Convolutional Neural Networks

Unlocking data's hidden patterns

236504 : Anushree Bhuskute
236517 : Satyam Gaikwad

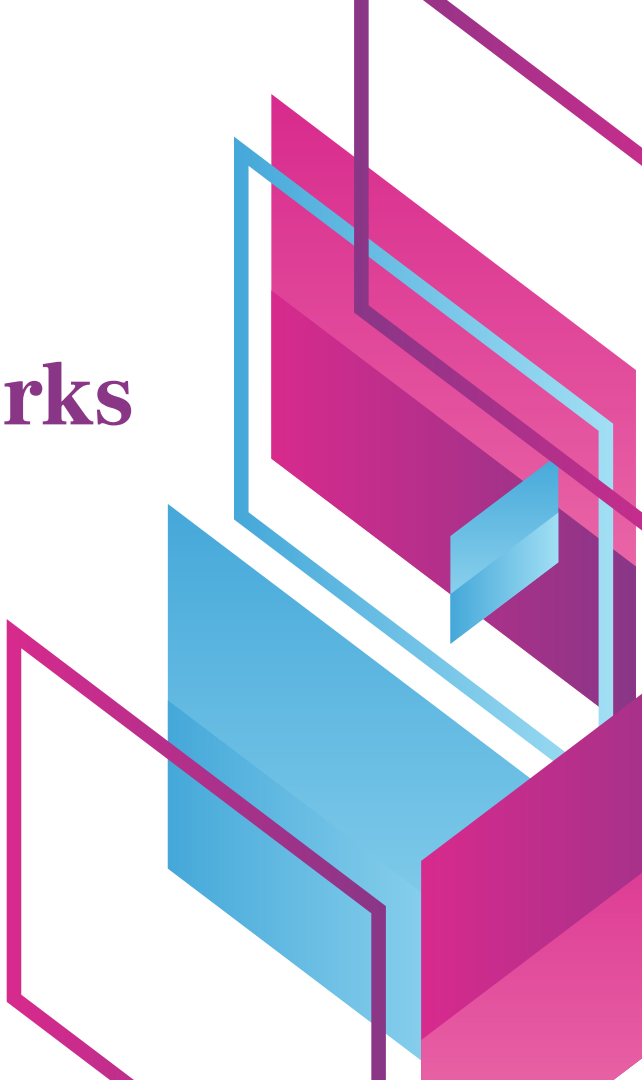
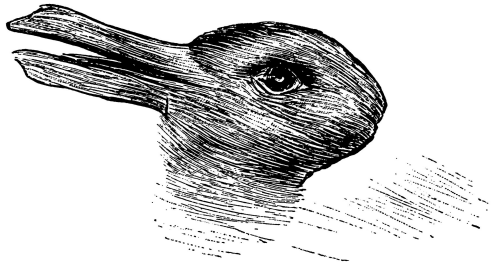


Table of contents

01

Introduction to *convolutional neural networks*

What can you see from these images?



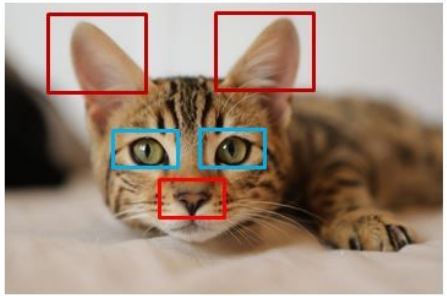
The elements that we perceive are **features.**

Convolutional neural networks

- A specialised type of deep learning neural network.
- Recognises and classifies images based on their features.
- Primarily utilized in computer vision tasks.
- Excel in recognizing patterns and structures within images.
- Revolutionized tasks like image classification, object detection, and facial recognition.

In simple terms :

- If we want our program to identify a cat.
- Then the CNN, will identify different features of the like eyes, ears, fur pattern, etc.
- It will put these features together and say “Its a CAT.”



Ears filter

Eyes filter

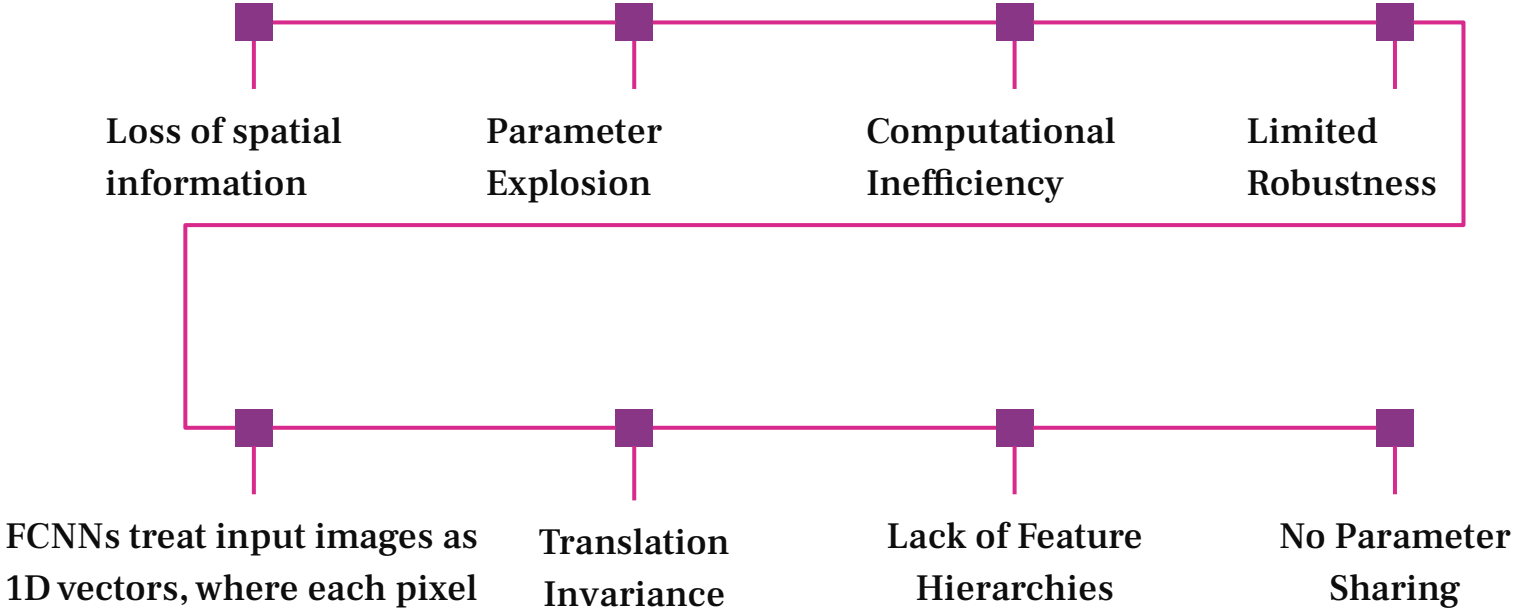
Nose filter

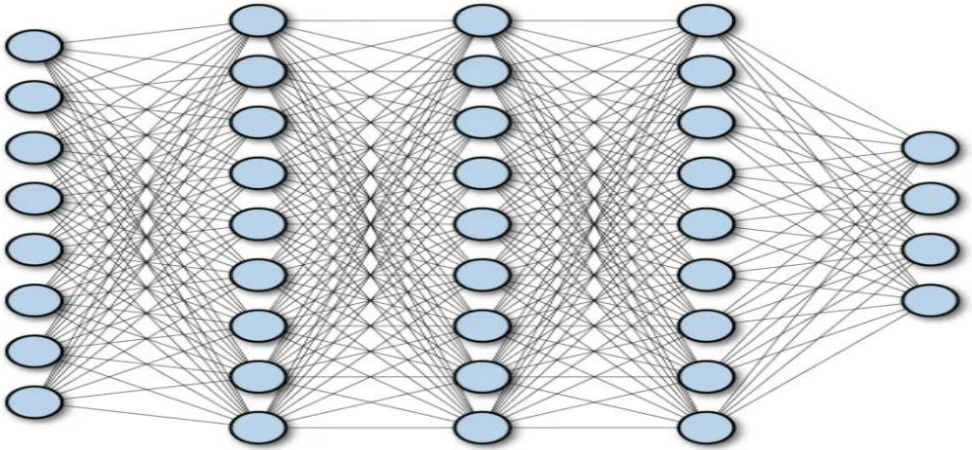
02

Why *convolutional neural networks?*

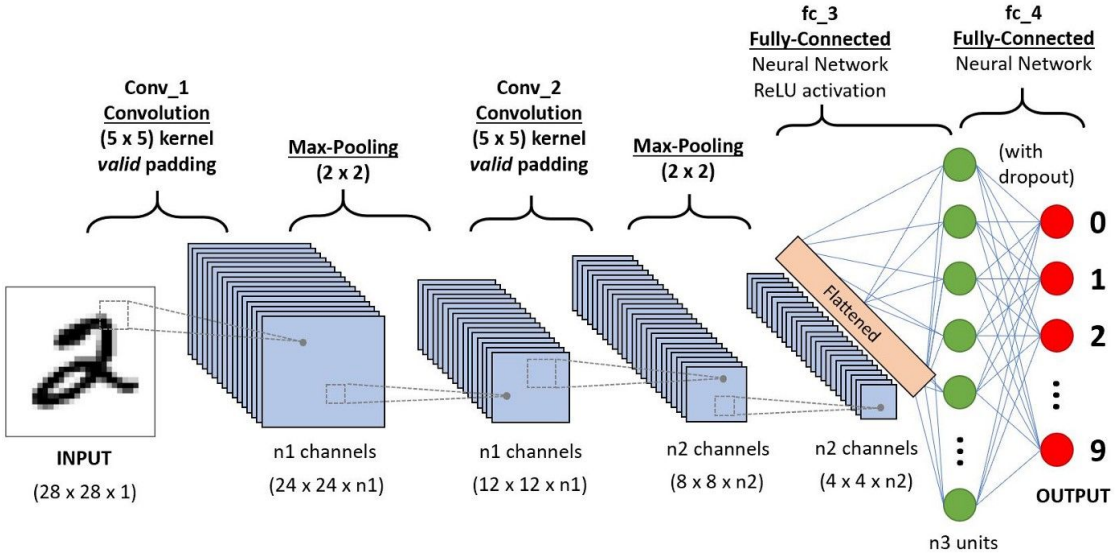
Why Convolutional Neural Networks ?

Rather Why not Fully connected NN?





Fully Connected NN

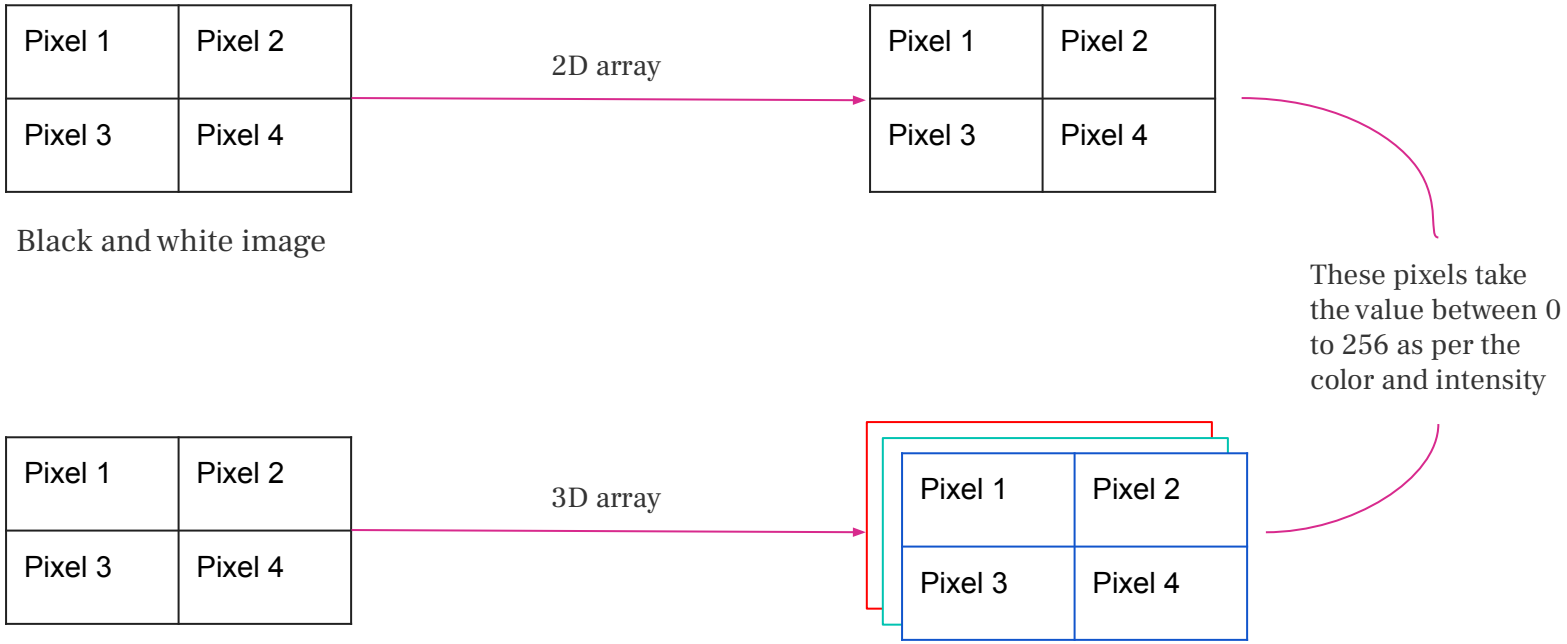


Convolutional NN

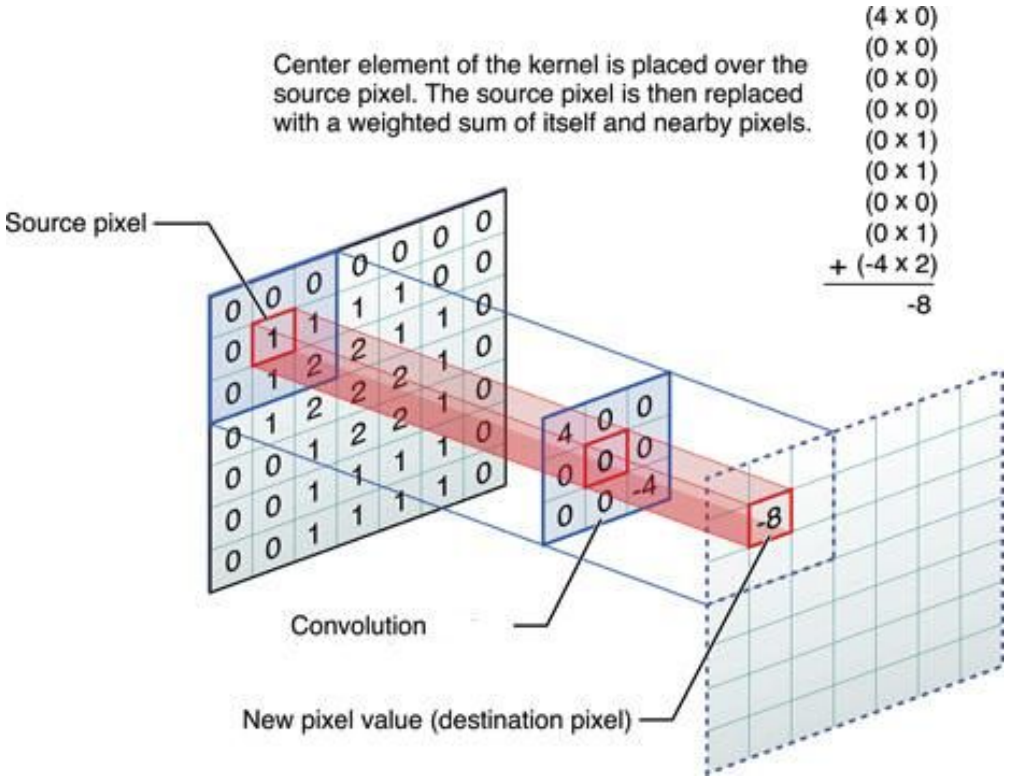
03

Layers of *convolutional neural networks*

How does CNN work?



Convolutional layer



Convolutional layer

0	0	0	0	0	0
0	1	1	1	1	0
0	1	2	2	1	1
0	1	2	2	2	1
0	0	1	2	2	1
0	0	1	1	1	1

Original image

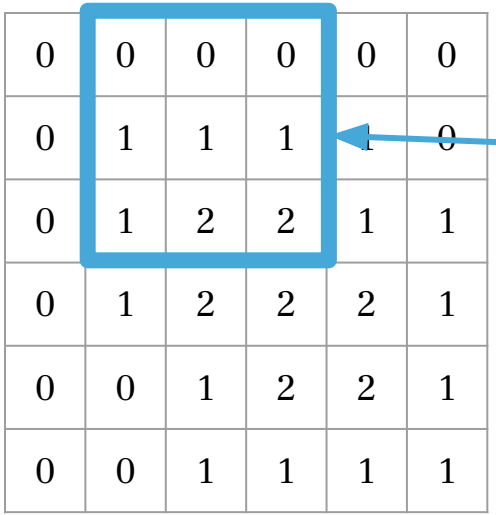
4	0	0
0	0	0
0	0	-4

Kernel / Filter / Feature detector

After applying kernel :

(0*4) + (0*0) + (0*0) +
(0*0) + (1*0) + (1*0) +
(0*0) + (1*0) + (2*-4) = -8

Convolutional layer



0	0	0	0	0	0
0	1	1	1	1	0
0	1	2	2	1	1
0	1	2	2	2	1
0	0	1	2	2	1
0	0	1	1	1	1

Original image

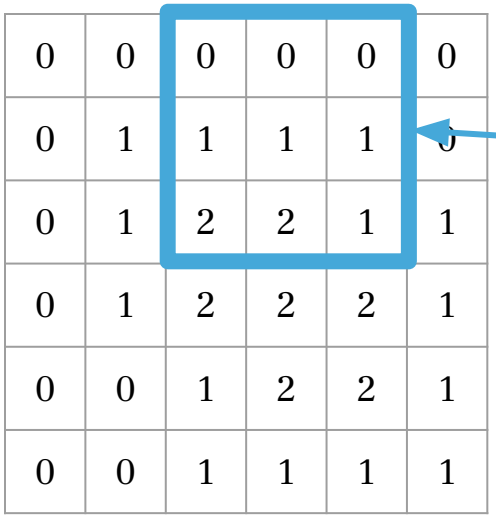
4	0	0
0	0	0
0	0	-4

Kernel / Filter / Feature detector

After applying kernel :

(0*4) + (0*0) + (0*0) +
(1*0) + (1*0) + (1*0) +
(1*0) + (2*0) + (2*-4) = -8

Convolutional layer



0	0	0	0	0	0
0	1	1	1	1	0
0	1	2	2	1	1
0	1	2	2	2	1
0	0	1	2	2	1
0	0	1	1	1	1

Original image

4	0	0
0	0	0
0	0	-4

Kernel / Filter / Feature detector

After applying kernel :

(0*4) + (0*0) + (0*0) +
(1*0) + (1*0) + (1*0) +
(2*0) + (2*0) + (1*-4) = -4

And so on …

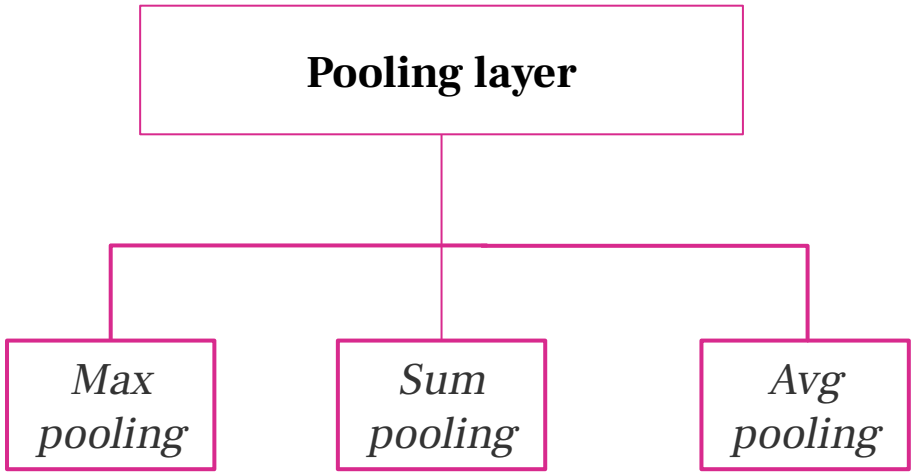
Convoluted image

-8	-8	-4	-4
-8	-4	-4	0
-4	-4	0	4
-4	0	4	4

- We get the convoluted image or the feature map after applying the kernel.
- The first convolutional layer consists of many such feature maps, created with different kernels.

Pooling layer

- Reducing the dimensions of feature map
- Helps decrease computations and reduces overfitting.



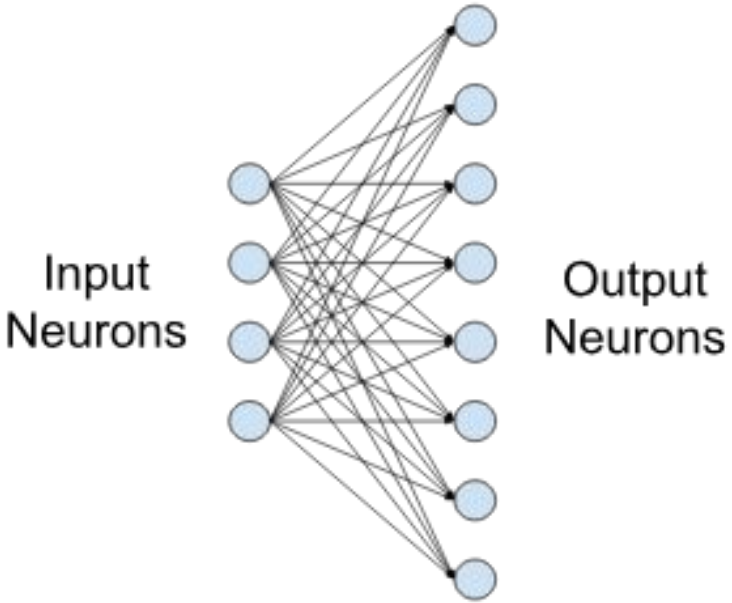
Pooling layer

Flattening

- Used to convert multi dimensional feature map into 1 dimensional array.
- The flattened feature vector serves as the input to one or more fully connected layers.

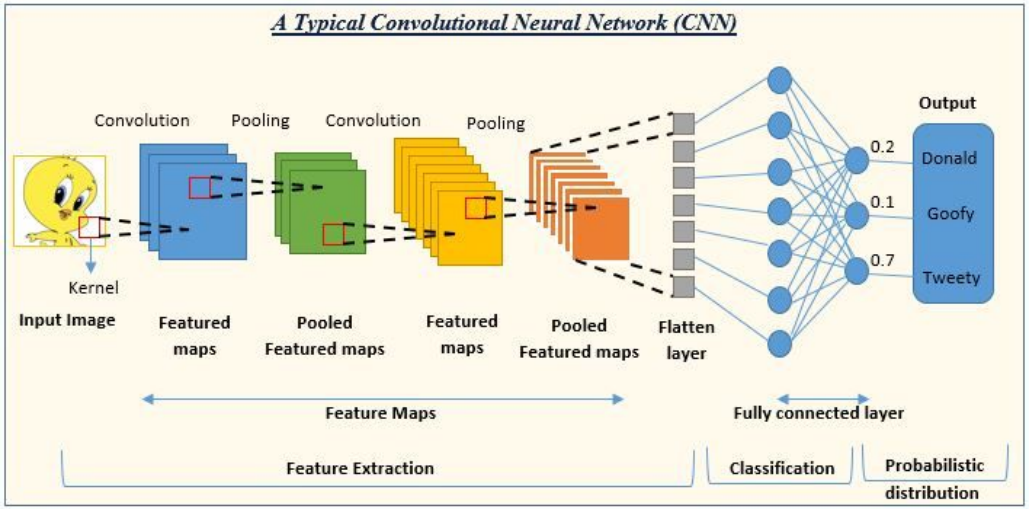
Fully connected layer

- It's like a hub where all the information from the previous layers comes together.
- It takes all the information it gets and tries to make sense of it. It looks for patterns and relationships between the features.



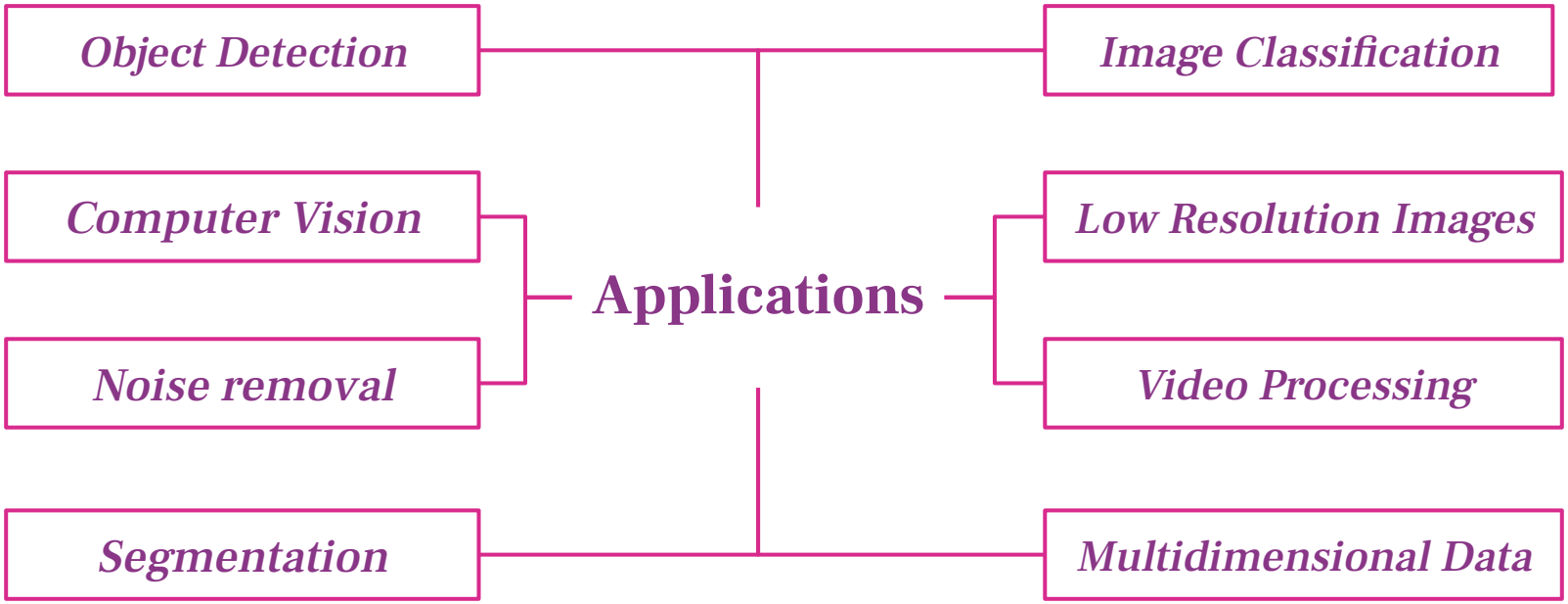
Output layer

- The output layer is like the decision maker of the network. It gives us the final answer based on everything the network has learned from the input data.
- For example, if the CNN is trained to classify images of animals, the output might say, "This image is most likely a cat with 80% confidence."



04

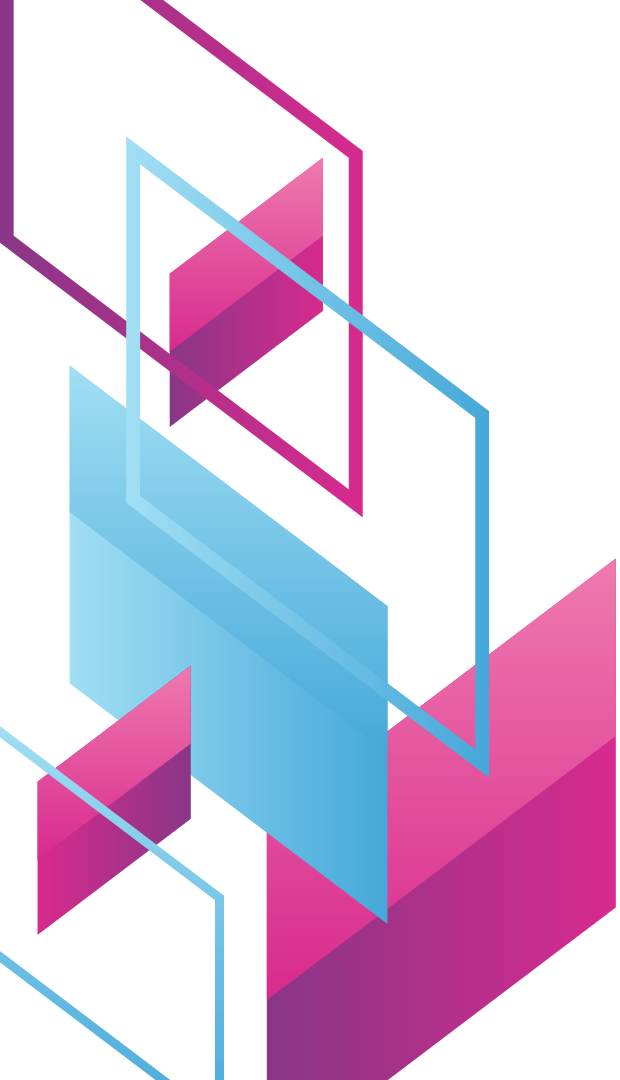
Illustrative example of *convolutional neural networks*


Object Detection
Computer Vision
Noise removal
Segmentation
Applications
Image Classification
Low Resolution Images
Video Processing
Multidimensional Data

Example

- The example below uses MNIST dataset.
- It is widely used for handwriting recognition

- https://adamharley.com/nn_vis/cnn/2d.html



Thank You!

Do you have any questions?