MNIST Handwritten Digit Classification using Convolutional Neural Network (CNN)

Presented By:
Sujan Kumar Roy, Ph. D. (Australia), M. A. Sc. (Canada)
Department of Computer Science and Engineering
University of Rajshahi, Bangladesh

http://rurfid.ru.ac.bd/ru_profile/public/teacher/

Outlines

- ❑ What is CNN?
- ❑ Why CNN over Simple Feed-Forward NN (FFNN)
- ❑ MNIST Digit Classification using CNN
  - ○ CNN Architecture Details
  - ○ Convolution Operation
  - ○ Pooling Operation
- ❑ Task to be completed Today.

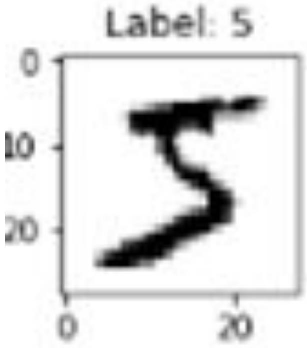
Ref:https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

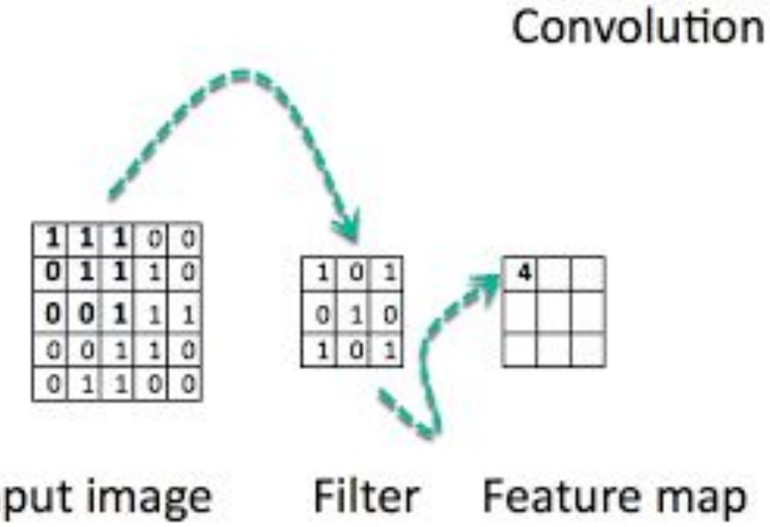
What is CNN?

- CNN, is a well-known method in computer vision applications.
- It is a class of deep neural networks that are used to analyze visual imagery.
- This type of architecture is dominant to recognize objects from a picture or video.
- It is used in applications like image or video recognition, neural language processing, etc.

Why CNN Over FNN?

- In cases of extremely basic binary images, FFNN might show an average precision score while performing prediction of classes.
- when it comes to complex images having pixel dependencies throughout (shapes/textures)– FFNN reduces prediction accuracy.
- A CNN is able to **successfully capture the Spatial and Temporal dependencies** in an image through the application of relevant filters.





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

1	0	1
0	1	0
1	0	1

4		

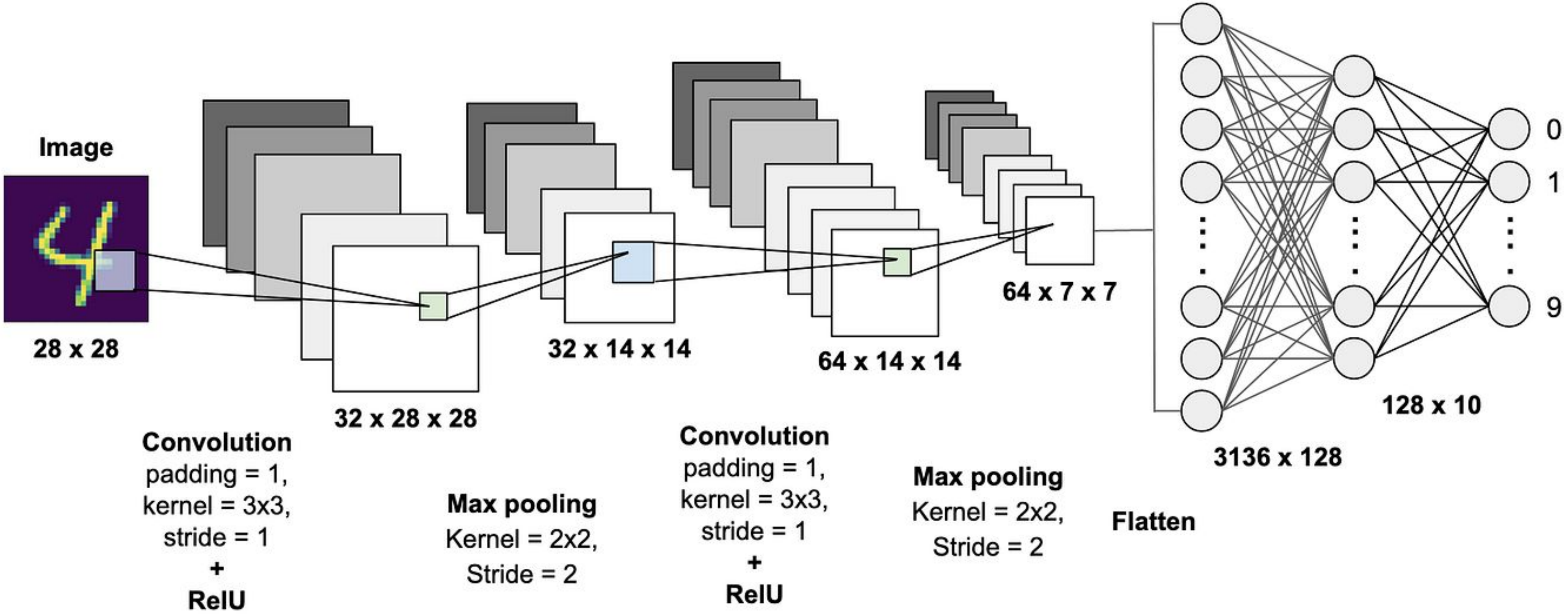
MNIST Digit Classification with CNN (1/9)

❑ **You will follow the steps below for image classification using CNN:**

- Step 1: Upload Dataset
- Step 2: Input layer
- Step 3: Convolutional layer
- Step 4: Pooling layer
- Step 5: Second Convolutional Layer and Pooling Layer
- Step 6: Dense layer
- Step 7: Logit Layer

MNIST Digit Classification with CNN (2/9)

Graphical View of CNN steps for handwritten digit classification:



MNIST Digit Classification with CNN (3/9)

Conv2D Function in TensorFlow

```
tf.keras.layers.Conv2D(
    filters,
    kernel_size,
    strides=(1, 1),
    padding='valid',
    data_format=None,
    dilation_rate=(1, 1),
    ……
)
```

- **Filters:** The number of output filters in the convolution– filters can be different, such as, edge detect, horizontal info detect, mixed, etc. (familiar with image processing with filters/kernels).
- **kernel_size:** An integer or tuple/list of 2 integers, specifying the height and width of the 2D convolution window, (3,3) or (5,5), etc.
- Etc.

MNIST Digit Classification with CNN (4/9)

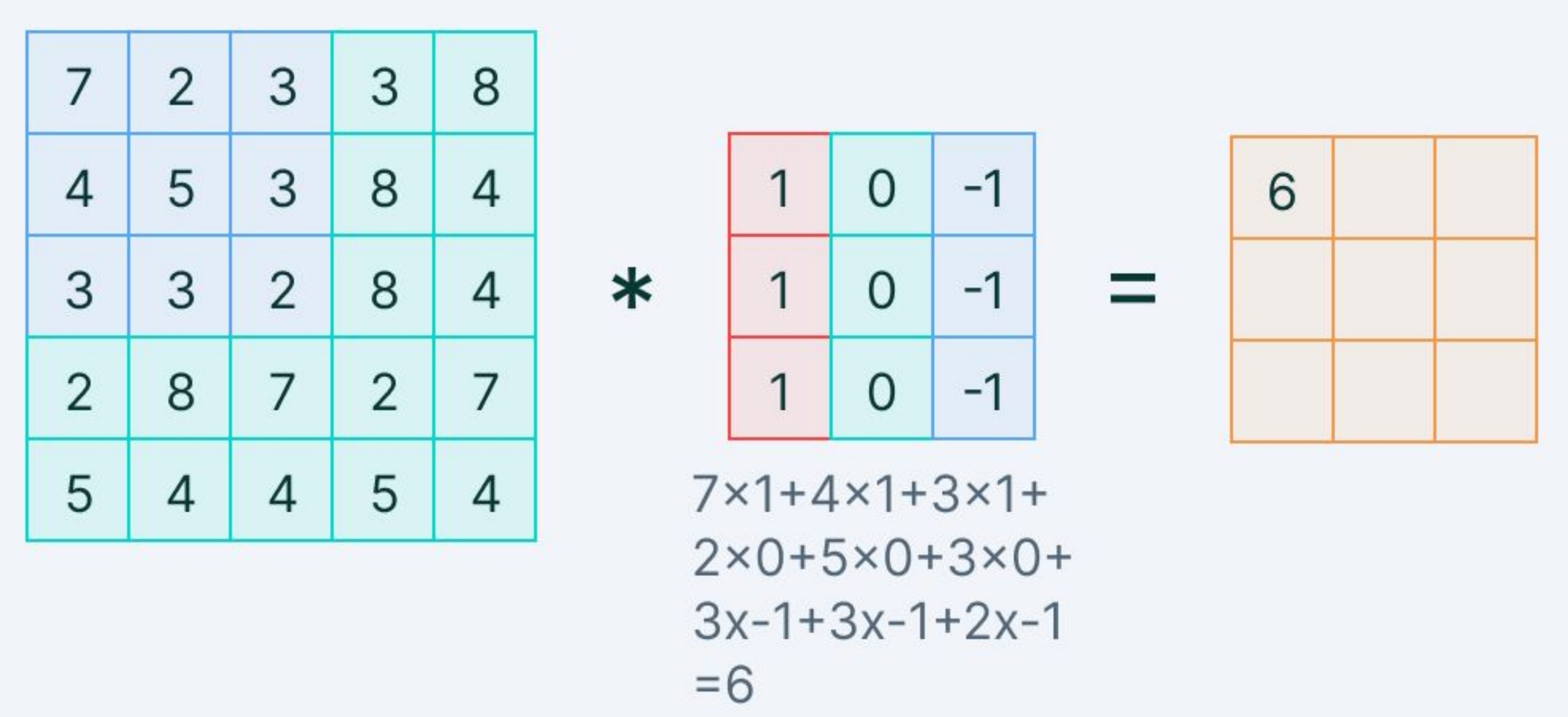
Components of Convolutional Neural Network (ConvNet or CNN)

❑ There are four components of a Convnets

  - Convolution
  - Non Linearity (ReLU)
  - Pooling or Sub Sampling
  - Classification (Fully Connected Layer)

MNIST Digit Classification with CNN (5/9)

Convolution Operation

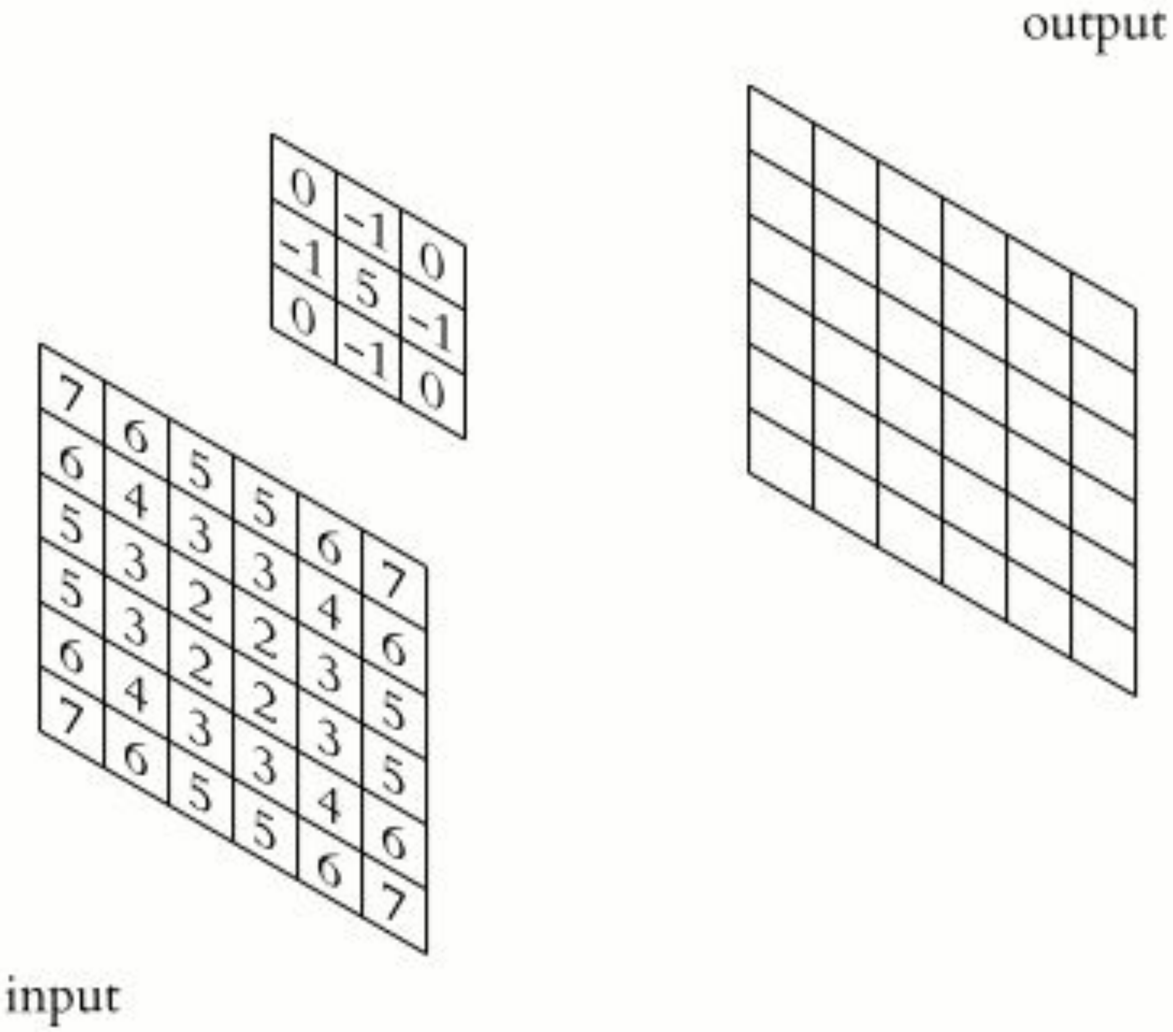


$$\begin{bmatrix}7&2&3&3&8\\4&5&3&8&4\\3&3&2&8&4\\2&8&7&2&7\\5&4&4&5&4\end{bmatrix} * \begin{bmatrix}1&0&-1\\1&0&-1\\1&0&-1\end{bmatrix} = \begin{bmatrix}6&&\\&&\\&&\end{bmatrix}$$

7×1+4×1+3×1+
2×0+5×0+3×0+
3x-1+3x-1+2x-1
=6

MNIST Digit Classification with CNN (6/9)

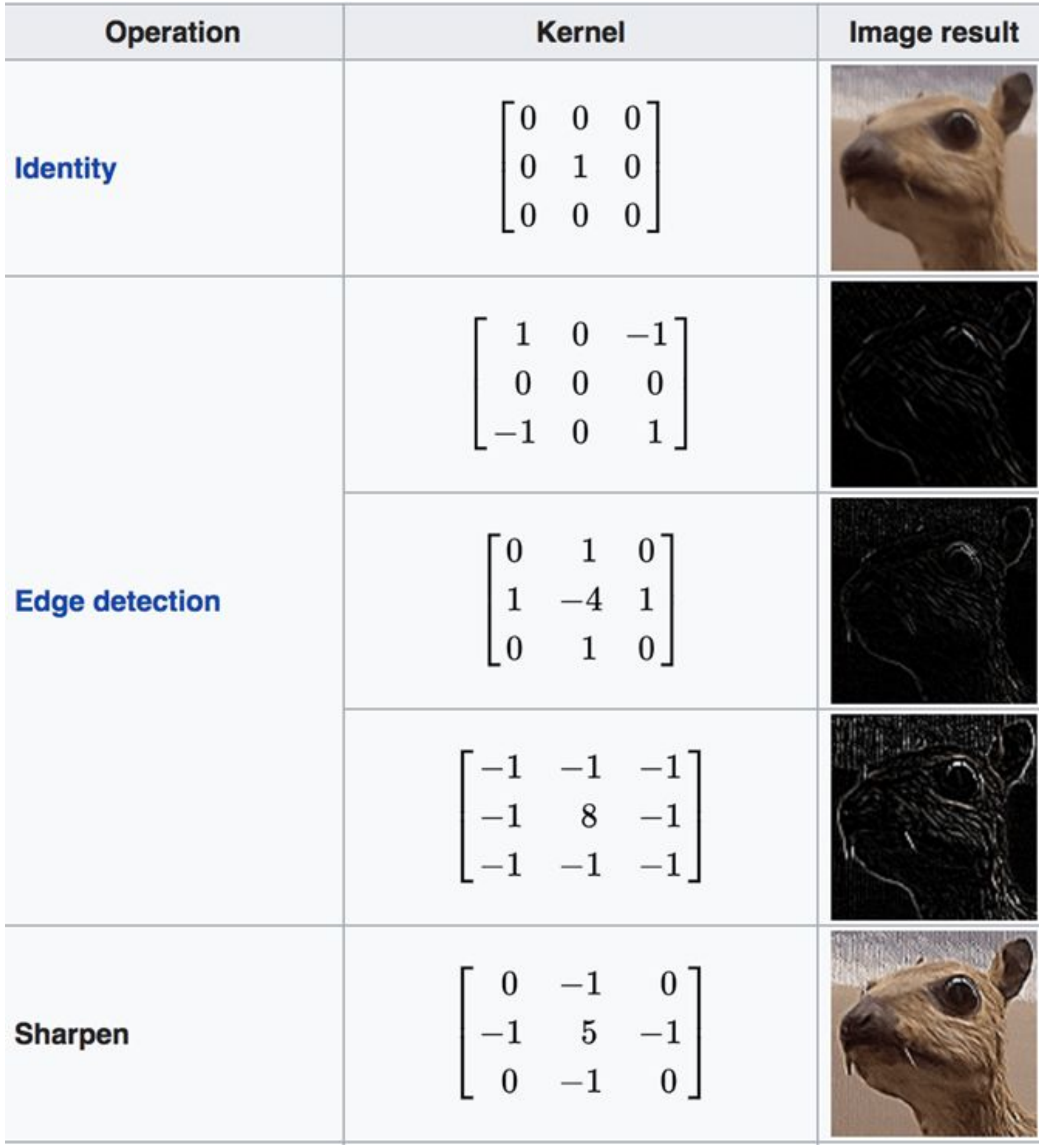
Animated Example of Convolution Operation



0	-1	0
-1	5	-1
0	-1	0

7	6	5	5	6	7
6	4	3	3	4	6
5	3	2	2	3	5
5	3	2	2	3	5
6	4	3	3	4	6
7	6	5	5	6	7

MNIST Digit Classification with CNN (7/9)

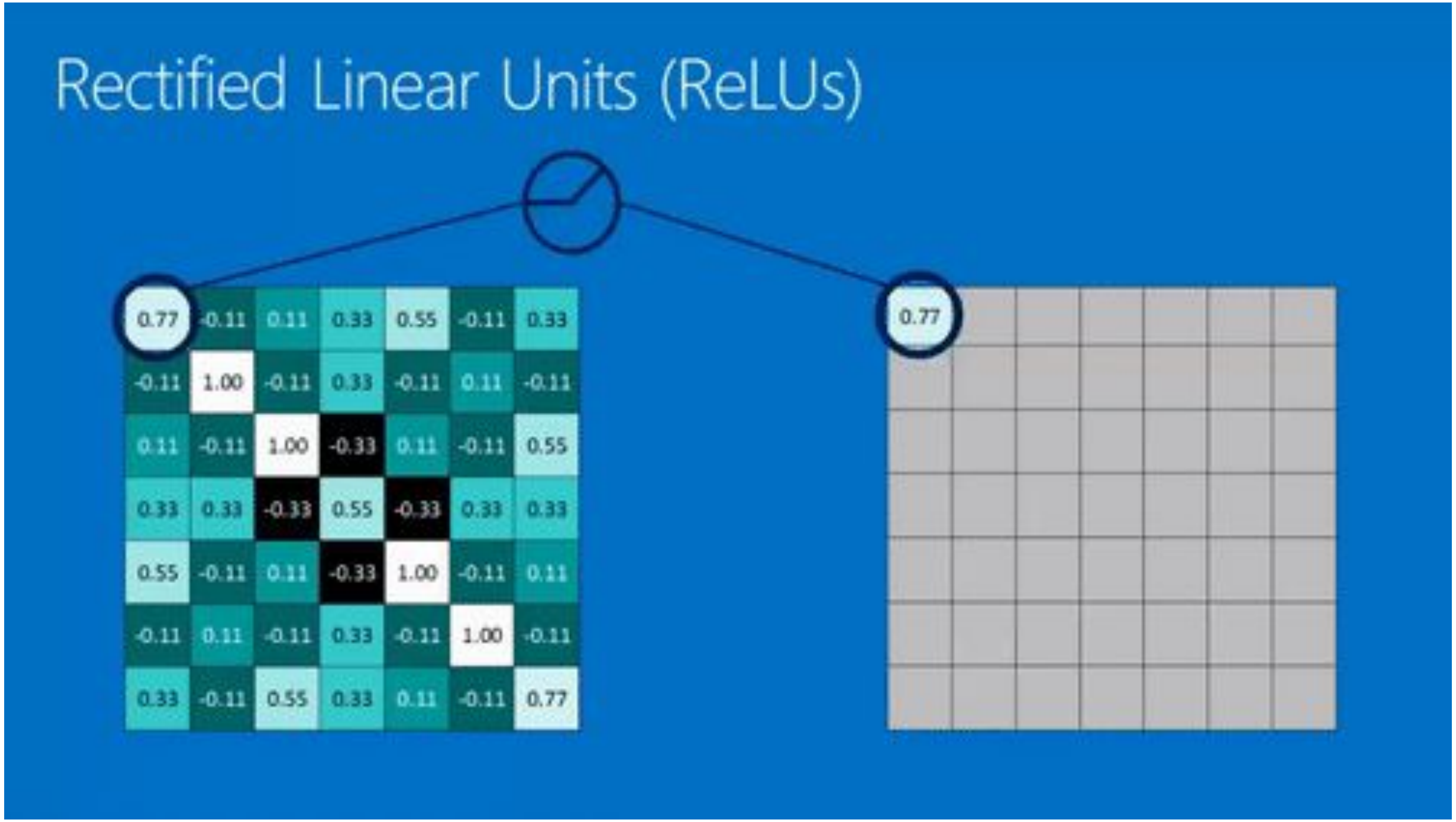


Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	

Examples of Different Convolution Kernels Capturing Complex Patterns (Edge, Identity)

MNIST Digit Classification with CNN (8/9)

ReLU Activation Function



Rectified Linear Units (ReLUs)

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.77						

MNIST Digit Classification with CNN (9/9)

Pooling Operation

Task to be Completed Today

Design a Customize CNN for Handwritten Digit Classification with the following specifications:

1. Generate a CNN model with:

   i. Two CNN hidden layers (Conv2D) of sizes 32, 64 followed by

   ii. ReLU Activation and

   iii. MaxPooling2D with Kernel size (3, 3), and

   iv. Stride= (1,1)

2. Use Flatten Layers to convert the feature map into 1D

3. Use simple Dense layer of size 64 followed by output Dense Layer of size 10 with SoftMax Activation Function

4. Use the MNIST database for training and testing

5. Carefully read the problem specifications and implement the CNN accordingly.