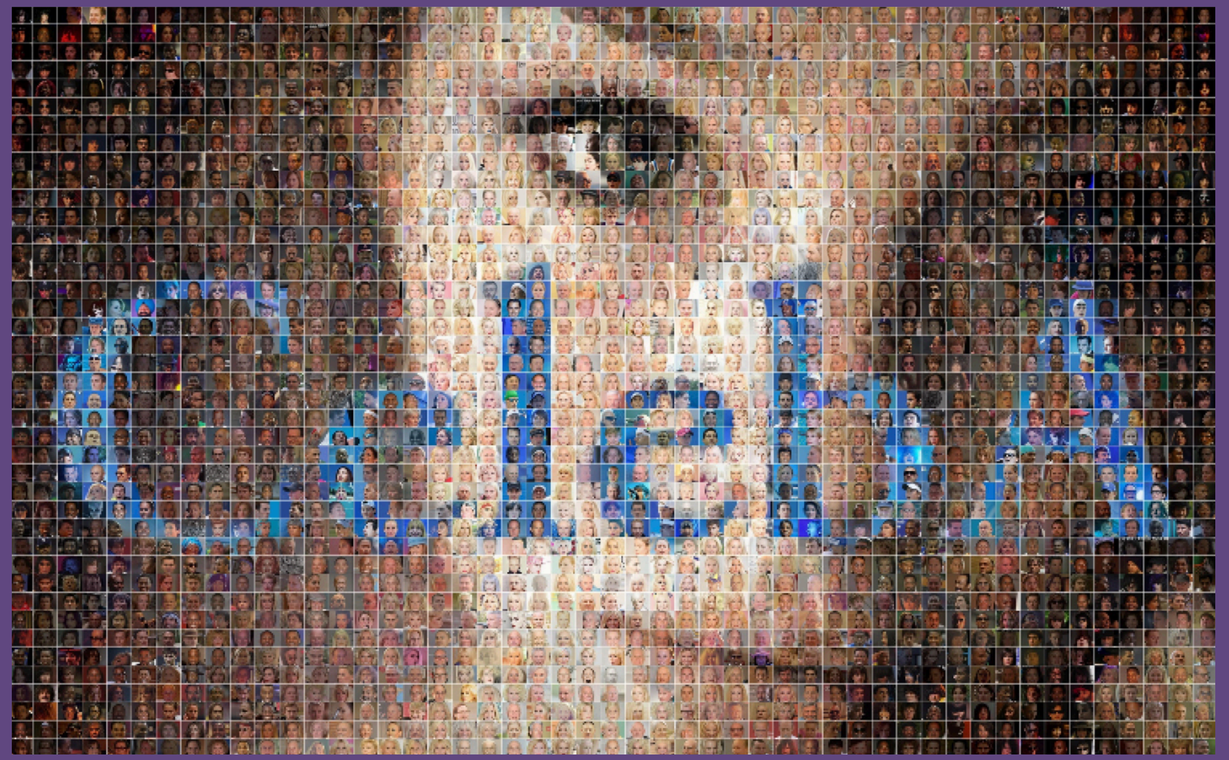
Tutorial 5: DCGAN



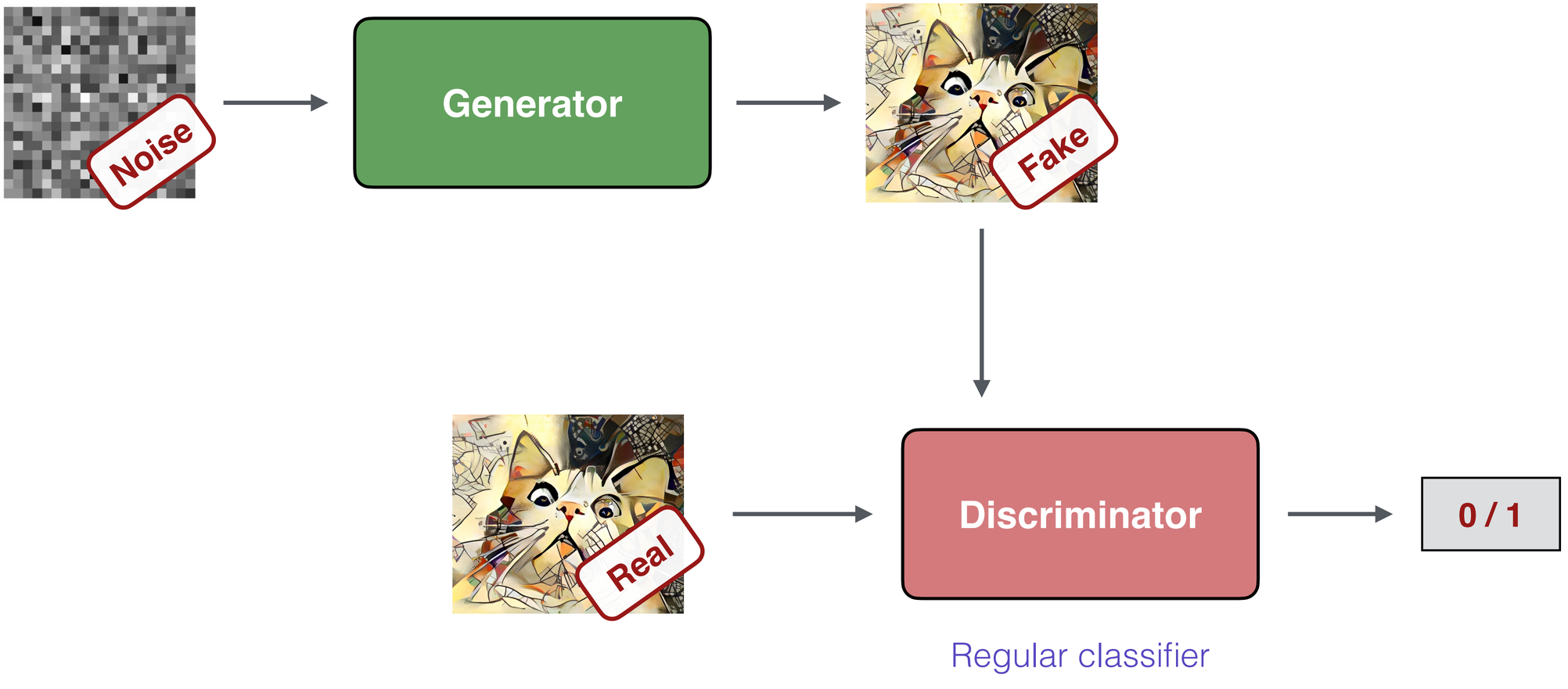
Practical Deep Learning for Science
24 April, 2025

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

- Nilotpal

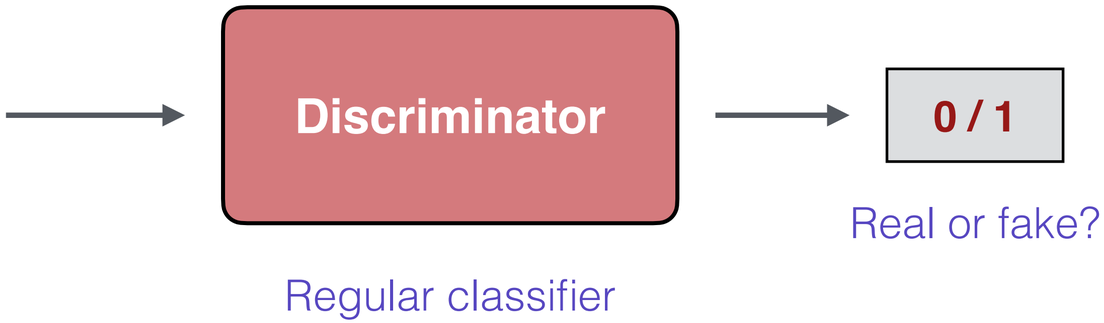
- We'll generate some celebrity faces
- In general, GANs are
  - hard to train (can be unstable)
  - Confusing to understand (loss etc.) You've probably seen in Eilam's lecture
- Here's some notes on the *confusing part*

GAN


Noise
Generator
Fake
Real
Discriminator
0 / 1
Regular classifier

Loss? It’s a bit tricky

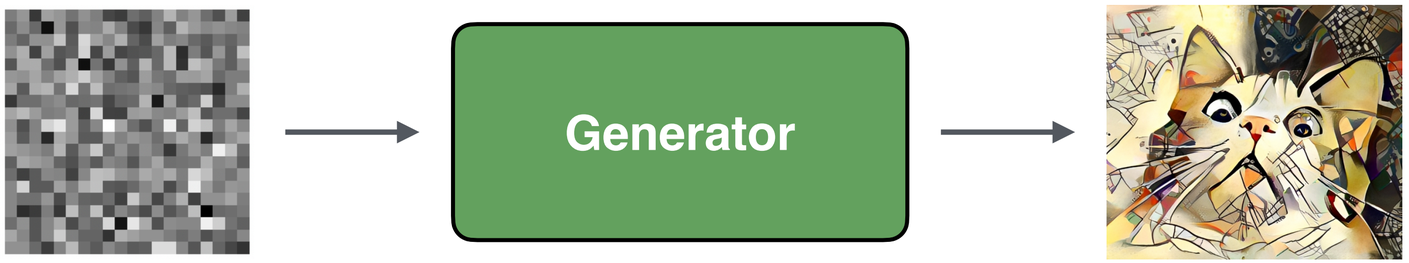
Discriminator is a regular binary classifier



Regular classifier

Binary Cross entropy loss

Generator loss?



- Exploit the adversary
- Generate a ***FAKE*** image
  - Ask the Discriminator how ***REAL*** does it look?
  - “How REAL does it look?” → “What’s the loss if the target label is REAL?”
  - *We flip the label to be REAL (1) and compute the loss*

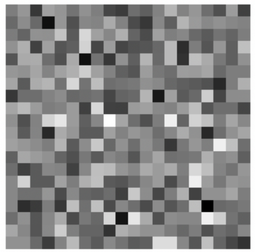
GAN recipe

- Train Discriminator
  - Get real images, generate fake images
  - Compute **discriminator loss**; update ***discriminator***

- Train Generator
  - Label the fake images as real
  - Compute **discriminator loss;** update ***generator***

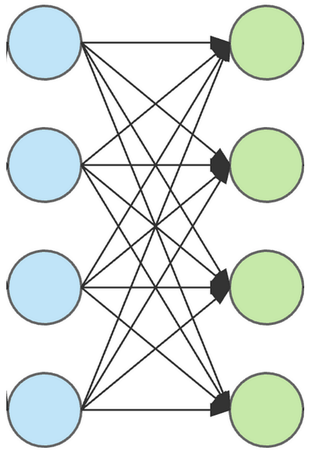
Always computing Discriminator loss, but updating generator when the labels are flipped

.detach()



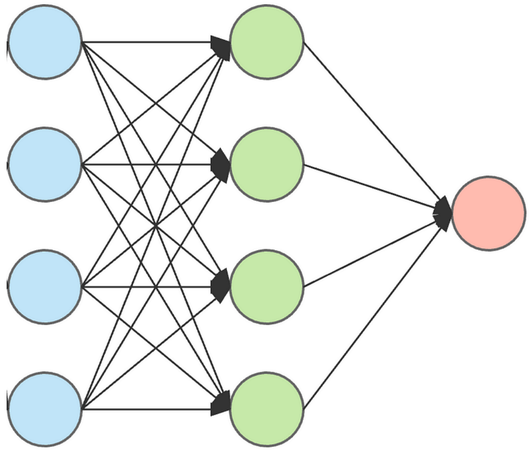
Noise

Generator



Fake image

Discriminator



`Optimizer_G`

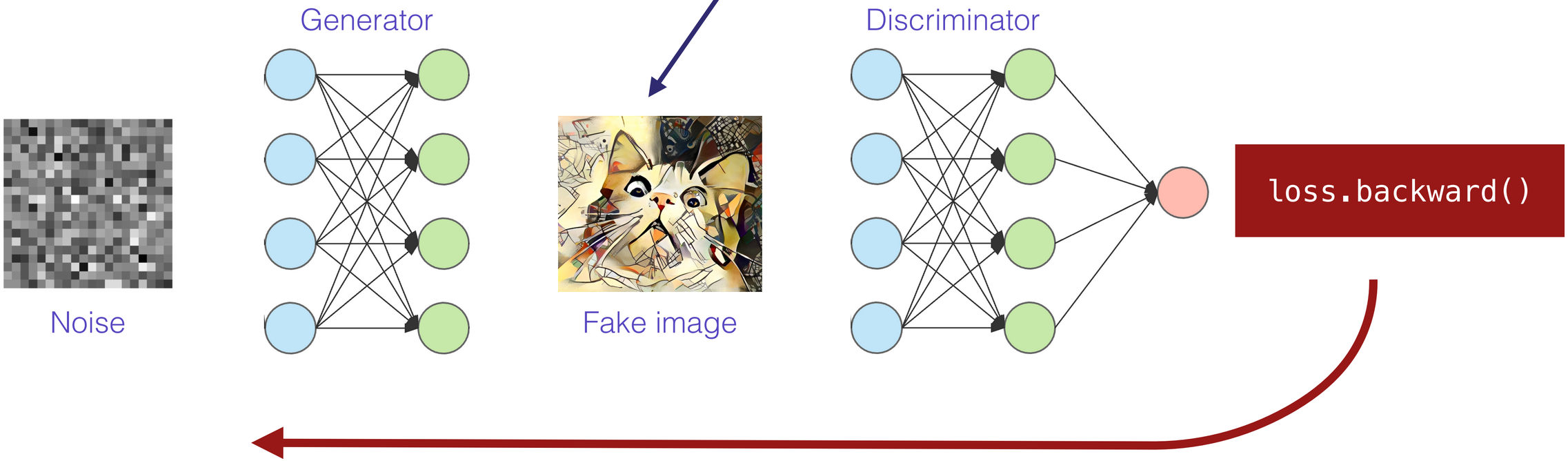
`Optimizer_D`

- Two networks → Two optimizers
  - ➡ We decide which one to update when

The two networks actually looks like
a one big network

```
fake_img.detach()
```

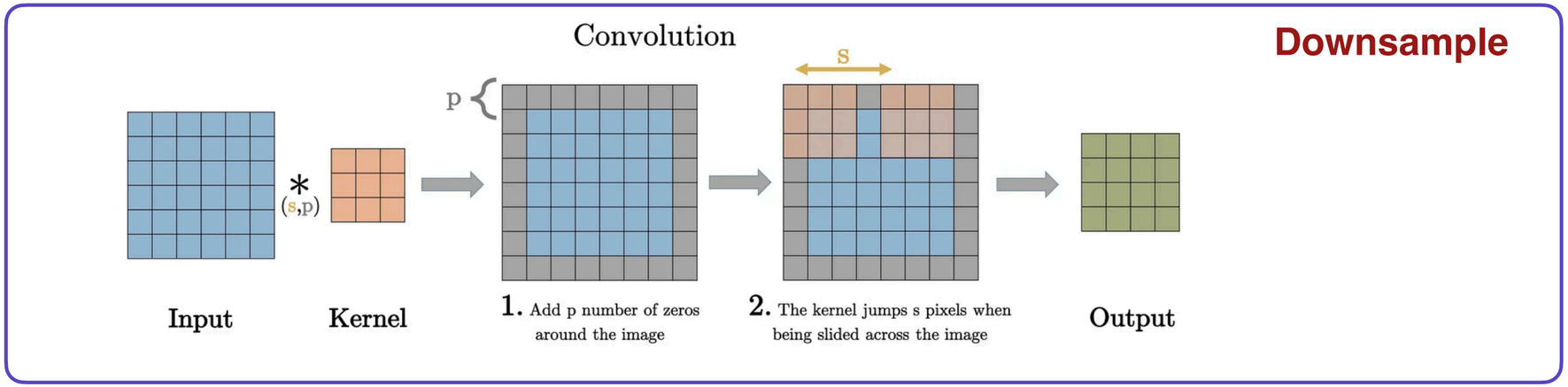
Will stop the gradients from flowing back to Generator
Can train the Discriminator in isolation

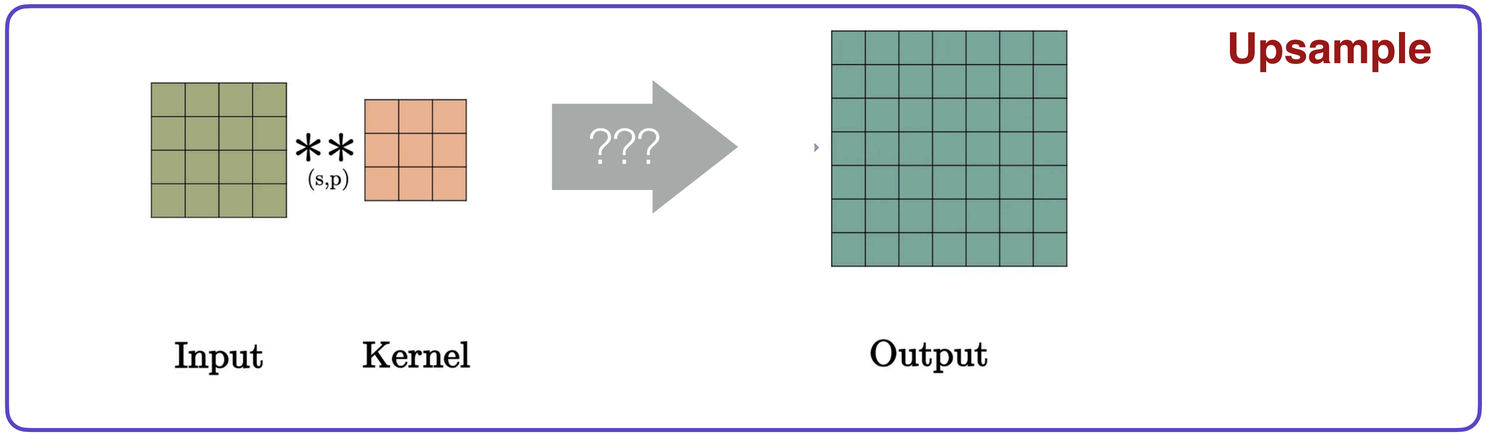


Gradients will always flow all the way back to the generator

Transposed Convolution

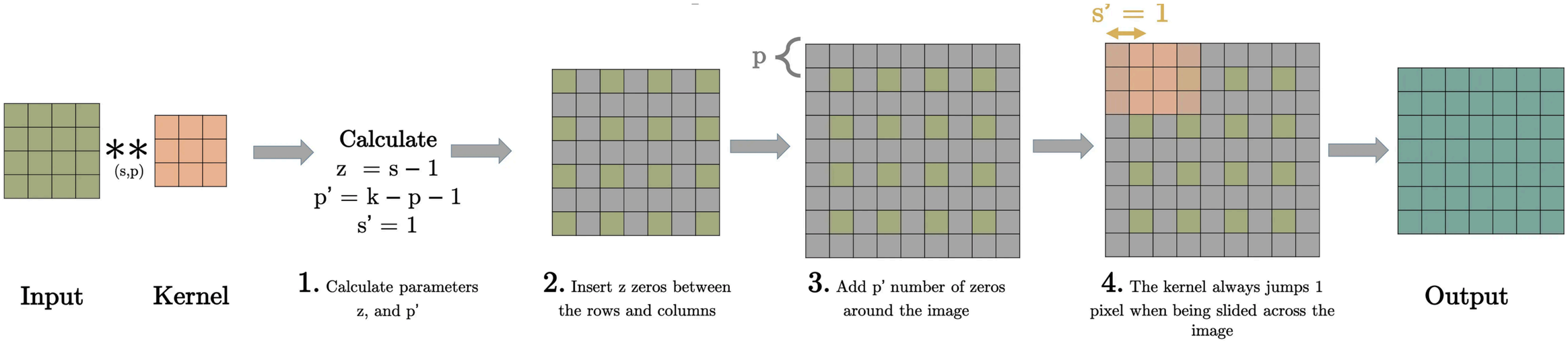
Cons vs TransConv





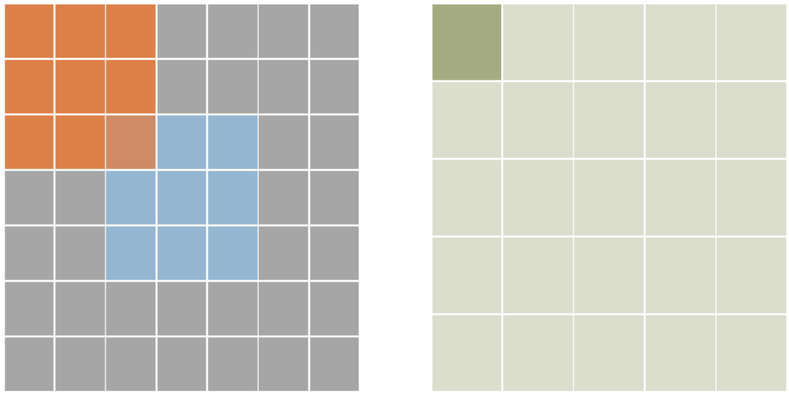
Checkout this article - https://towardsdatascience.com/what-is-transposed-convolutional-layer-40e5e6e31c11
(The graphics are taken from there)

TransConv

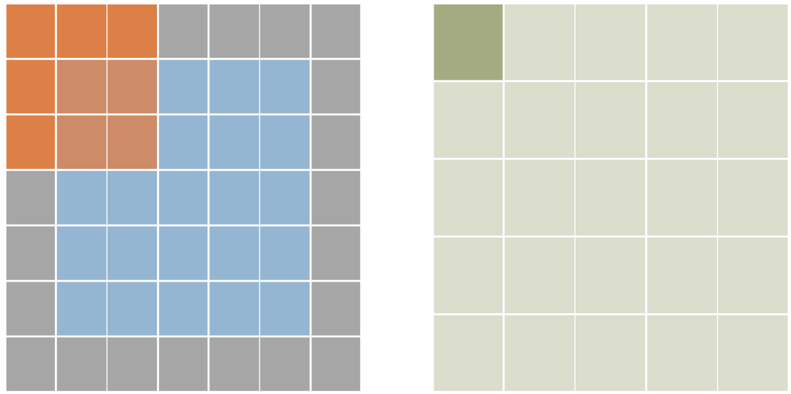


Checkout this article - https://towardsdatascience.com/what-is-transposed-convolutional-layer-40e5e6e31c11
(The graphics are taken from there)

Type: transposed conv - Stride: 1 Padding: 0

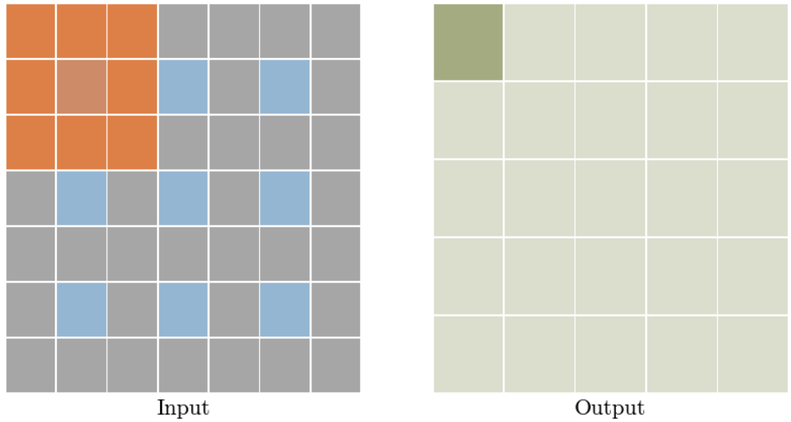


Type: transposed conv - Stride: 1 Padding: 1



Type: transposed conv - Stride: 2 Padding: 0

Type: transposed conv - Stride: 2 Padding: 1



Compute

$$z = s - 1$$
$$p' = k - p - 1$$
$$s' = 1$$

Output size

$$o = (i - 1) \times s + k - 2p$$

k = kernel
p = padding
s = stride