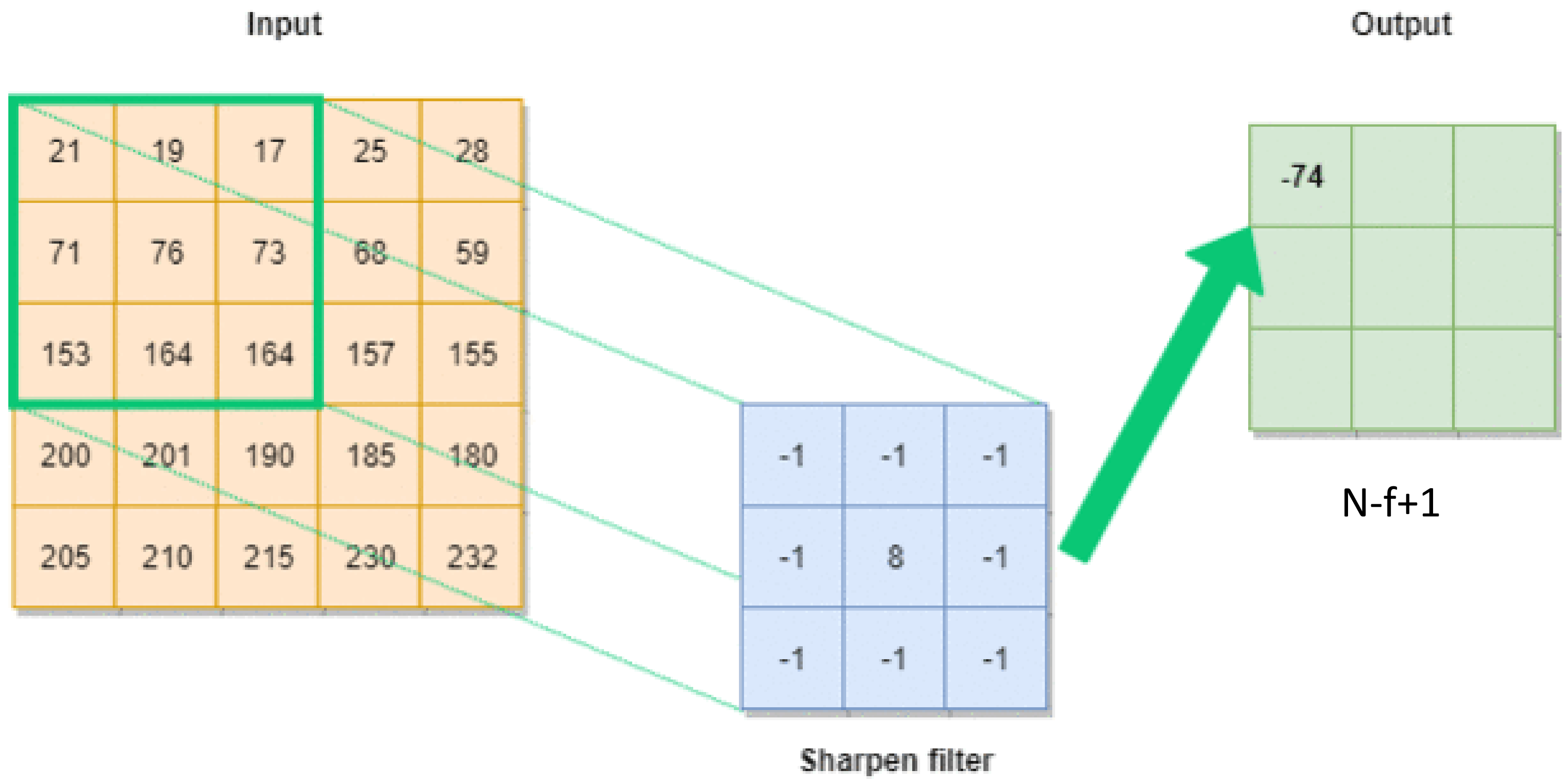
Padding & Strides

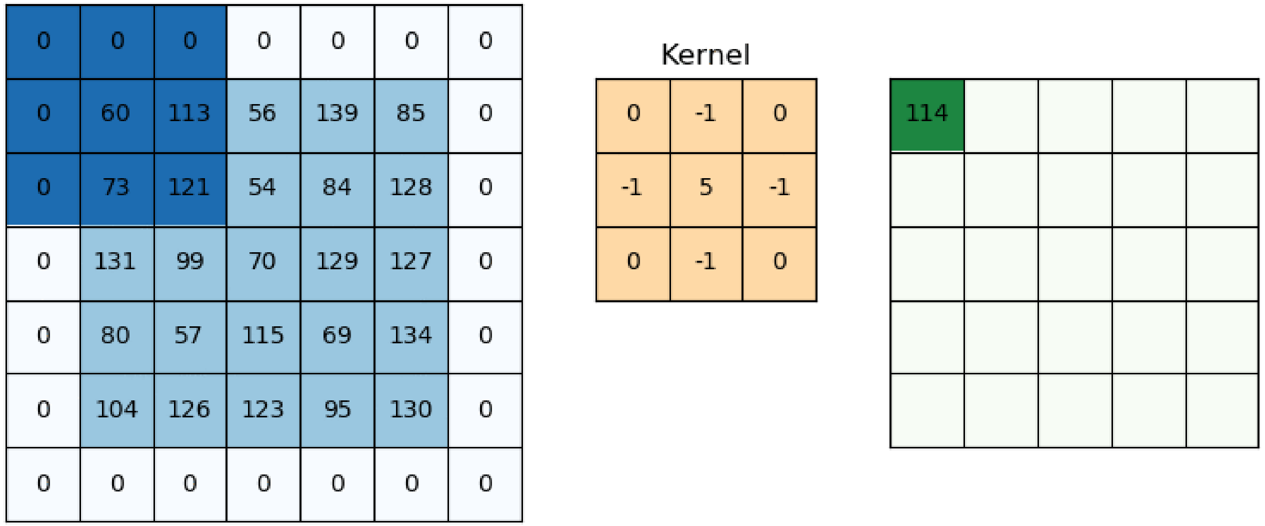
MUKESH KUMAR



Input
21 19 17 25 28
71 76 73 68 59
153 164 164 157 155
200 201 190 185 180
205 210 215 230 232
-1 -1 -1
-1 8 -1
-1 -1 -1
Sharpen filter
Output
-74

N-f+1

Padding/Zero Padding



0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

Kernel

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

114				

Output = N+2p-f+1

Reasons for Using Padding

Preserve Spatial Dimensions:

- Without padding, the spatial dimensions of the output feature map are reduced after each convolution operation. This happens because the filter cannot be applied to the borders of the input image.
- By adding padding, the input image size can be maintained after the convolution. This is particularly useful in deep networks where you want to preserve the original spatial dimensions to maintain the resolution of the feature maps.

Enable Convolution at Borders:

- Padding allows the filter to be applied to the edges of the input image. Without padding, pixels at the border would be used fewer times compared to the central pixels, potentially leading to a loss of important information at the edges.

- By adding padding, each pixel, including those at the borders, gets an equal opportunity to be included in multiple **receptive fields.**

Control Output Size:

- Padding helps in controlling the size of the output feature maps. For instance, by using “same” padding, the output dimensions can be kept the same as the input dimensions, which can simplify the architecture design.

- “Valid” padding (no padding) reduces the size of the feature map, while “same” padding maintains the size.

Improvement in Feature Extraction:

- Padding can improve the performance of the convolution operation by allowing the filter to better capture features at the edges of the input image. This is crucial for tasks where edge features are important, such as in image recognition or object detection.

Preventing Information Loss:

- When multiple convolutional layers are stacked without padding, the reduction in spatial dimensions can be significant, leading to potential loss of important spatial information.

- Padding helps mitigate this issue by maintaining a larger spatial dimension across layers, preserving more information through the network.

STRIDES

Definition

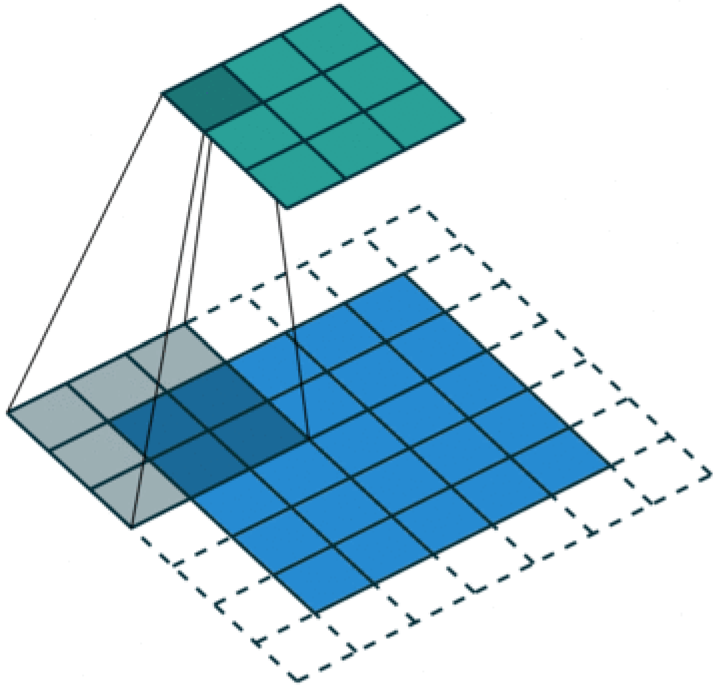
- The stride specifies the number of pixels by which the filter (kernel) moves (or "strides") over the input image during the convolution operation.
- A stride of 1 means the filter moves one pixel at a time, while a stride of 2 means the filter moves two pixels at a time, and so on.

Stride

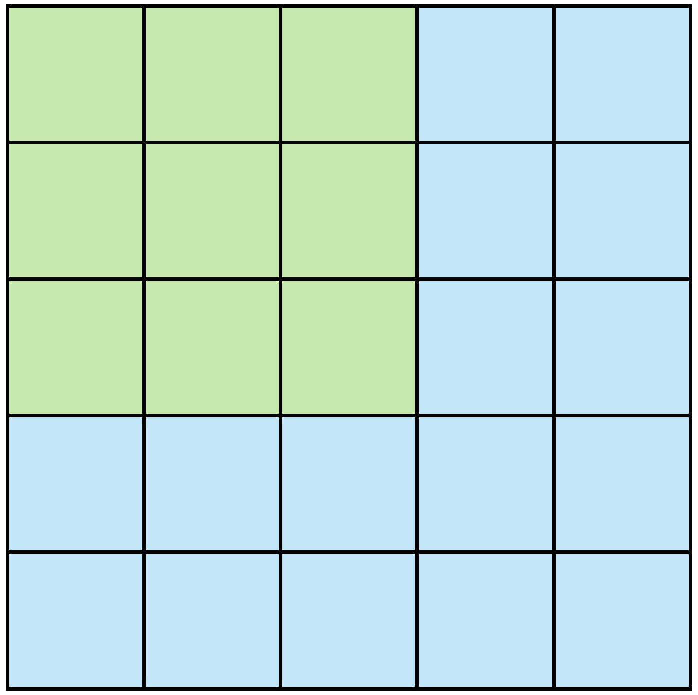
- Output = (N+2p-f/2)+1

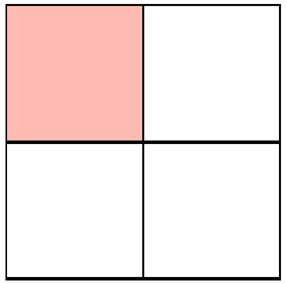
$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$

n_{in}:	number of input features
n_{out}:	number of output features
k:	convolution kernel size
p:	convolution padding size
s:	convolution stride size



Stride = 2


Stride 2


Feature Map

Why Stride is Important

Control Over Output Size:

- The stride directly affects the spatial dimensions (width and height) of the output feature map.

- By increasing the stride, you reduce the size of the output feature map because the filter covers the input image more quickly.

- This can be important for reducing the computational load and managing memory, especially for large input images.

Dimensionality Reduction:

- Using larger strides can help reduce the spatial dimensions of the feature maps more quickly than pooling layers.
- This can be useful for progressively reducing the size of feature maps in deeper layers of a CNN, which helps in managing computational complexity and preventing overfitting.

Efficient Feature Extraction:

- Larger strides allow the network to scan the image more efficiently by taking larger steps. This can be useful when the relevant features in the input data are spread out and do not require examining every single pixel.

- Striding can help in identifying patterns that are more spread out in the input image, which can be beneficial for certain types of images or signals

Speed and Performance:

- Using strides greater than 1 reduces the number of convolution operations because the filter is applied fewer times across the input image. This speeds up the training and inference processes, making the CNN more efficient.

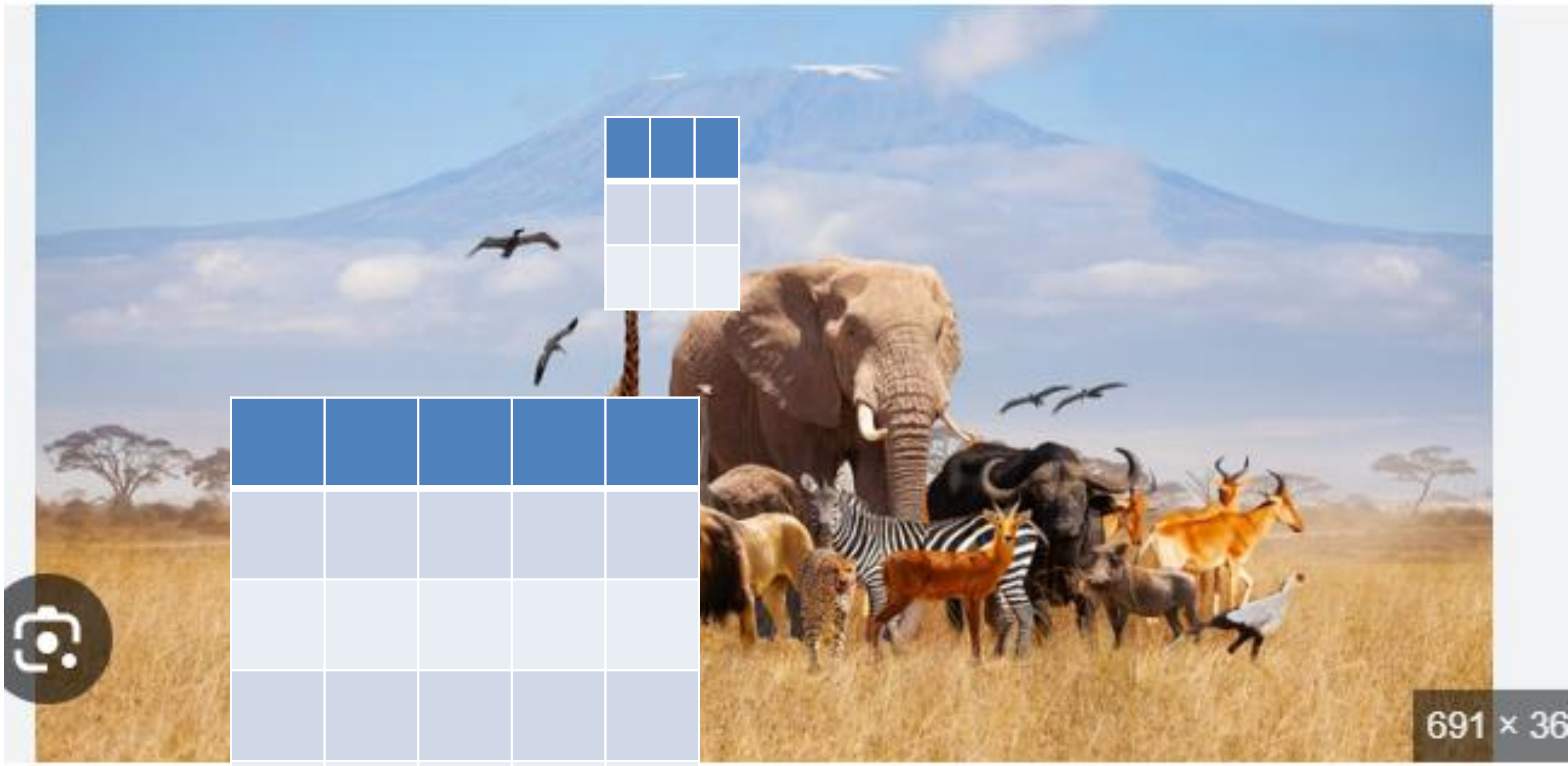
- This is crucial for real-time applications or when working with limited computational resources.

Effects of Different Stride Values

Stride of 1:

- The filter moves one pixel at a time, resulting in a highly detailed feature map that retains much of the spatial resolution of the input.

- This is useful for tasks requiring fine-grained details, but it is computationally more intensive.


691 × 36

Stride Greater than 1:

- The filter moves multiple pixels at a time, producing a feature map with reduced spatial dimensions.
- This results in less detailed, but more computationally efficient, feature maps.
- Helps in faster convergence during training and can be useful for larger images where fine-grained detail is less important.

- Stride reduces the feature map size
- Stride increases information loss
- Also reduces computational power requirements