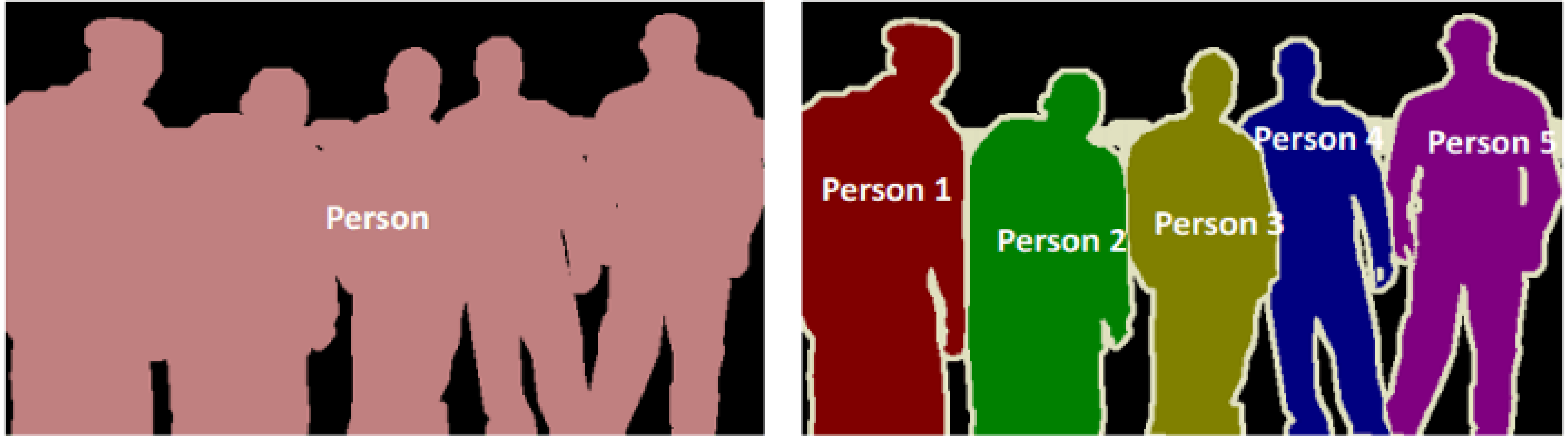
Unet for Semantic segmentation

. **Architecture of Unet segmentation**
. **Difference between upsampling and transpose convolution**

- **by Ahmed *Ibrahim***

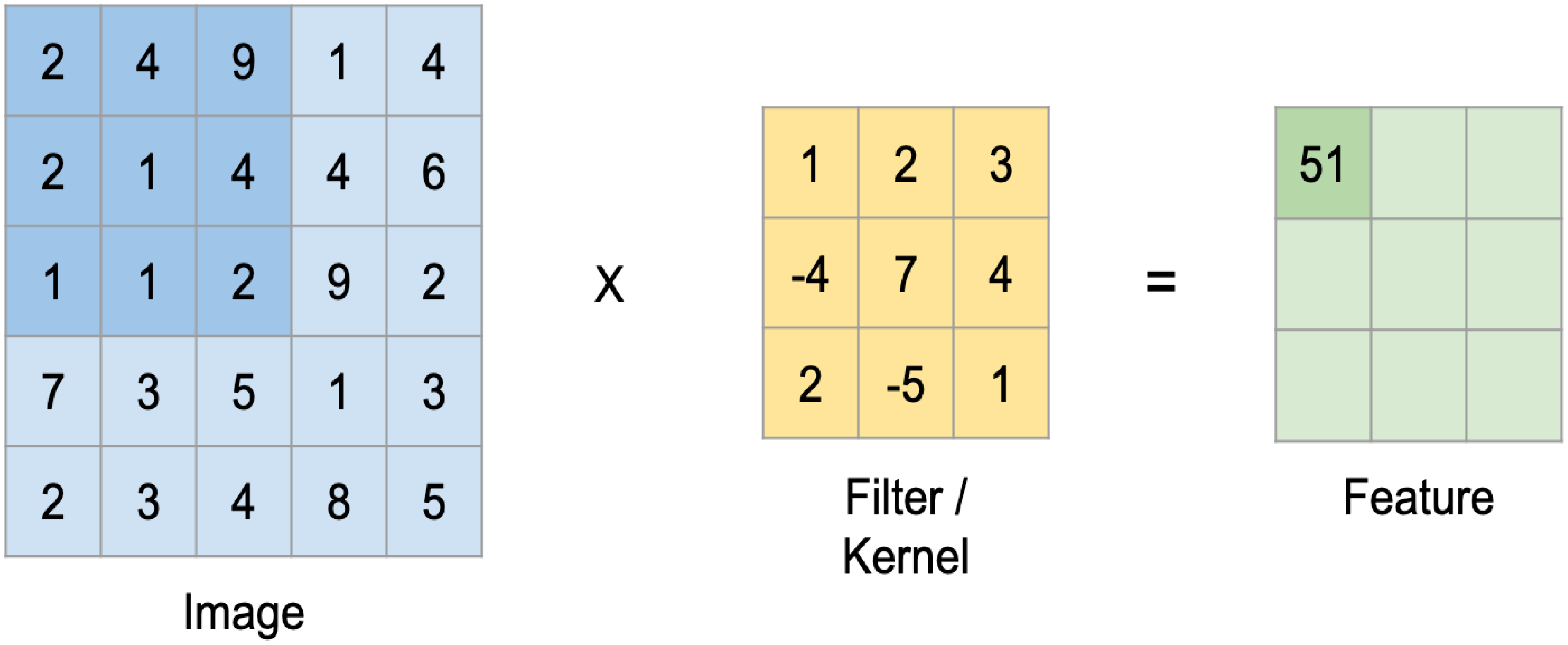
Difference between SS and IS



Semantic Segmentation

Instance Segmentation

Convolutional for downsampling



2	4	9	1	4
2	1	4	4	6
1	1	2	9	2
7	3	5	1	3
2	3	4	8	5

Image

X

1	2	3
-4	7	4
2	-5	1

Filter / Kernel

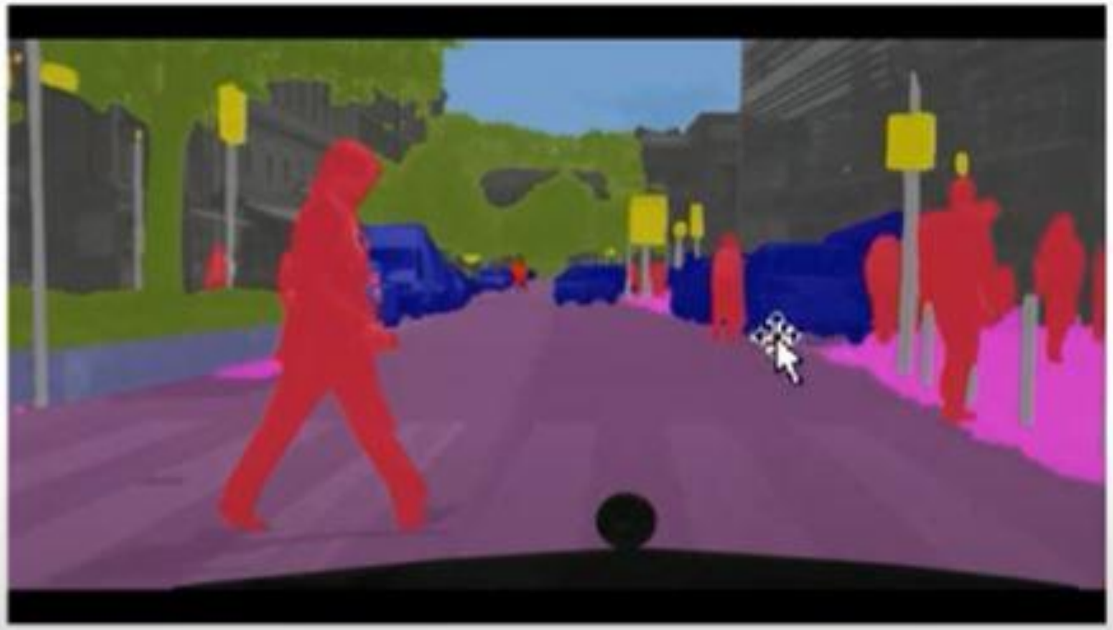
=

51		

Feature

Transposed Convolutions

- Super Resolution -> Upscaling the input image to higher resolutions.
- Semantic Segmentation -> From RGB image input to class-based visualisation.



1. upsampling

- Repeats the rows and columns of the data by size[0] and size[1] respectively.

- No trainable parameters.

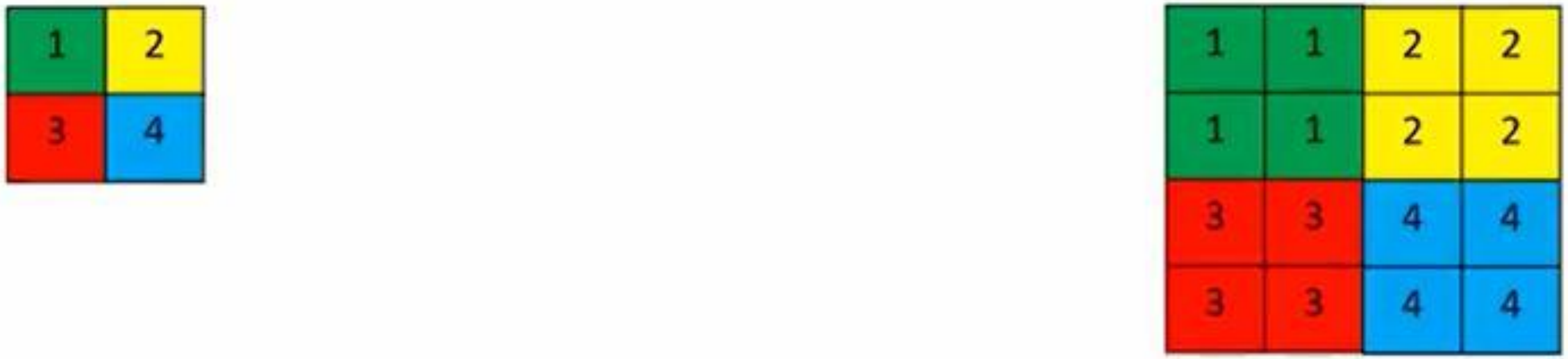
1. upsampling

1	2
3	4

1	2
1	2
3	4
3	4

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

1. upsampling



1	2
3	4

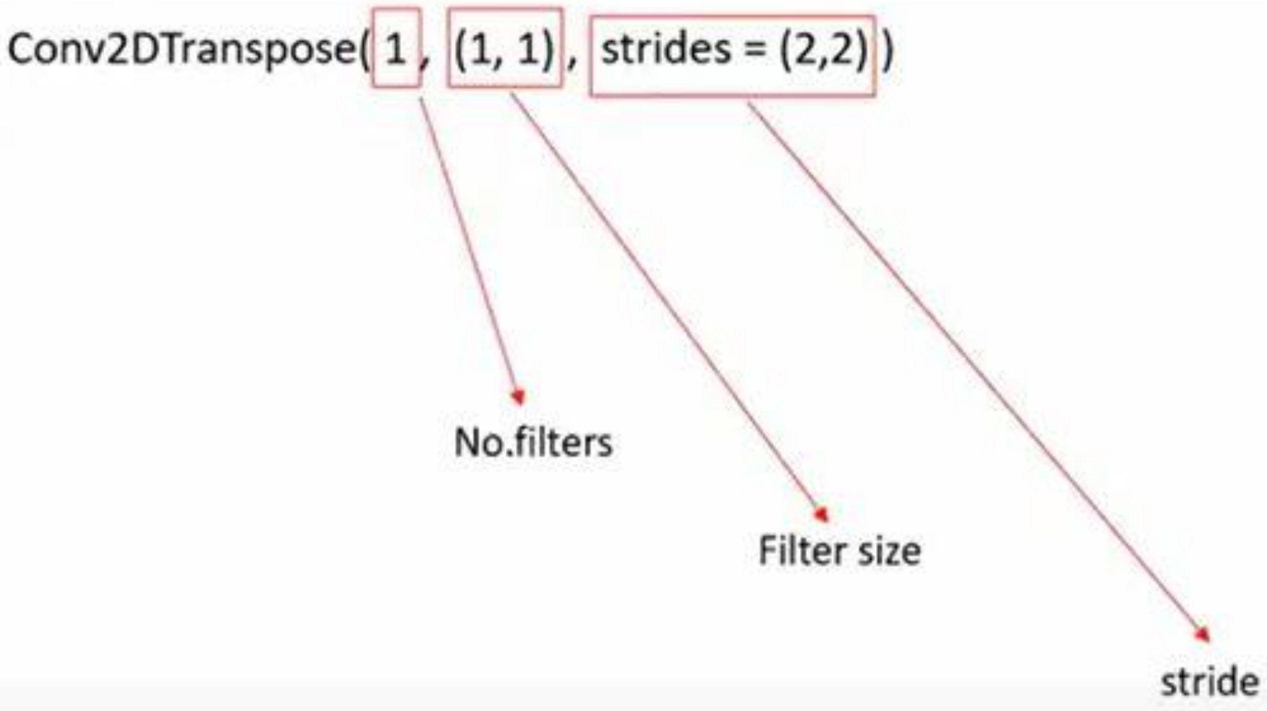
1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

2. transpose convolutional

- Convolution and upsampling.

- Trainable parameters.

2. transpose convolutional



```
Conv2DTranspose( 1 , (1, 1) , strides = (2,2) )
```

No.filters

Filter size

stride

2. transpose convolutional

```
Conv2DTranspose( 1 ,  (1, 1) ,  strides = (2,2) )
```

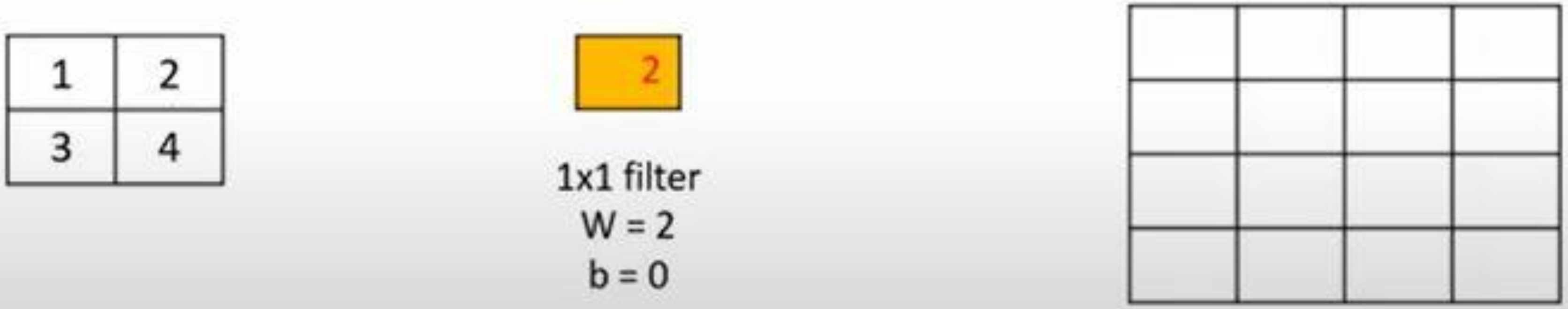
1	2
3	4

(w,b)

1x1 filter
W = 2
b = 0

2. transpose convolutional

Conv2DTranspose(1 , (1, 1) , strides = (2,2))

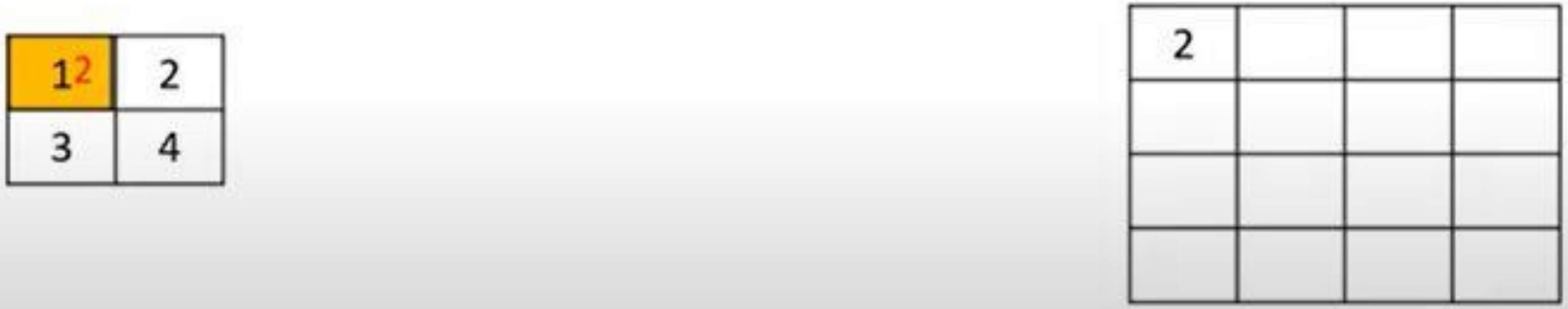


1	2
3	4

1x1 filter
W = 2
b = 0

2. transpose convolutional

```
Conv2DTranspose( 1 ,  (1, 1) ,  strides = (2,2) )
```

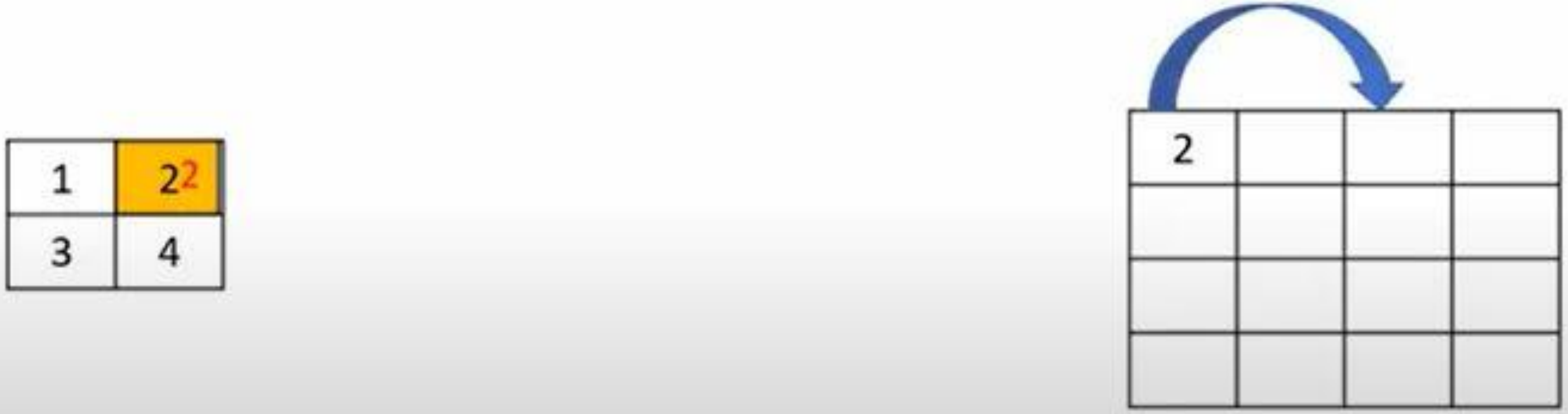


12	2
3	4

2			

2. transpose convolutional

```
Conv2DTranspose( 1 ,  (1, 1) ,  strides = (2,2) )
```

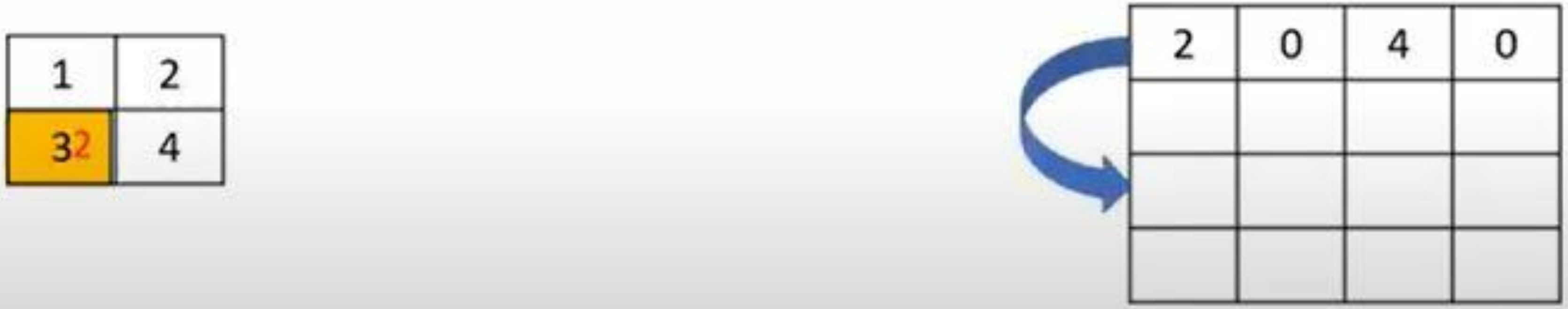


1	22
3	4

2			

2. transpose convolutional

```
Conv2DTranspose( 1 ,  (1, 1) ,  strides = (2,2) )
```



1	2
32	4

2	0	4	0

2. transpose convolutional

Conv2DTranspose(1 , (1, 1) , strides = (2,2))

1	2
3	42

2	0	4	0
0	0	0	0
6	0	8	0
0	0	0	0

2. transpose convolutional

```
Conv2DTranspose( 1 ,  (1, 1) , strides = (1,1) )
```

1	2
3	4

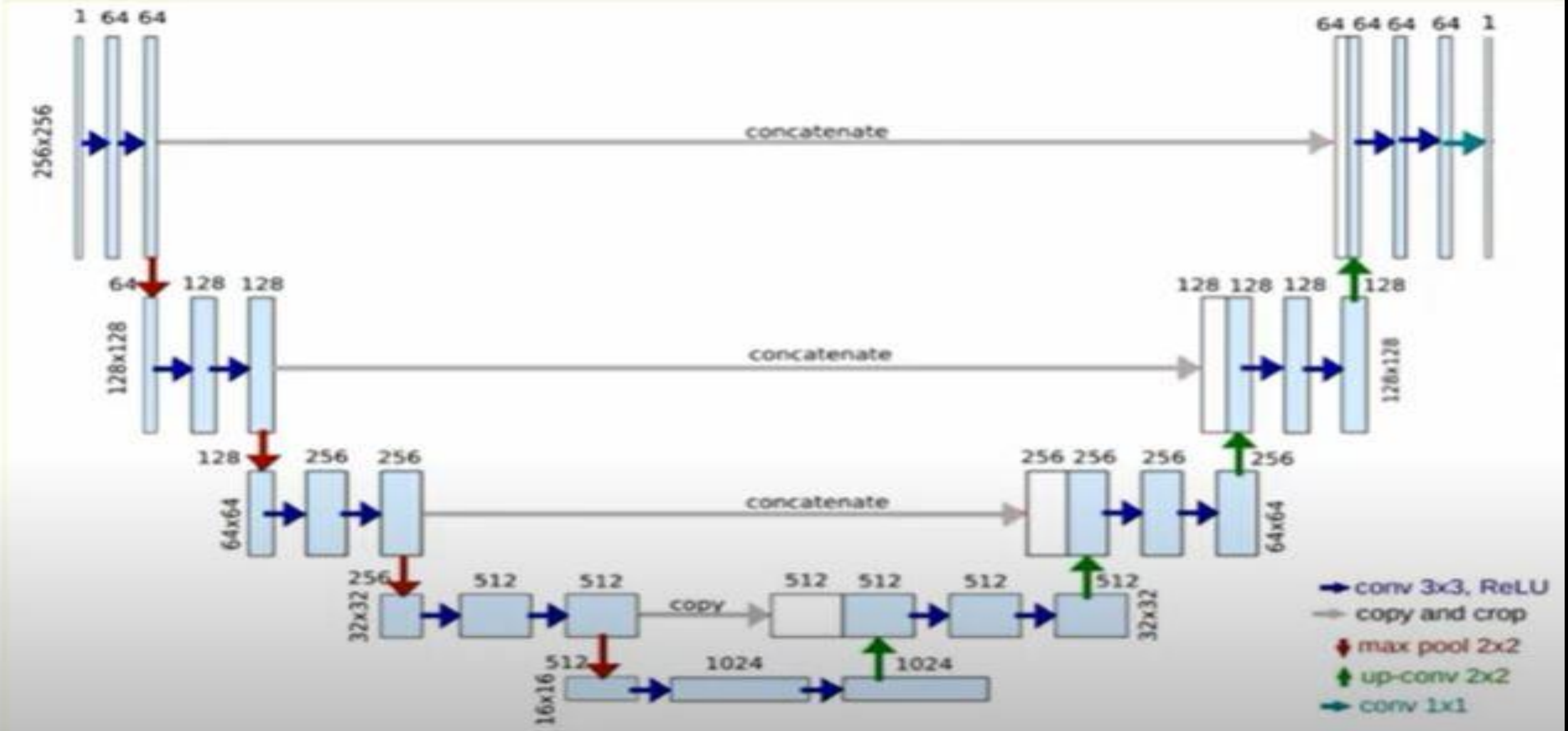
(w,b)

1x1 filter
W = 2
b = 0

2	4
6	8

transpose convolution with stride 1 = no upsampling

U-Net


1
64
64
256x256
concatenate
64
64
64
64
1
64
128
128
128x128
concatenate
128
128
128
128
128x128
128
256
256
64x64
concatenate
256
256
256
256
64x64
256
512
512
32x32
copy
512
512
512
512
32x32
512
1024
1024
16x16
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1