Introduction to Robotics

CSE 461

Lecture 13: Introduction to CNNs and Object Detection

Niloy Irtisam
Lecturer, Dept. of Computer Science and Engineering
Brac University

Last Class

What is Machine Learning

Neural Network

Mother Law of Machine Learning

$$w1 * x1 + w2 * x2 + w3 * x3 = y$$

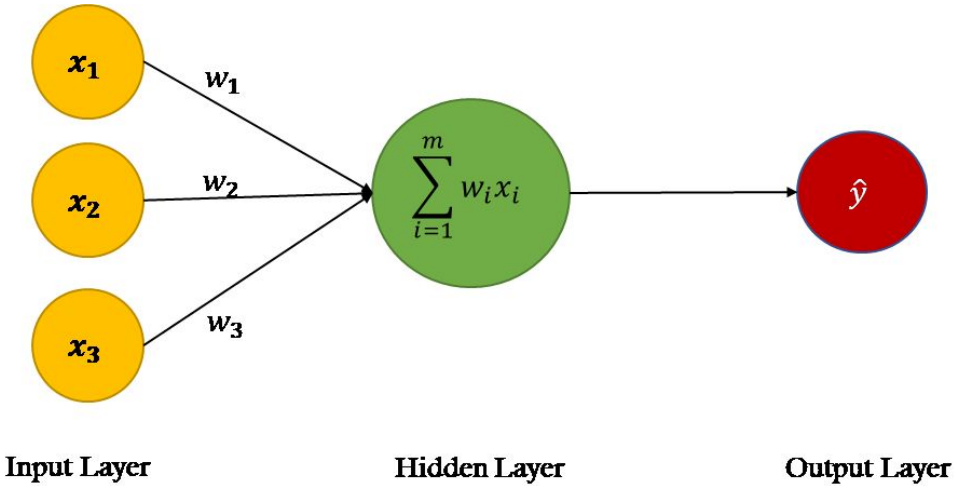
x1, x2, ..., xn = Features

w1, w2, ..., wn = Weights

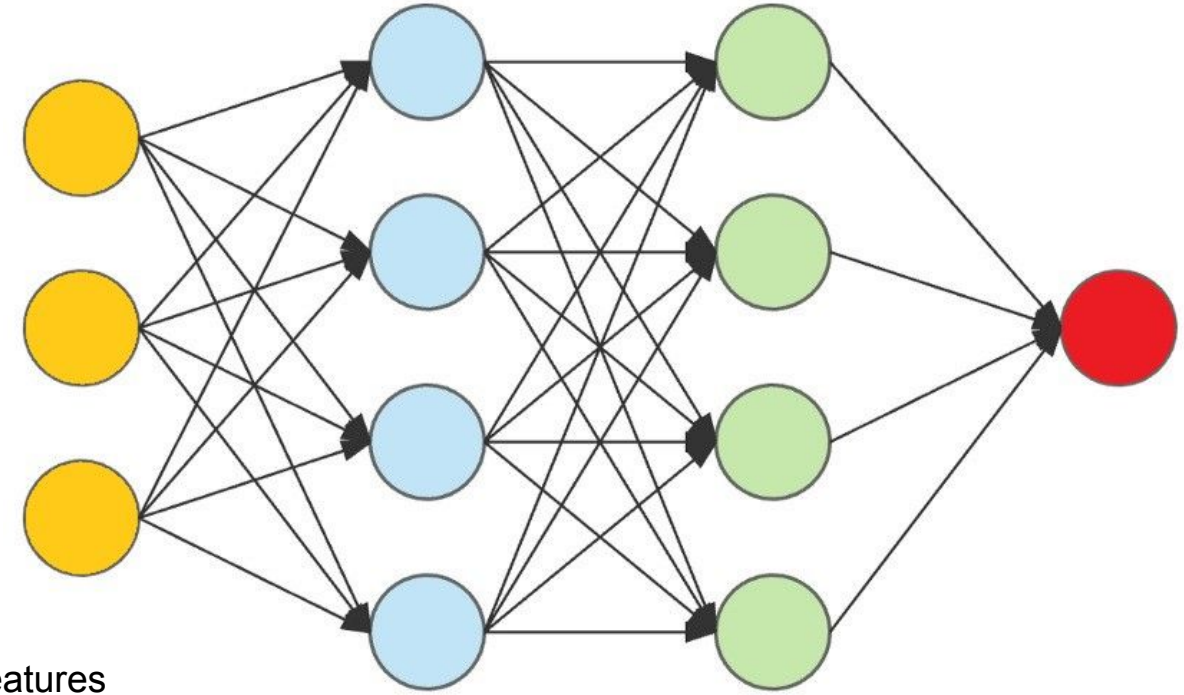
y = output/ target

Neural Network

$$w1 * x1 + w2 * x2 + w3 * x3 = y$$



Neural Network



Predictions/
decisions

Features

Hidden Layers

Two types

1. Features are given (Handcrafted Features)
2. Raw Data is given not features

Out[33]=

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

1. Some data column may not a good feature.
2. Some data column may need to be transformed.

Out[33]=

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

1. Some data column may not a good feature.
2. Some data column may need to be transformed.

Out[33]=

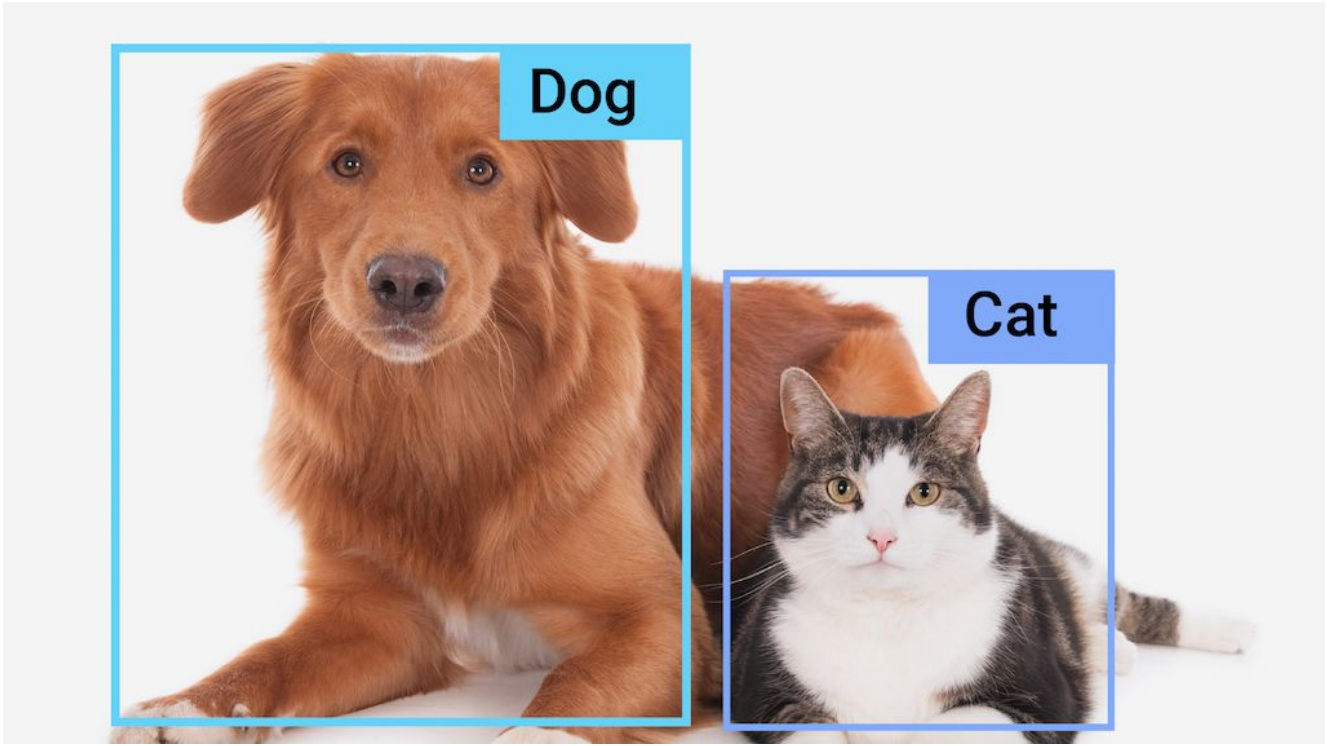
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

1. We can do it by our hand (Handcrafted)
2. Neural network does this automatically (Deep Learning)

Deep Learning

Deep learning is a type of machine learning based on artificial neural networks in which multiple layers of processing are used to extract progressively higher level features from data.

Let’s Do object detection



Out[33]=

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

Out[33]=

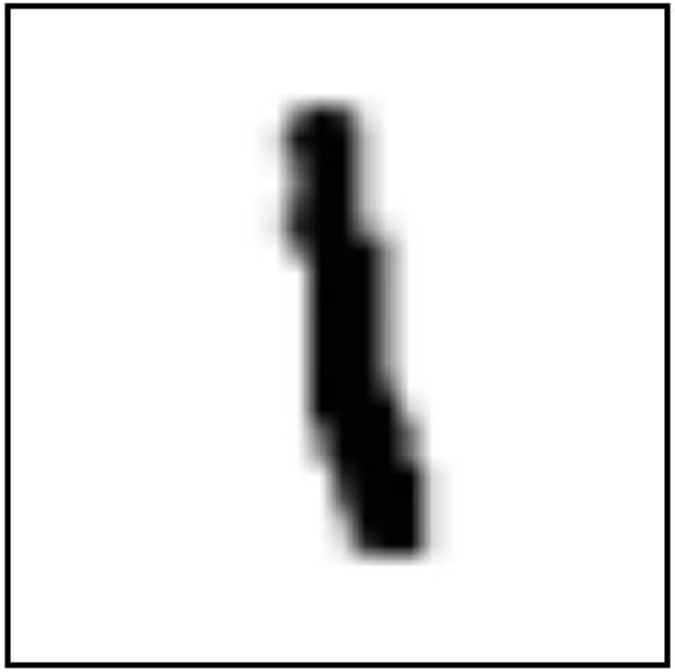
Cat

Cat

Dog

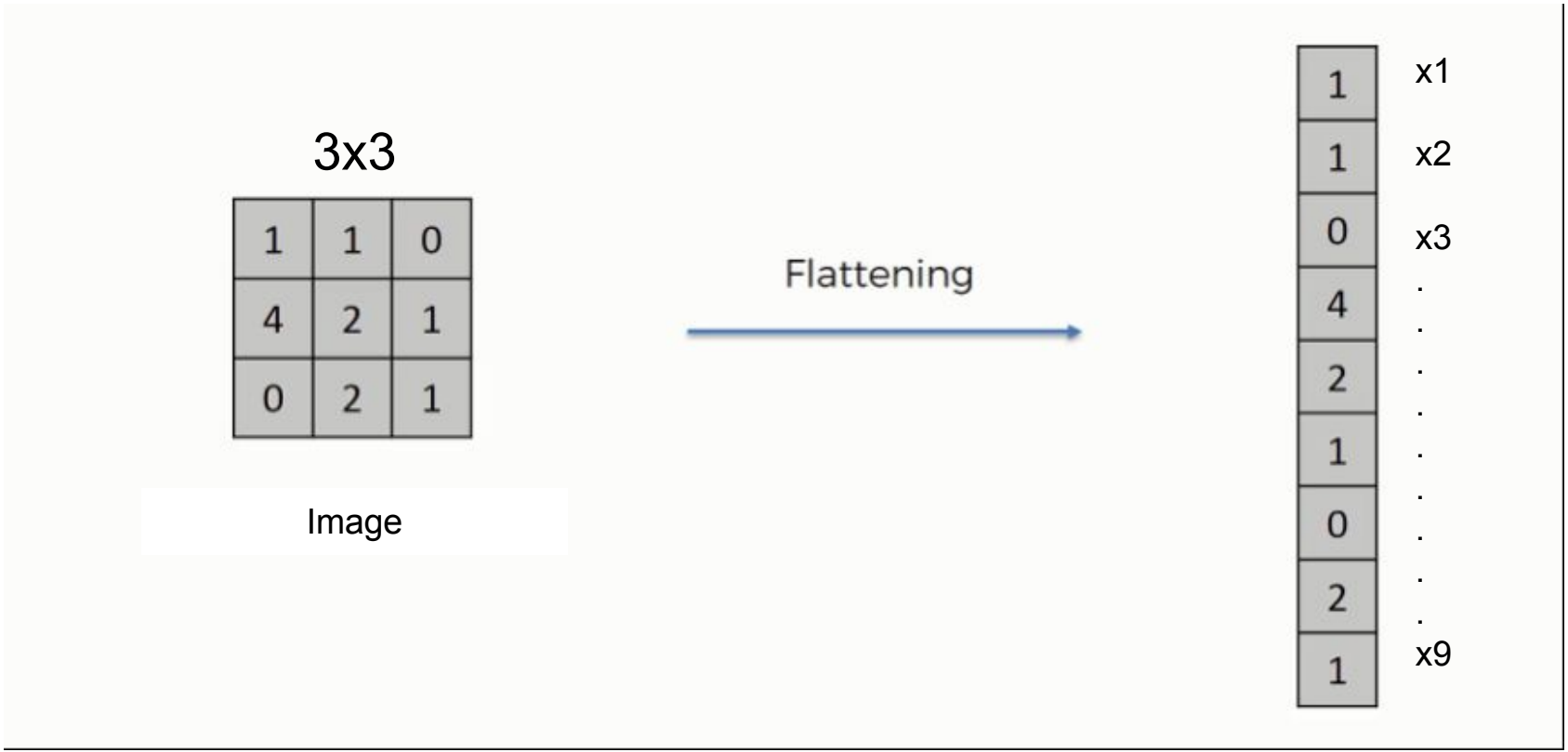
Dog

Features ??

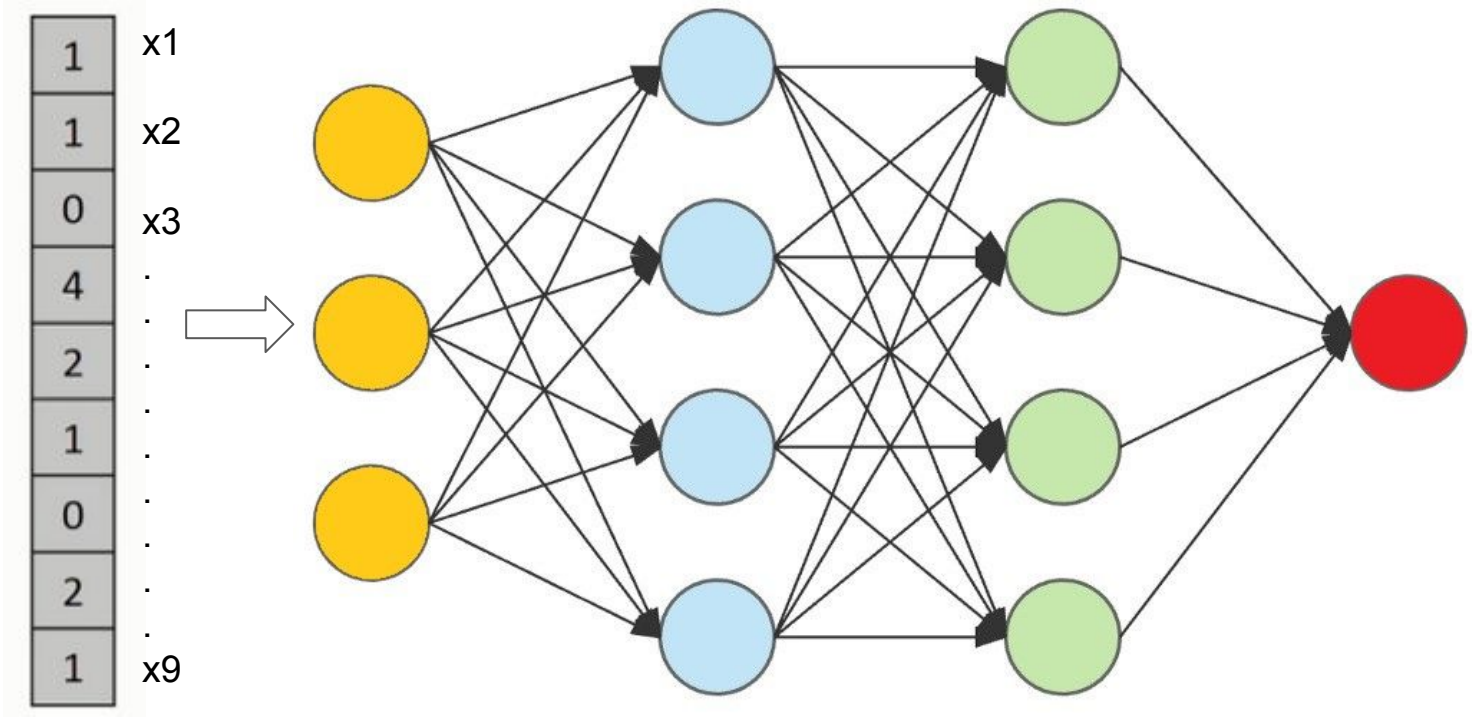


$\approx$

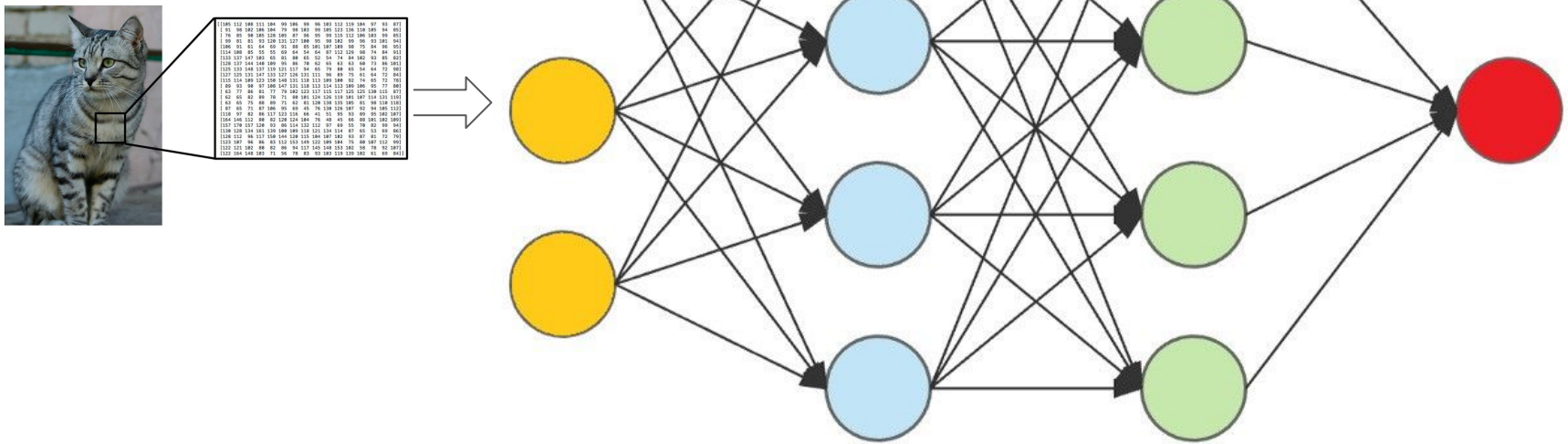
$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .6 & .8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .5 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .9 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .3 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$


3x3
1 1 0
4 2 1
0 2 1
Image
Flattening
1
1
0
4
2
1
0
2
1
x1
x2
x3
x9

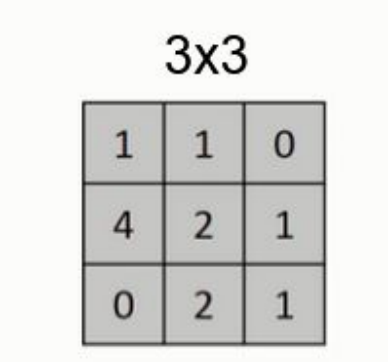
Pixels as Features



Pixels as Features



Problem with “Pixels as features”



3*3 = 9

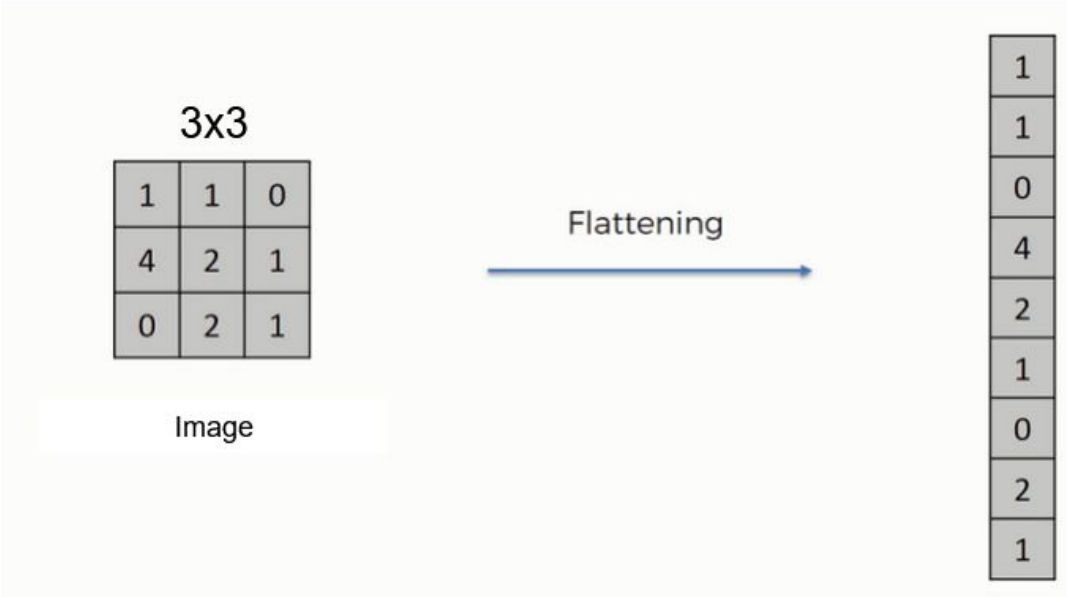
12x12 → 12*12 = 144

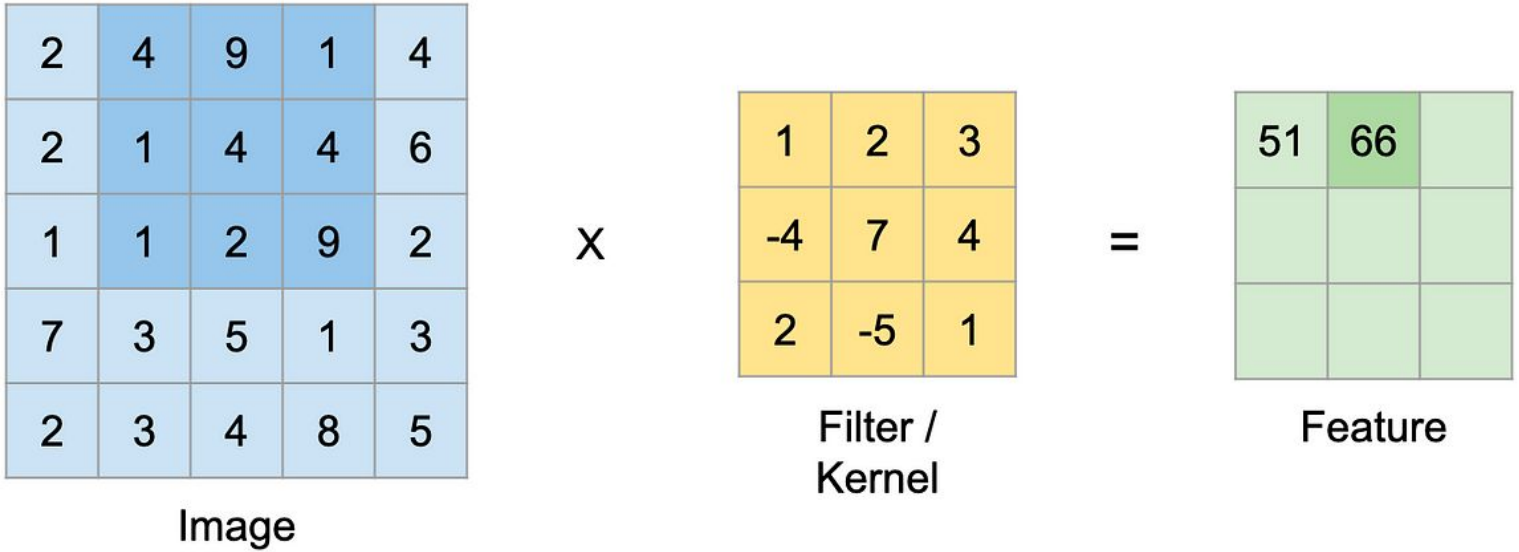
680x420 → 680*420 = 285,600

680x420x3 → 680*420*3 = 856,800

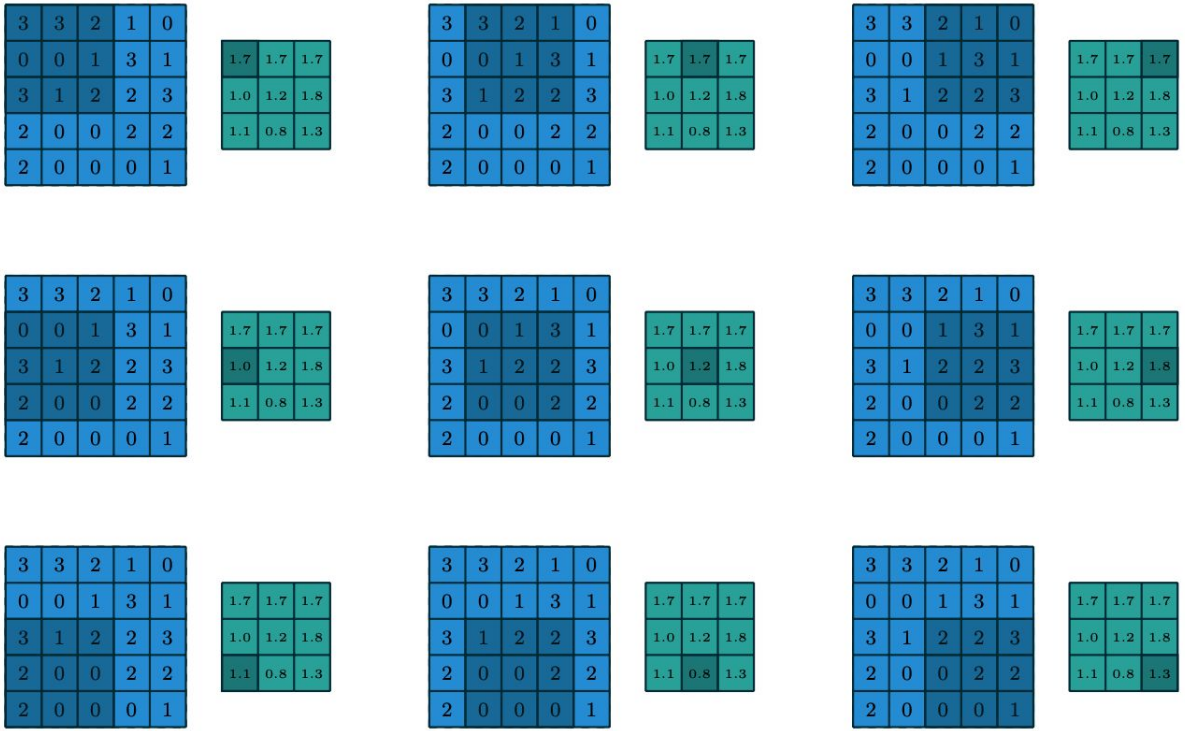
Problem with “Pixels as features”

No Structural Features

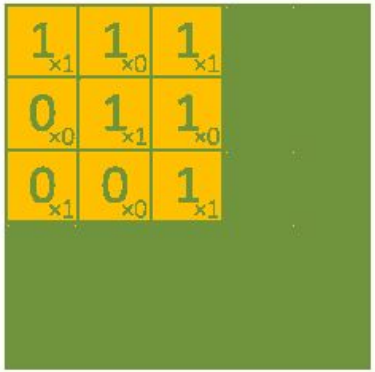



2 4 9 1 4
2 1 4 4 6
1 1 2 9 2
7 3 5 1 3
2 3 4 8 5
Image
X
1 2 3
-4 7 4
2 -5 1
Filter / Kernel
=
51 66
Feature

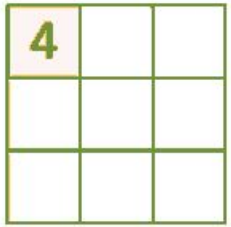
Convolution



Kernels



Image

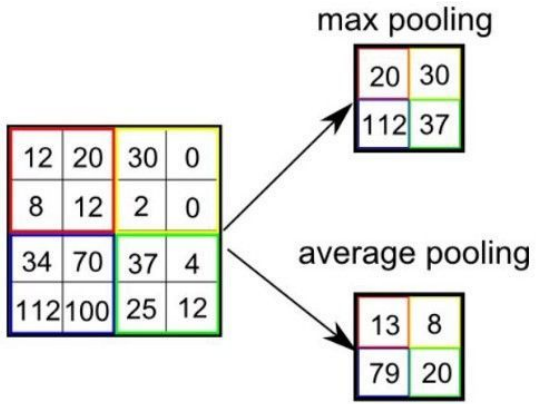


Convolved Feature

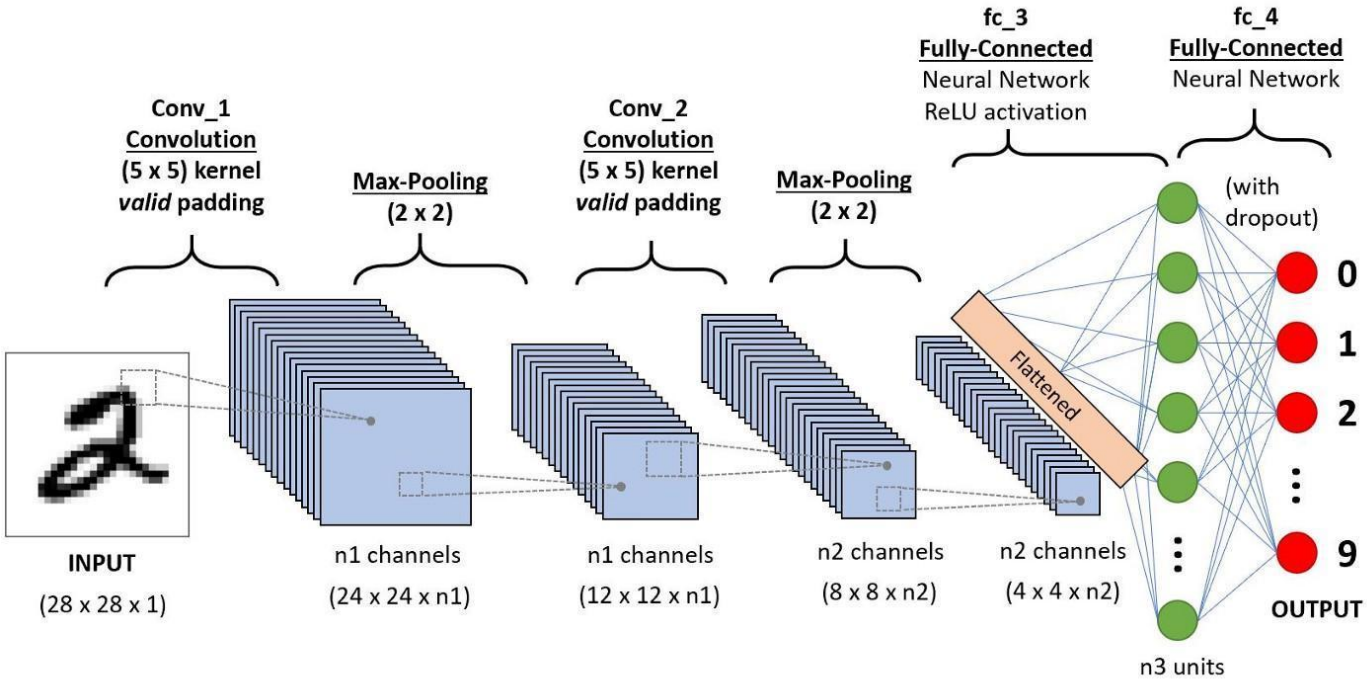
Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Pooling Layer

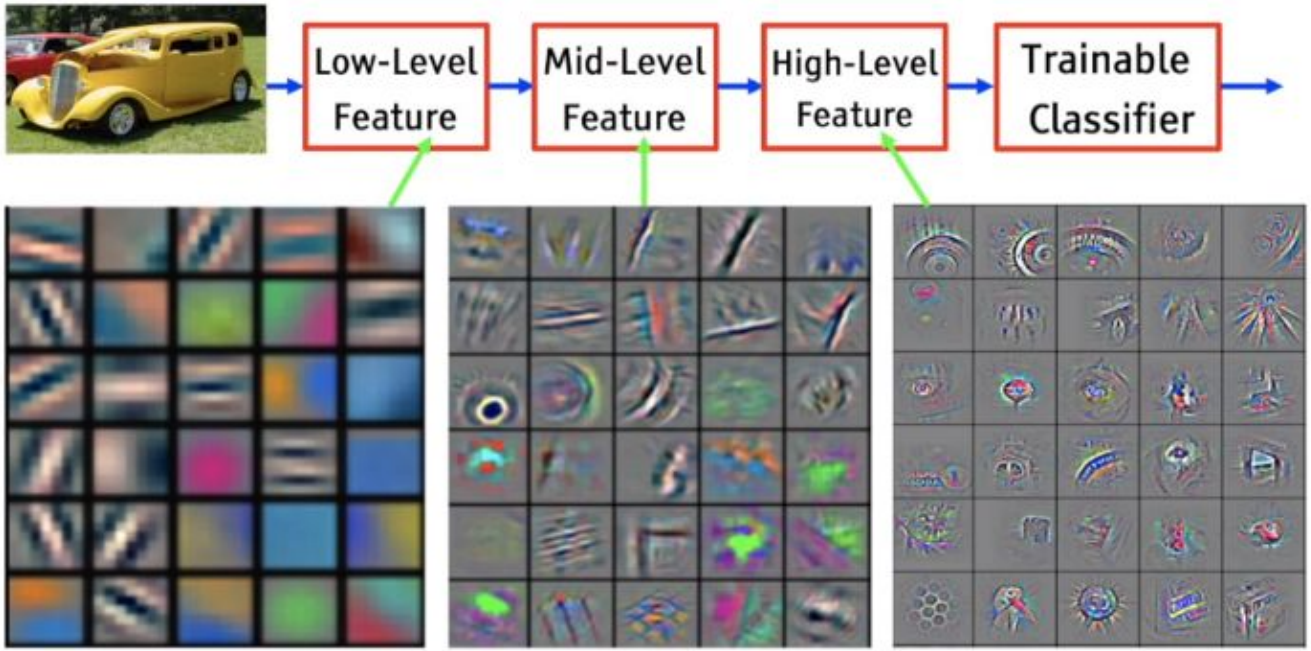
- Used for reducing the number of parameters in case of large images
- Also called Subsampling or Down sampling
- Retains major information
- Max pooling and average pooling are two types of pooling that are used



Architecture of CNNs



In the eyes of the CNN



https://miro.medium.com/max/2400/1*bIY_8FE7dJaGk1kjU7iVSg.png

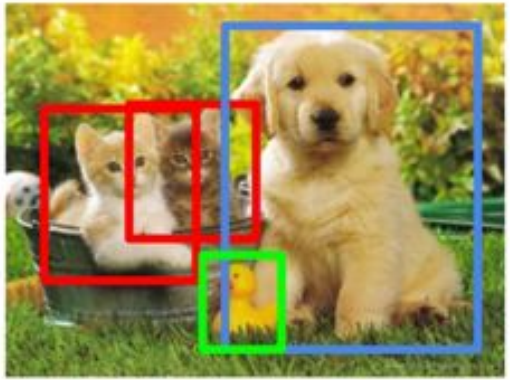
Classification

CAT

Classification + Localization

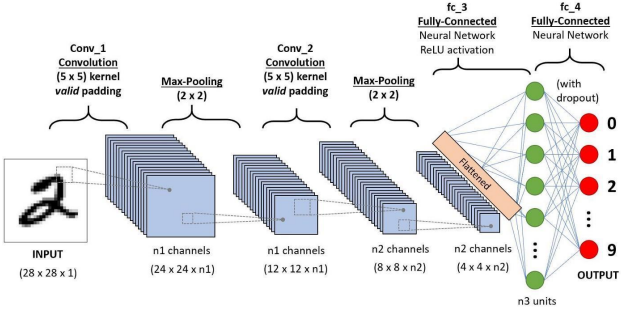
CAT

Object Detection



CAT, DOG, DUCK

Some Popular Algorithms



- Solved most of the challenges except for the speed:
  - Faster R-CNN
  - Single Shot MultiBox Detector (SSD)
  - Retina Net
- YOLO algorithm was able to solve the object detection speed issue. Some of it's features are:
  - **Speed**: This algorithm improves the speed of detection because it can predict objects in real-time.
  - **High accuracy**: provides accurate results with minimal background errors.
  - **Learning capabilities**: The algorithm has excellent learning capabilities.
  - It has multiple versions: YOLO V1, YOLO 9000, YOLO V3, YOLO V4
  - However, it has lower accuracy than state-of-the-art object detection algorithms

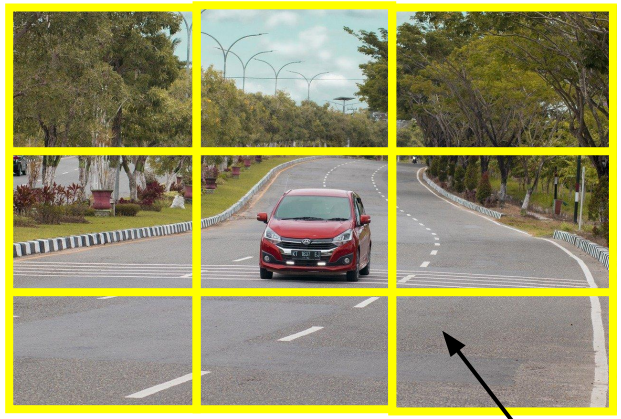
We will discuss YOLO in the upcoming slides

YOLO (**Y**ou **O**nly **L**ook **O**nce)

- The YOLO framework takes the entire image in a single instance and predicts the bounding box coordinates and class probabilities for these boxes
- It uses the following techniques:
  - Residual blocks
  - Bounding box regression
  - Intersection Over Union (IOU)

Basic Working Principle

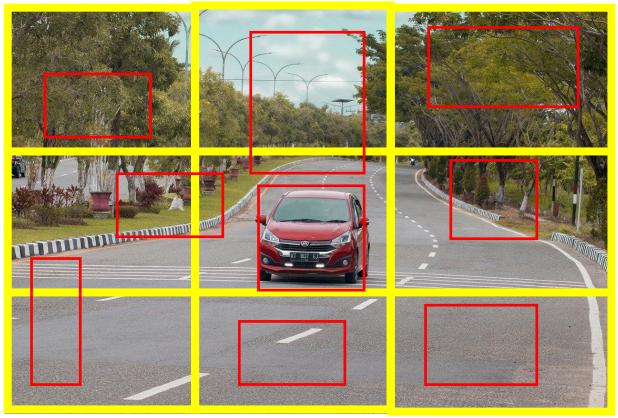
1.Divide the image into S x S grids



S=3

Grid

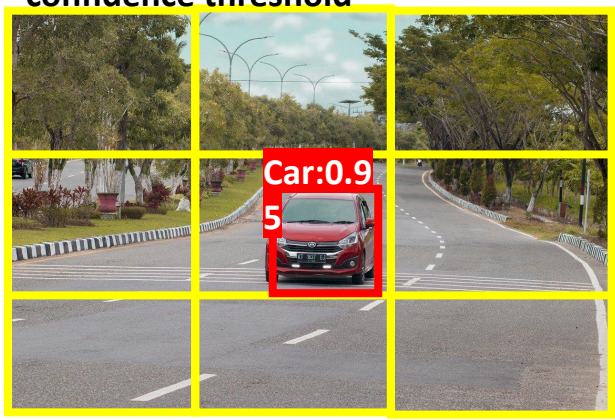
2.Each grid predicts B bounding boxes



B=1

If the center of a object falls into a grid cell, that grid cell is responsible for detecting that object

3.Return bounding boxes above confidence threshold



All other bounding boxes have confidence. Threshold below , suppose 0.90. So they were removed