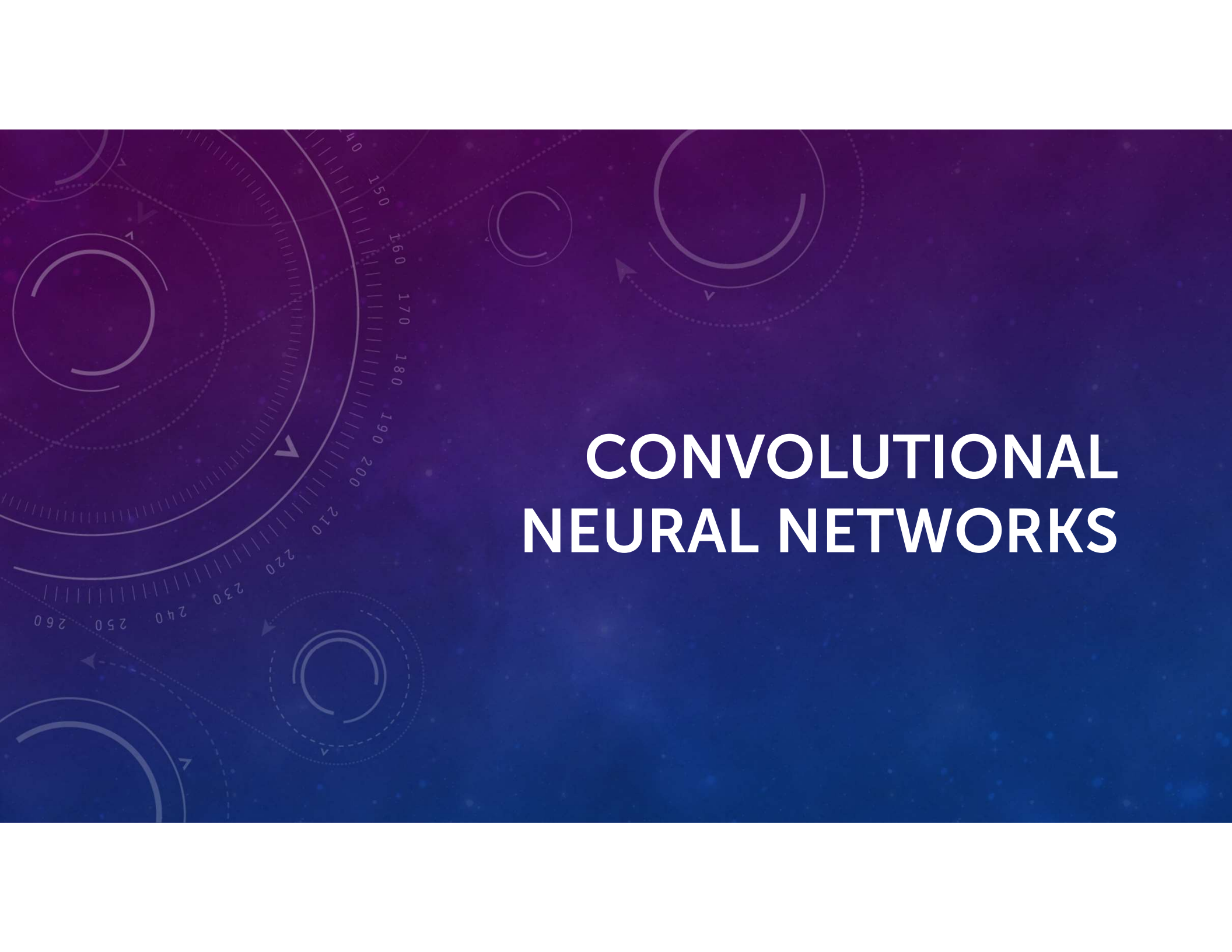




CONVOLUTIONAL NEURAL NETWORKS

OVERVIEW

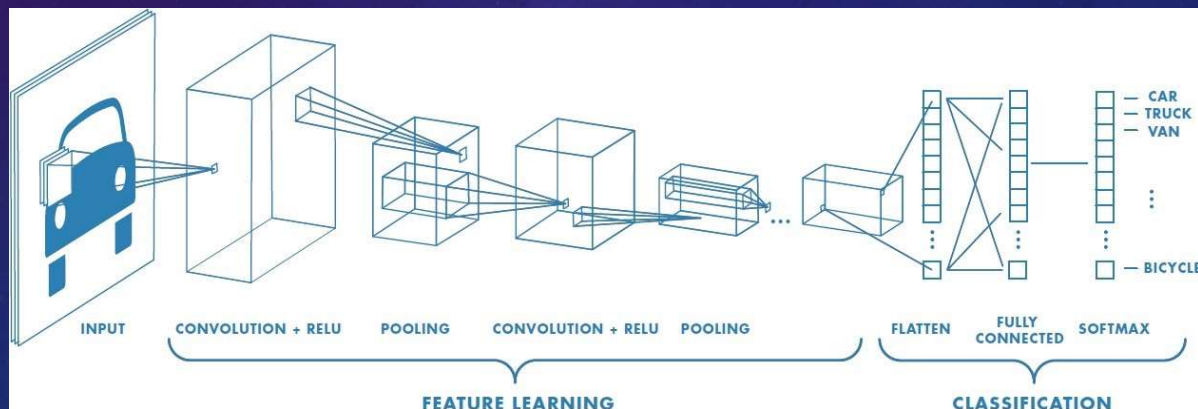
- The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution.
- Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers.
- In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

HOW IT WORKS

- CNN image classifications takes an input image, process it and classify it under certain categories (E.g., Dog, Cat, Tiger, Lion).
- Computers sees an input image as array of pixels, and it depends on the image resolution.
- Based on the image resolution, it will see $h \times w \times d$ (h = Height, w = Width, d = Dimension).
E.g., An image of $6 \times 6 \times 3$ array of matrix of RGB (3 refers to RGB values) and an image of $4 \times 4 \times 1$ array of matrix of grayscale image.

HOW IT WORKS (CONT.)

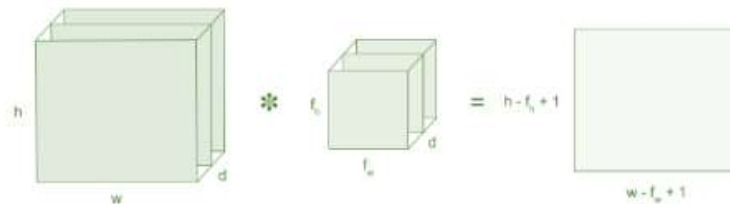
- Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1.
- The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.



CONVOLUTION LAYER

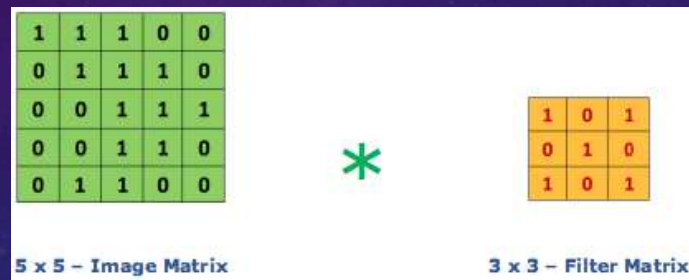
- Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

- An image matrix (volume) of dimension $(h \times w \times d)$
- A filter $(f_h \times f_w \times d)$
- Outputs a volume dimension $(h - f_h + 1) \times (w - f_w + 1) \times 1$



CONVOLUTION LAYER (CONT.)

- Consider a 5 x 5 whose image pixel values are 0, 1 and filter matrix 3 x 3 as shown in below



The diagram illustrates a convolution operation. On the left is a 5x5 Image Matrix with green cells containing the values: $\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$. In the center is a green asterisk representing multiplication. On the right is a 3x3 Filter Matrix with orange cells containing the values: $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$. Below the matrices are the labels "5 x 5 - Image Matrix" and "3 x 3 - Filter Matrix".

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

5 x 5 - Image Matrix

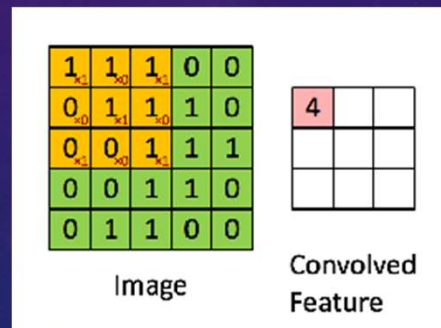
*

1	0	1
0	1	0
1	0	1

3 x 3 - Filter Matrix

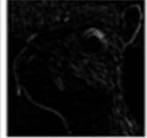
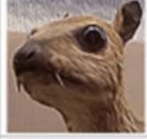
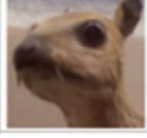
CONVOLUTION LAYER (CONT.)

- Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called "Feature Map" as output shown in below



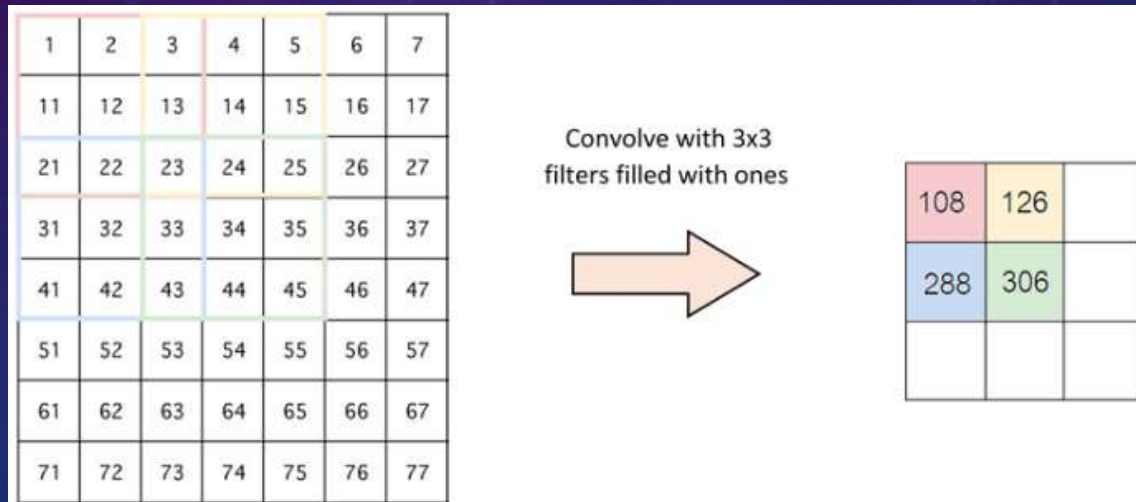
CONVOLUTION LAYER (CONT.)

- Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters.
- The below next shows various convolution image after applying different types of filters (Kernels).

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)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

STRIDES

- Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2.

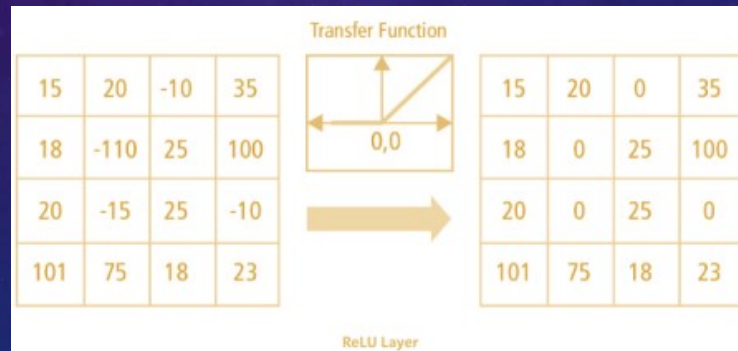


PADDING

- Sometimes filter does not fit perfectly fit the input image. We have two options:
 - Pad the picture with zeros (zero-padding) so that it fits
 - Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.

NON LINEARITY (RELU)

- ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = \max(0, x)$.
- Why ReLU is important : ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real-world data would want our ConvNet to learn would be non-negative linear values.

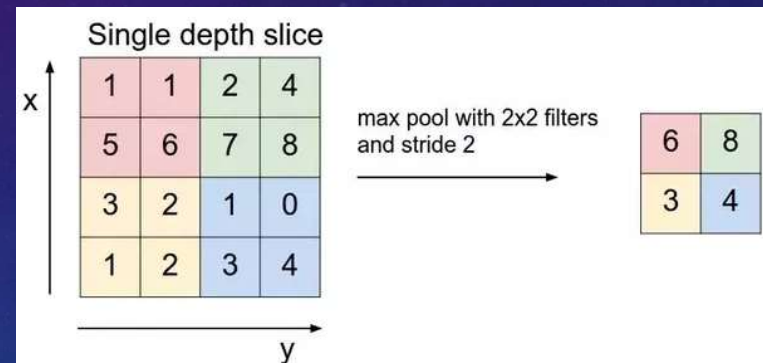


POOLING LAYER

- Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

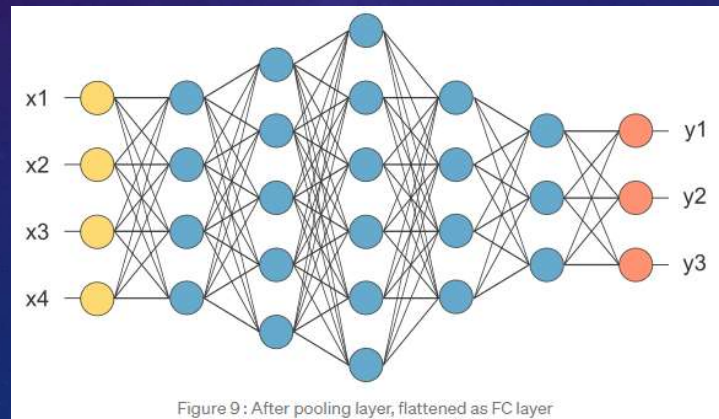
- Max Pooling
- Average Pooling
- Sum Pooling

- Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.



FULLY CONNECTED LAYER

- The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like a neural network.
- In the diagram below, the feature map matrix will be converted as vector ($x_1, x_2, x_3, \dots$). With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as softmax or sigmoid to classify the outputs as cat, dog, car, truck etc.,



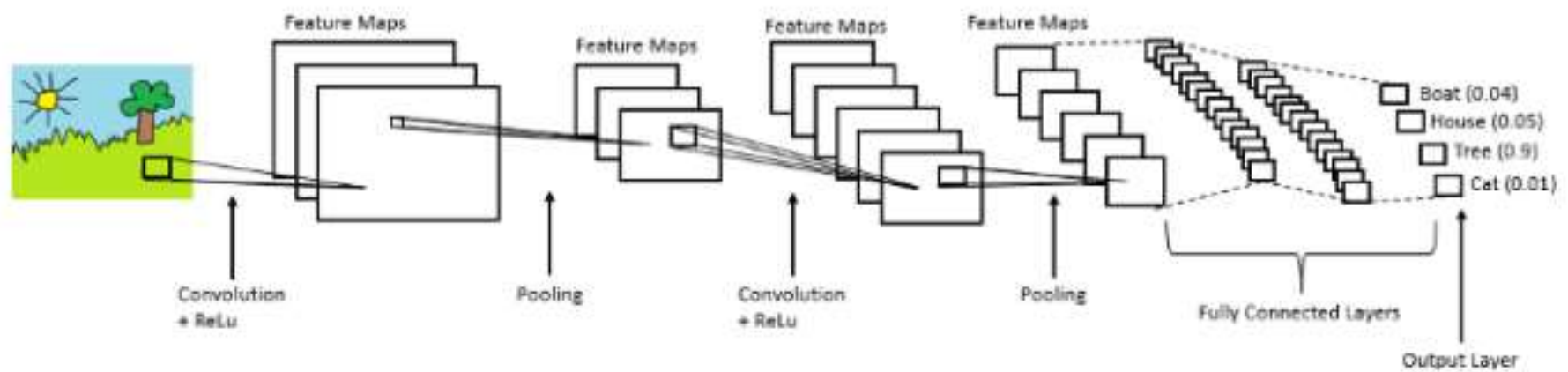
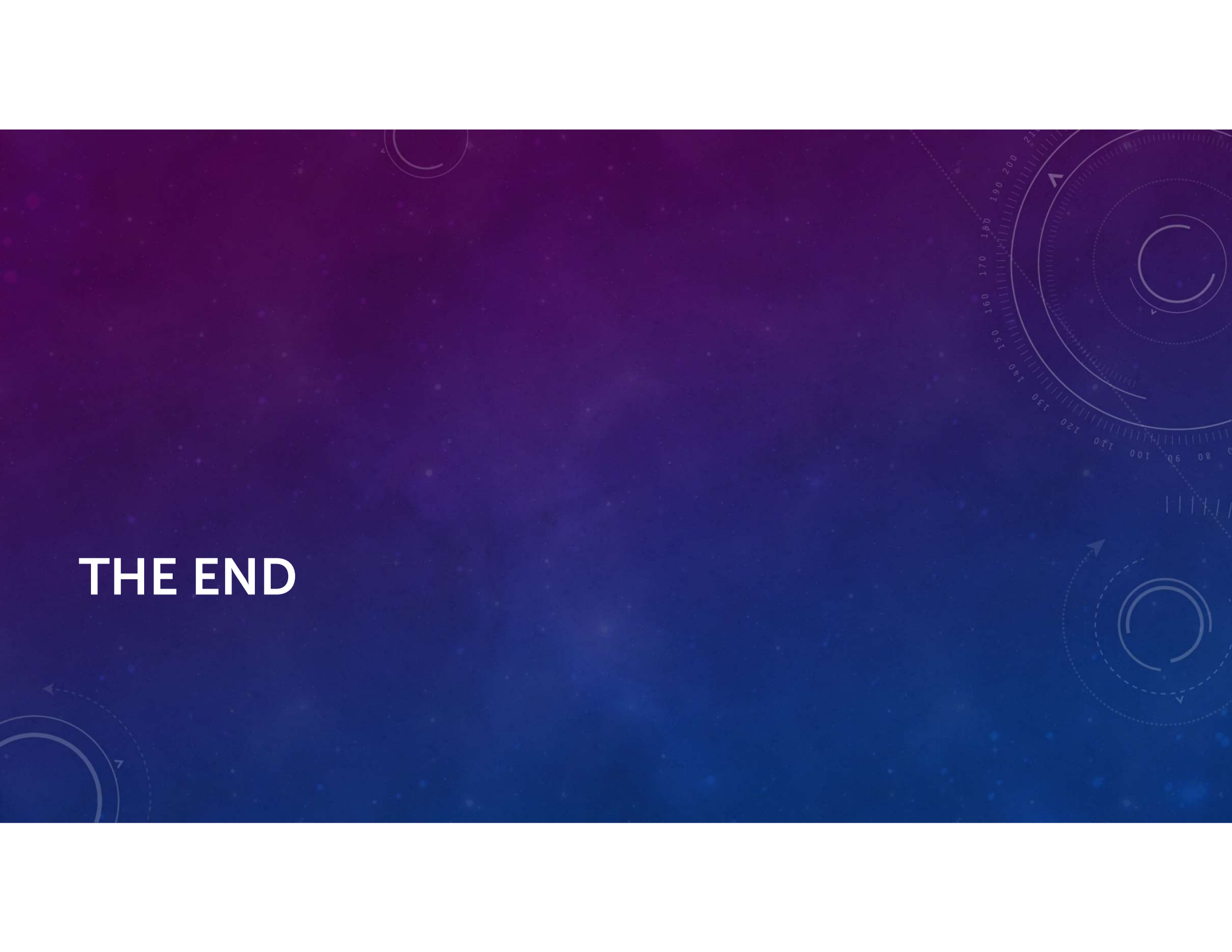


Figure 10 : Complete CNN architecture



THE END