Summer of Code

Artificial Intelligence

(Machine Learning & Deep Learning)

Instructor
Wajahat Ullah
- *Research Assistant* (DIP Lab)

Duration
03 Months
(September – November)

Day 01 – Deep Learning
(Convolutional Neural Networks)

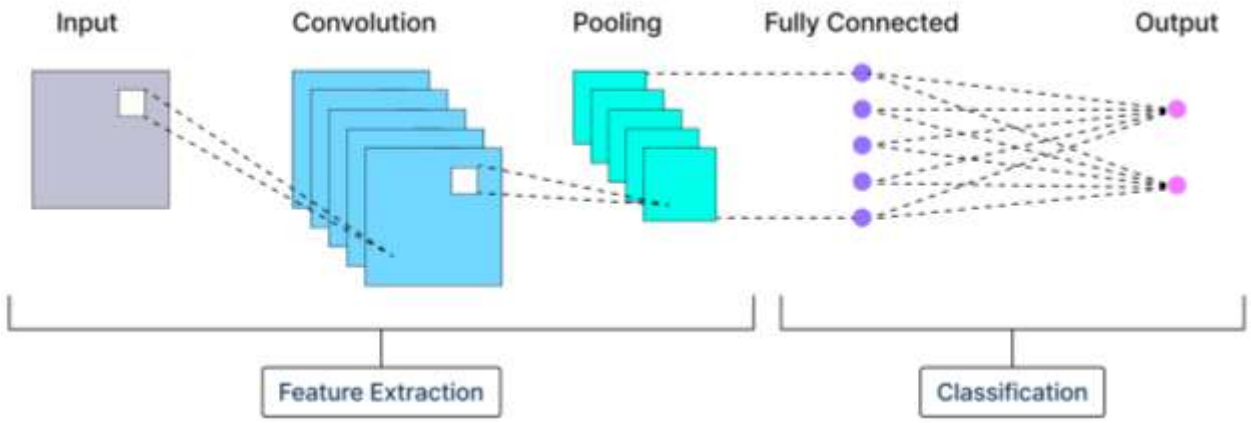
Objectives:

- Convolution
- Pooling
- Convolutional Neural Networks

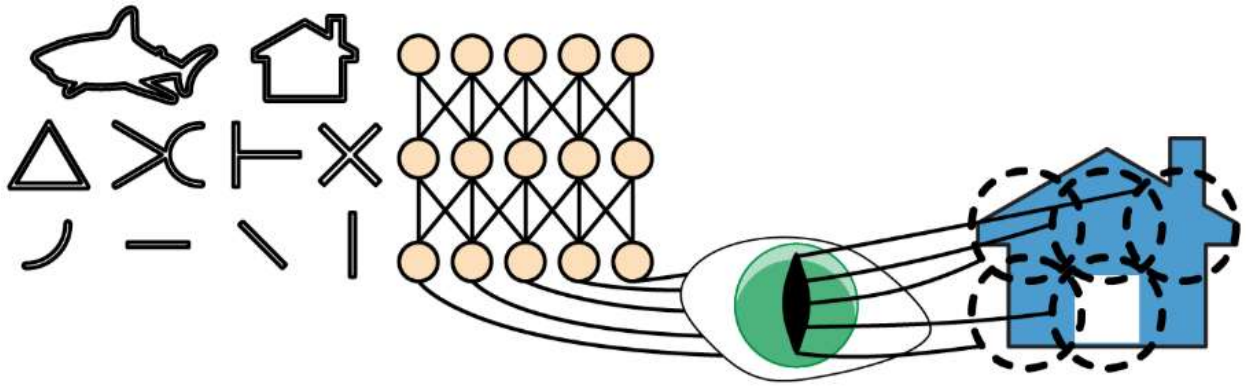
Convolutional Neural Networks

- A class of neural networks primarily used for analyzing visual data such as images and videos.
- Learn spatial hierarchies of features through convolutional operations.
- The fundamental unit of such networks is a small bundle of computation called an artificial neuron.
- Modeled after the visual cortex of the human brain as neurons respond to specific regions of the visual field.

CNNs

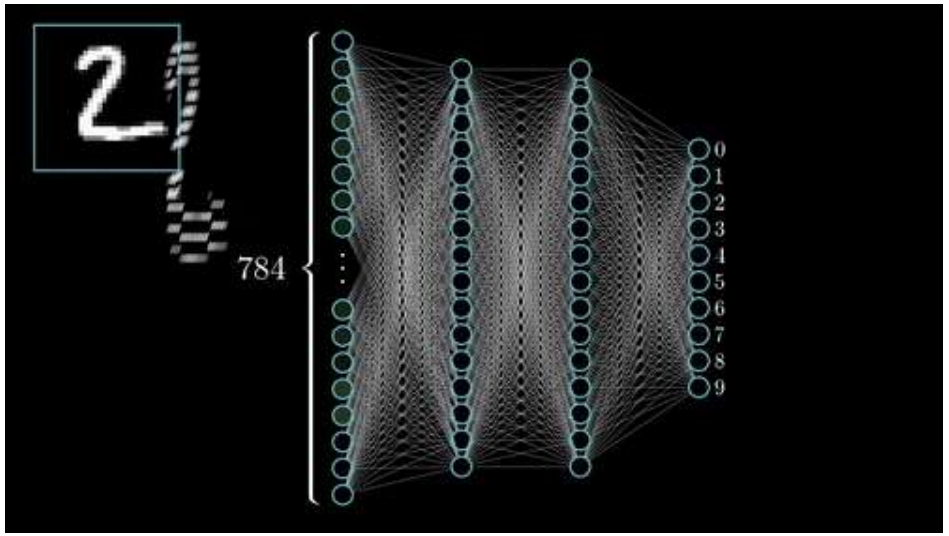


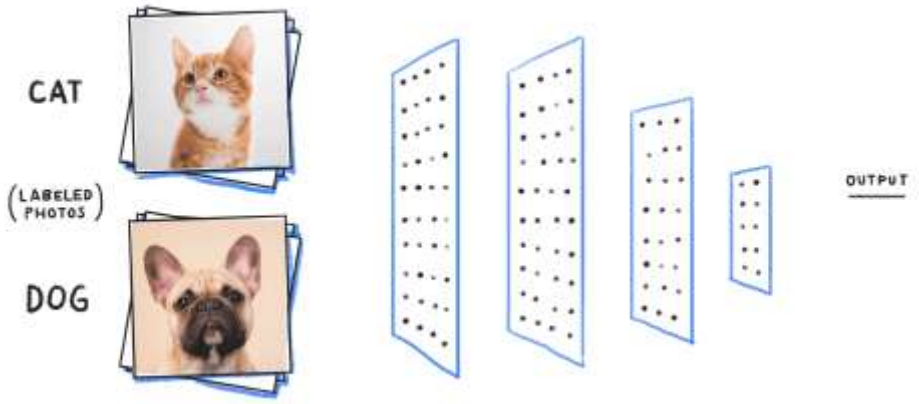
Human Visual Cortex



Convolutional vs Fully Connected Layers

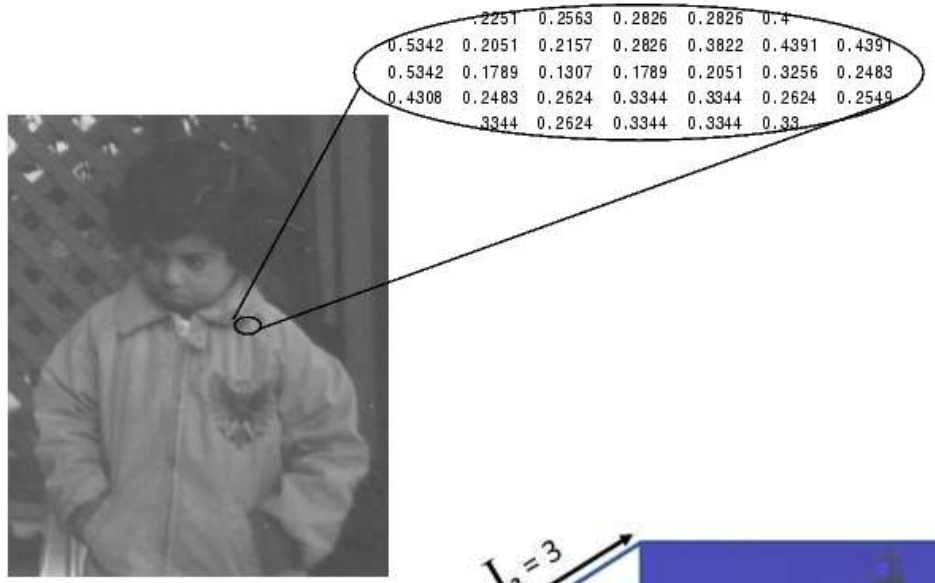
- Each neuron connected to a small range of input, instead of all the inputs.
- Performs the convolution operation to extract features.
- It has fewer parameters compared to fully connected layers because of weight sharing.
- Primarily used for feature extraction

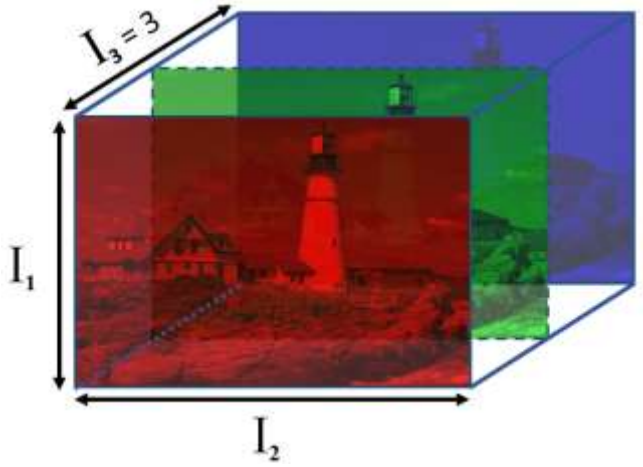


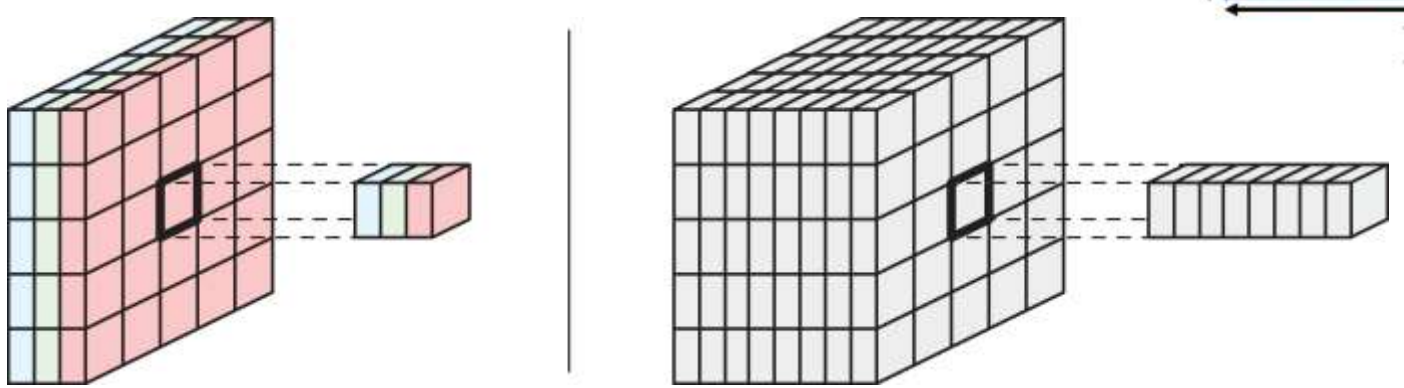


Images and Tensors

- A digital image is a grid of pixel values, each pixel containing intensity information.
- Grayscale images are represented as a 2D matrix of pixels (Height x Width), while color images have channels as well (Height, Width, Channels).
- Tensors are generalization of matrices to higher dimensions.





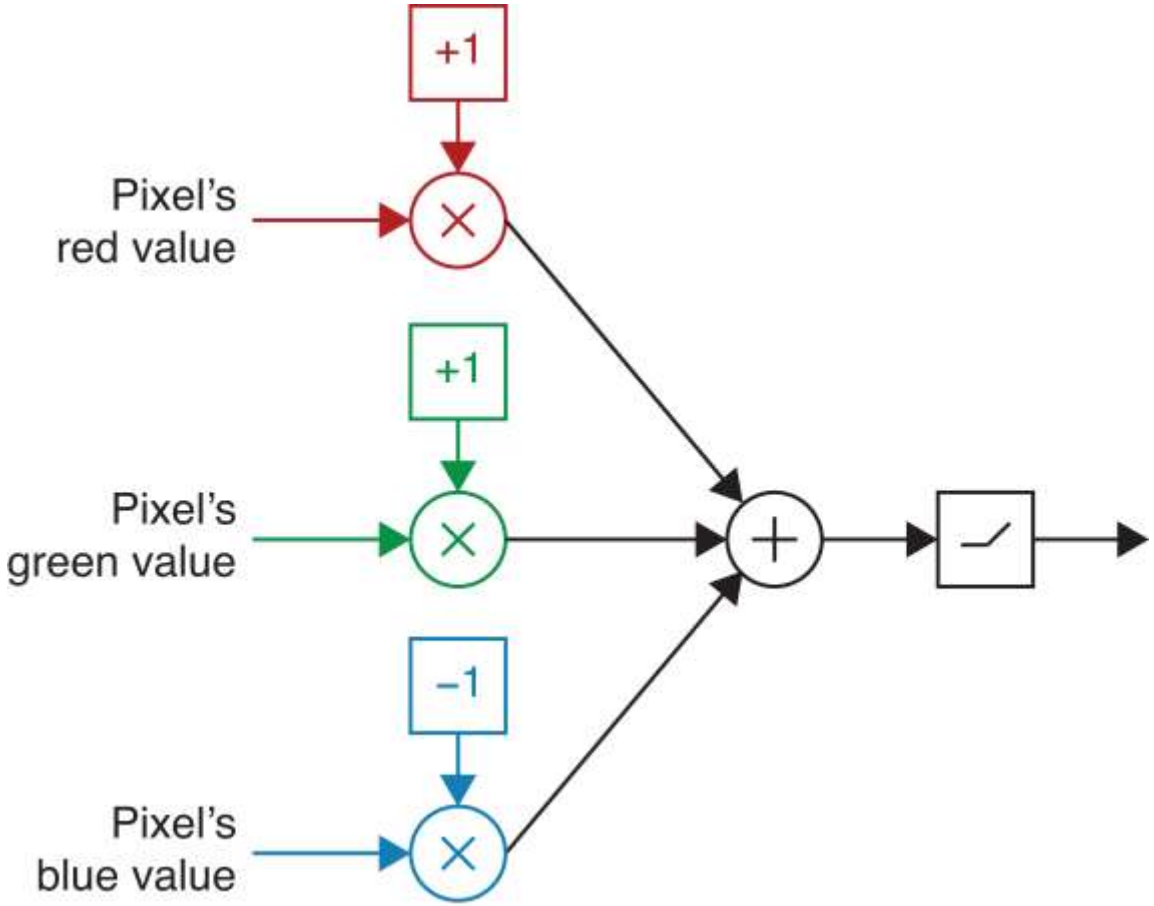


Convolution

Let’s say we want to detect the intensity of yellow color in each pixel

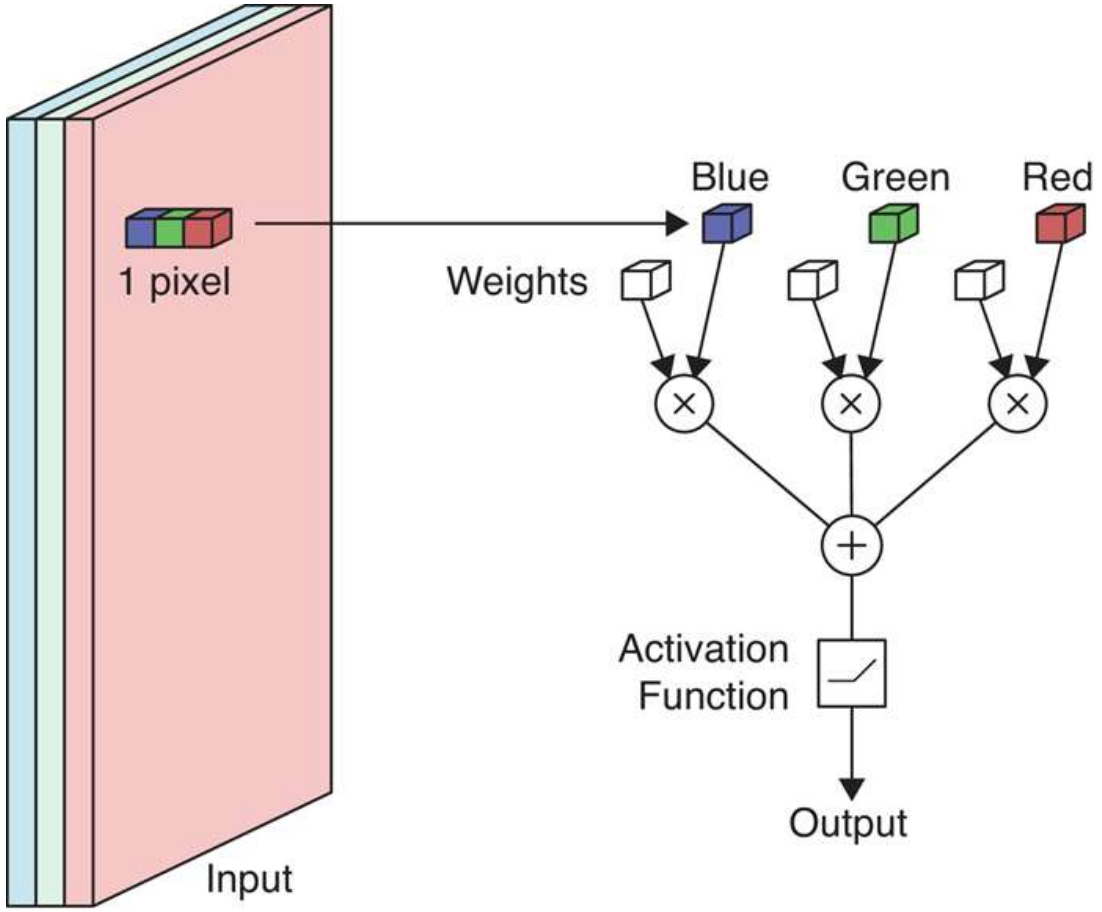
Convolution

We can create a yellow detector as an artificial neuron like so



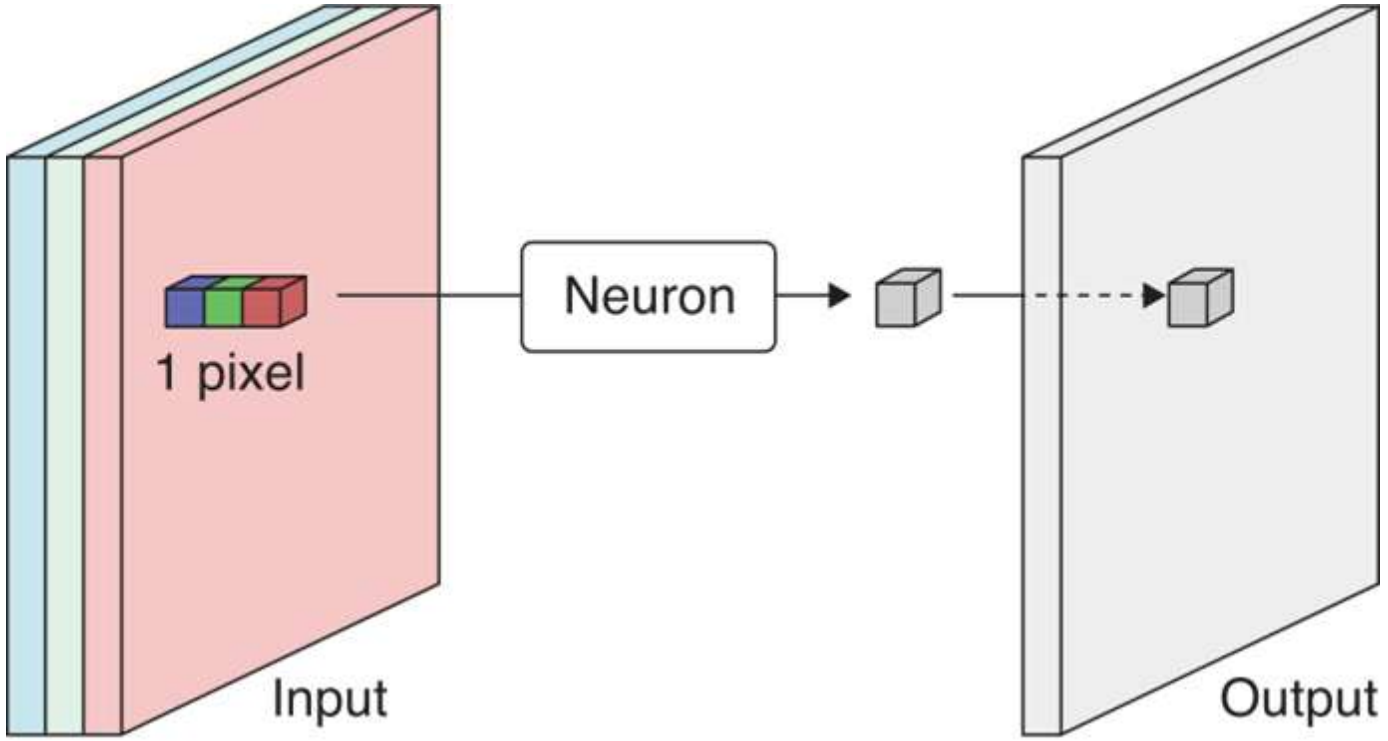
Convolution

We can apply the yellow detector to every pixel in the color input image



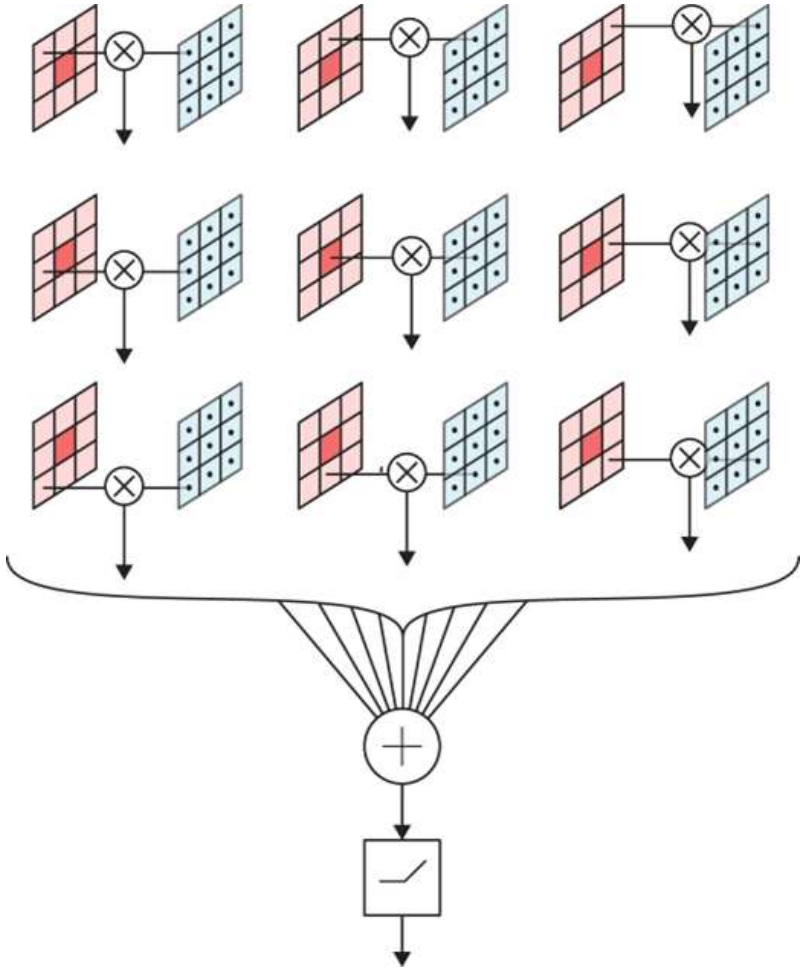
Convolution

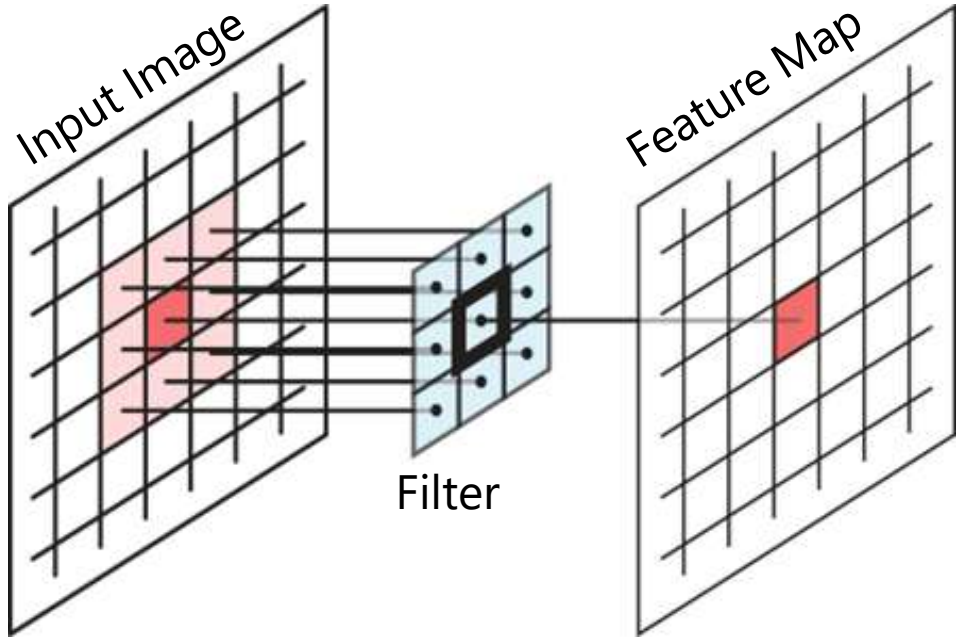
As a result, we get a grayscale image where each pixel corresponds to the intensity of yellow color in the input image



Convolution

Instead of processing a single pixel at a time, we can process a small region around the pixel





Convolution as Feature Extractor

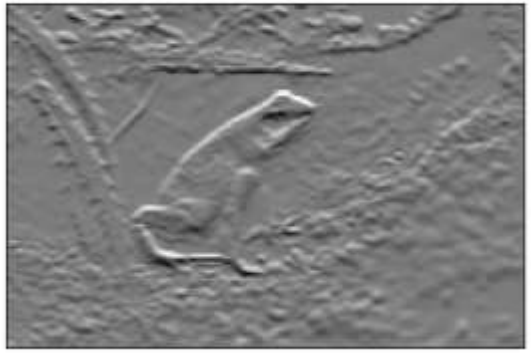
We sweep the filter across the entire image to detect these different patterns as

Blur

1	1	1
1	1	1
1	1	1

Horizontal Edges

1	1	1
0	0	0
–1	–1	–1

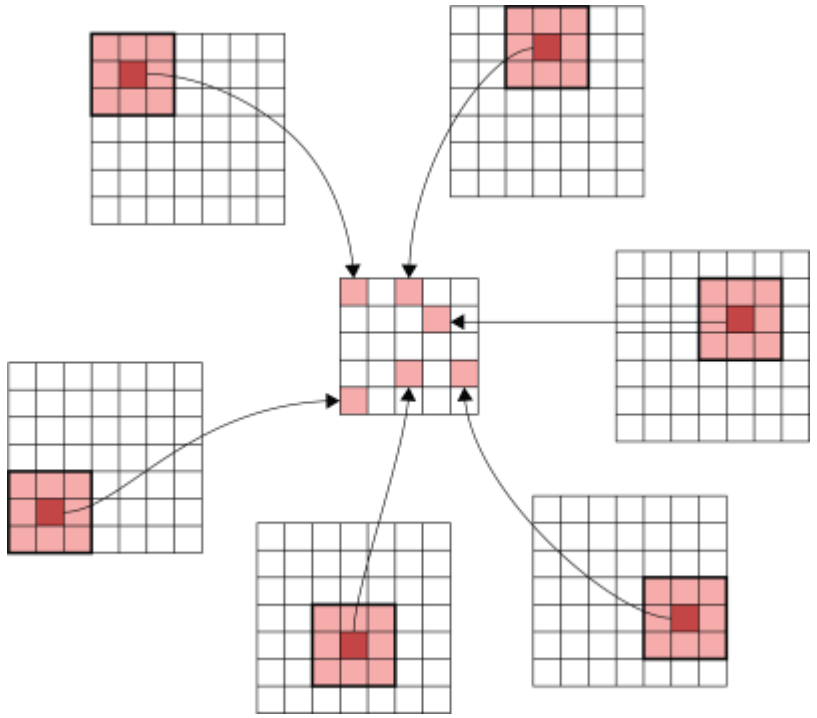


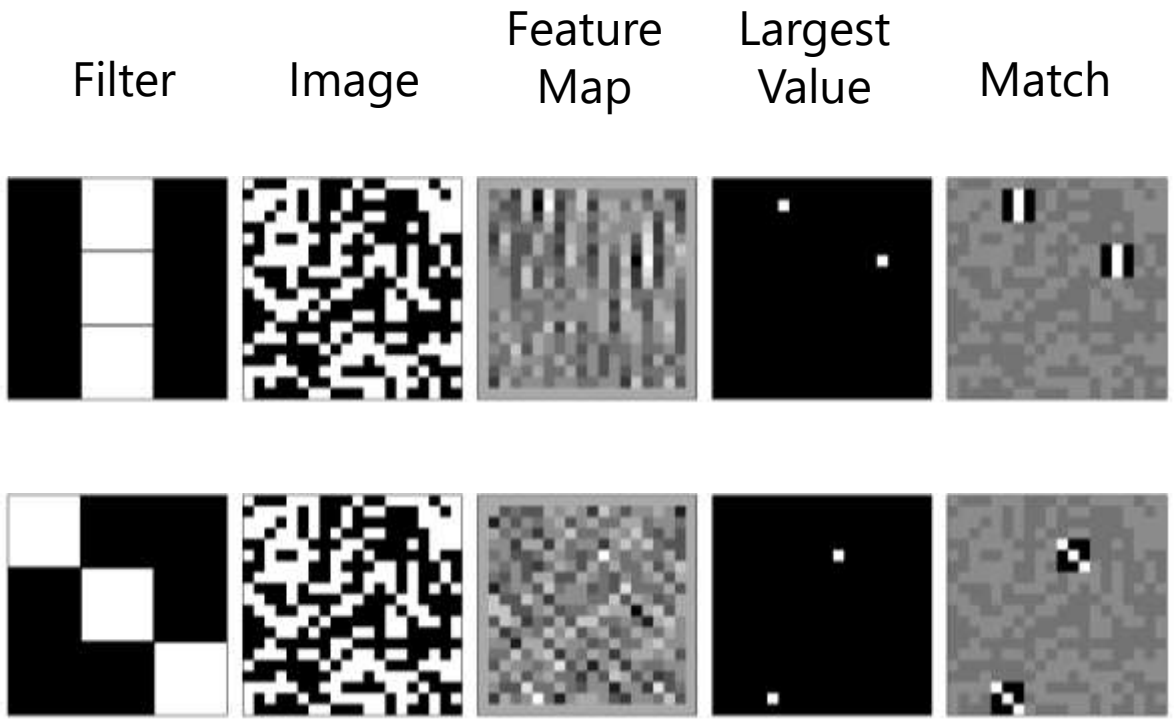
Vertical Edges

–1	0	1
–1	0	1
–1	0	1

Convolution as Feature Extractor

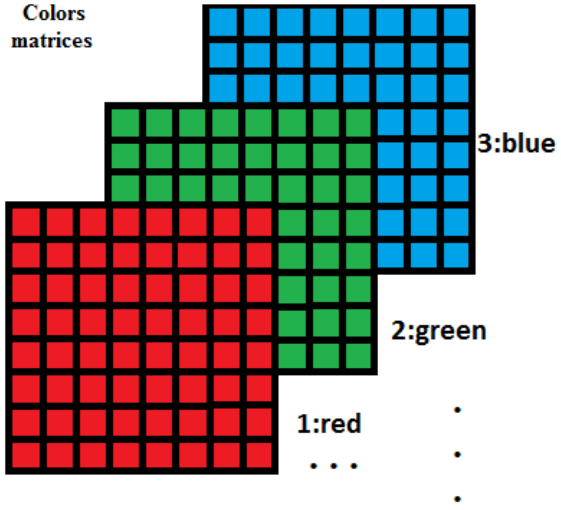
This Process of sweeping the filter across the image and getting a feature map is called convolution



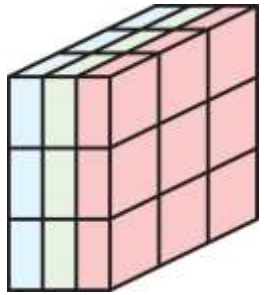


Multichannel Convolution

While performing convolution on input with more than one channel, our filter must also have the same number of channels



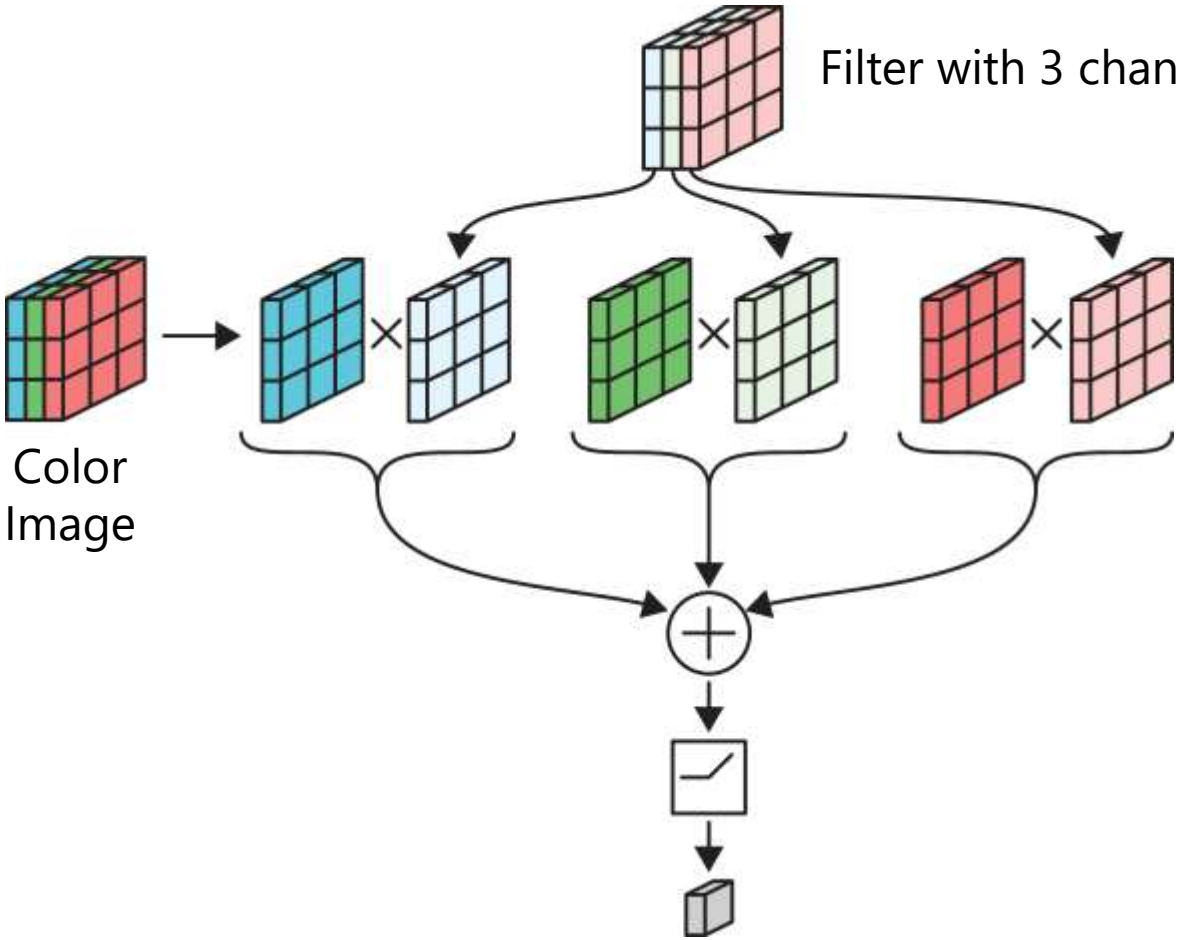
Color Image



Filter with 3 channels

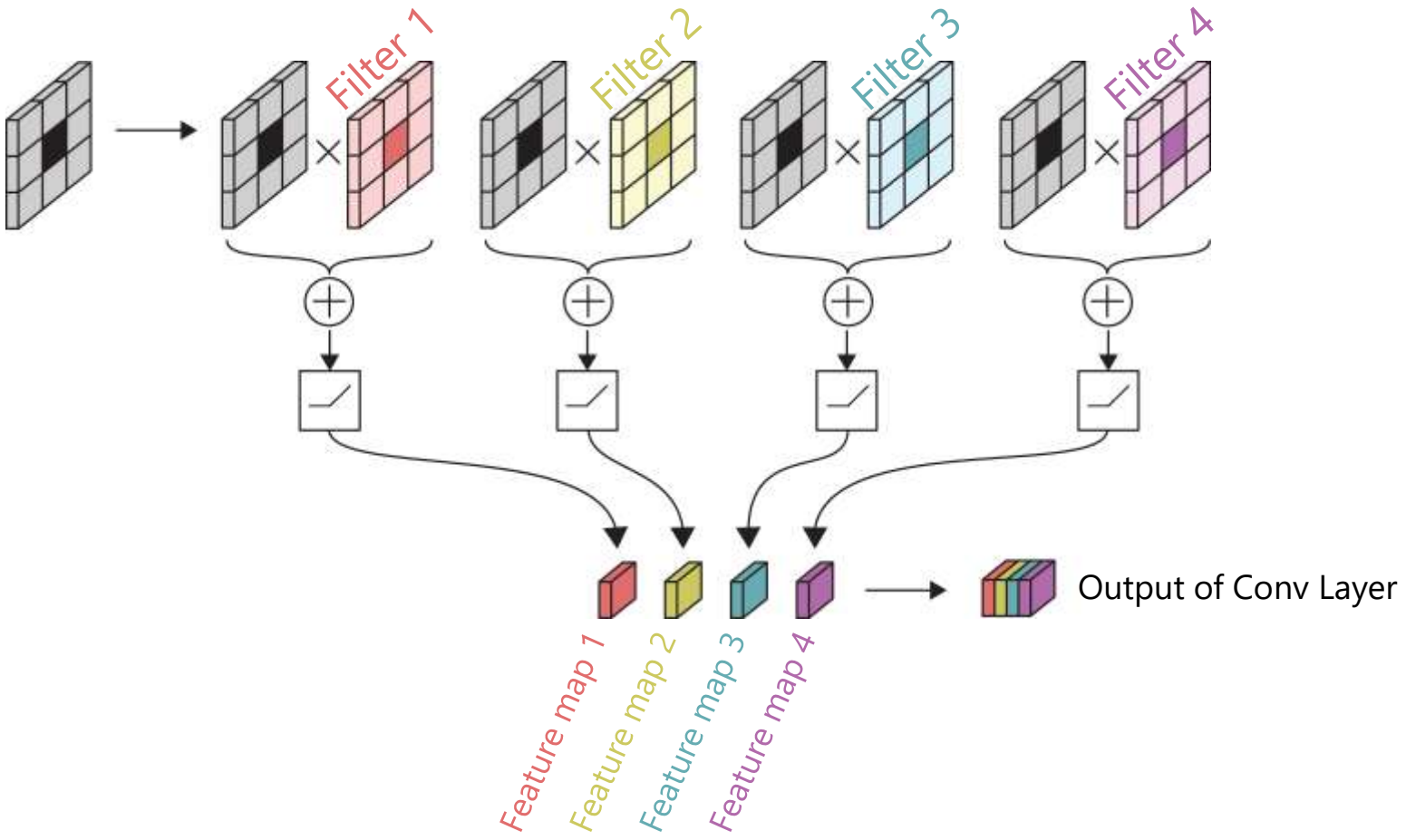
Multichannel Convolution

Each filter will then apply to the corresponding channel and results are added together



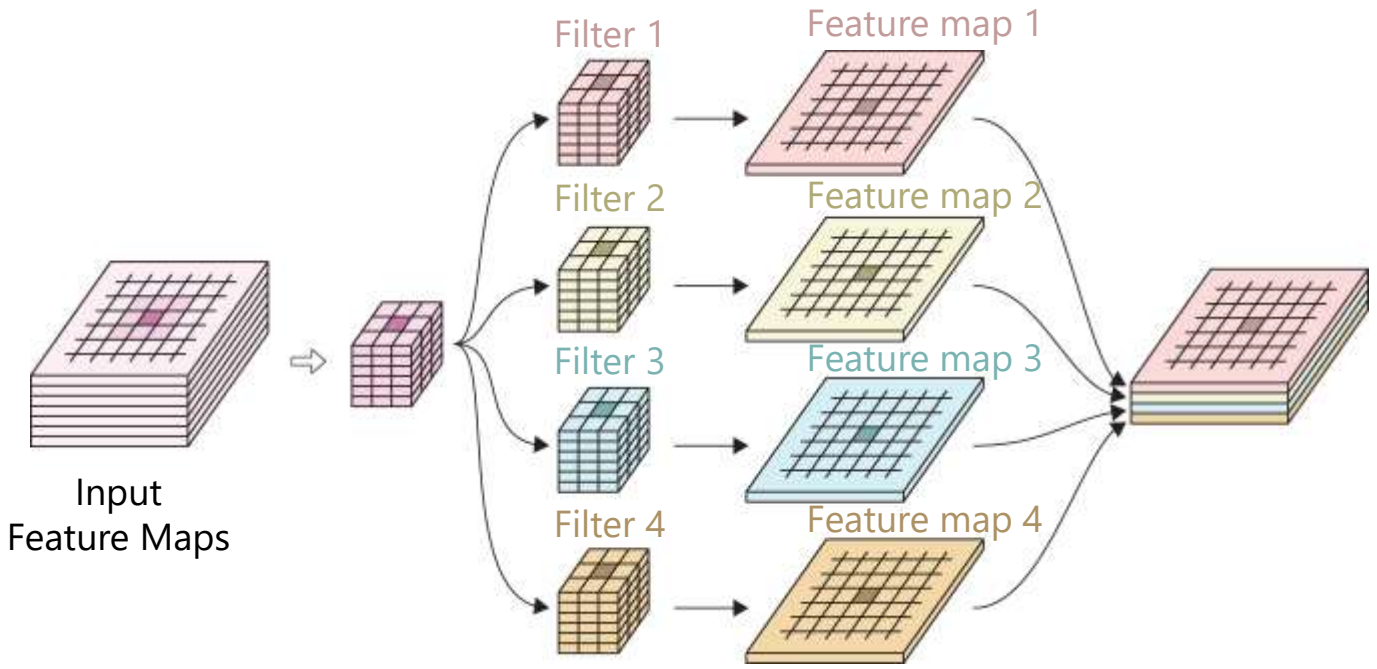
Convolutional Layer

We bundle up multiple filters together, each looking for a different pattern, forming a convolutional layer



Convolutional Layer

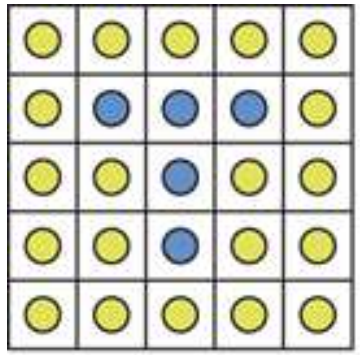
A convolutional layer can have many filters, but each one will produce a single feature map



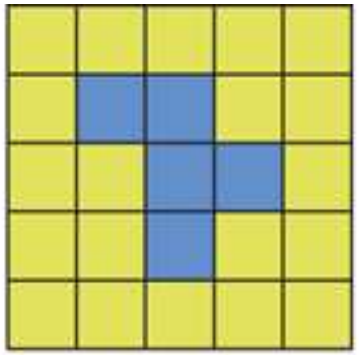
Output of Conv Layer
As many feature maps as there are filters

Pooling Layer

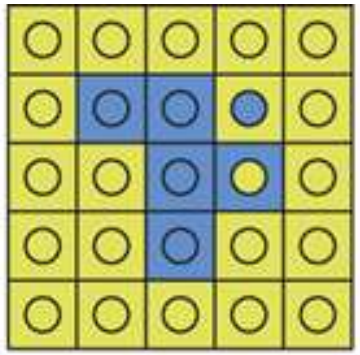
The convolutional filters look for exact match, so a slight change in the input will result in mismatch



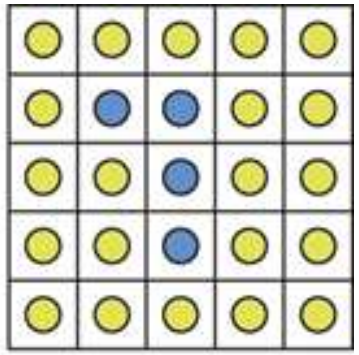
perfect filter



perfect image



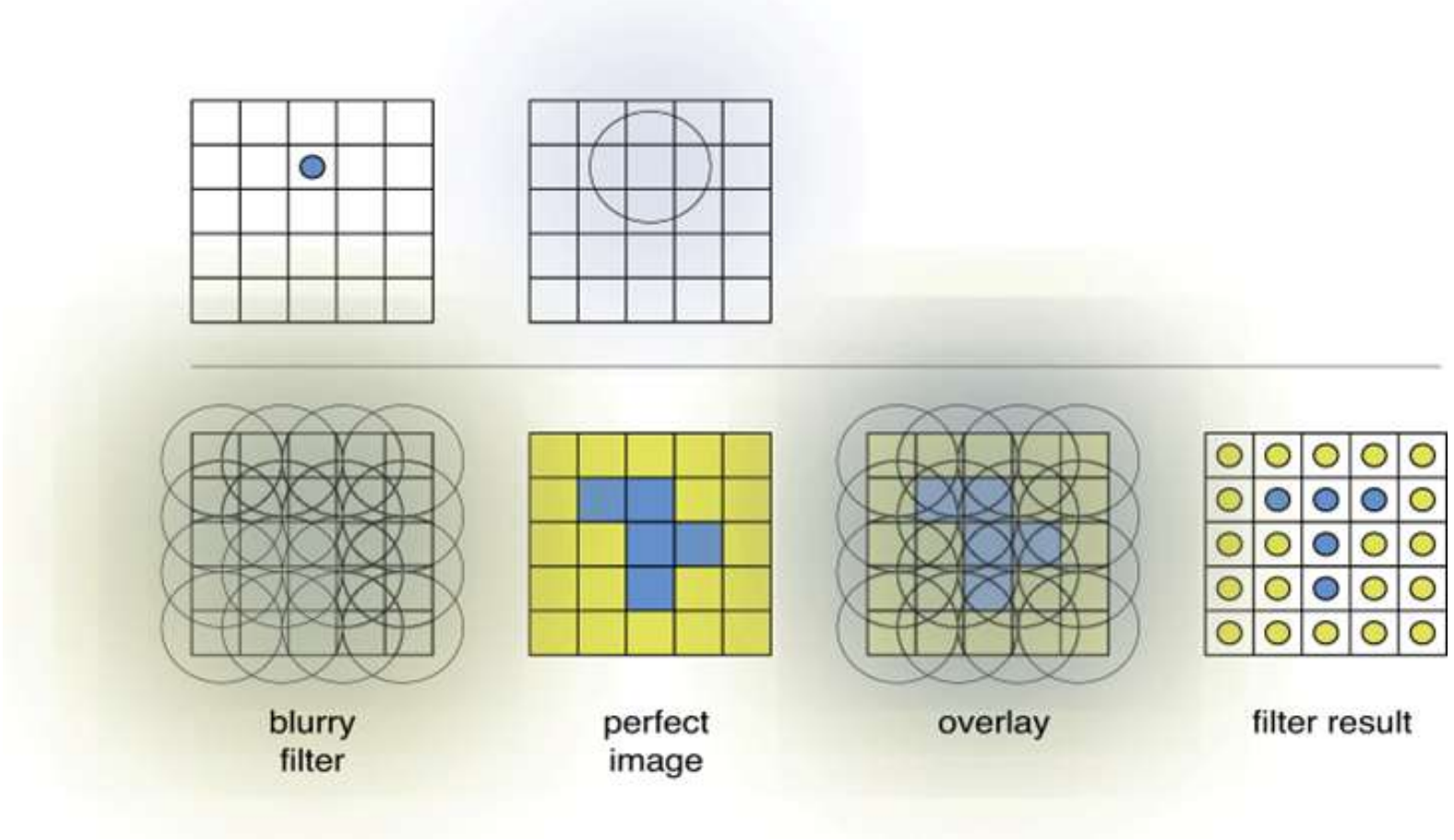
overlay



filter result

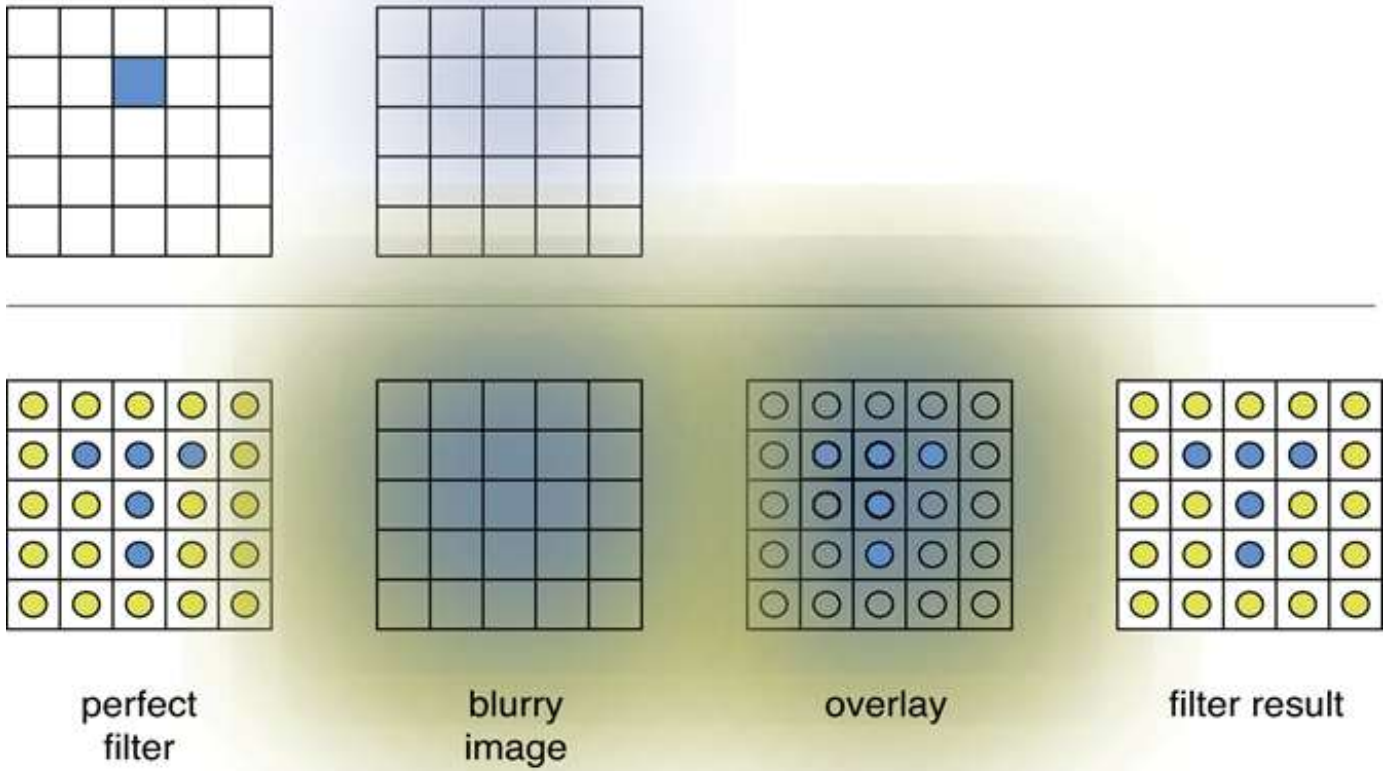
Pooling Layer

Is there a way for our filters to match the input despite slight translations?
we can make the filters look in its surrounding as well



Pooling Layer

A better way would be to blur the input based on small regions. This is called pooling



Pooling Layer

Pooling frees our filters to be precisely in the right place and also reduce the feature map size

3	2	–3	2
1	6	20	5
4	–13	2	6
–2	3	9	3

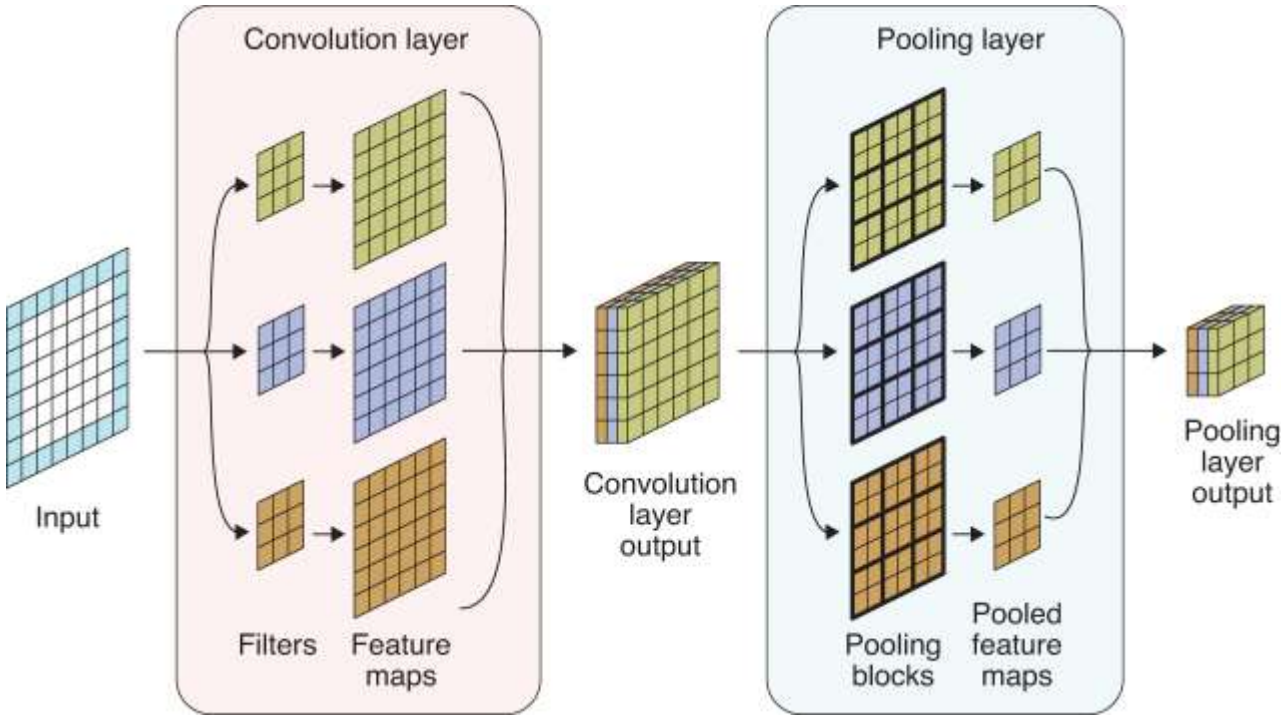
3	2	–3	2
1	6	20	5
4	–13	2	6
–2	3	9	3

3	6
–2	5

Average

6	20
4	9

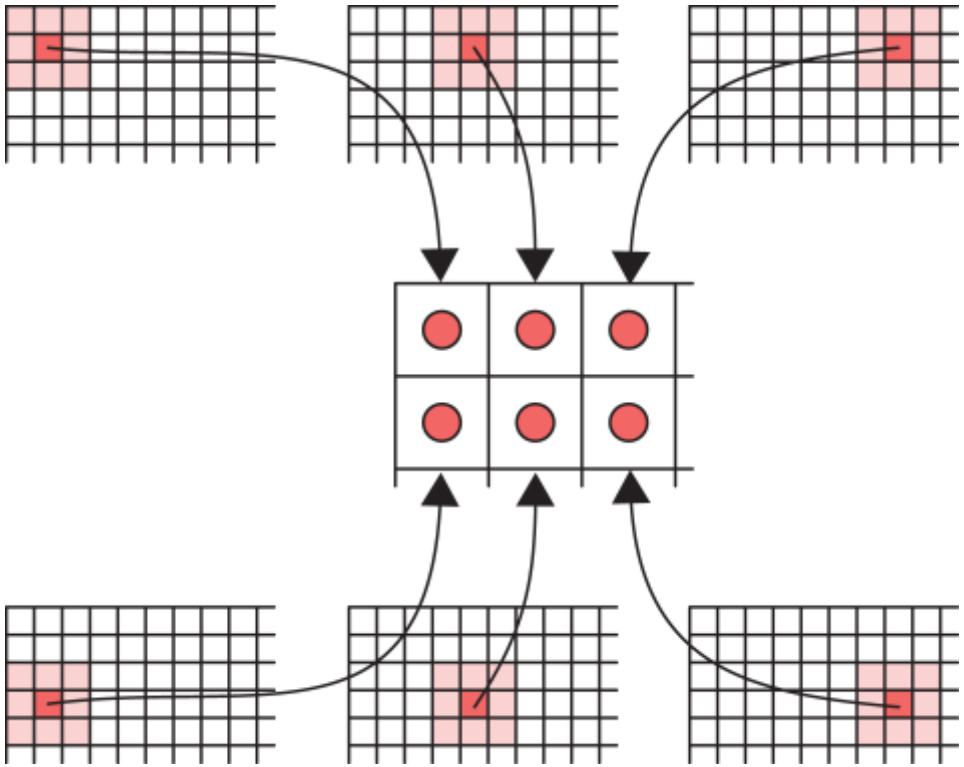
Max



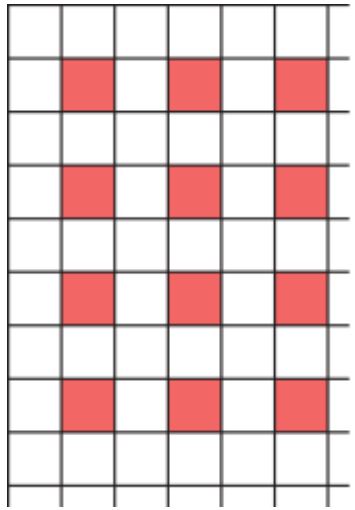
Striding

Instead of sweeping our filters over one pixel at a time, we can move more than one pixel

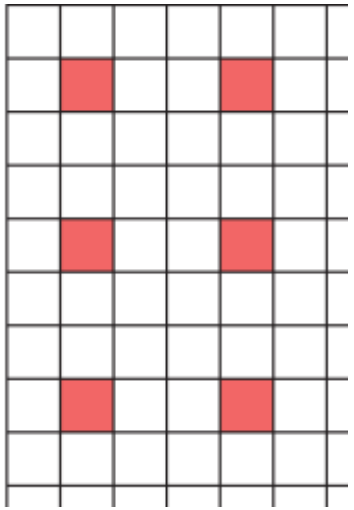
Moving 3 pixels while moving to the right



Moving 2 pixels while moving down



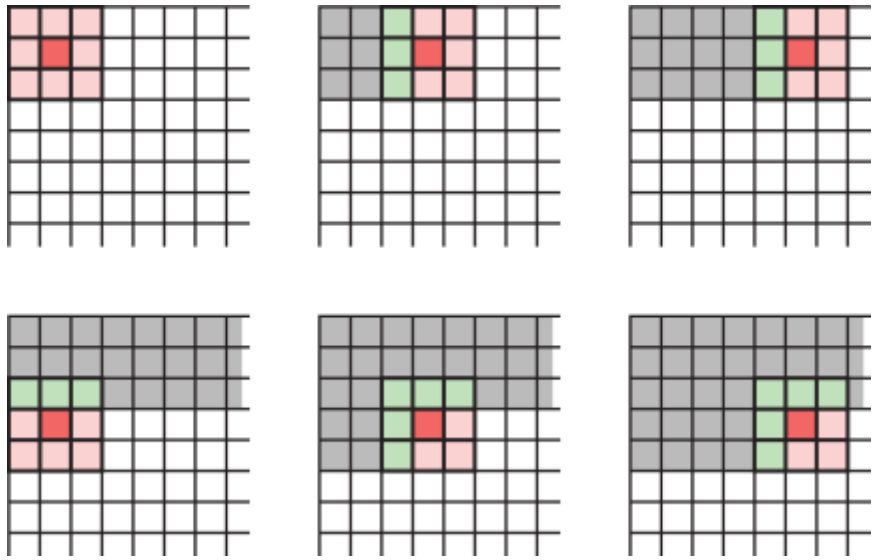
Stride = 2
Center Filter over every other pixel



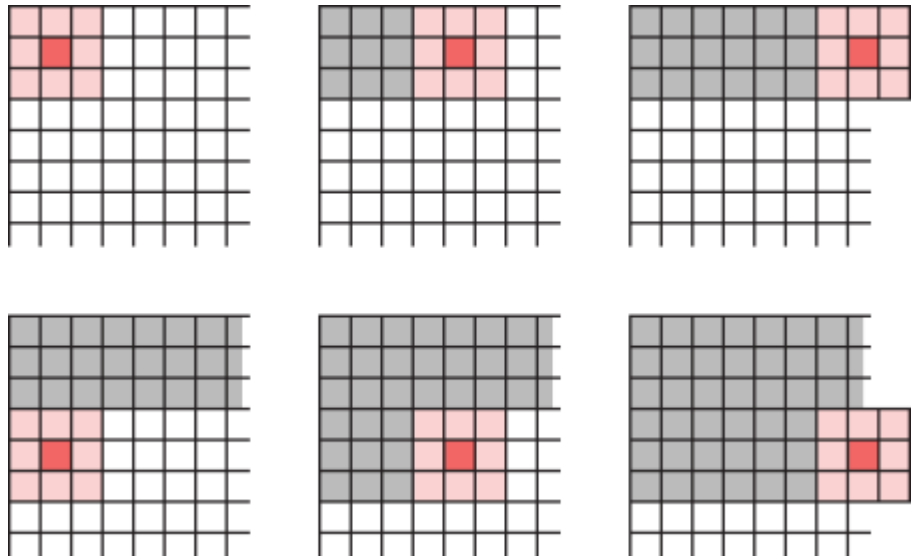
Stride = 3
Center Filter over every third pixel

Striding

Instead of sweeping our filters over one pixel at a time, we can move more than one pixel



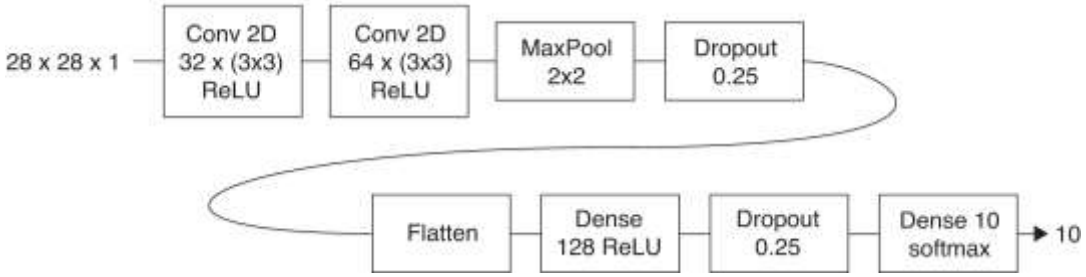
Stride = 2
Center Filter over every other pixel



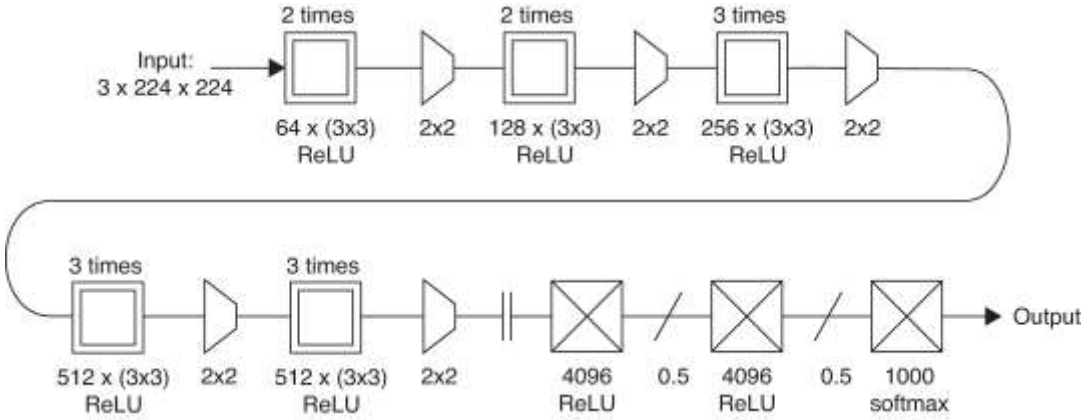
Stride = 3
Center Filter over every third pixel

Convnets in Practice

Convolutional Neural Networks (CNNs) are made of convolutional layers and pooling layers (along with others)



LeNet Architecture

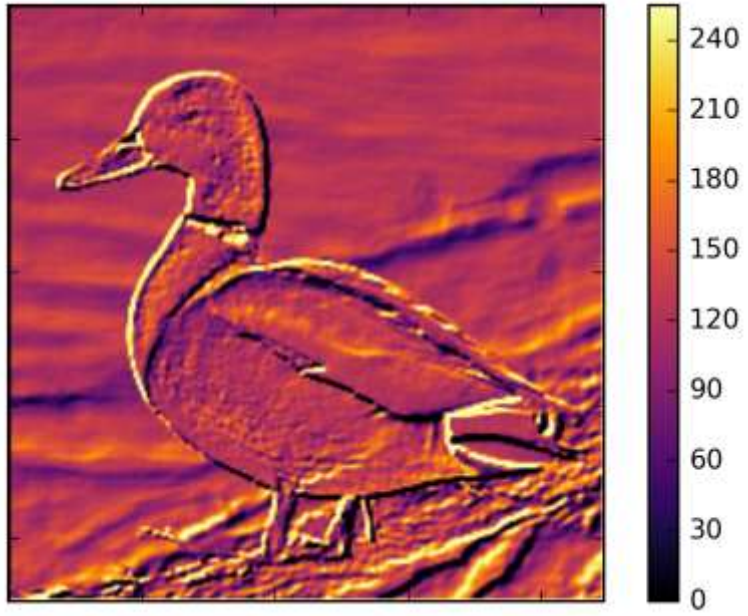


VGG16 Architecture

Visualizing Feature Maps

Let's visualize the feature maps obtained from various CNN blocks of VGG16

Input Image



Feature map of a single filter

Visualizing Feature Maps

Let's visualize the feature maps obtained from various CNN blocks of VGG16

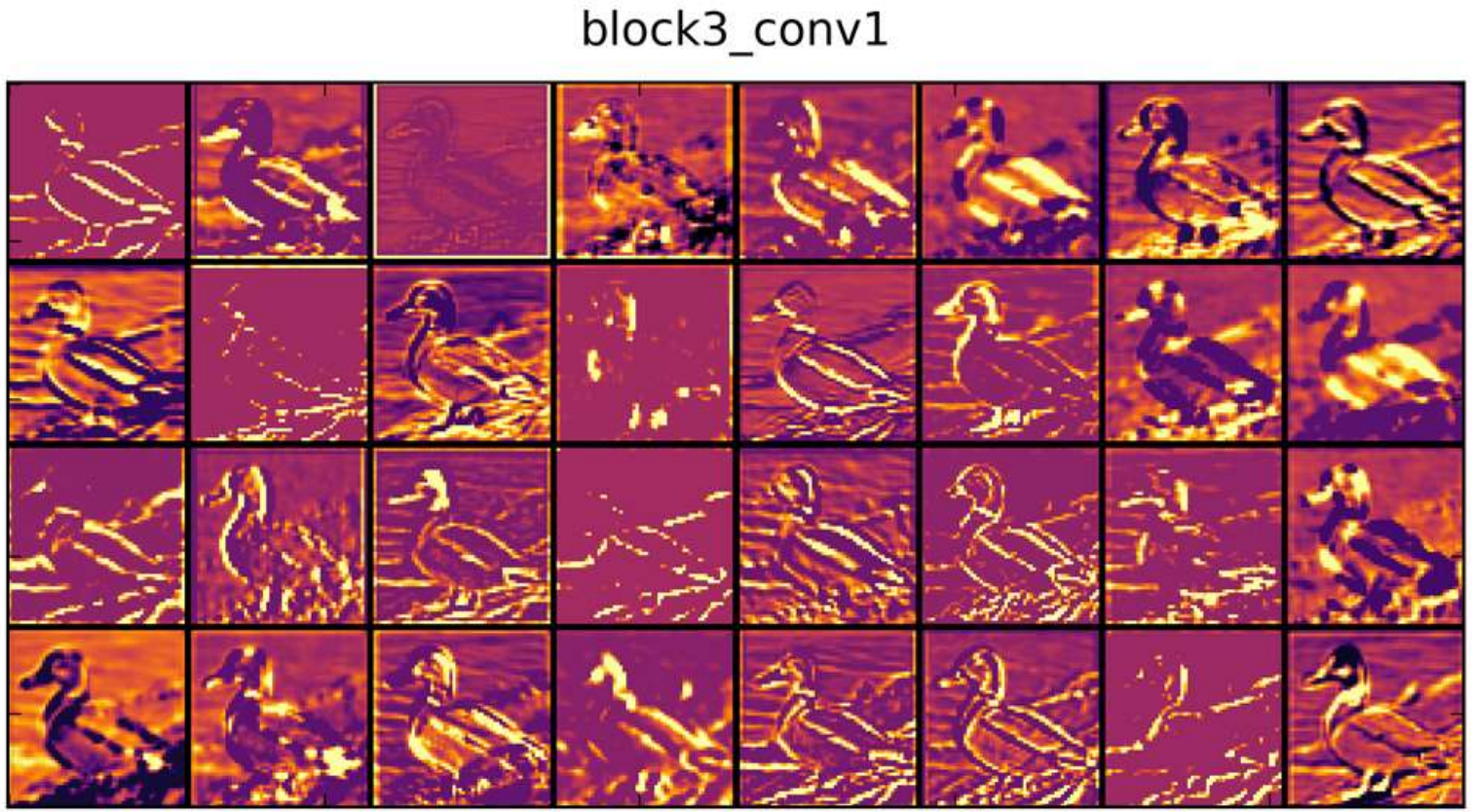
Input Image

block1_conv1

Visualizing Feature Maps

Let's visualize the feature maps obtained from various CNN blocks of VGG16

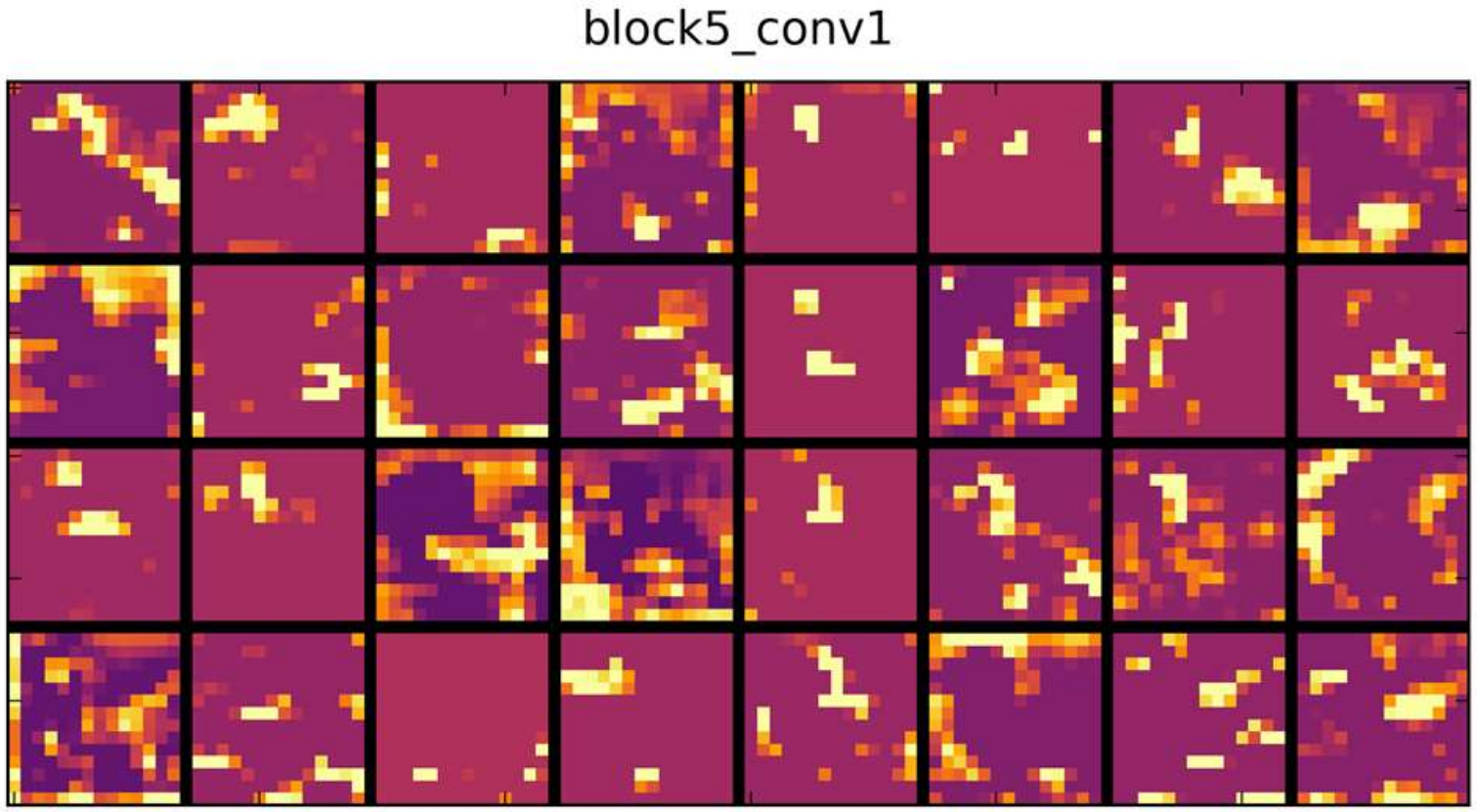
Input Image



Visualizing Feature Maps

Let's visualize the feature maps obtained from various CNN blocks of VGG16

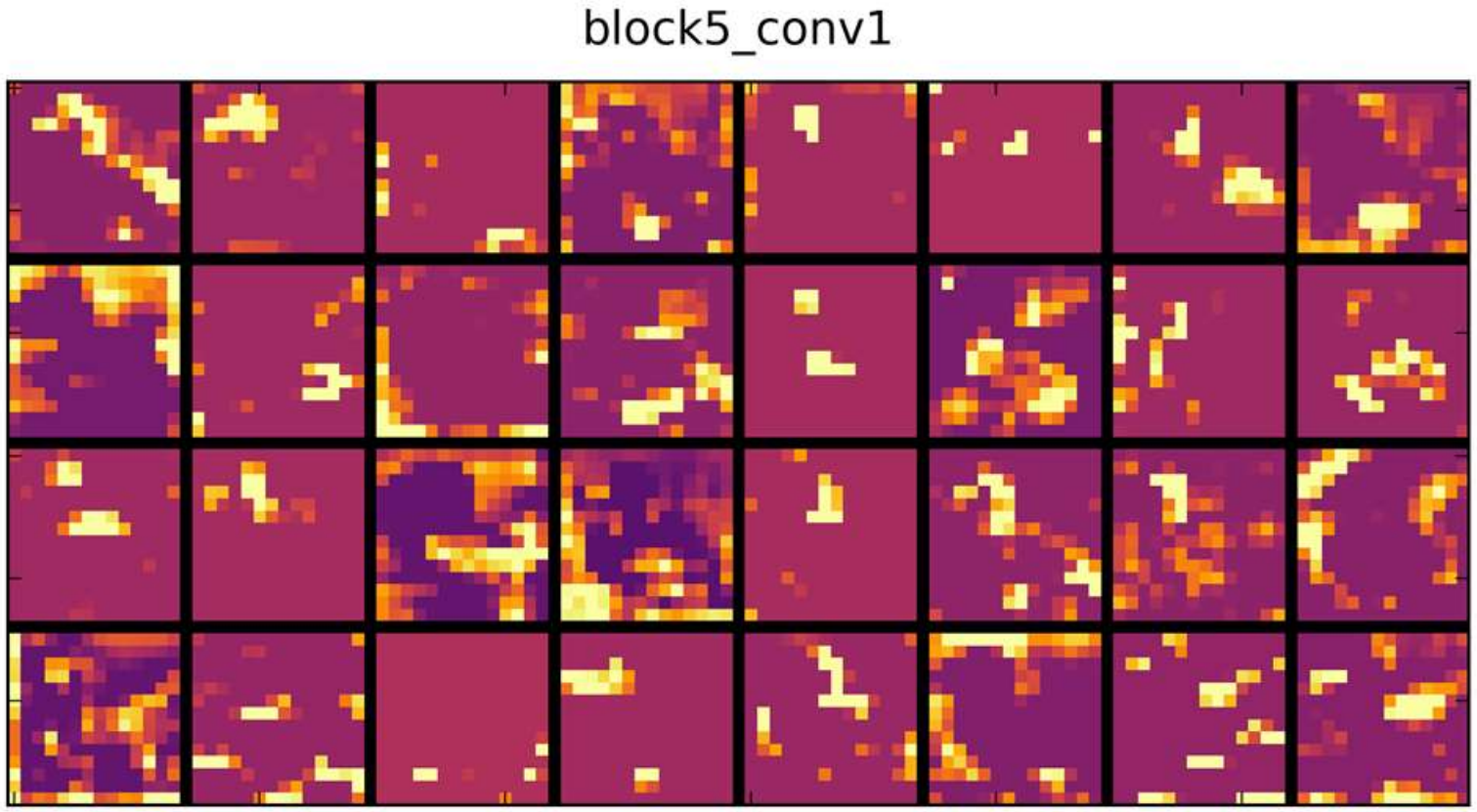
Input Image



Visualizing Feature Maps

Let's visualize the feature maps obtained from various CNN blocks of VGG16

Input Image



Thank You