

STABLE DIFFUSION



STRUCTURE

WHAT IS A STABLE DIFFUSION
AND HOW DOES IT WORK?

HANDS ON PART

CONCLUSIONS



1. STABLE DIFFUSION

“Teddy bears wearing suit discussing a business proposal around office table”

Stable
Diffusion



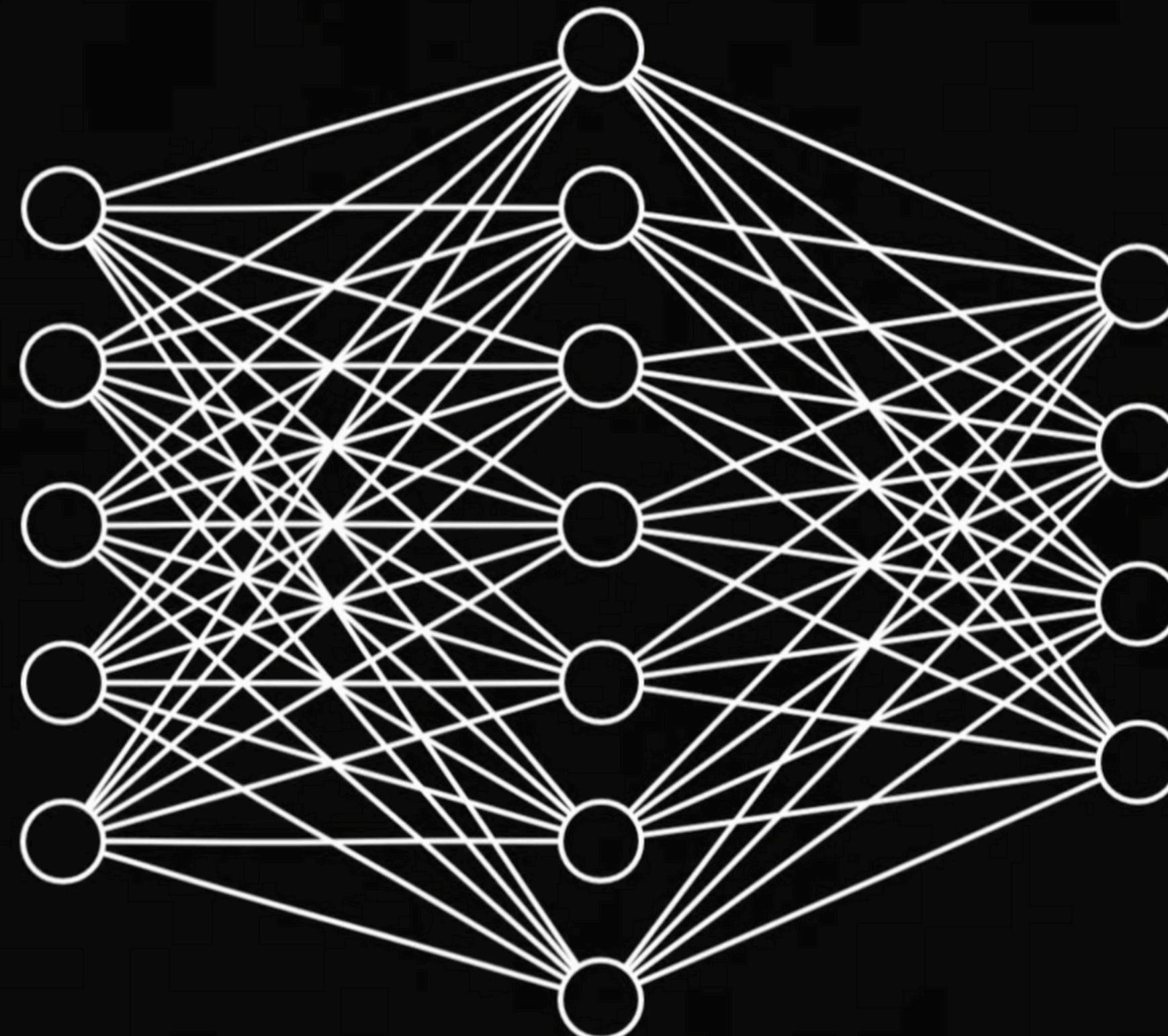
WHAT IS IT?

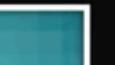
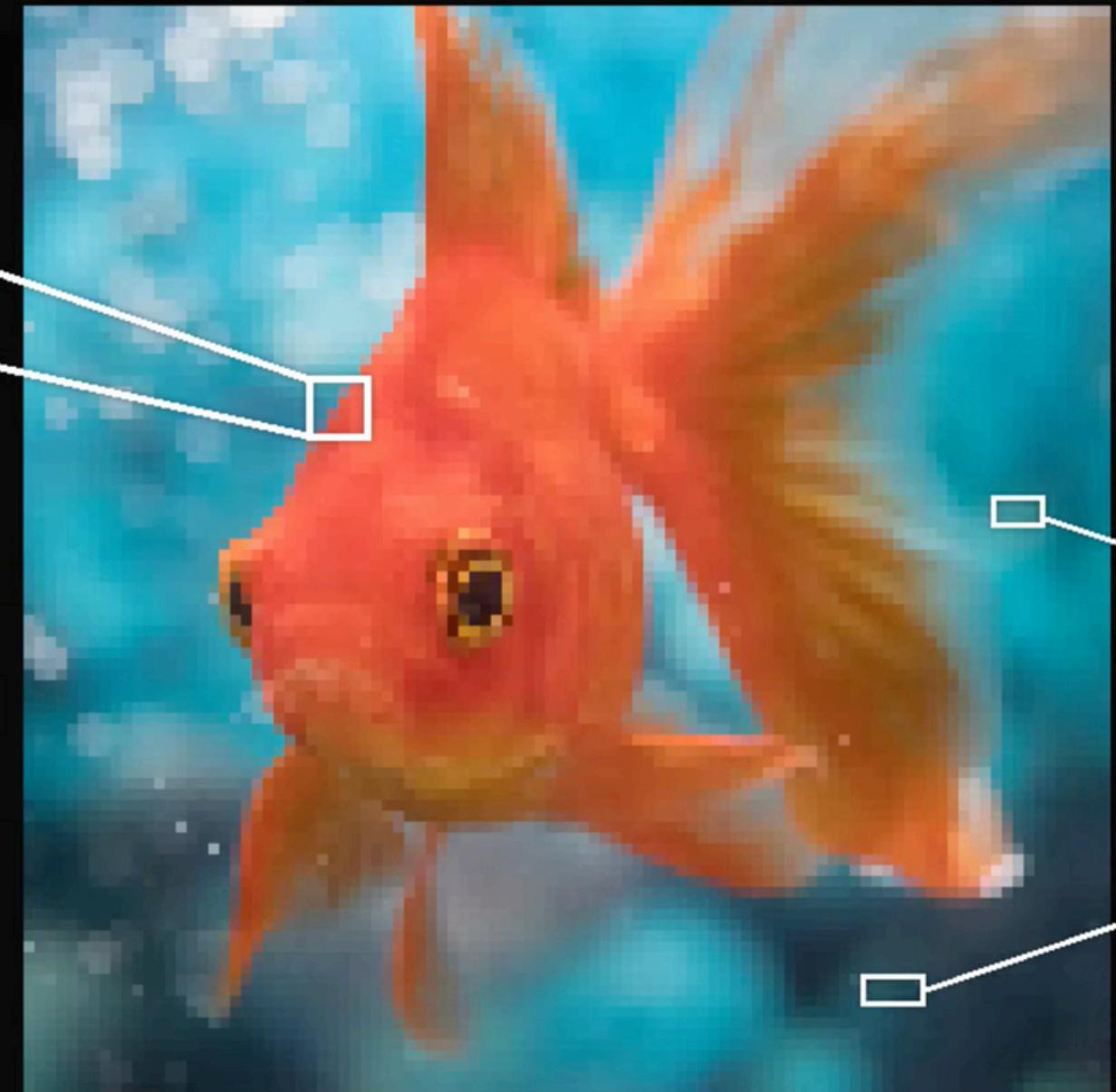
2 special types of layers

convolutional
layer

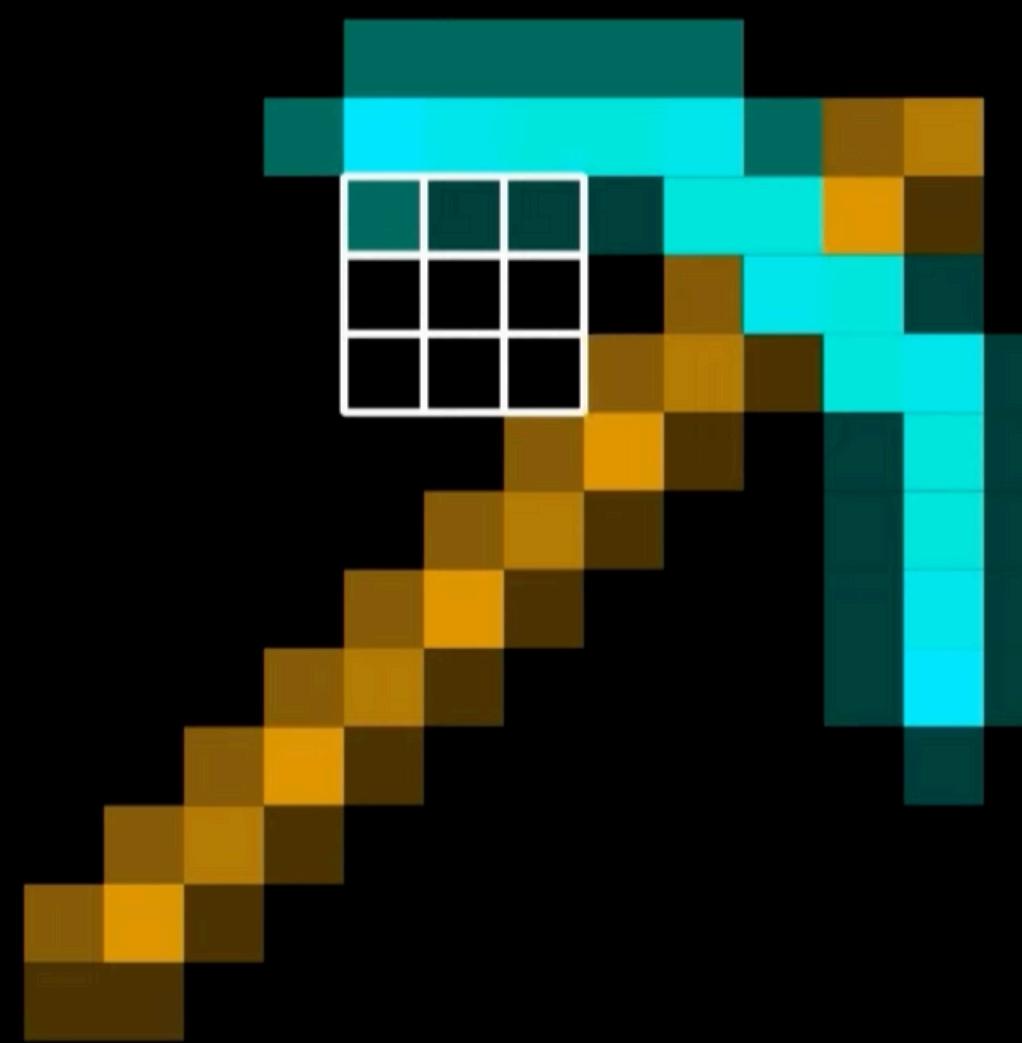
?

fully connected layer



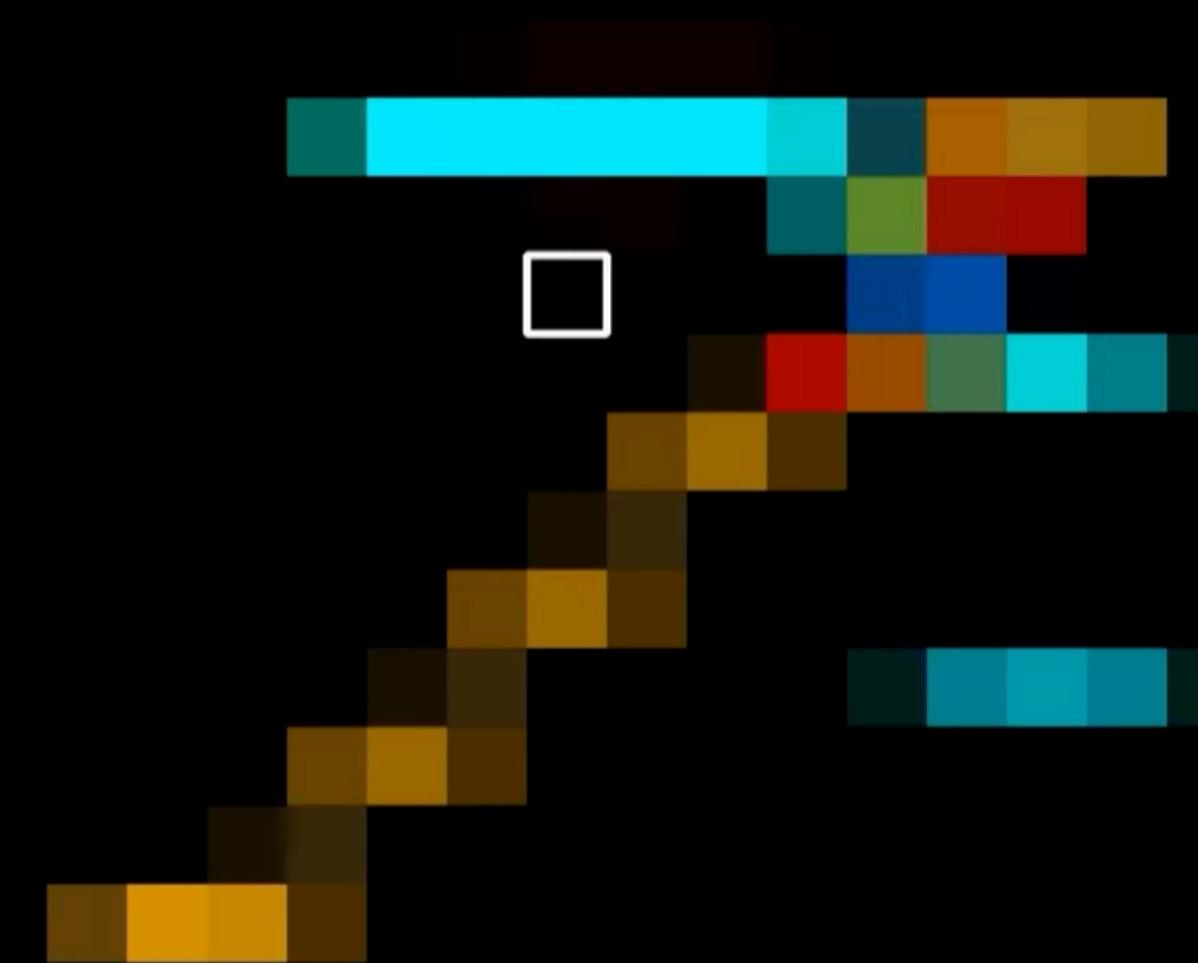


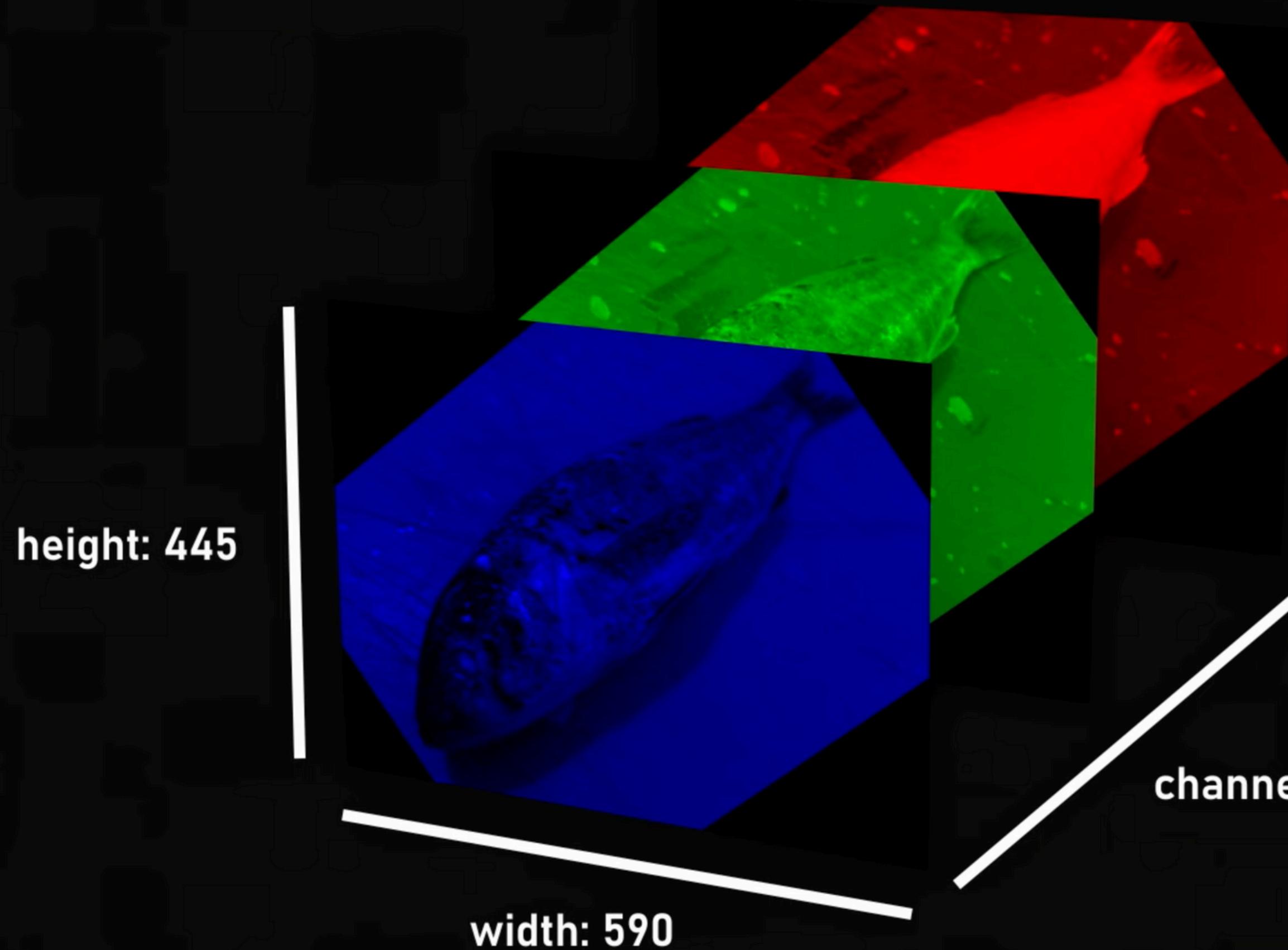
convolutional layer



-0.5	-0.5	-0.5
1	1	1
-0.5	-0.5	-0.5

kernel





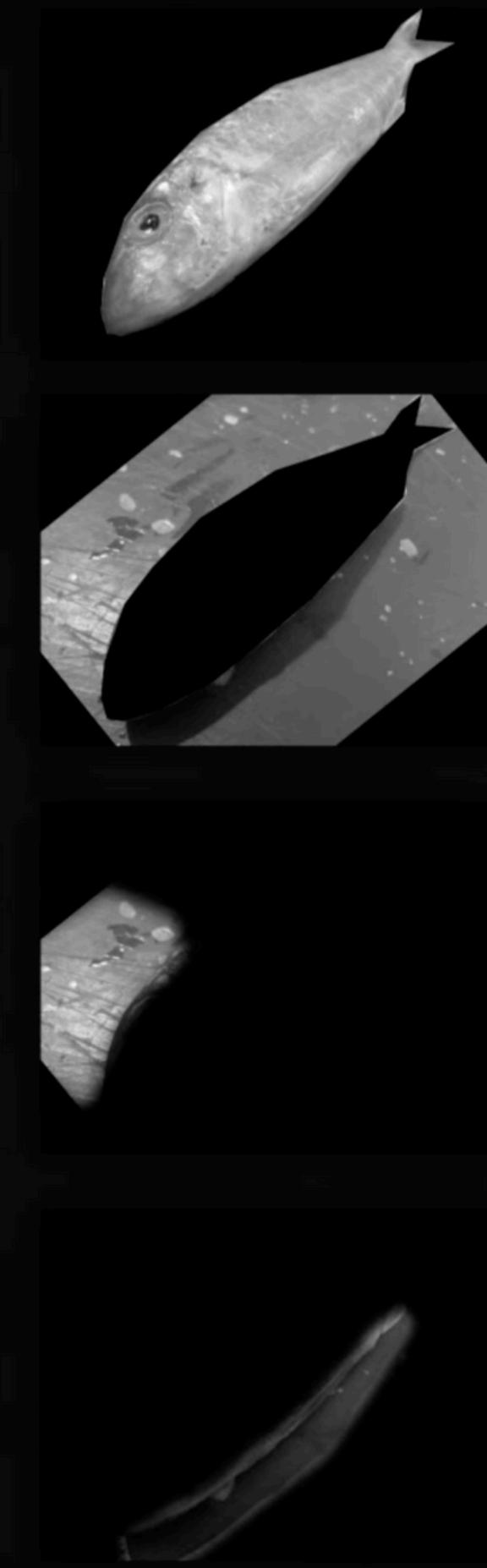
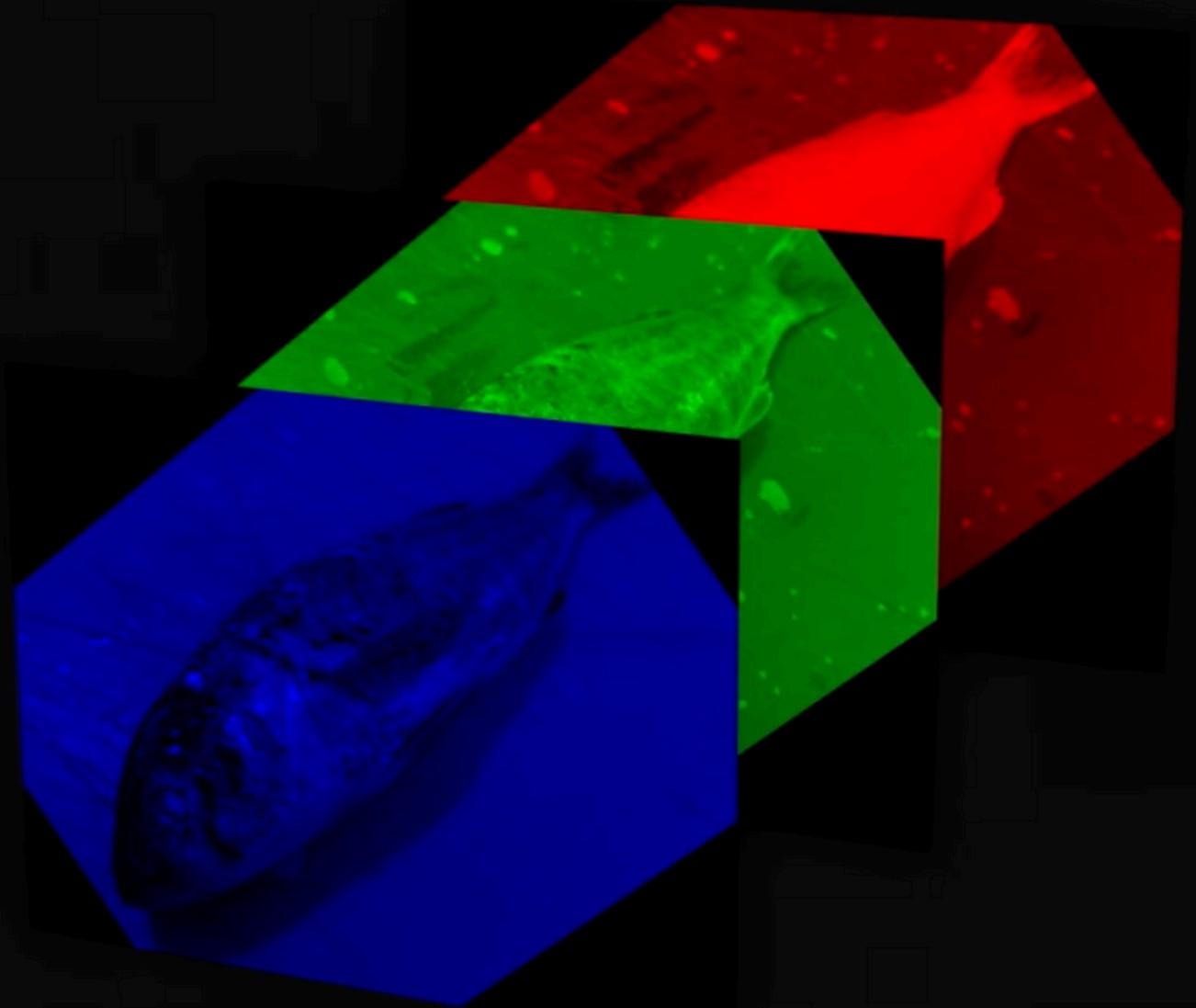
3D tensor with shape (3, 445, 590)

*PyTorch image tensors are (C, H, W)

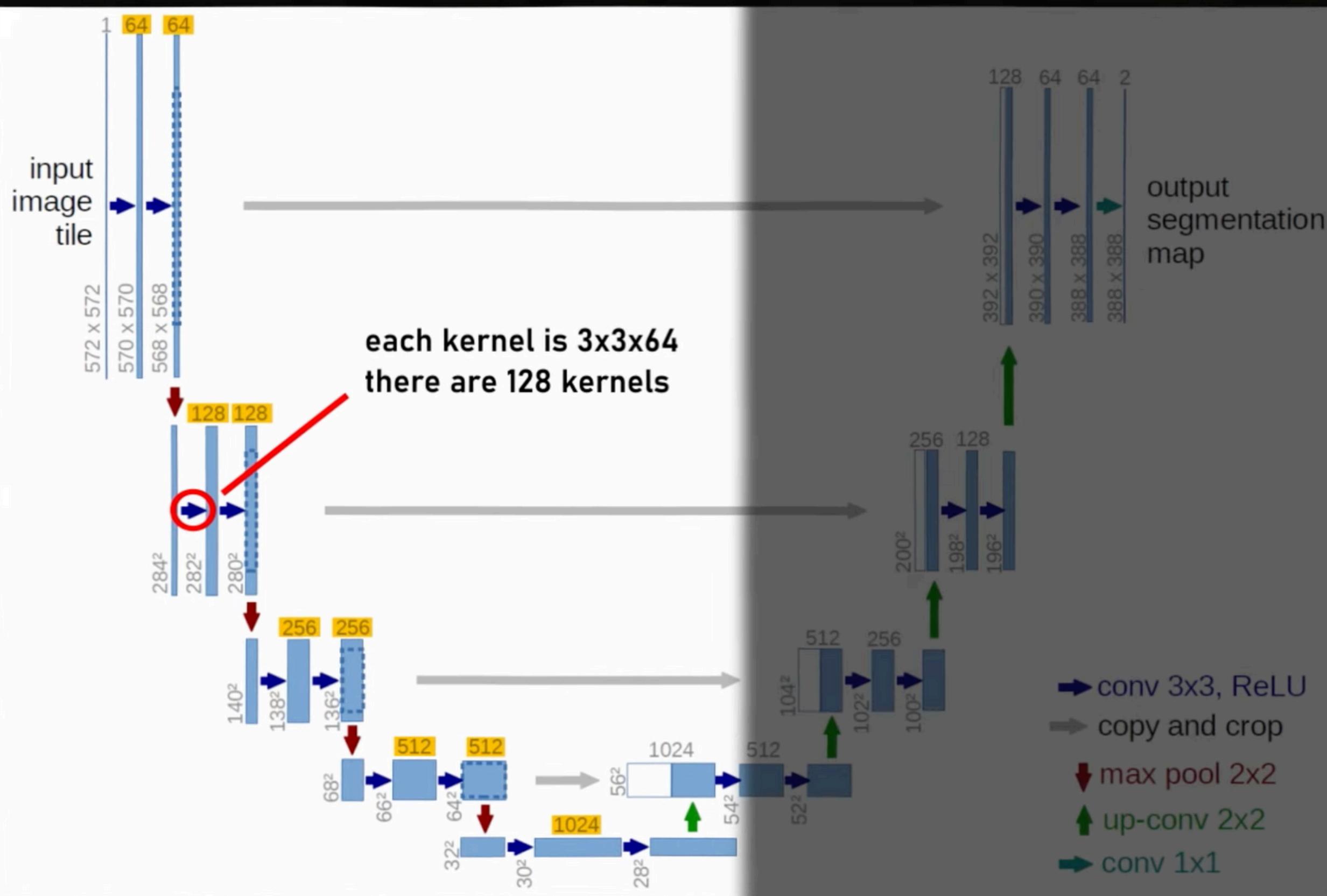
height: 445

channels: 3

width: 590



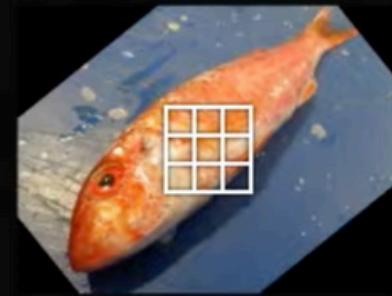
U-Net





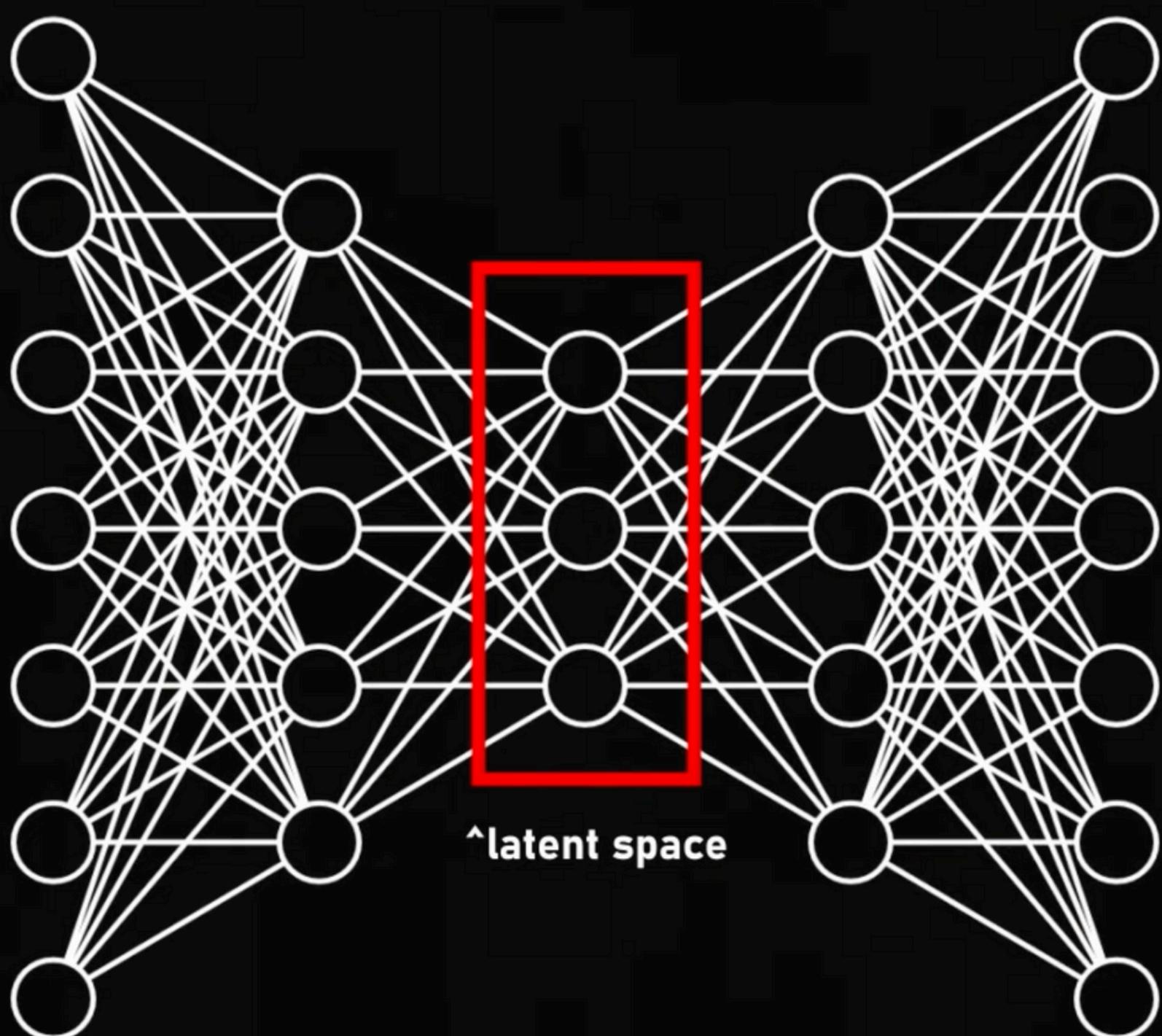


$$\begin{aligned}3 \times 3 &= 9 \\5 \times 5 &= 25 \\7 \times 7 &= 49\end{aligned}$$

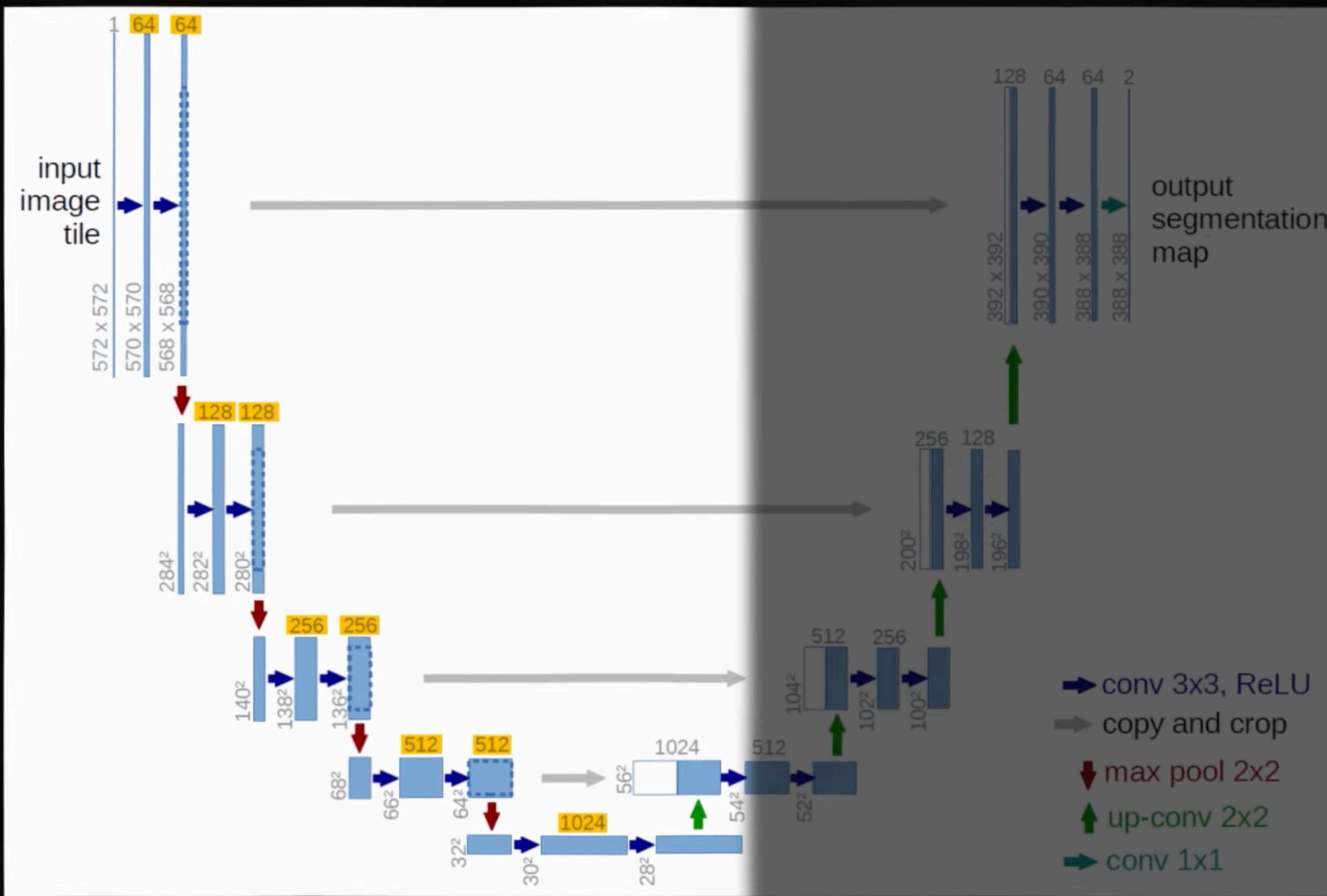




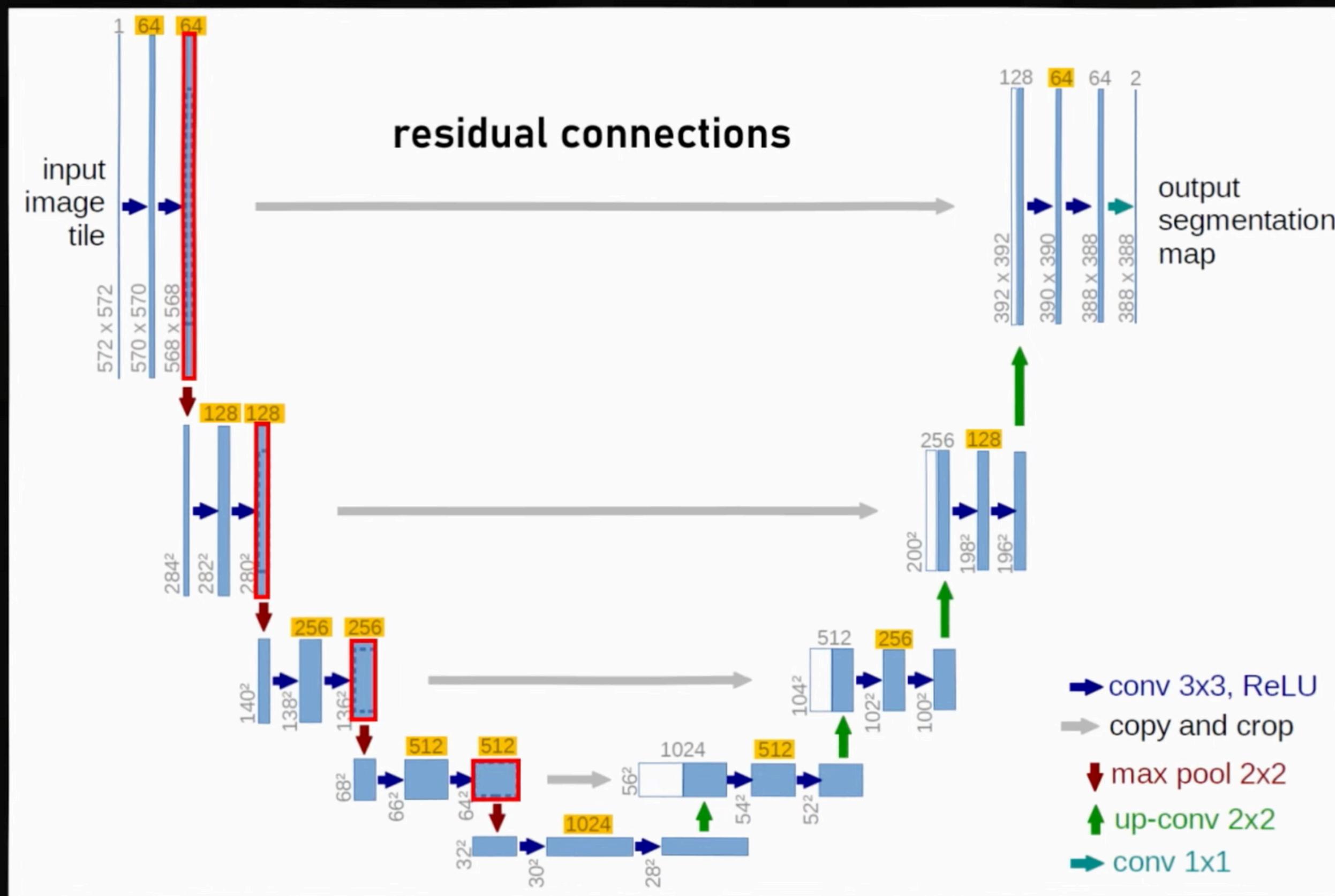
autoencoder



U-Net



U-Net





<https://n8python.github.io/mnistLatentSpace/>

UNet training step

1

Pick a training example
from the training dataset

2

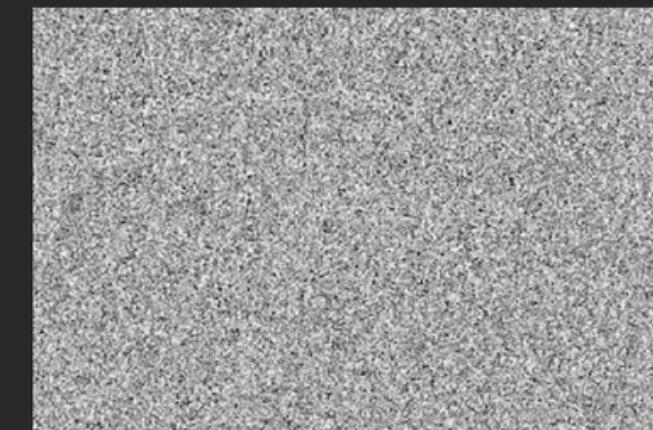
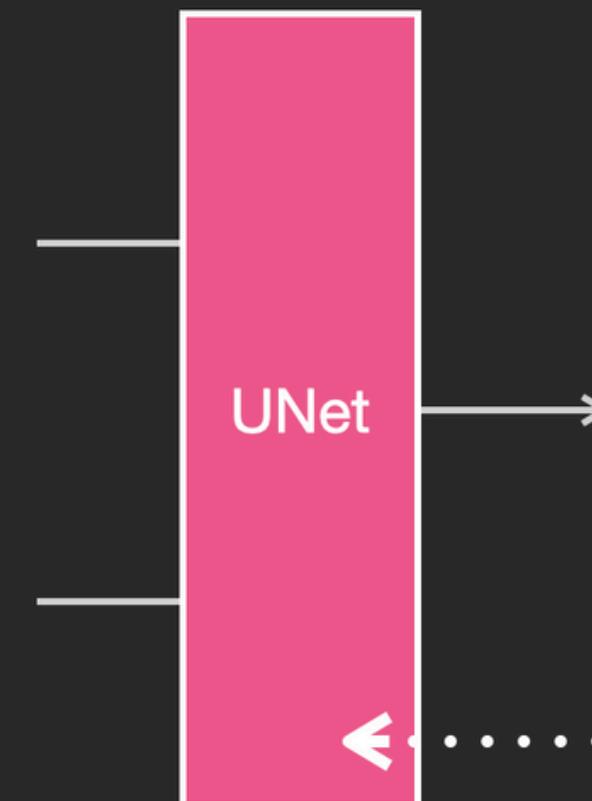
Predict the noise

3

Compare to actual
noise (calculate loss)

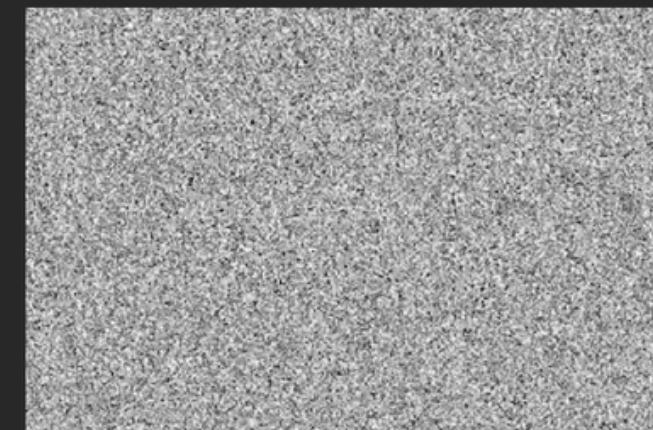


Noise
amount:
3



Unet Prediction

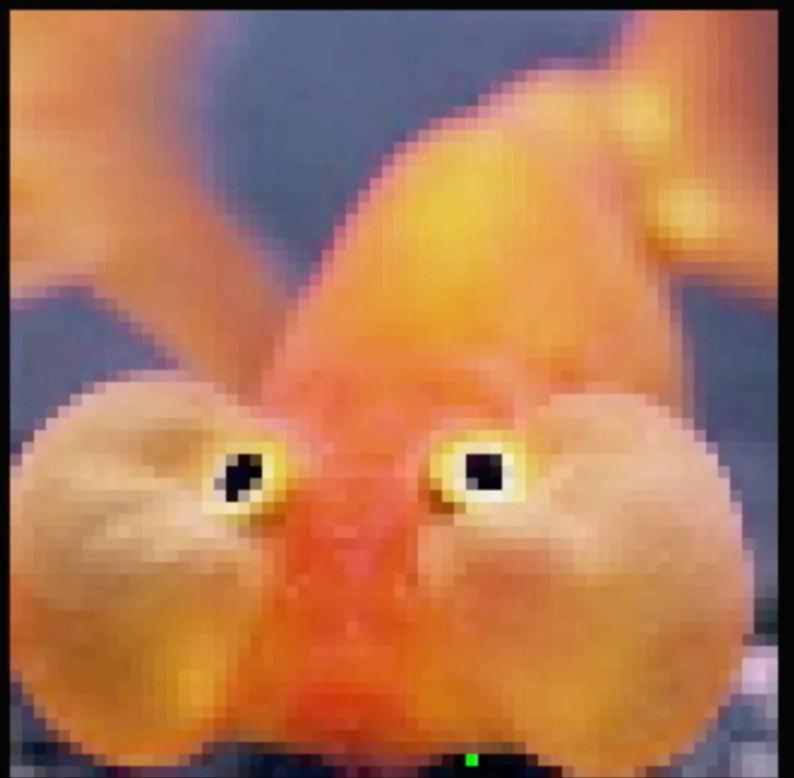
Actual noise (Label)



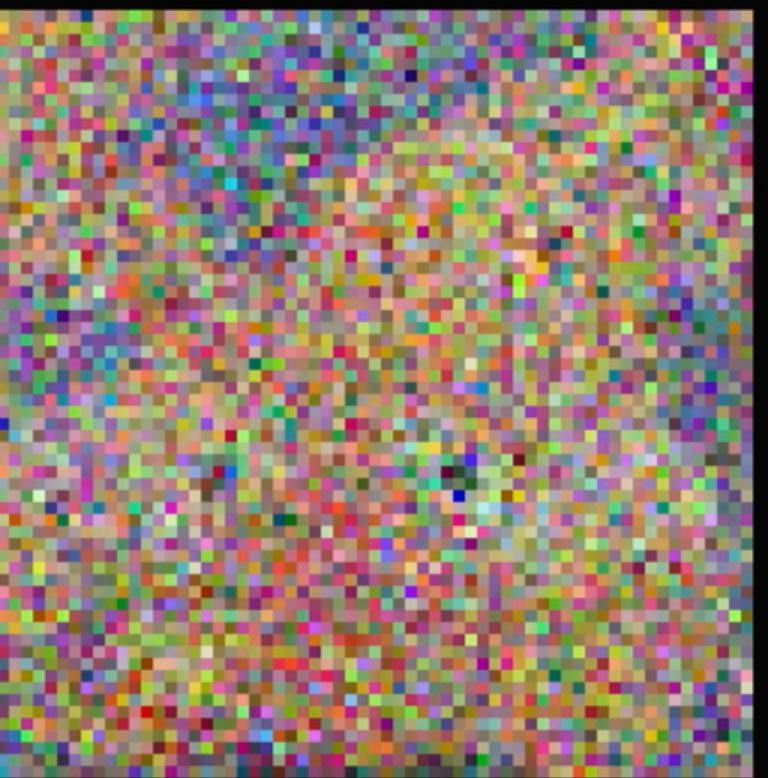
4

Update model
(backprop)

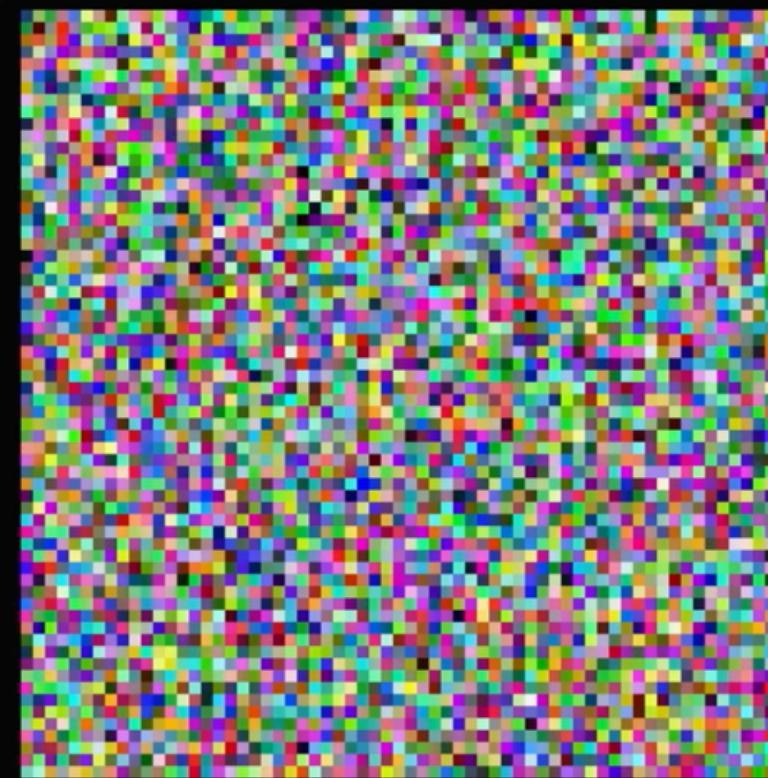
denoising

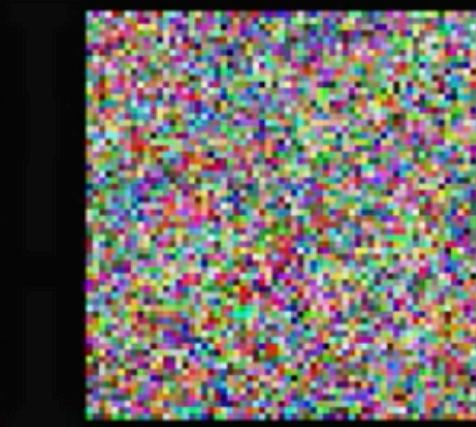
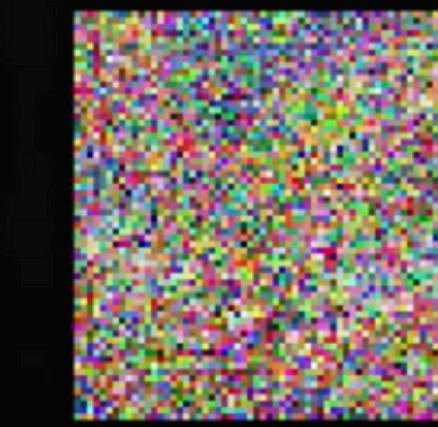
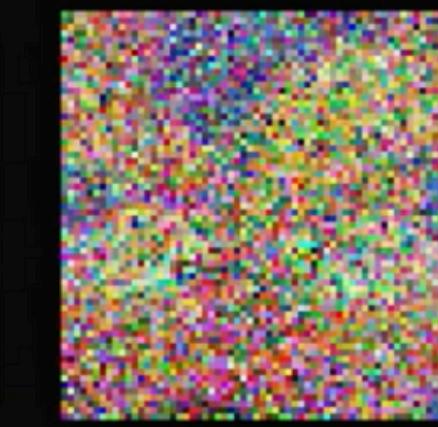
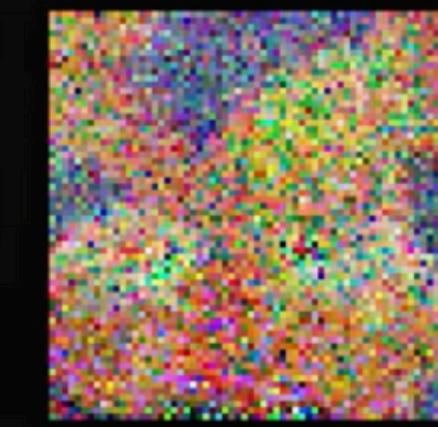
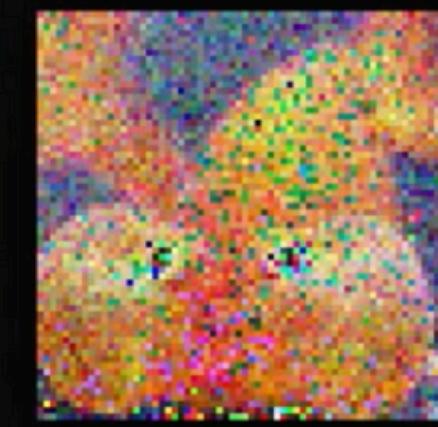
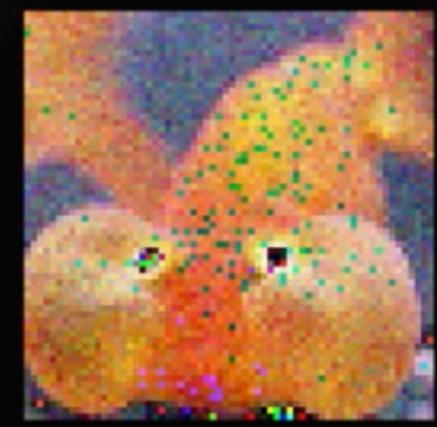


=



-





0

5

10

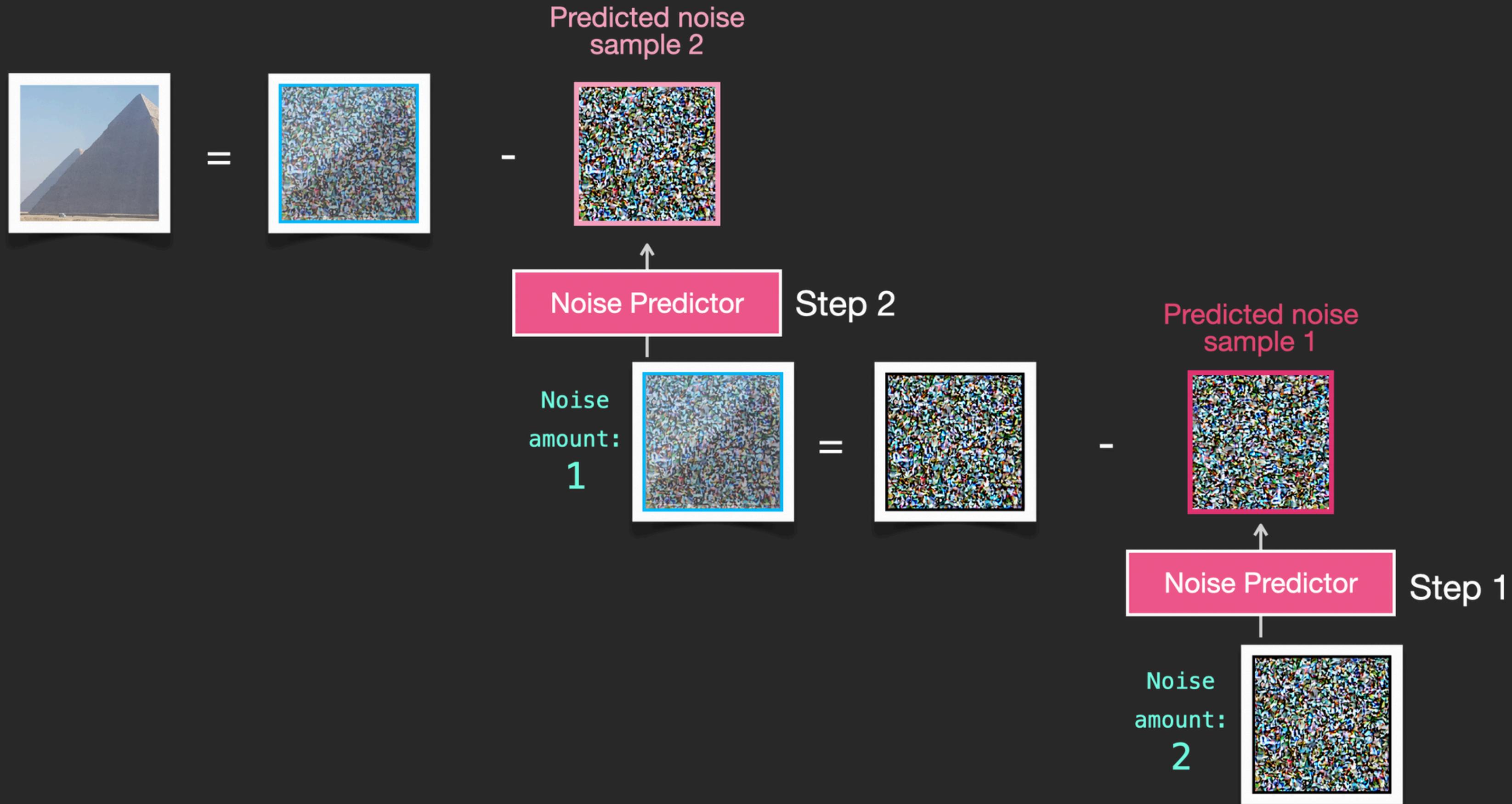
15

20

25

30

Image Generation by Reverse Diffusion (Denoising)



2 special types of layers

**convolutional
layer**

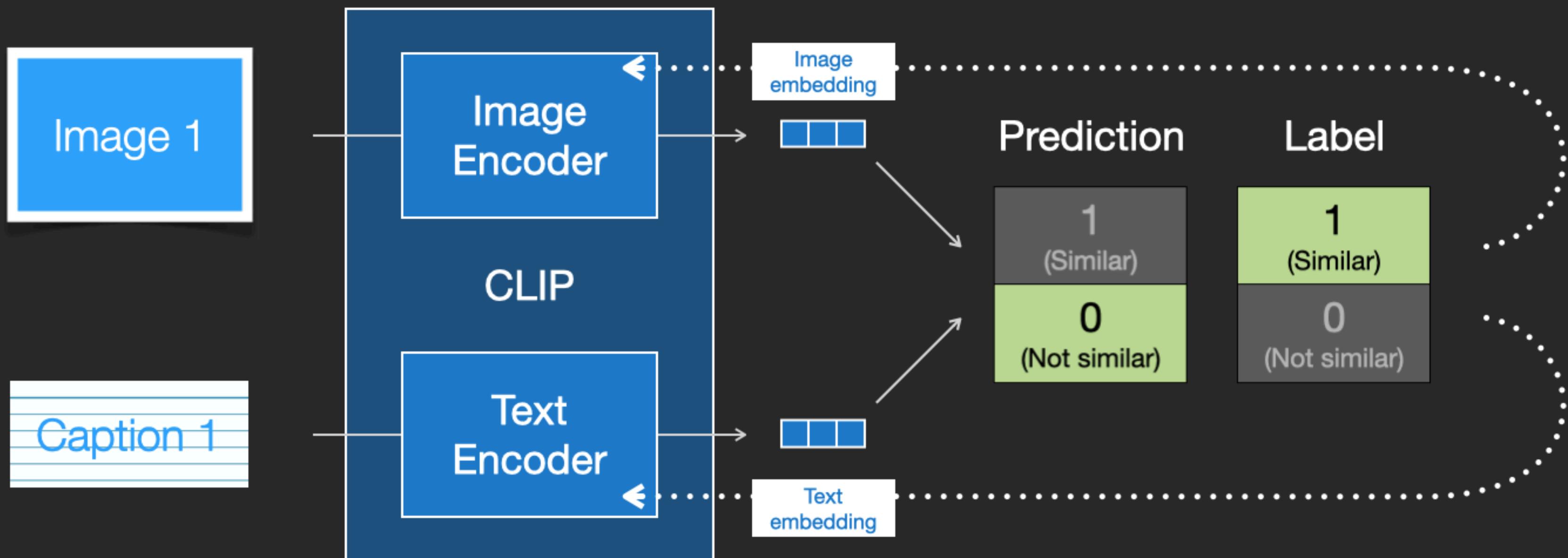
**self attention
layer**

convolution - extract features from **images** using relationships between **pixels**, depending on their **relative spatial position**

self attention - extract features from **phrases** using relationships between **words**, depending on their **embedding vectors**

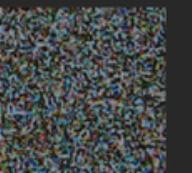
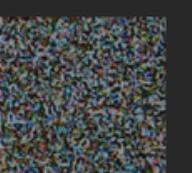
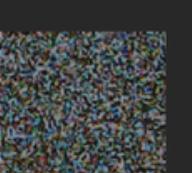
CONDITIONING

- 1 Embed image and text
- 2 Compare the embeddings
- 3 Update the models

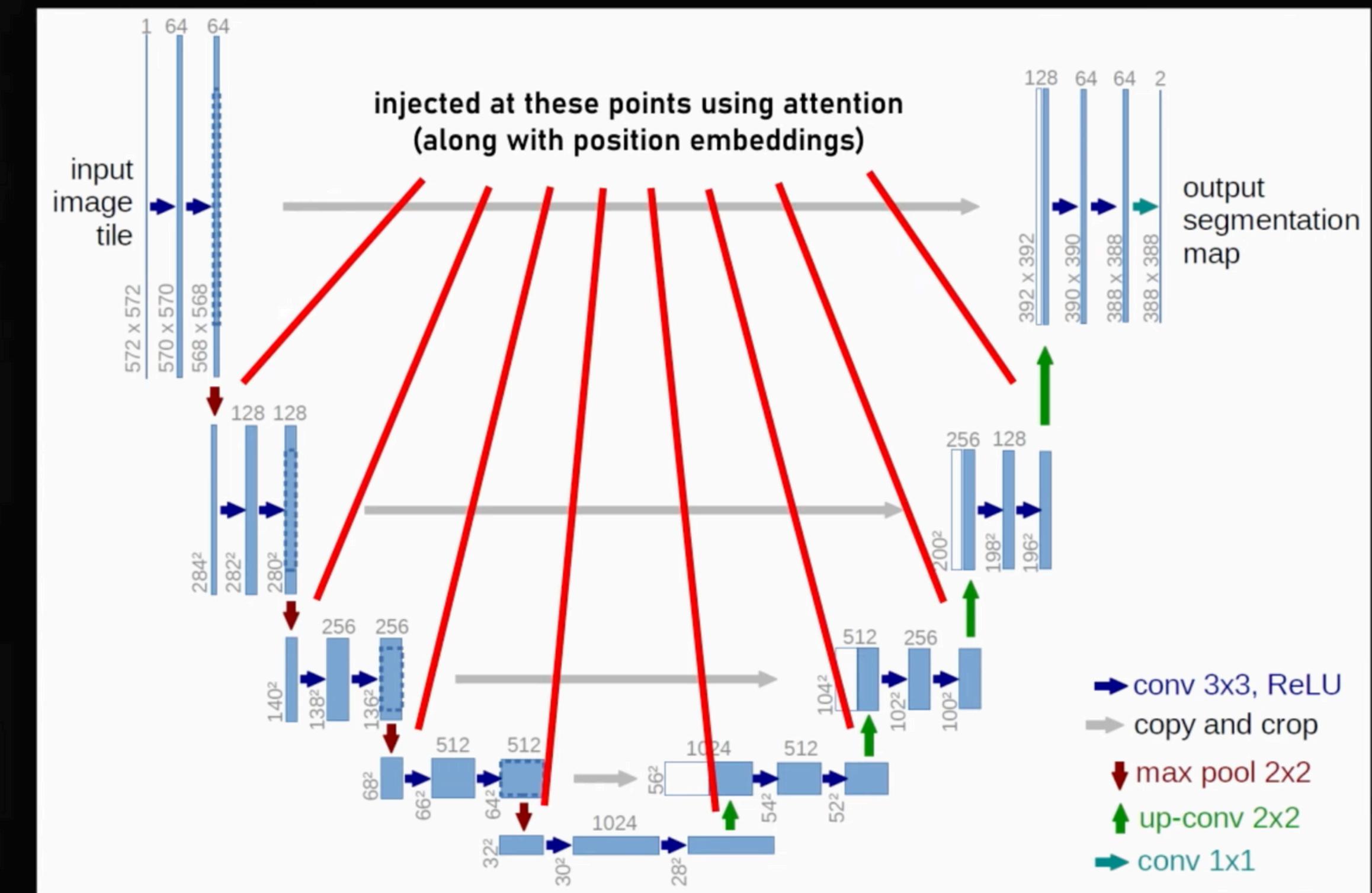
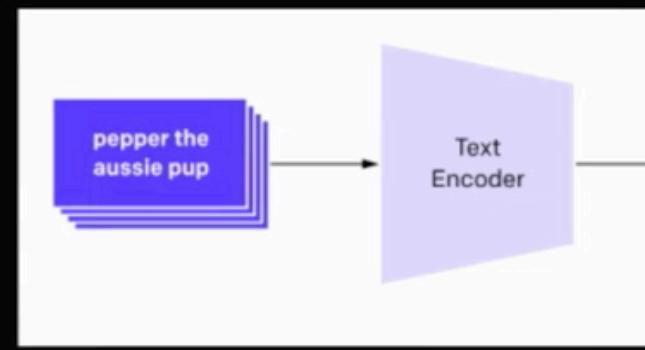


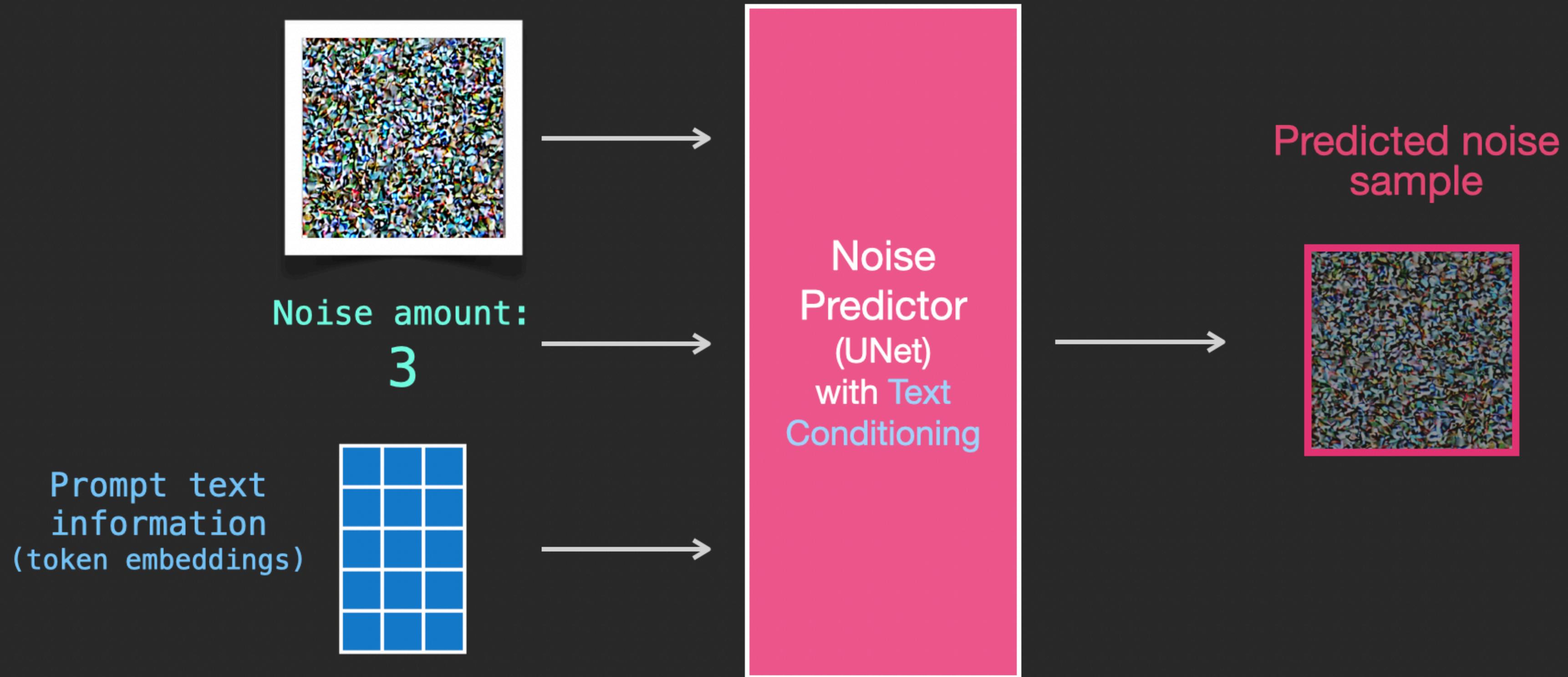
DATASET

MODEL

Step	INPUT		noise sample
	Image	Text	
3			
14			
7			
42			
2			
21			

Noise Predictor
(UNet)
with Text
Conditioning





2.CREATING AI INFLUENCERS



2.1 DALLE-2 INPAINTING

[HTTPS://GITHUB.COM/PROFI7/DALLE2APP](https://github.com/profi7/dalle2app)

2.2 STABLE DIFFUSION INPAINTING

[HTTPS://GITHUB.COM/PROFI7/IMAGE_INPAINTER](https://github.com/profi7/IMAGE_INPAINTER)

2.2 STABLE DIFFUSION LORA

HTTPS://GITHUB.COM/PROFI7/DALLE2-STABLEDIFFUSION-LORA

CONCLUSIONS