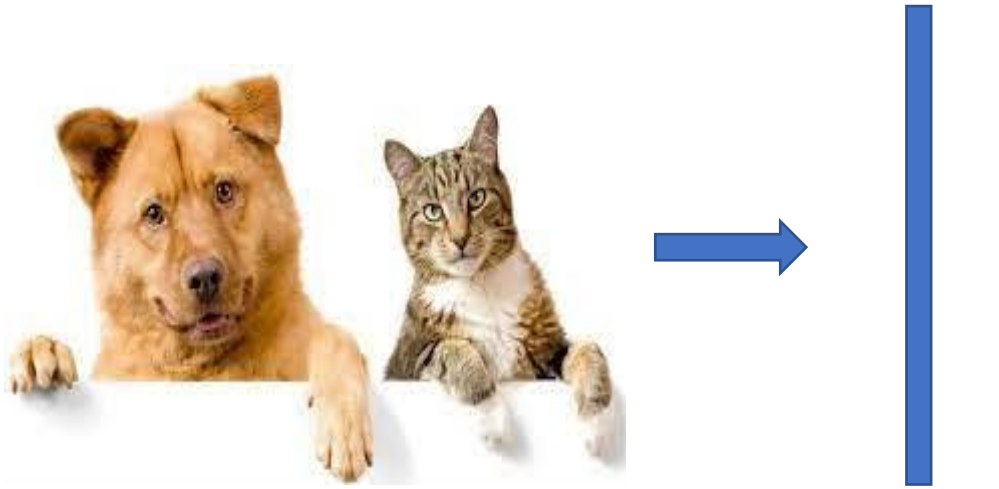
Convolutional Neural Network(CNN)

Two Concerns

- Large number of parameters while applying MLP on domains like images
- Need of extracting high level localized features

Large number of parameters

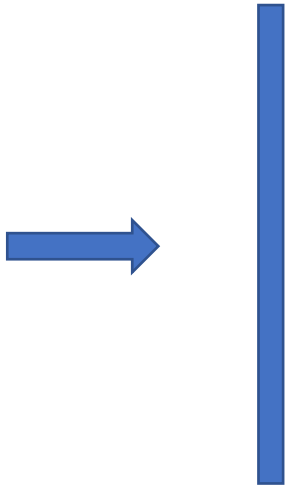


1024x1024x3

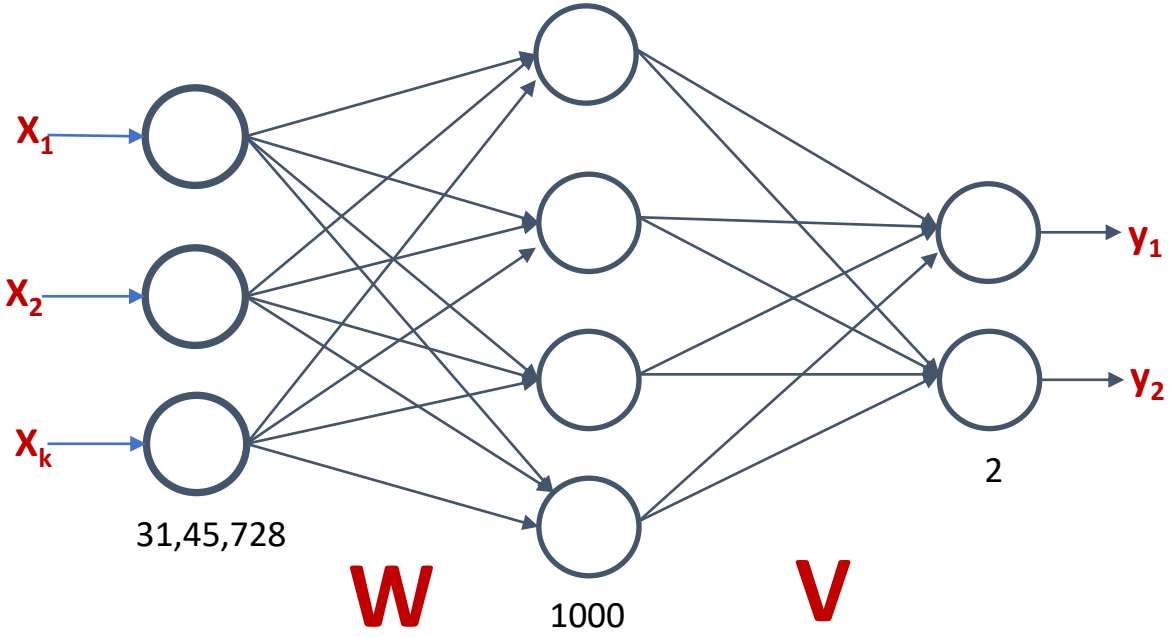
31,45,728

Large number of parameters

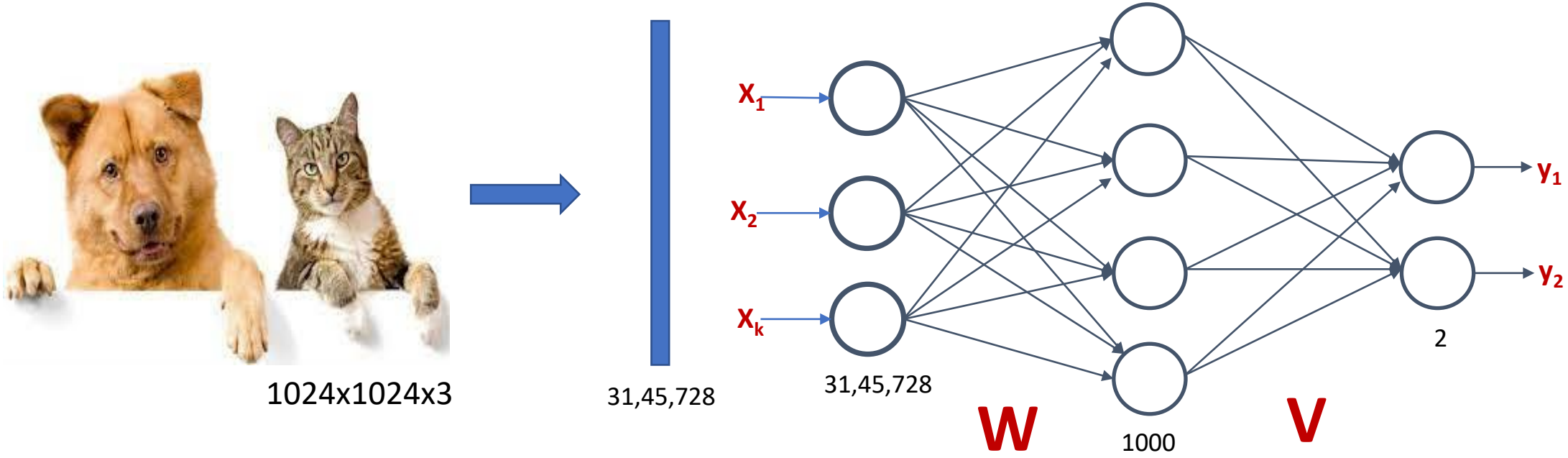
1024x1024x3



31,45,728



Large number of parameters



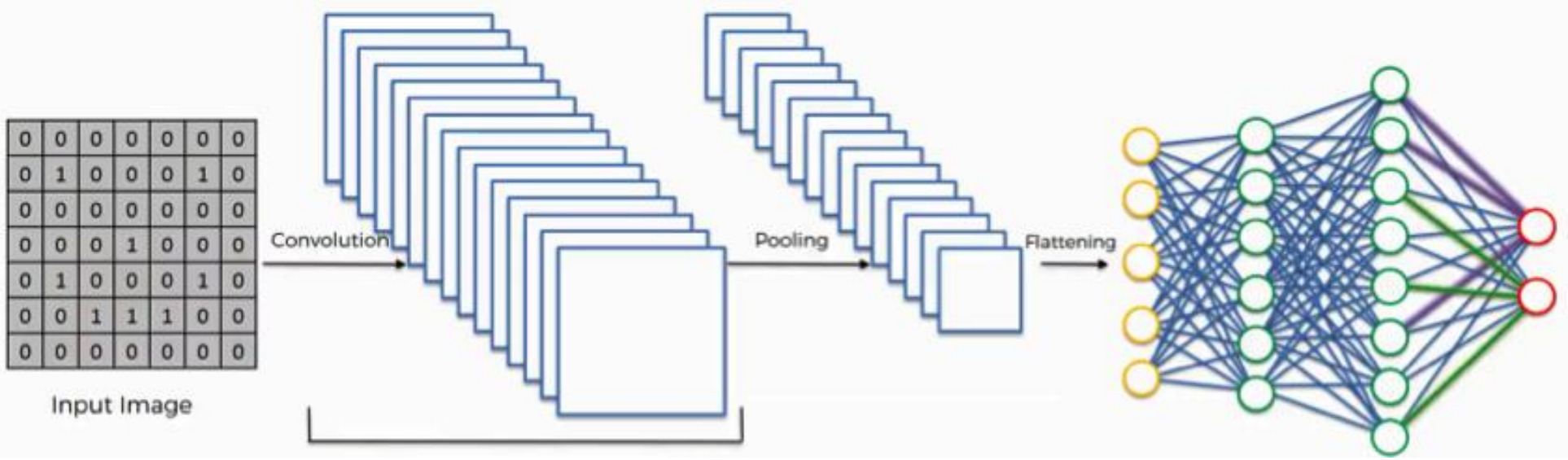
Parameters : **W** matrix of order **3145728x1000** and **V** matrix of the order **1000x2**

Need of extracting features from localize region

Need of extracting features from localize region

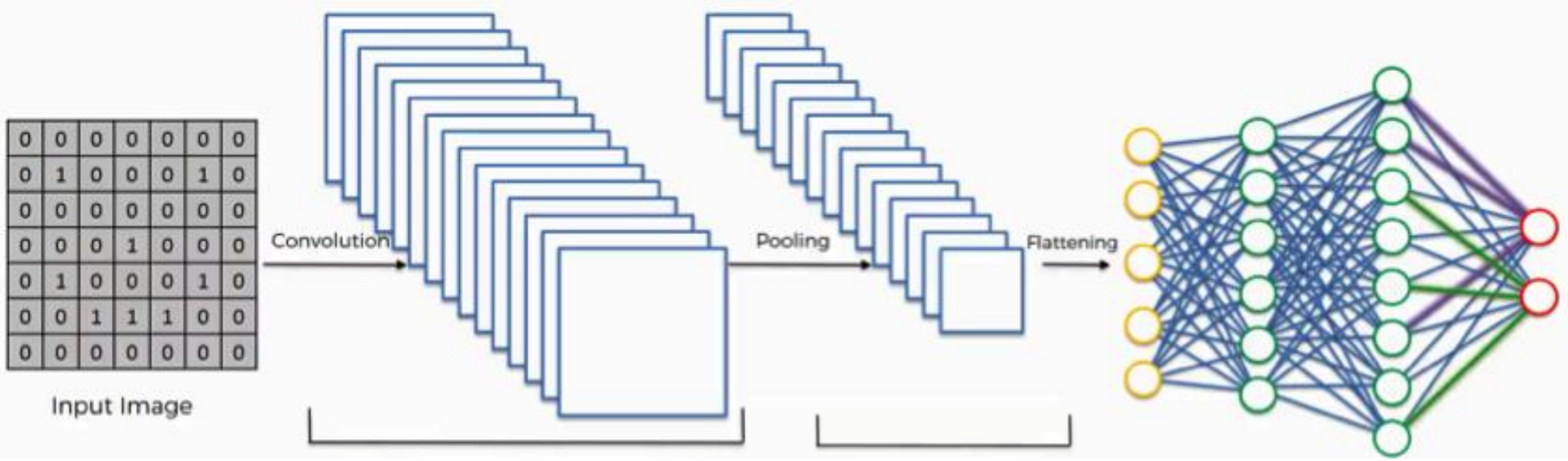
CNN tried to address these two concerns by extracting localized features.

CNN has three basic layers



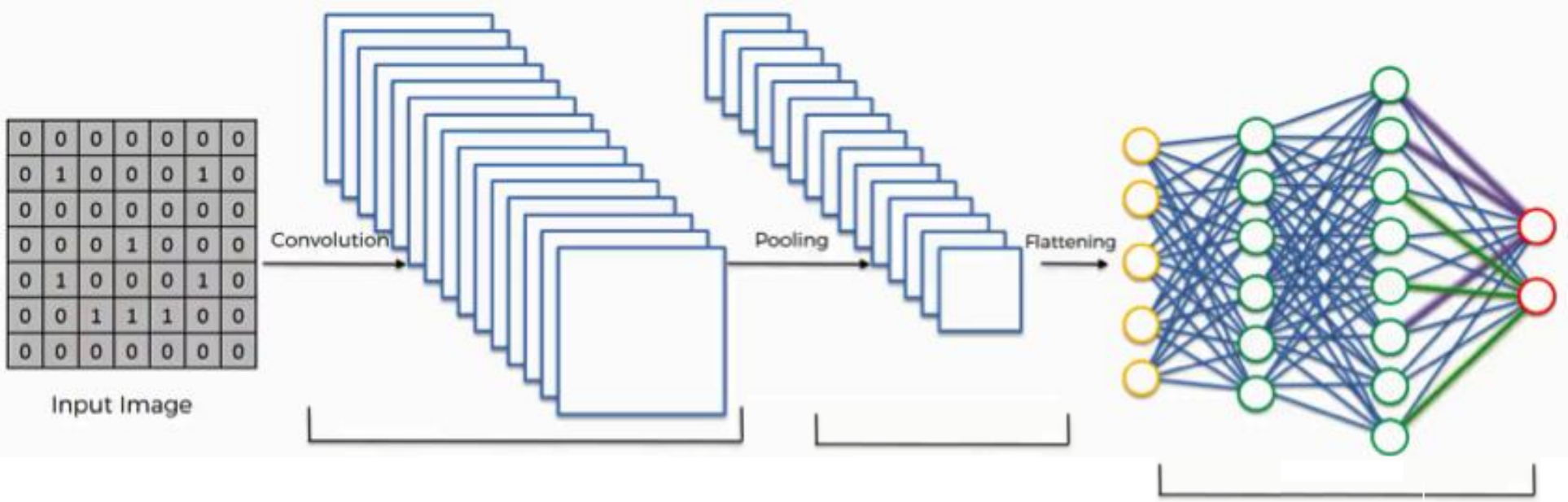
Convolutional Layer

CNN has three basic layers



Convolutional Layer **Pooling Layer**

CNN has three basic layers

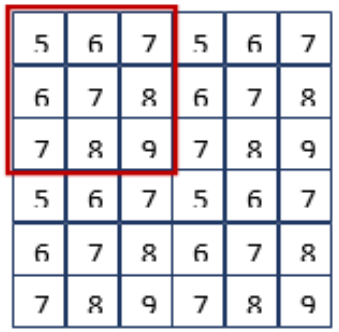


Convolutional Layer

Pooling Layer

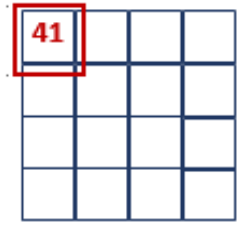
Dense Layer

Convolution Layer



Input Matrix A

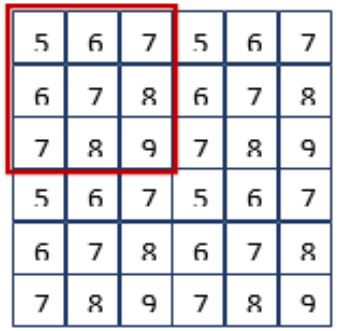
Filter F



Convolution C

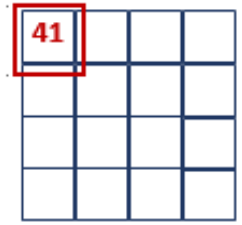
- Filters/Kernels define the local region for extracting localized features.
- They move around the input image to high level extract features

Convolution Layer



Input Matrix A

Filter F

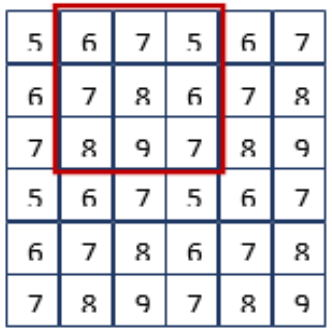


Convolution C

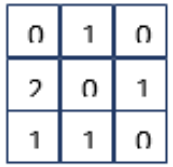
$$
\begin{aligned}
&C_{00} \\
&= A_{00}F_{00} + A_{01}F_{01} + A_{02}F_{02} \\
&+ A_{10}F_{10} + A_{11}F_{11} + A_{12}F_{12} \\
&+ A_{20}F_{20} + A_{21}F_{21} + A_{22}F_{22}
\end{aligned}
$$

Extracted feature is defined by the summation of the element wise multiplication between the filter and the local region

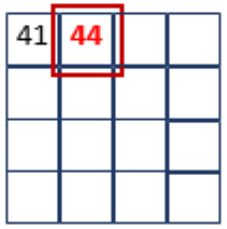
Convolution Layer



Input Matrix A



Filter F

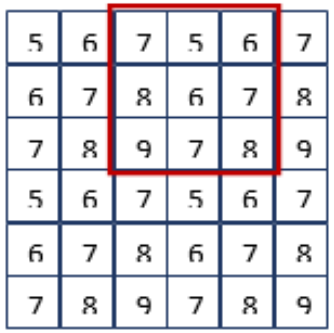


Convolution C

$$C_{01} = A_{01}F_{00} + A_{02}F_{01} + A_{03}F_{02} + A_{11}F_{10} + A_{12}F_{11} + A_{13}F_{12} + A_{21}F_{20} + A_{22}F_{21} + A_{23}F_{22}$$

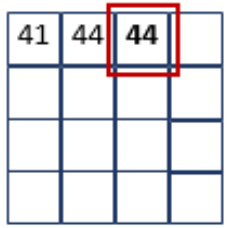
- The position of the shift of the filter is known as **Stride.**
- This example has **stride length =1**

Convolution Layer



Input Matrix A

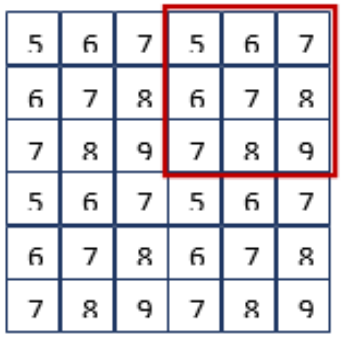
Filter F



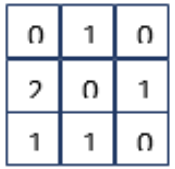
Convolution C

$$C_{02} = A_{02}F_{00} + A_{03}F_{01} + A_{04}F_{02} + A_{12}F_{10} + A_{13}F_{11} + A_{14}F_{12} + A_{22}F_{20} + A_{23}F_{21} + A_{24}F_{22}$$

Convolution Layer



Input Matrix A

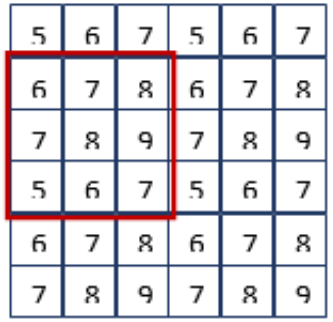


Filter F

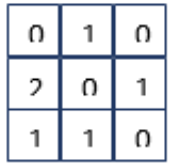
Convolution C

$$
\begin{aligned}
&C_{03} \\
&= A_{03}F_{00} + A_{04}F_{01} + A_{05}F_{02} \\
&+ A_{13}F_{10} + A_{14}F_{11} + A_{15}F_{12} \\
&+ A_{23}F_{20} + A_{24}F_{21} + A_{25}F_{22}
\end{aligned}
$$

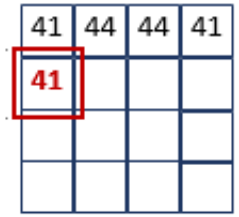
Convolution Layer



Input Matrix A



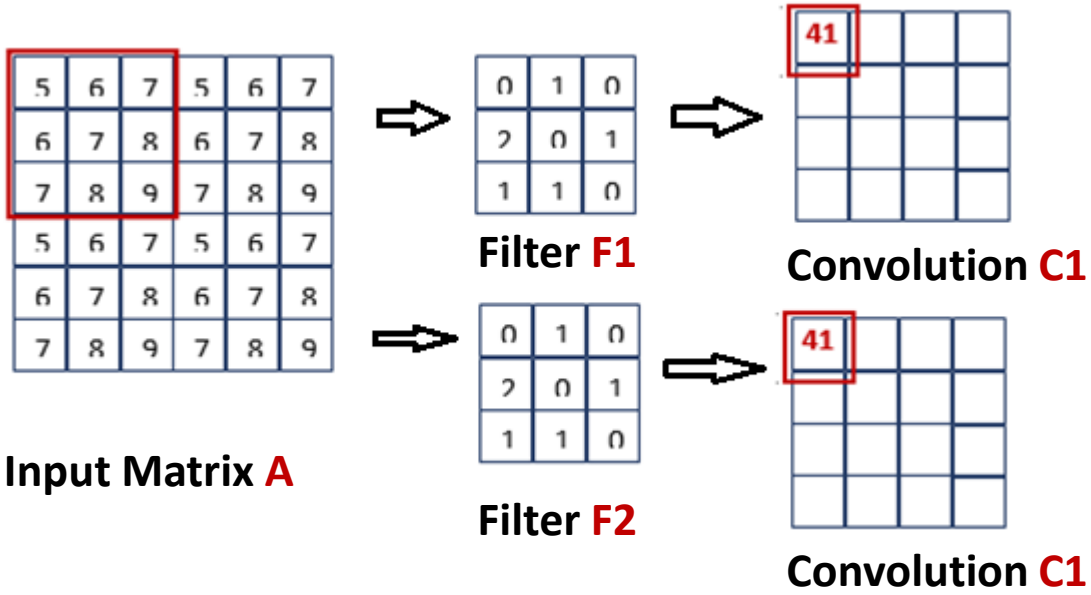
Filter F



Convolution C

$$
\begin{aligned}
&C_{10} \\
&= A_{10}F_{00} + A_{11}F_{01} + A_{12}F_{02} \\
&+ A_{20}F_{10} + A_{21}F_{11} + A_{22}F_{12} \\
&+ A_{30}F_{20} + A_{31}F_{21} + A_{32}F_{22}
\end{aligned}
$$

Multiple Filters

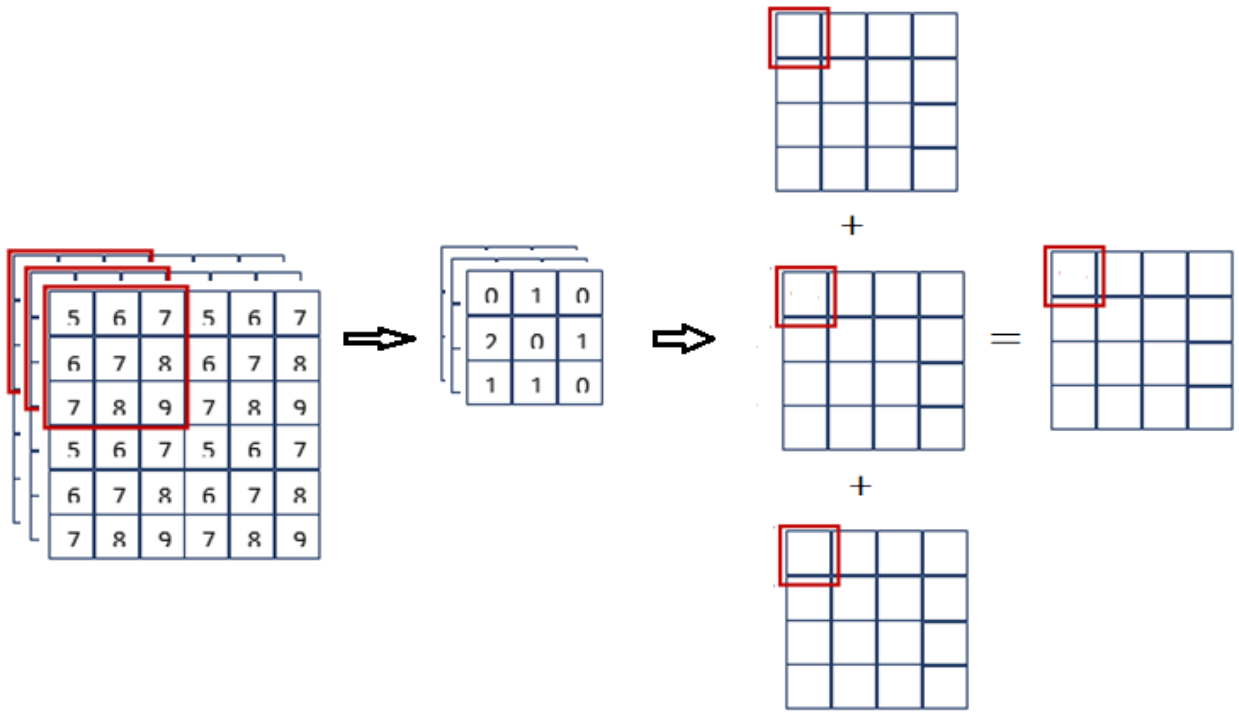


$$C1_{10} = A_{10}F1_{00} + A_{11}F1_{01} + A_{12}F1_{02} + A_{20}F1_{10} + A_{21}F1_{11} + A_{22}F1_{12} + A_{30}F1_{20} + A_{31}F1_{21} + A_{32}F1_{22}$$

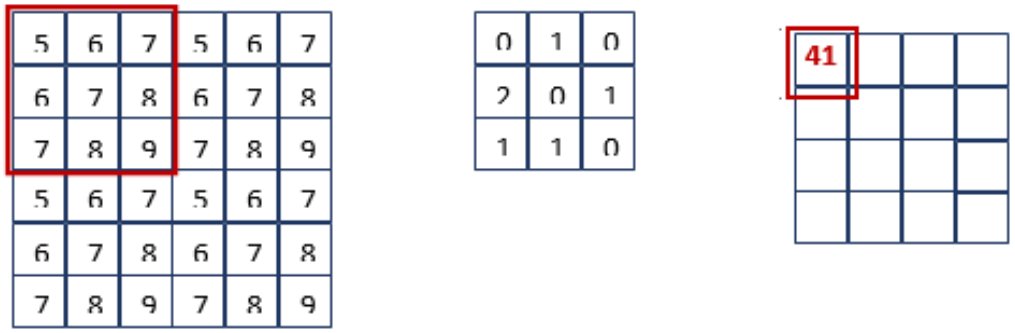
$$C2_{10} = A_{10}F2_{00} + A_{11}F2_{01} + A_{12}F2_{02} + A_{20}F2_{10} + A_{21}F2_{11} + A_{22}F2_{12} + A_{30}F2_{20} + A_{31}F2_{21} + A_{32}F2_{22}$$

- **Multiple filters of different sizes can be applied.**
- **Each filter focuses on extracting features of different types (Say, eyes, mounth)**

Input Matrix could be multi dimensional



Output of the convolution Layer

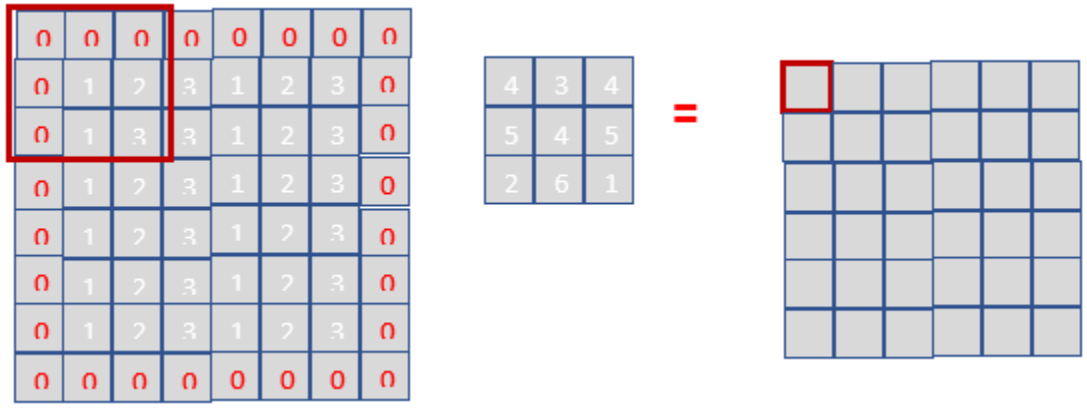


Input Matrix = 6 x 6
Filter = 3 x 3
Stride = 1
Padding = 0

Output Matrix = 4 x 4 (the number of shifts)

Size of the output matrix is smaller than the input.

Padding – add Zeros



Input Matrix = 6 x 6
Filter = 3 x 3
Stride = 1
Padding = 1

Output Matrix = 6 x 6 (the number of shifts)

Pooling

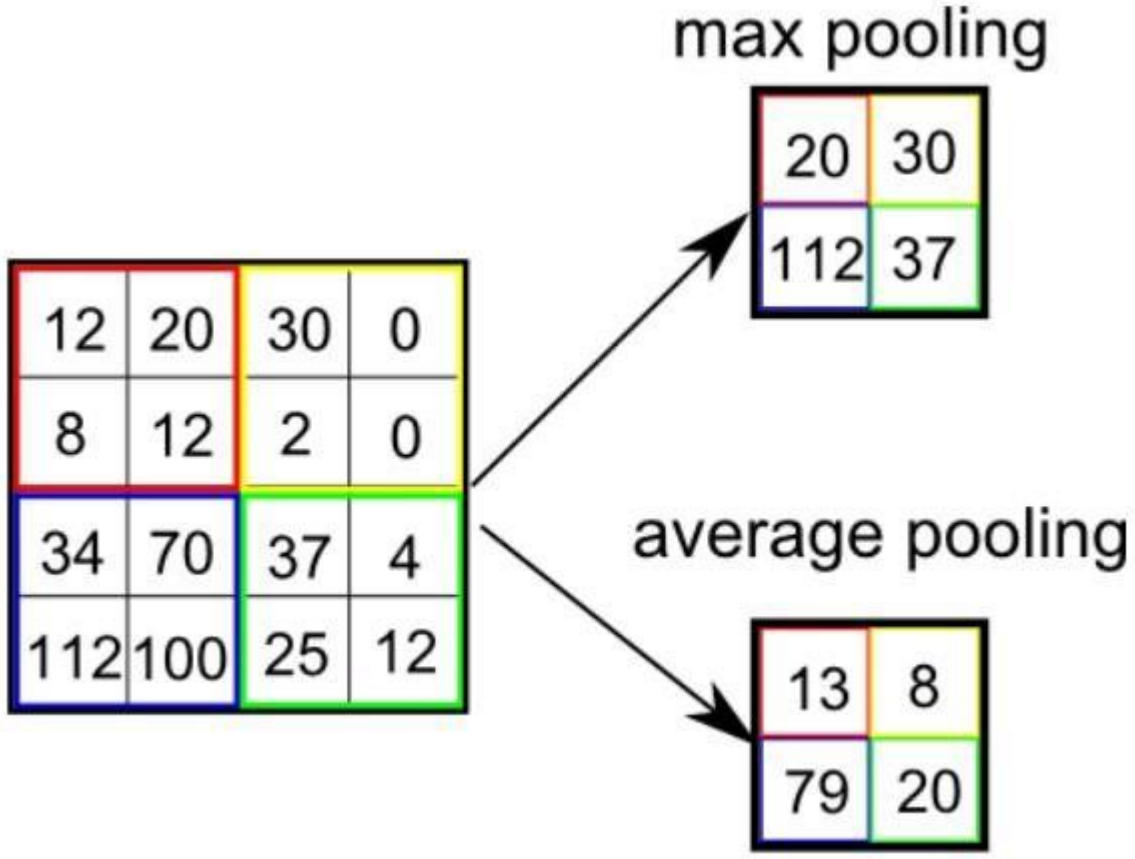
- **Reduces** the spatial size of the Convolved Features.
- **Decreases** the computational power required to process the data**.**
- **Extracts dominant features** which are rotational and positional invariant

Two types of Pooling

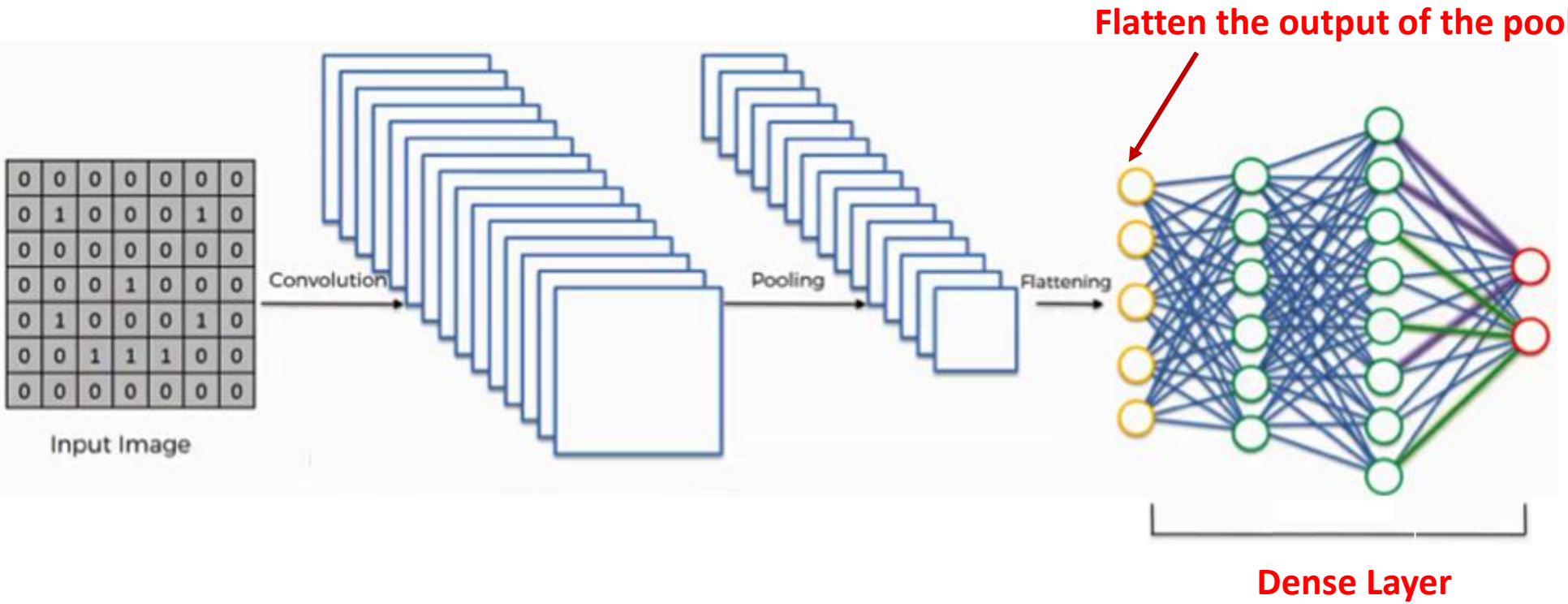
- **Max Pooling : maximum value** from the portion covered by the Kernel
  - Remove noise by removing non-dominant features

- **Average Pooling : average of all the values** from the portion covered by the Kernel
  - Remove noise by averaging

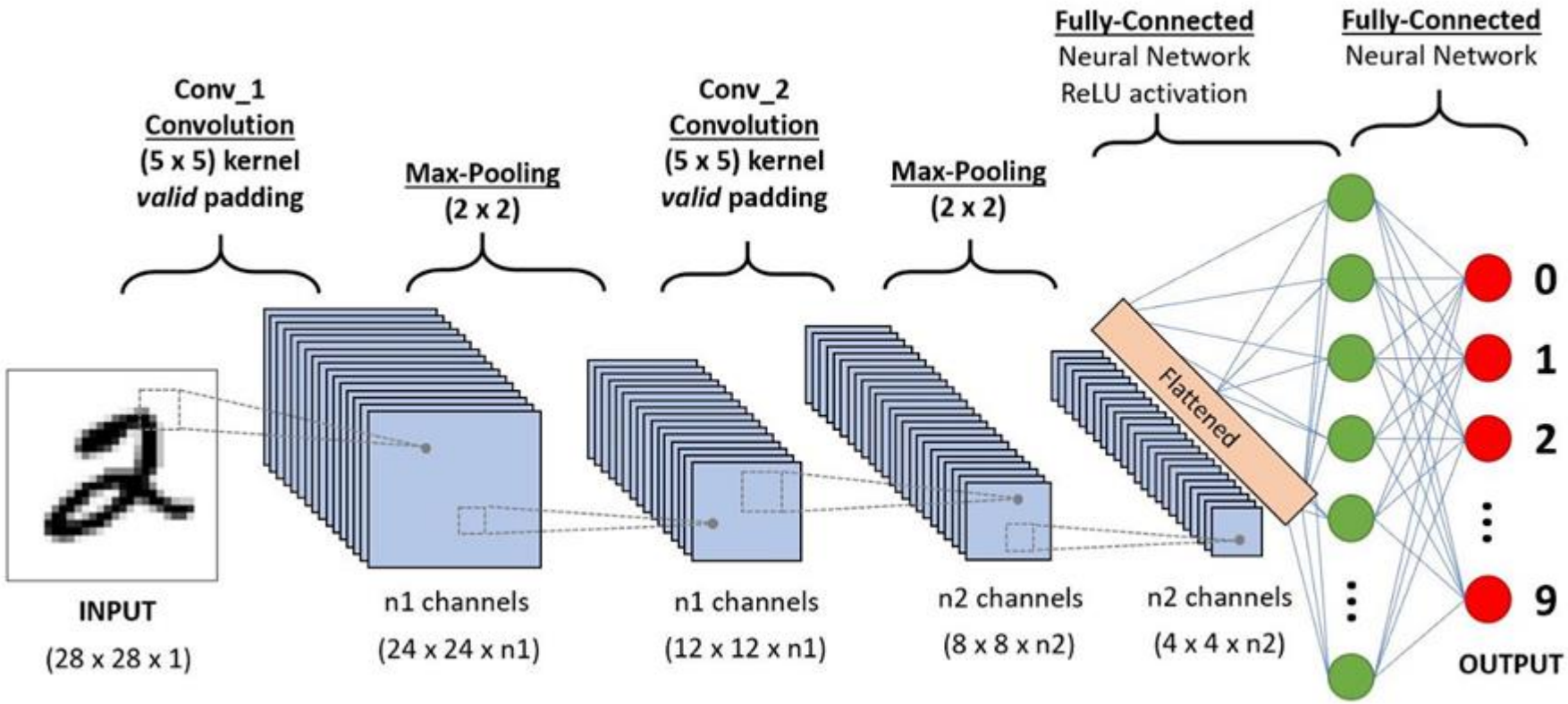
Two types of Pooling



Flattening and Dense Layer



Deep CNN


Fully-Connected
Neural Network
ReLU activation
Fully-Connected
Neural Network
Conv_1
Convolution
(5 x 5) kernel
valid padding
Max-Pooling
(2 x 2)
Conv_2
Convolution
(5 x 5) kernel
valid padding
Max-Pooling
(2 x 2)
Flattened
0
1
2
9
INPUT
(28 x 28 x 1)
n1 channels
(24 x 24 x n1)
n1 channels
(12 x 12 x n1)
n2 channels
(8 x 8 x n2)
n2 channels
(4 x 4 x n2)
OUTPUT

Summary

Three Components

- **Convolution:** Extract high level features from local regions
- **Pooling:** Reduce dimension and denoise
- **Flattening and Dense layer**: Design the network for the underlying task