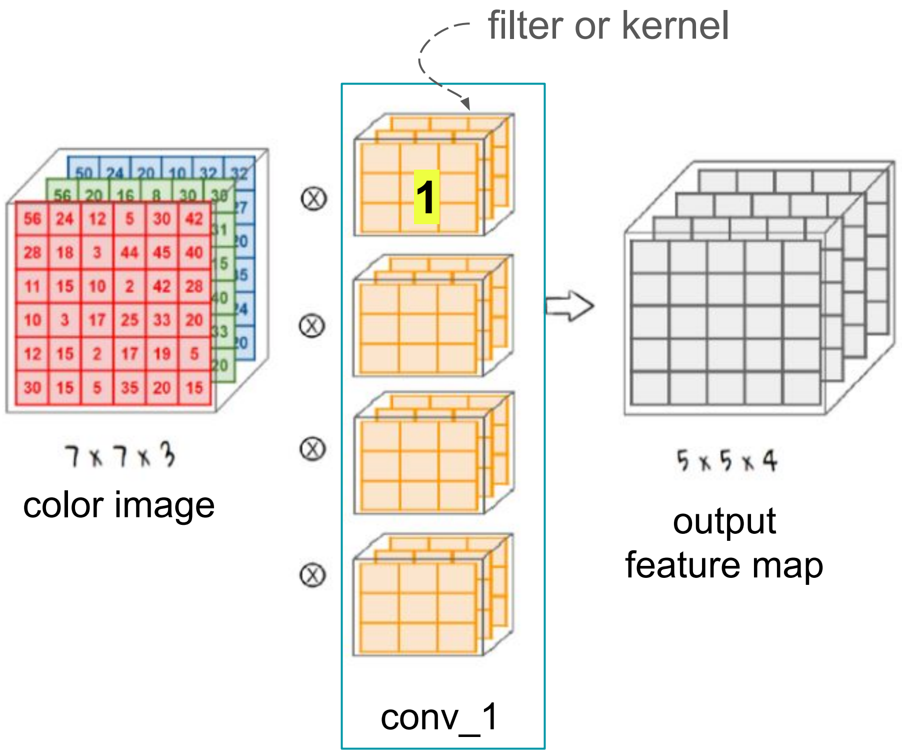


Example

Given a color image that has 7x7 pixel (this means: the input has a 7x7x3 volume), use one convolutional layer with 4 filters (size 3x3) to detect certain features.

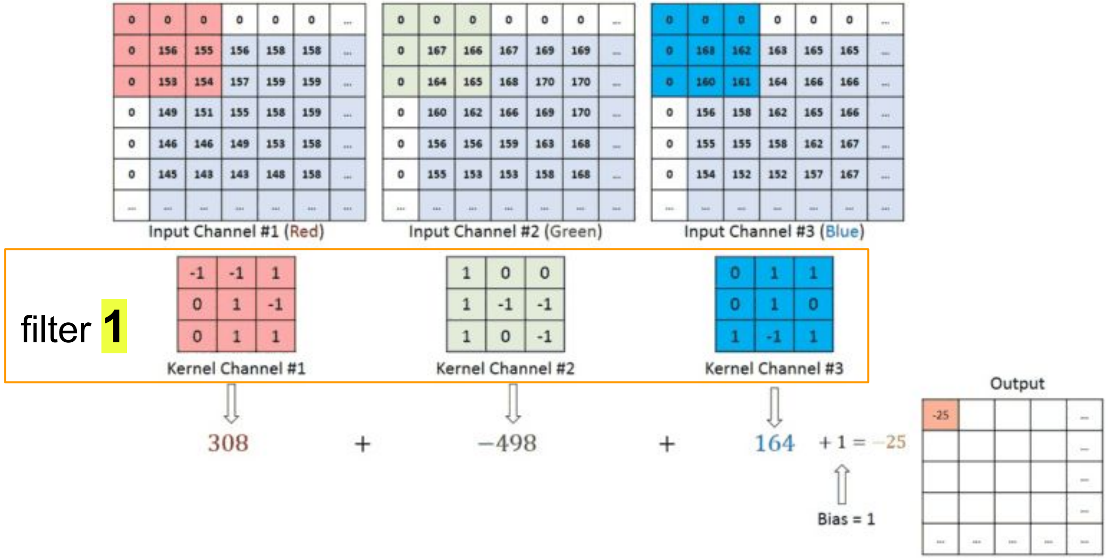
Pay attention to the volume of the output: the number of filters will determine the depth of the output.

Example (2)



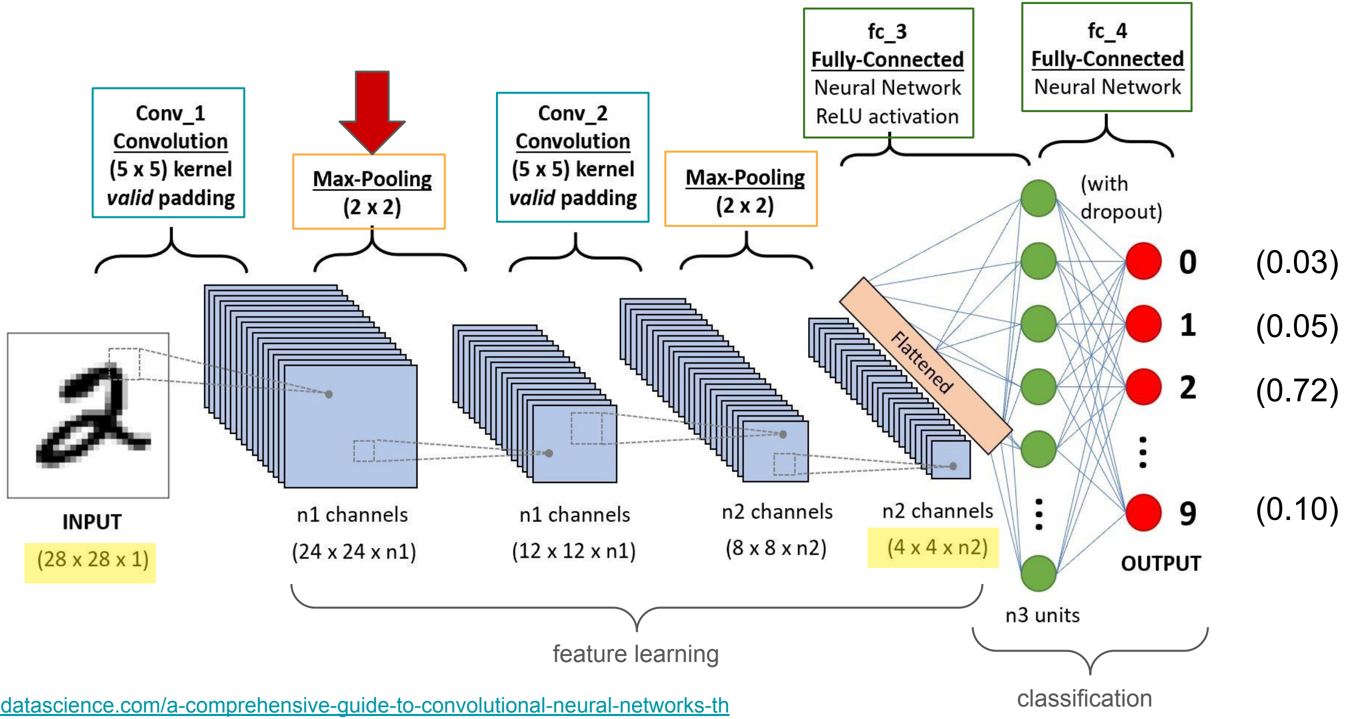
https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480

Example (3)



https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Typical architecture for image classification

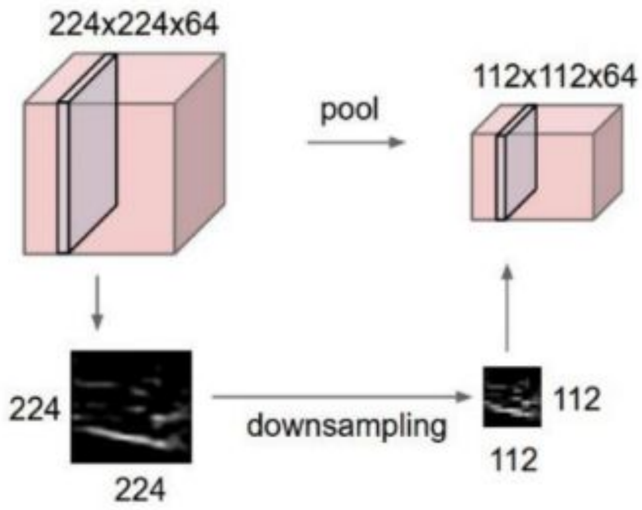


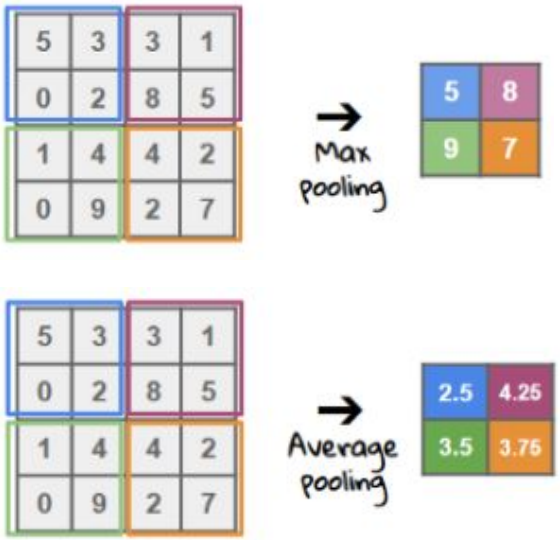
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Pooling layer

- Commonly used between convolutional layers
- Reduces spatial dimensions (width x height) >> decreases the computational power required
- Extracts dominant features >> translation invariant
- Depth of the output volume equal to input volume

Pooling layer (2)



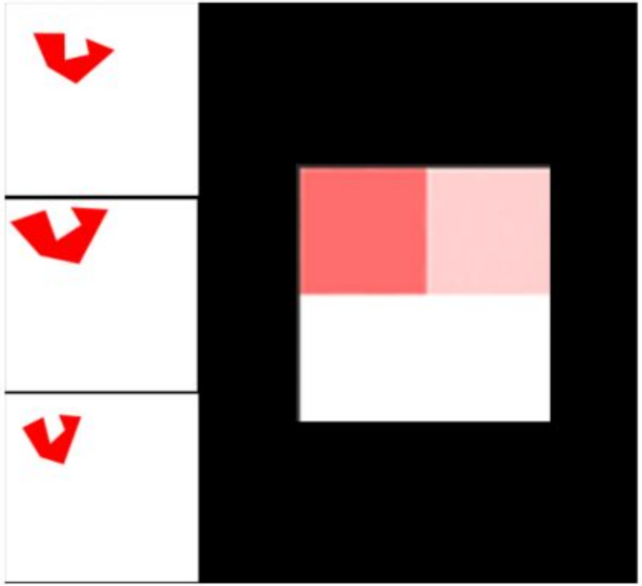


https://cs231n.github.io/convolutional-networks/#convert

https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480

Pooling (intuition)

- The three images on the left give the same image on the right.
- The model using only the right image knows that there is the shape on the top left.



https://www.quora.com/How-exactly-does-max-pooling-create-translation-invariance

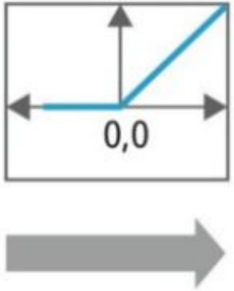
Activation function

Wherever a negative number occurs, swap it out for a 0.

it is easier to train and often achieves better performance.

Rectified Linear Unit (ReLU)

15	20	-10	35
18	-110	25	100
20	-15	25	-10
101	75	18	23

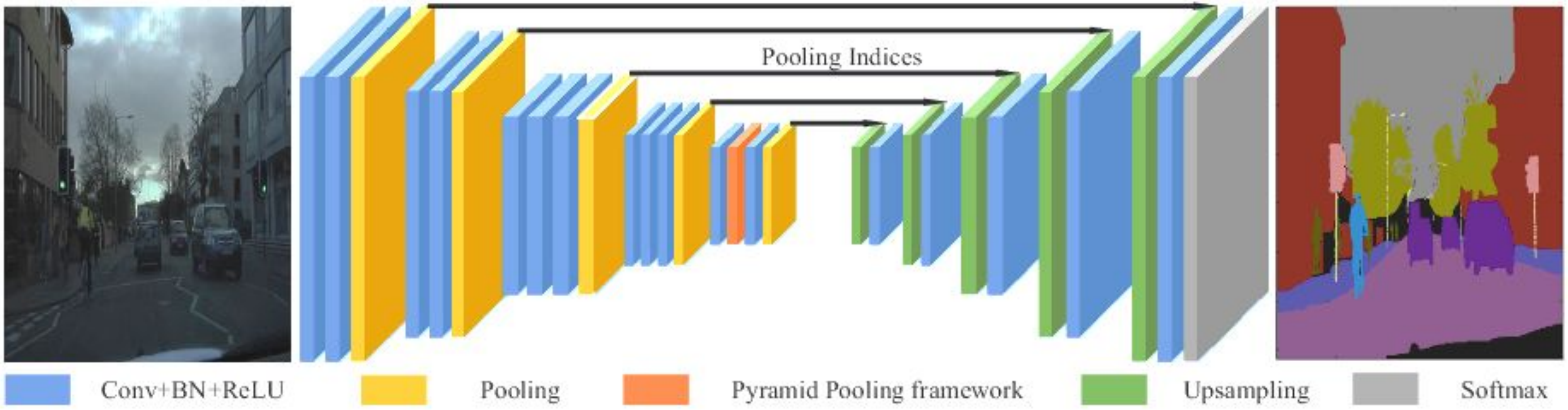


15	20	0	35
18	0	25	100
20	0	25	0
101	75	18	23

https://towardsdatascience.com/image-classifier-cats-vs-dogs-with-convolutional-neural-networks-cnns-and-google-colabs-4e9af21ae7a8

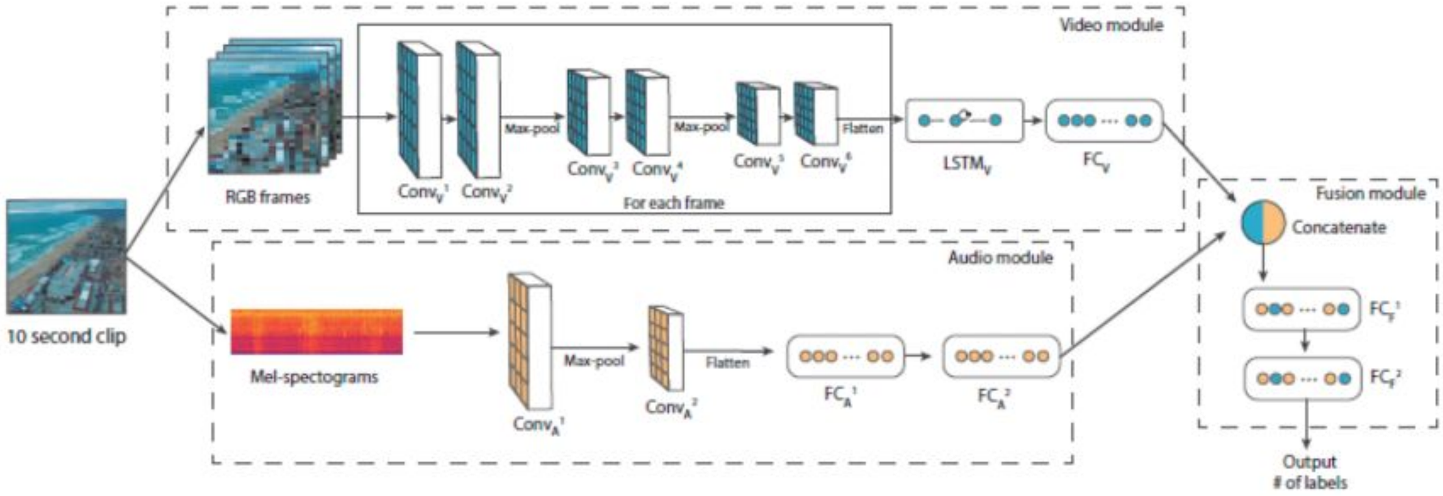
Some applications

Semantic segmentation (2D CNN)



[Tan et al., ICIG 2017]

Effects synchronization (2D CNN)



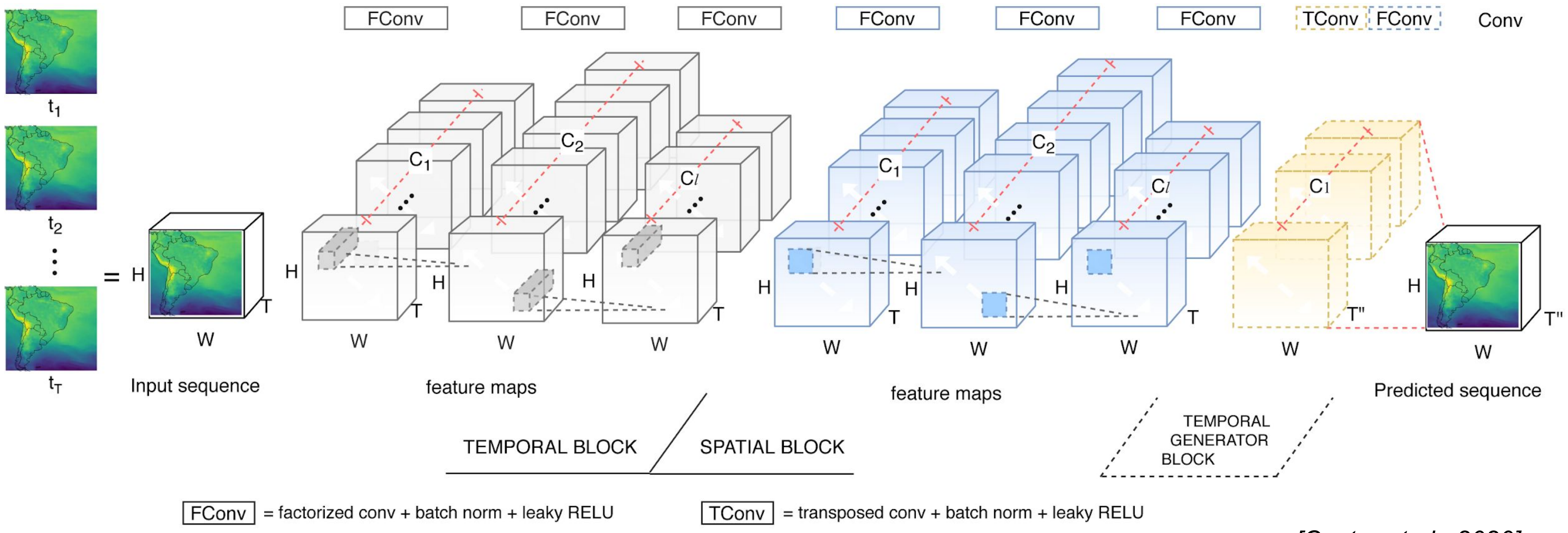
Bimodal architecture

[Abreu et al, IJCNN 2018]

Machine Translation (1D CNN)

[Gehring et al., 2017]

Spatiotemporal data forecasting ([3D CNN](https://github.com/MLRG-CEFET-RJ/stconvs2s))



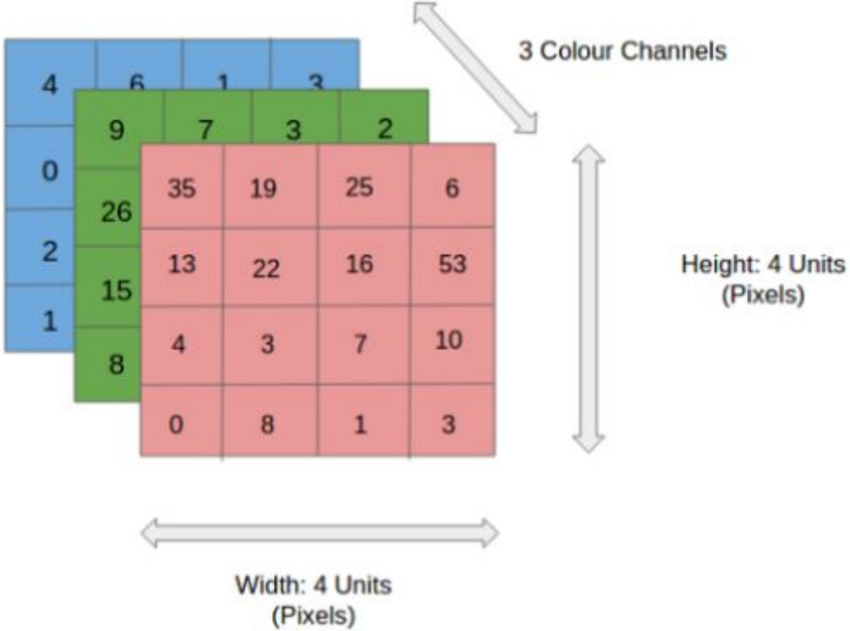
[Castro et al., 2020]

https://github.com/MLRG-CEFET-RJ/stconvs2s

Backup slides

Images are numbers

RGB image

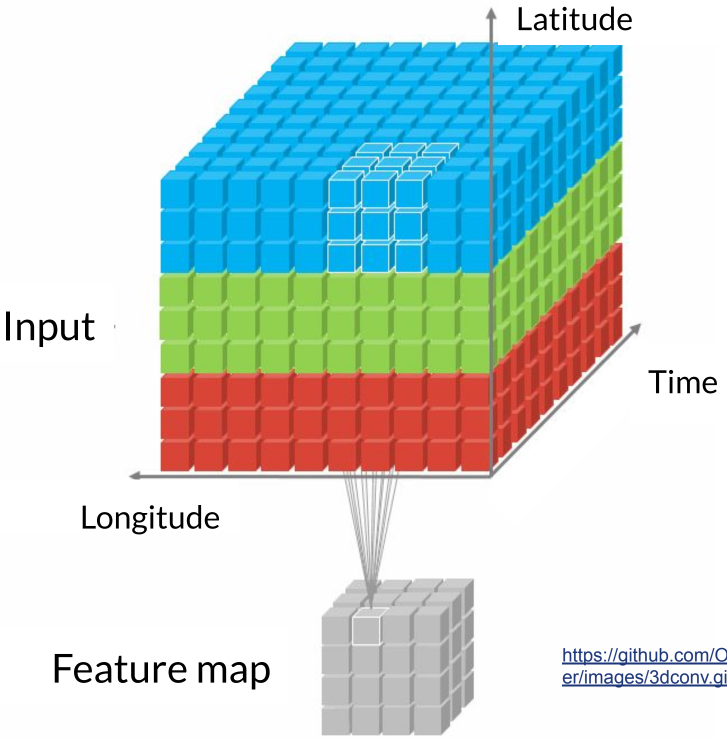


<< back

https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

3D CNN

kernel = 3 x 3 x 3
padding = 0
stride = 1



https://github.com/OValery16/Tutorial-about-3D-convolutional-network/blob/master/images/3dconv.gif

<< back