

# **Convolutional Neural Network**

**DSE 220**

# Topics

- Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

# Key Terminologies

- **Input Image**

- RGB image has 3 channels



What We See

08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08  
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00  
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65  
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91  
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80  
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50  
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70  
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21  
24 55 58 05 66 73 99 26 97 17 78 78 96 93 14 88 34 89 63 72  
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95  
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92  
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57  
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58  
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40  
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 99 66  
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69  
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36  
20 69 36 41 72 30 23 88 34 62 99 69 67 59 85 74 04 36 16  
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 37 05 54  
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48

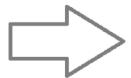
What Computers See

# Topics

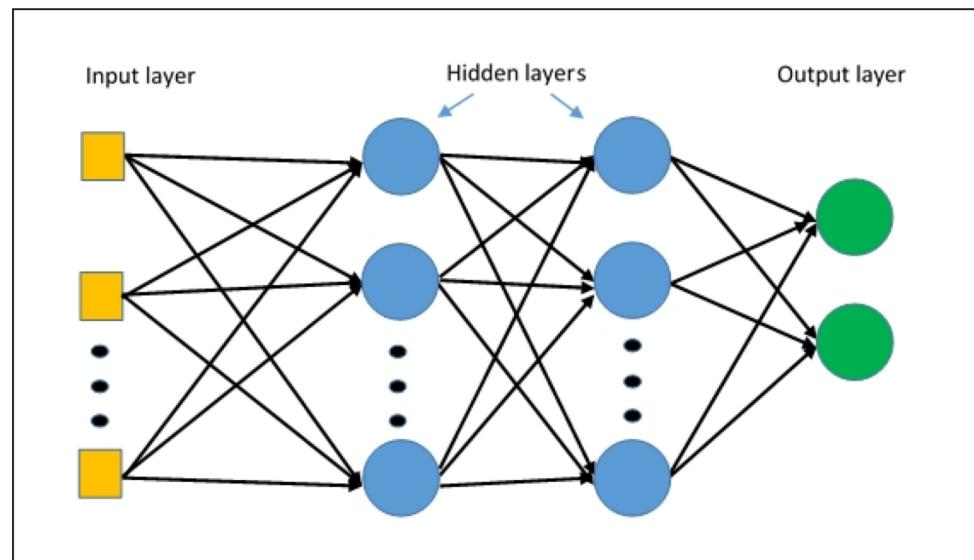
- Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

# Multilayer Perceptron

1	1	0
4	2	1
0	2	1



1
1
0
4
2
1
0
2
1



Flatten the image

Feed the image to MLP

## Why not Multilayer Perceptron ?

- If we flatten an image, we lose spatial information.
- Any pixel in image is related to its surrounding pixels.
- How to exploit the spatial information?

# Topics

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

# Key Terminologies

- Input Image
- **Kernel/Filter/Feature Extractor**
  - Filters are matrix of numbers.
  - They extract interesting features from the image.
  - The extracted features are called **Feature Map/Activation map**

1	0	1
0	1	0
1	0	1

# Topics

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

# Convolution - 2D

- Input Image
- Kernel/Filter/Feature Extractor
  - Filters are matrix of numbers.
  - They extract interesting features from the image.
  - Different filters extract different features.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input image

1	0	1
0	1	0
1	0	1

Kernel/Filter

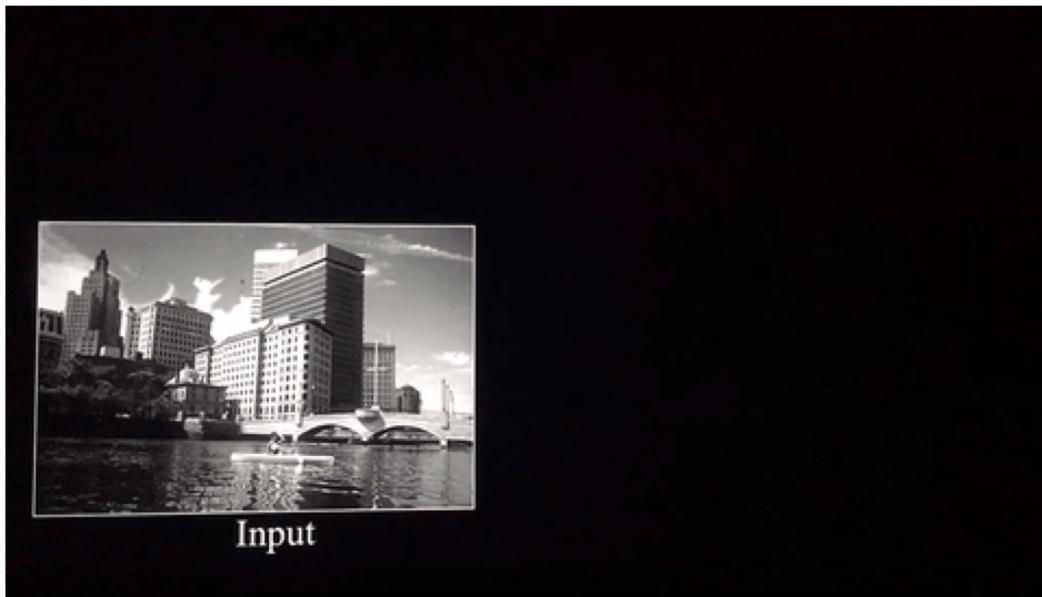
1 <small><math>\times 1</math></small>	1 <small><math>\times 0</math></small>	1 <small><math>\times 1</math></small>	0	0
0 <small><math>\times 0</math></small>	1 <small><math>\times 1</math></small>	1 <small><math>\times 0</math></small>	1	0
0 <small><math>\times 1</math></small>	0 <small><math>\times 0</math></small>	1 <small><math>\times 1</math></small>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

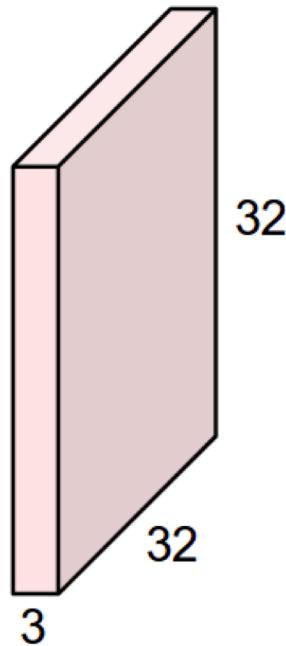
Convolved Feature

## Convolution - 2D



## Convolutional Neural Network(CNN): Architecture

32x32x3 image

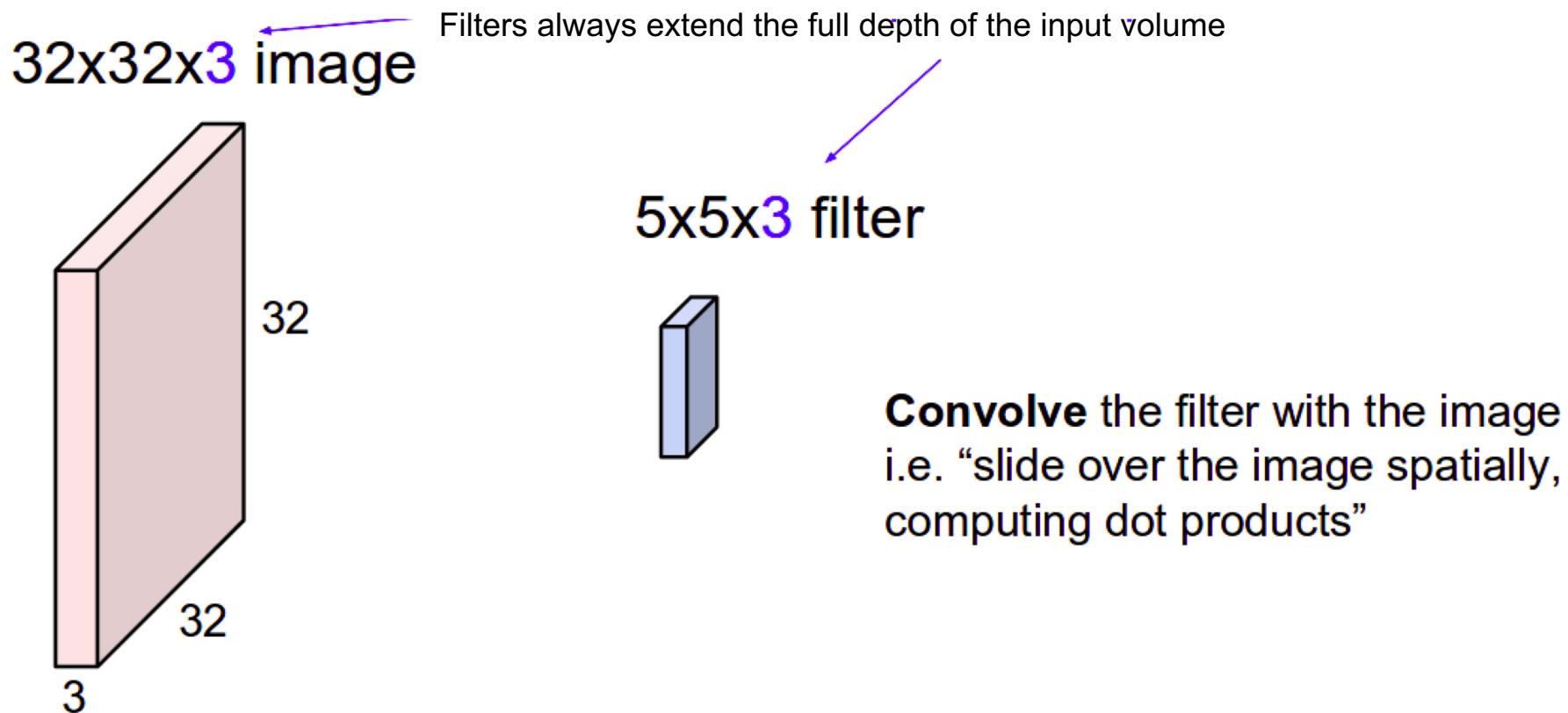


5x5x3 filter

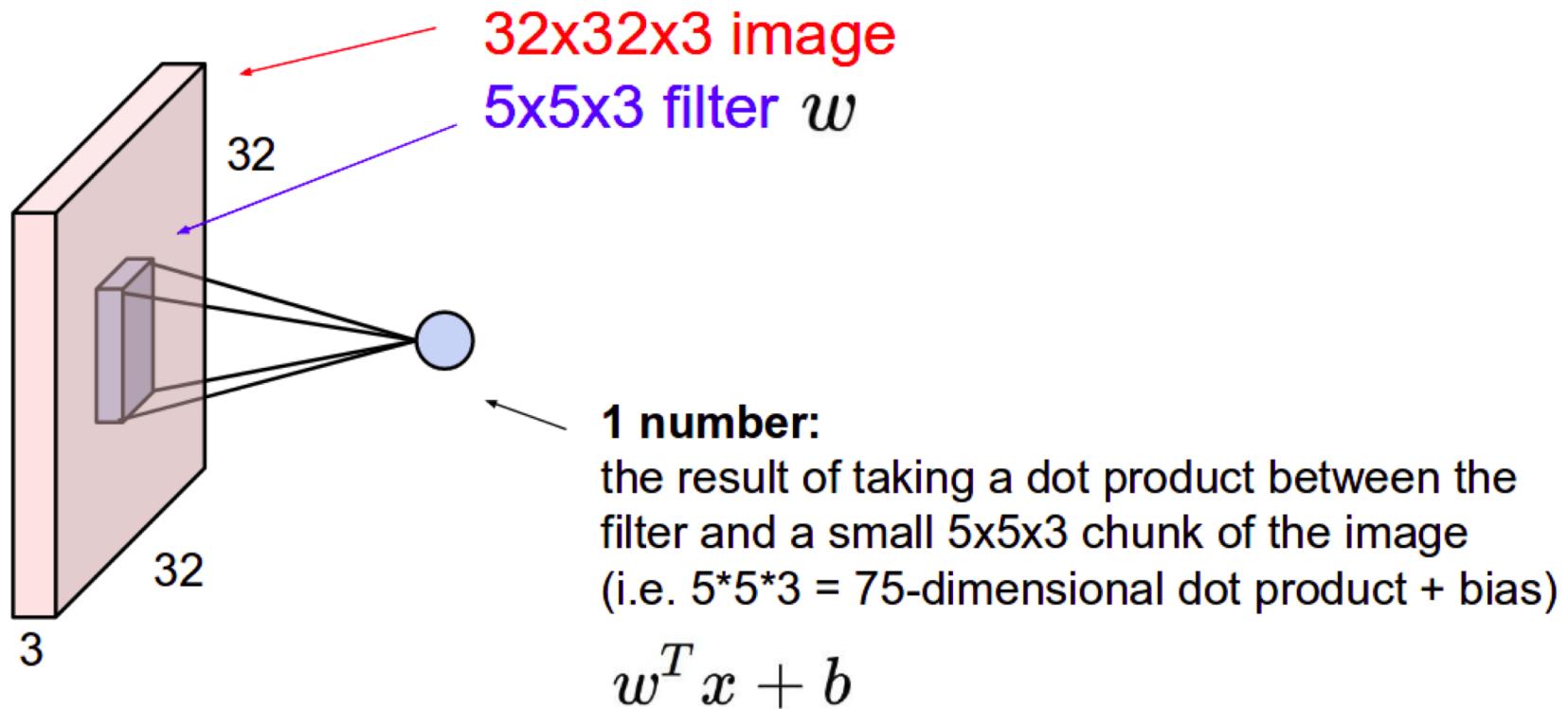


**Convolve** the filter with the image  
i.e. “slide over the image spatially,  
computing dot products”

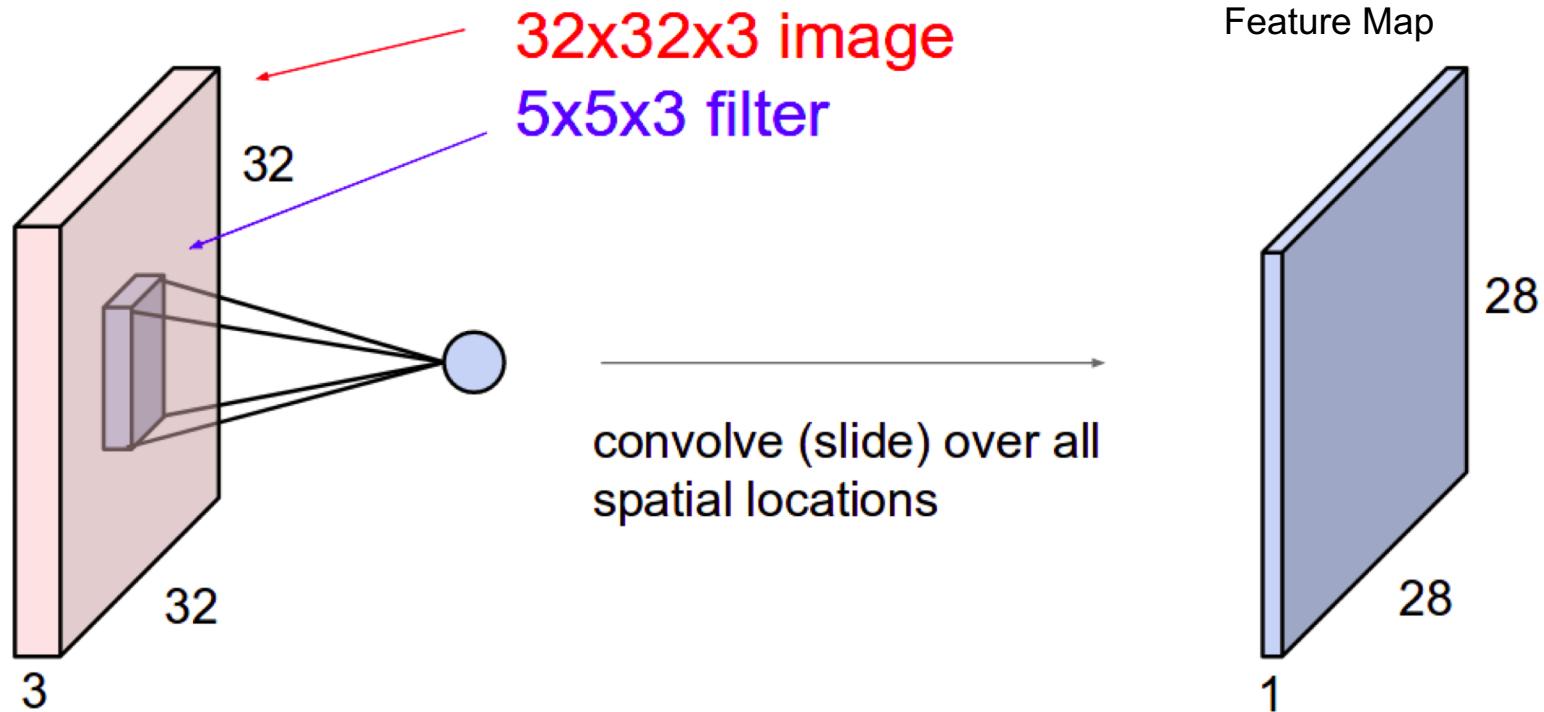
## CNN: Architecture



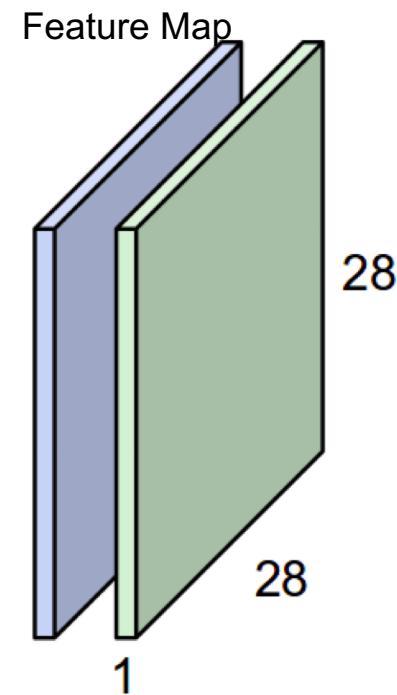
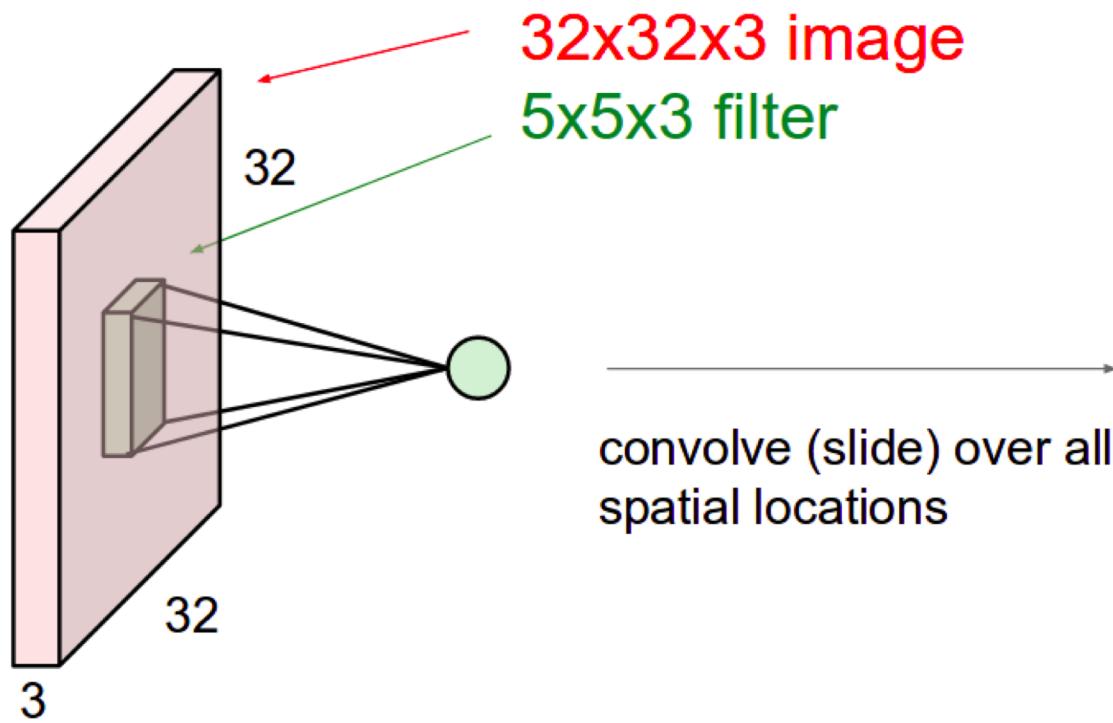
## CNN: Architecture



## CNN: Architecture

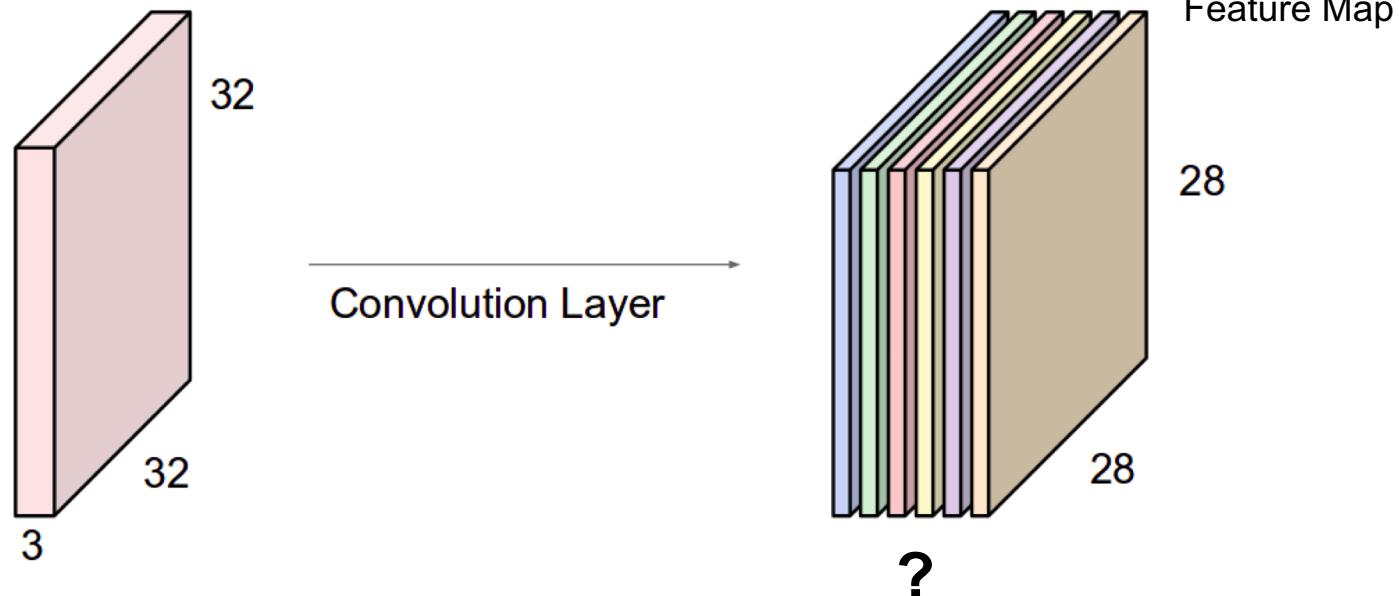


## CNN: Architecture



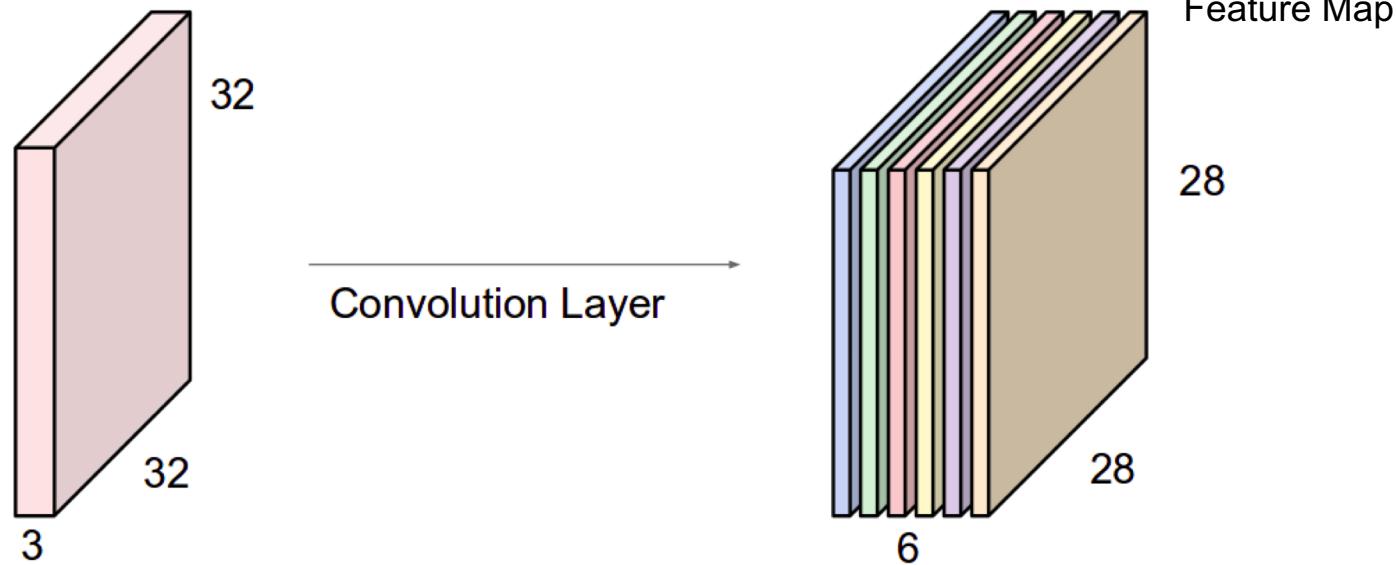
## CNN: Architecture

For eg, if we have 6 5x5x3 filters, how many feature maps do we get?



# CNN: Architecture

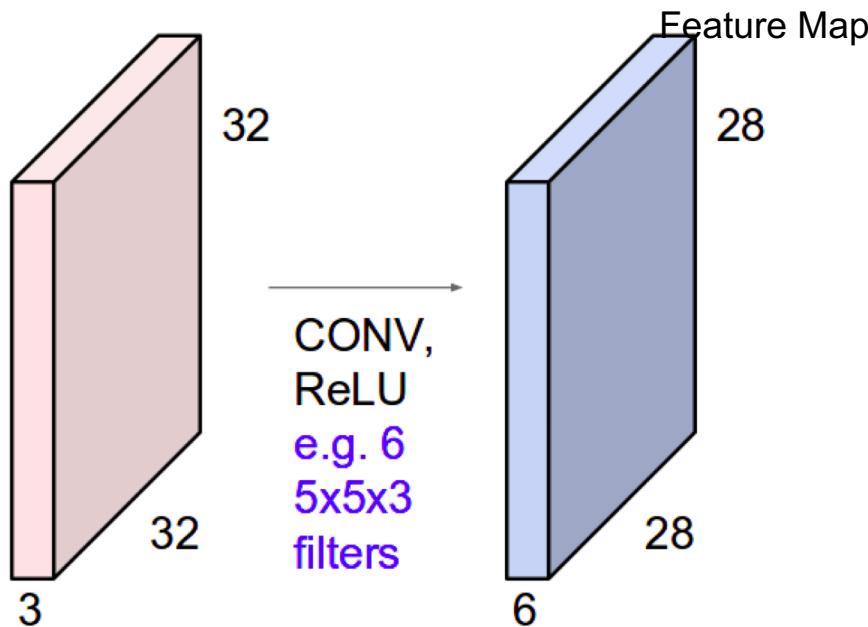
We get 6 feature maps!!!



We stack these up to get a “new image” of size 28x28x6!

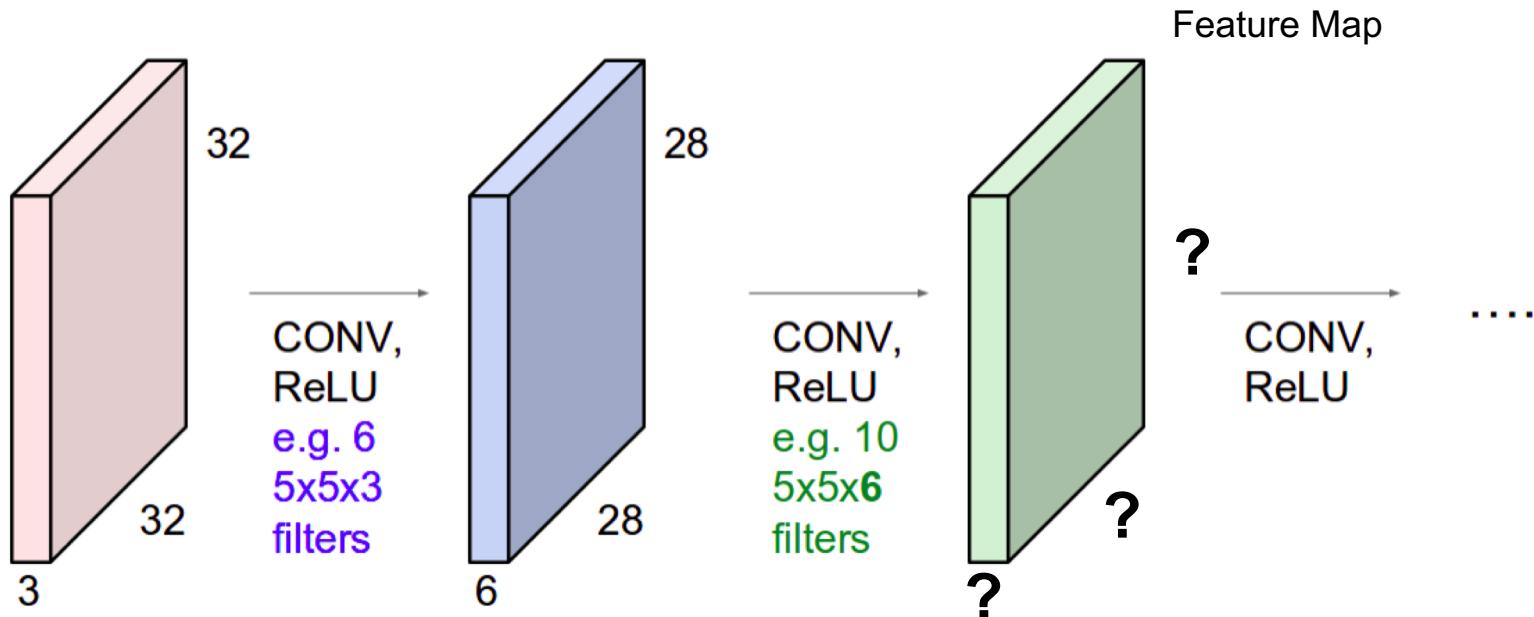
## CNN: Architecture

ConvNet is a sequence of Convolutional Layers, separated by activation functions



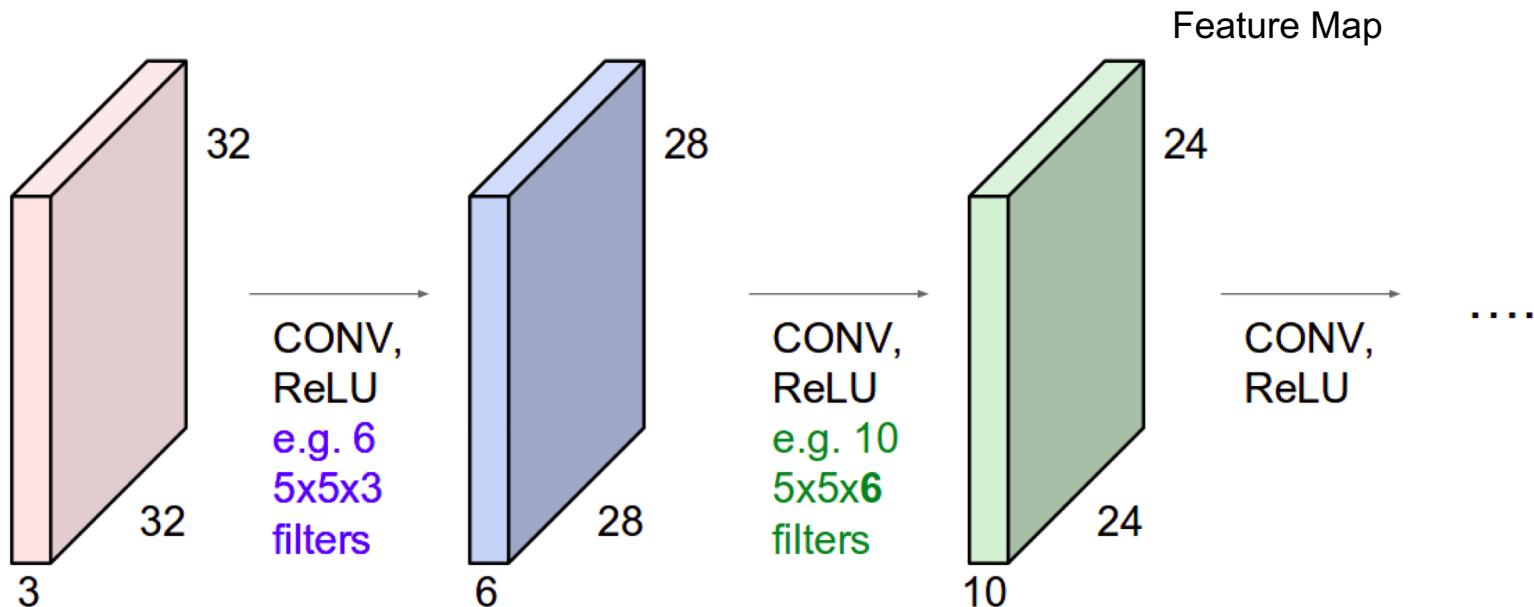
## CNN: Architecture

ConvNet is a sequence of Convolutional Layers, separated by activation functions



## CNN: Architecture

ConvNet is a sequence of Convolutional Layers, separated by activation functions



# Convolutional Neural Network : Key Idea

- The idea of convolution of a filter over image was always there.
- Earlier, these filters were manually designed.
- With CNN, these filters are learnt.

-1	0	1
-1	0	1
-1	0	1

Filter: Detects Vertical Edge

-1	-1	-1
0	0	0
1	1	1

Filter: Detects Horizontal Edge

# Convolutional Neural Network : Key Idea

- The idea of convolution of a filter over image was always there.
- Earlier, these filters were manually designed.
- With CNN, these filters are learnt.

-1	0	1
-1	0	1
-1	0	1

Filter: Detects Vertical Edge

**lena.gif**



**vertical edges**



**horizontal edges**



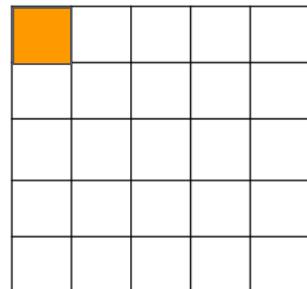
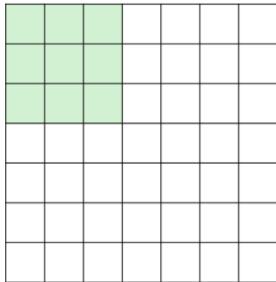
-1	-1	-1
0	0	0
1	1	1

Filter: Detects Horizontal Edge

# Topics

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

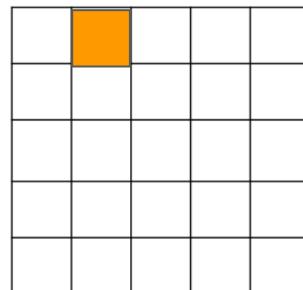
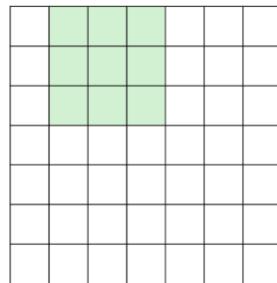
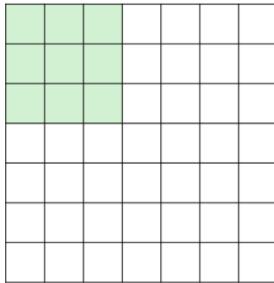
## Strides 1



Generated feature map

Input Image Size: 7x7

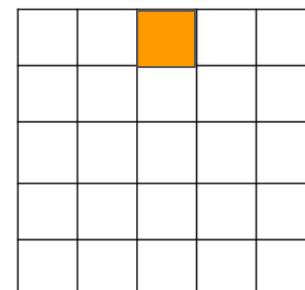
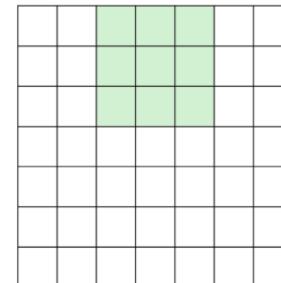
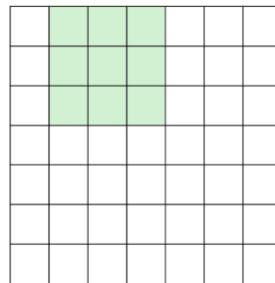
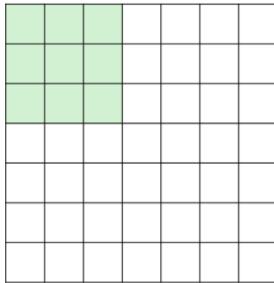
## Strides 1



Generated feature map

Input Image Size: 7x7

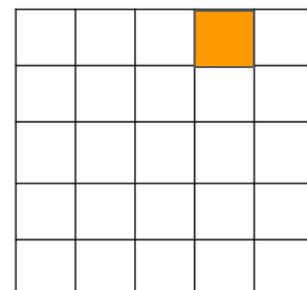
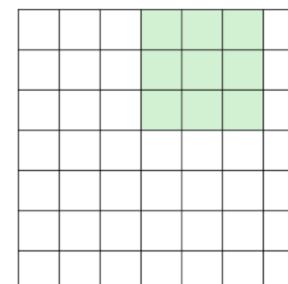
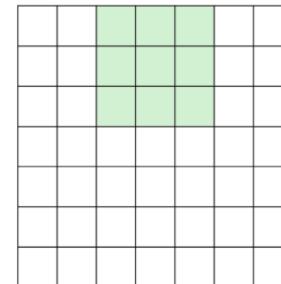
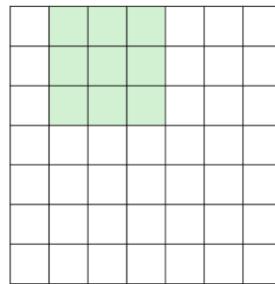
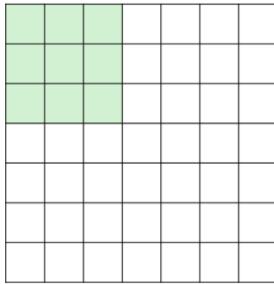
## Strides 1



Generated feature map

Input Image Size: 7x7

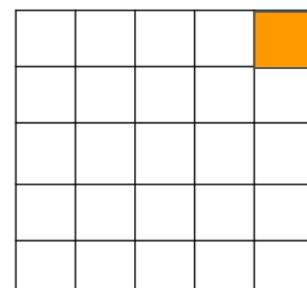
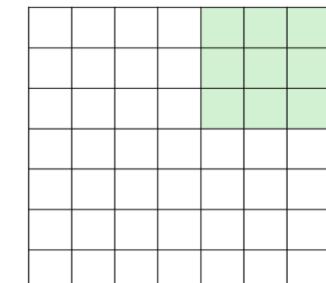
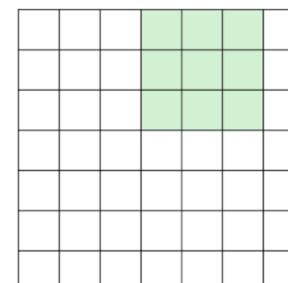
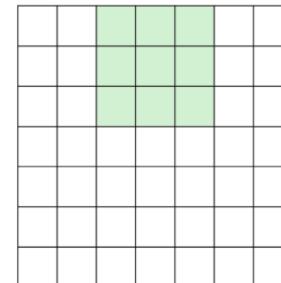
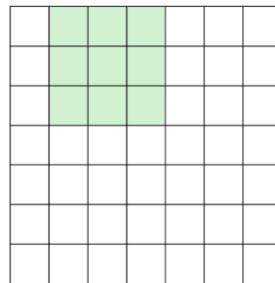
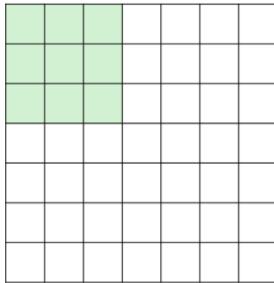
## Strides 1



Generated feature map

Input Image Size: 7x7

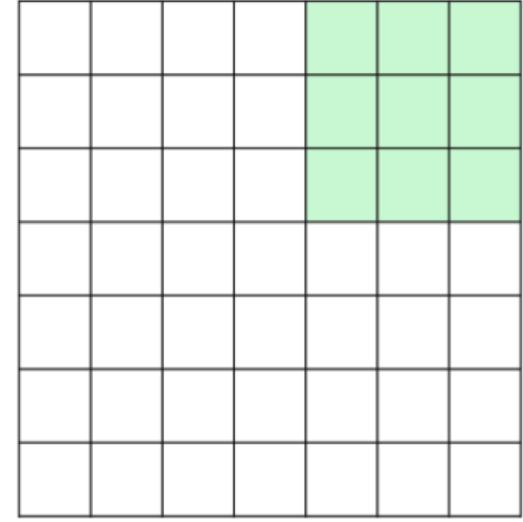
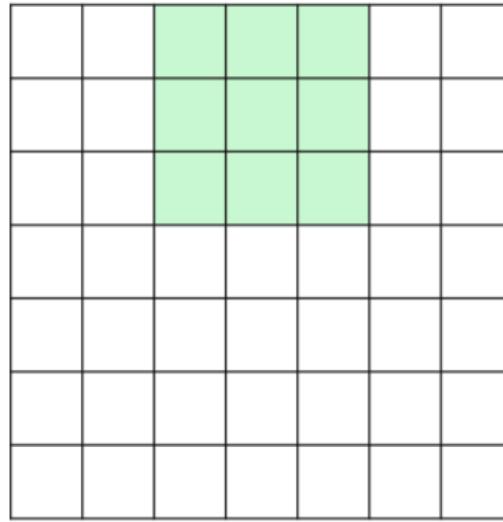
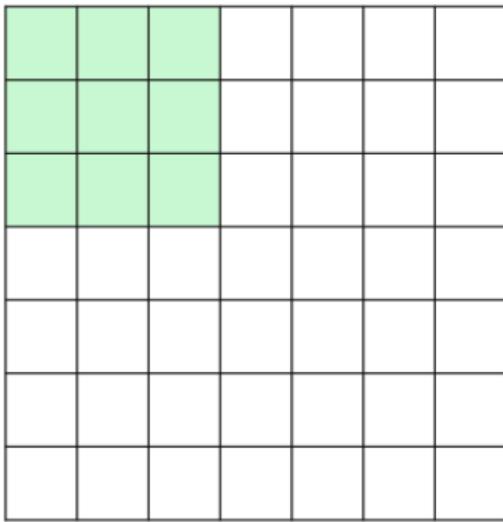
## Strides 1



Generated feature map

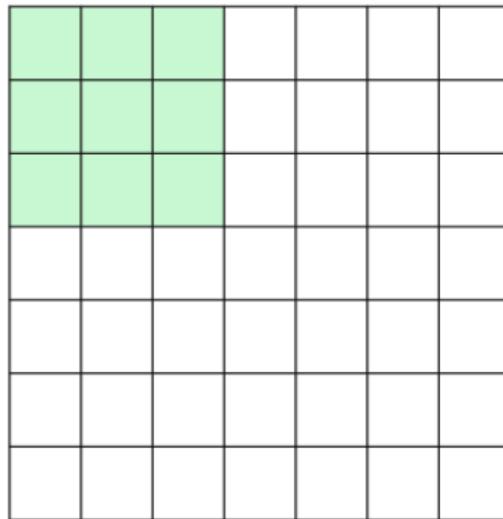
Input Image Size: 7x7  
Feature Map Size: 5x5

## Strides 2



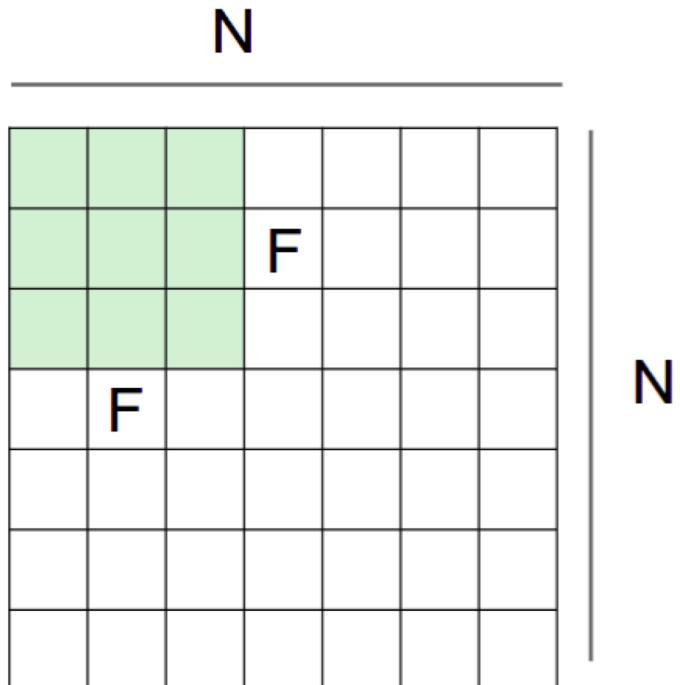
- Feature Map is of size: 3x3

## Strides 3



- Strided(3) convolution not possible on image of size 7x7

## Feature map dimensions



Output size:

$$(N - F) / \text{stride} + 1$$

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

$$\text{stride } 1 \Rightarrow (7 - 3)/1 + 1 = 5$$

$$\text{stride } 2 \Rightarrow (7 - 3)/2 + 1 = 3$$

$$\text{stride } 3 \Rightarrow (7 - 3)/3 + 1 = 2.33$$

# Topics

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

## Padding

In practice: Common to zero pad the border

0	0	0	0	0	0			
0								
0								
0								
0								

e.g. input 7x7

**3x3 filter, applied with **stride 1****

**pad with 1 pixel border => what is the output?**

**7x7 output!**

in general, common to see CONV layers with stride 1, filters of size FxF, and zero-padding with  $(F-1)/2$ . (will preserve size spatially)

e.g.  $F = 3 \Rightarrow$  zero pad with 1

$F = 5 \Rightarrow$  zero pad with 2

$F = 7 \Rightarrow$  zero pad with 3

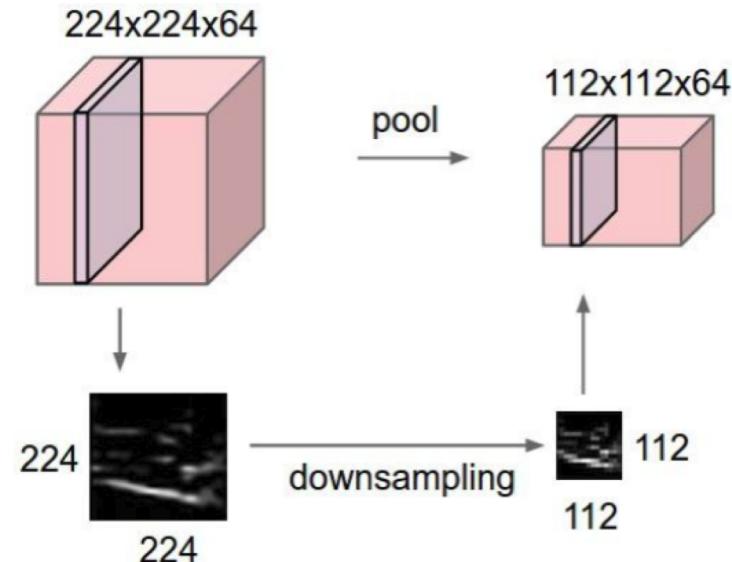
# Topics

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

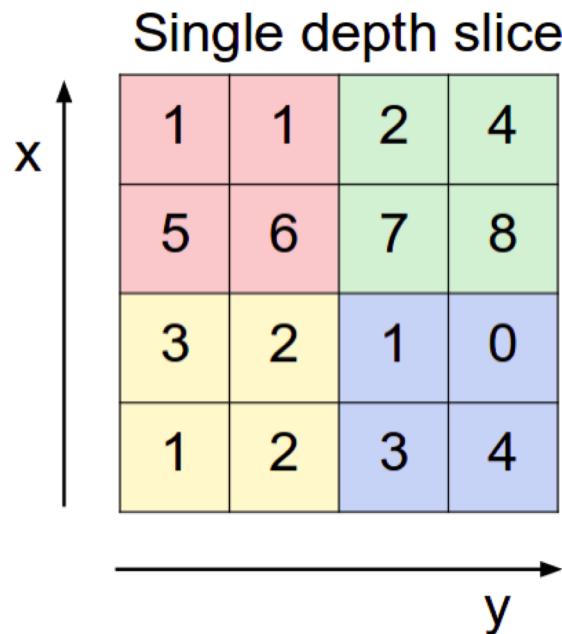
# Subsampling: Pooling

Pooling layer

- Makes representation smaller and more manageable
- Operates over each feature map independently.
- Has no activation function.



## Max-pooling



max pool with 2x2 filters  
and stride 2

6	8
3	4

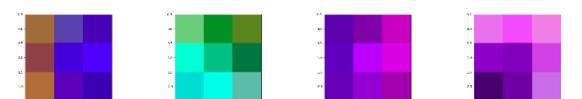
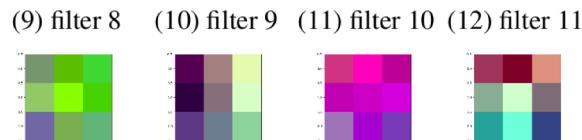
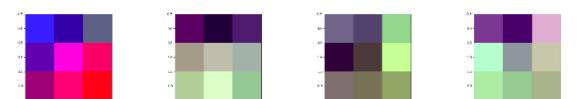
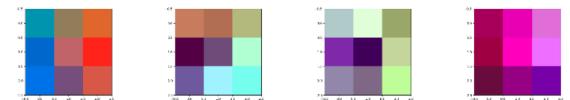
# Topics

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - **Feature Visualization**
  - What happens after Convolution
  - Overview
- Latest trends in CNN

# CNN: Visualization of filters

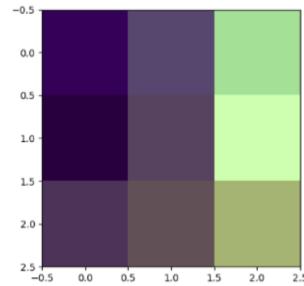


Input image

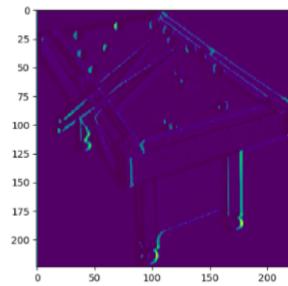


Feature map from first convolution layer

# CNN: Visualization of filters - Conv 1 layer

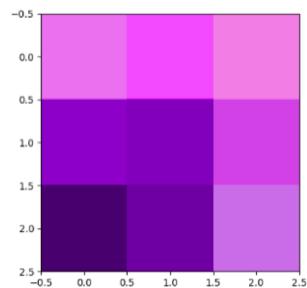


(1) Filter 6 (blue line)

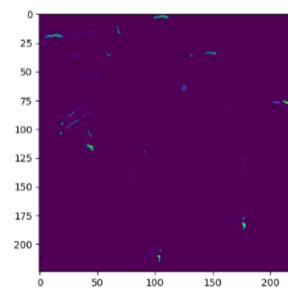


(2) Activation/Features from  
Filter 6 (dark edges)

# CNN: Visualization of filters - Conv 1 layer

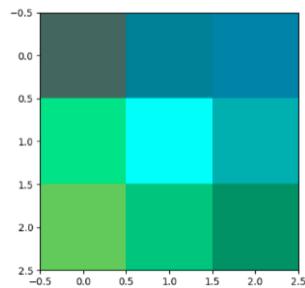


(3) Filter 27 (dark corner)

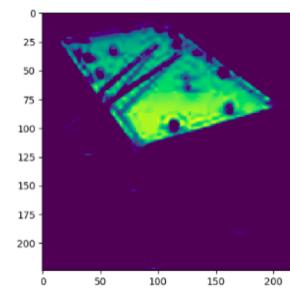


(4) Activation/Features from  
Filter 27

# CNN: Visualization of filters - Conv 1 layer

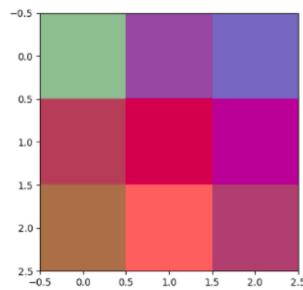


(5) Filter 15 (green surface)

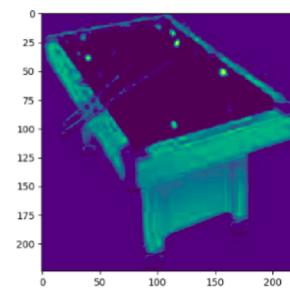


(6) Activation/Features from  
Filter 15

# CNN: Visualization of filters - Conv 1 layer



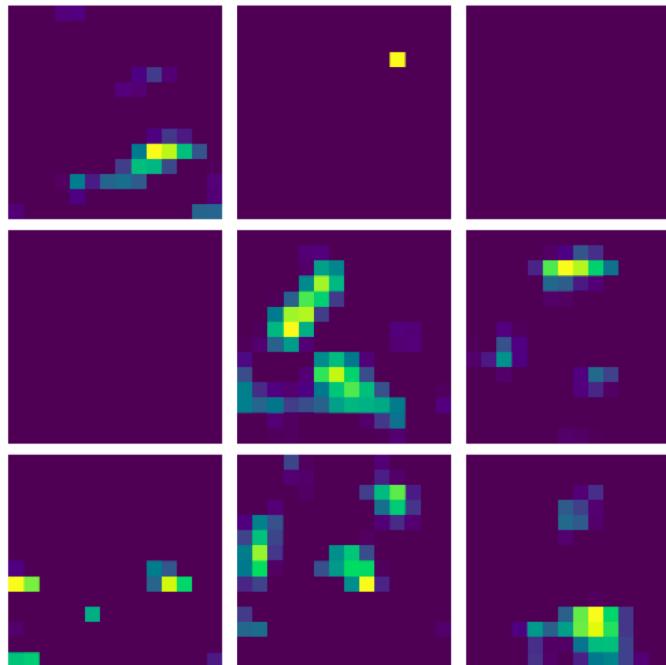
(7) Filter 47 (yellow surface)



(8) Activation/Features from  
Filter 47

## Deeper layers

Feature Maps from deeper layers are very difficult to interpret.



# Deep Visualization Toolbox

<https://www.youtube.com/watch?v=AgkfIQ4IGaM>

# Topics

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - **What happens after Convolution**
  - Overview
- Latest trends in CNN

# What happens after Convolution & Activation ?

- Feature extraction is done by Convolution
  - The extracted features are 3D (Feature Maps)

# What happens after Convolution & Activation ?

- Feature extraction is done by Convolution
  - The extracted features are 3D (Feature Maps)
- How to do classification using feature maps?

# What happens after Convolution & Activation ?

- Feature extraction is done by Convolution
  - The extracted features are 3D (Feature Maps)
- How to do classification using feature maps?
- For classification, we need to add fully connected layers (MLP)
  - After last Convolutional Feature Map
  - Flatten the image - i.e. convert the 3D image to a 1D vector
  - Add Dense/Fully Connected layers.
  - Add classification head (output layer)

# Topics

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - **Overview**
- Latest trends in CNN

# CNN Architecture: Overview

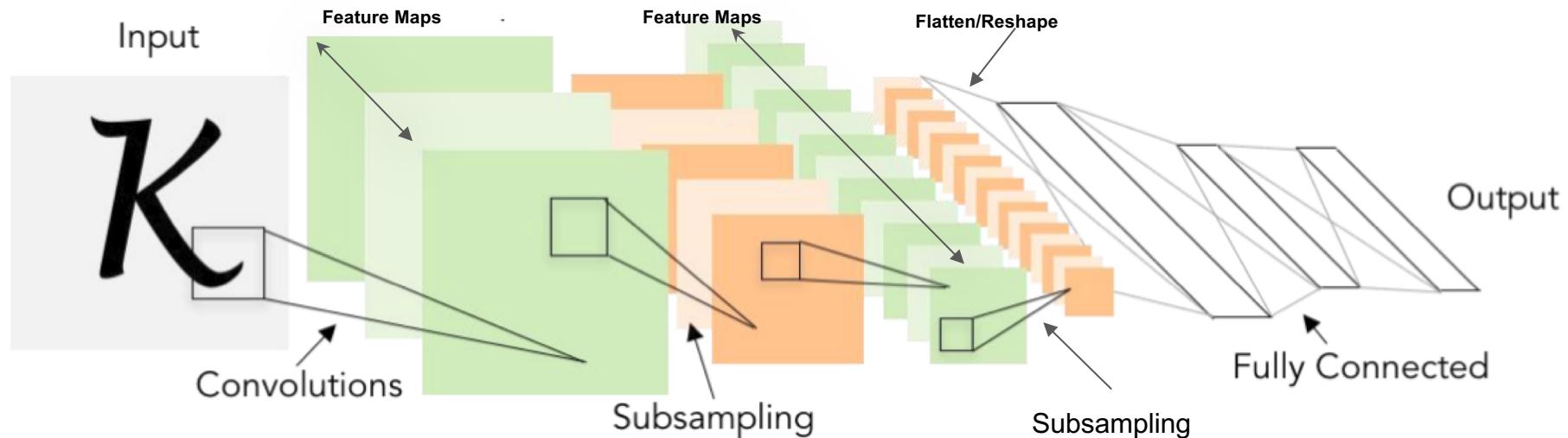
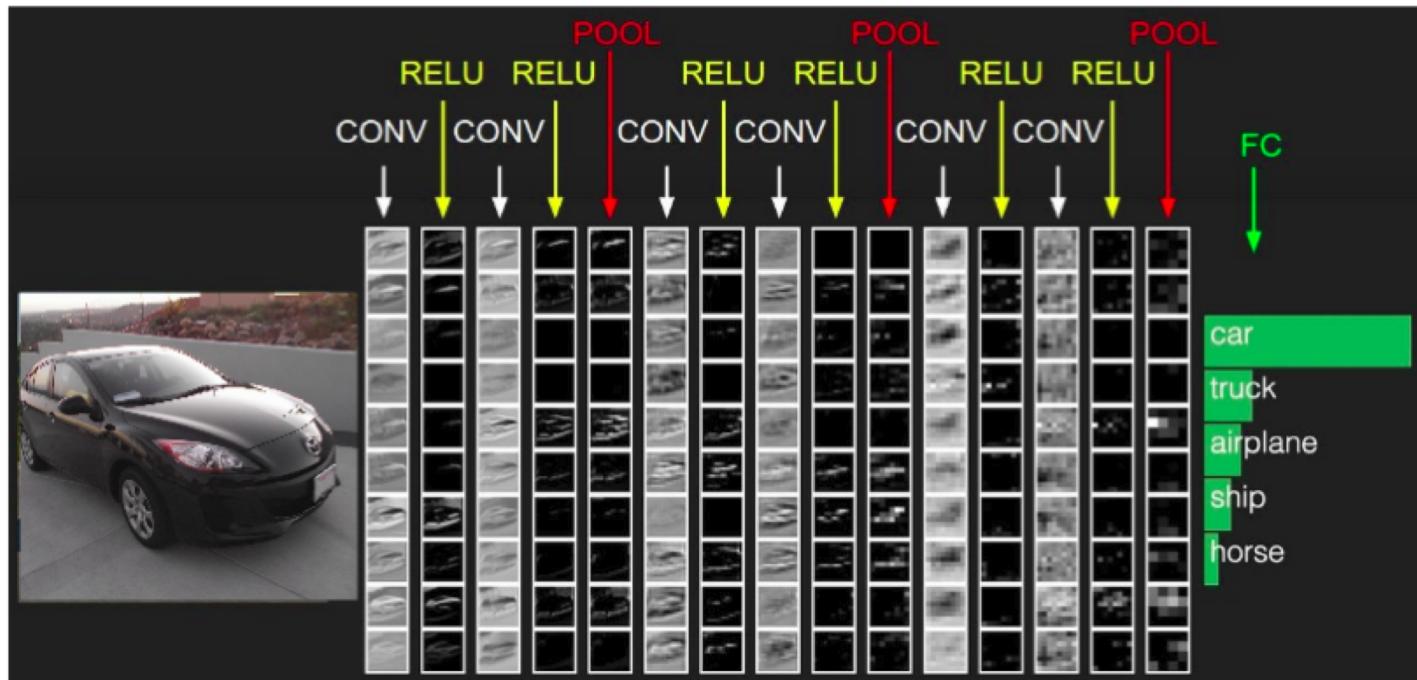


Illustration of LeCun et al. 1998 from CS231n 2017 Lecture 1

# Model Summary in Keras

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 26, 26, 32)	320
conv2d_2 (Conv2D)	(None, 24, 24, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 64)	0
dropout_1 (Dropout)	(None, 12, 12, 64)	0
flatten_1 (Flatten)	(None, 9216)	0
dense_1 (Dense)	(None, 128)	1179776
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 10)	1290
<hr/>		
Total params: 1,199,882		
Trainable params: 1,199,882		
Non-trainable params: 0		
<hr/>		

# Overall Architecture



# Demo

Demo

# Topics

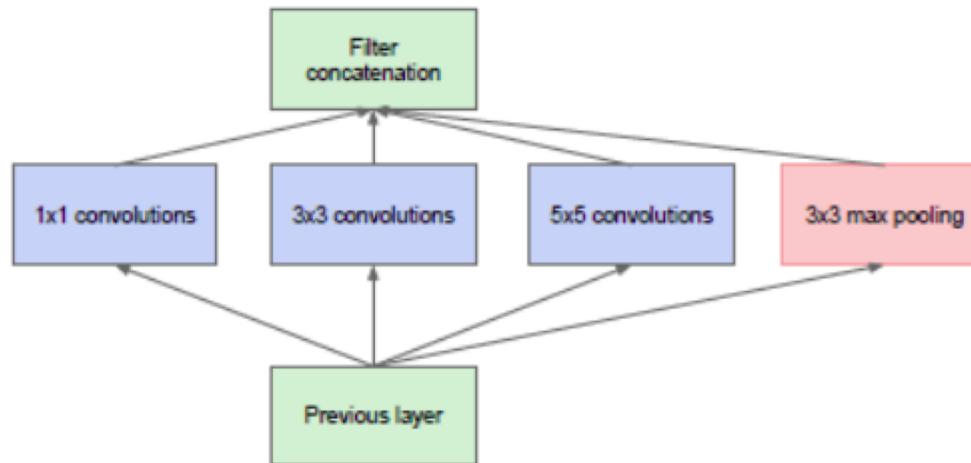
- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

## Latest trends

- In 2012, Neural Networks were typically ~ 8 layer deep.
- Now networks are more than 100 layers deep.
- Most commonly used networks are:
  - Google: GoogLeNet
  - Microsoft: ResNet

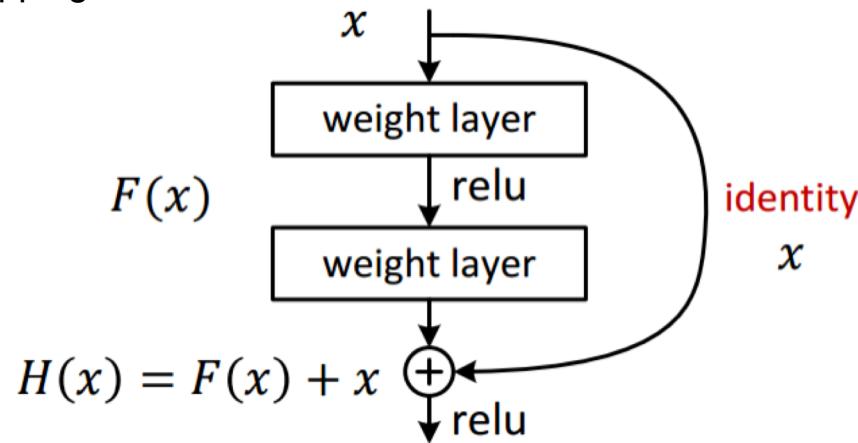
## GoogLe Net : Inception Module

- Let the network decide the best filter size by itself.



# Microsoft : Residual Net (ResNet)

- Enables deeper network
- Backpropagation is easier, and deals with Vanishing gradient problem
- Learns identity mapping



# Network Examples

