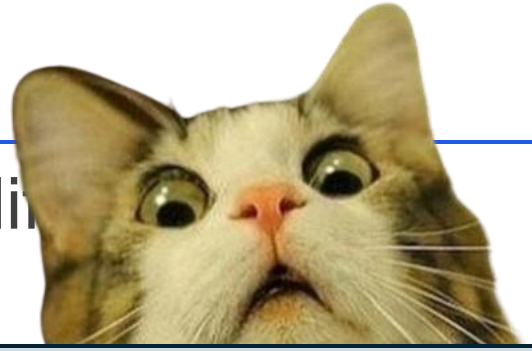
Building Blocks

ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

SCHMIDT FUTURES

Building Blocks

Here is the algorithm for di



Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \text{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on

$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$

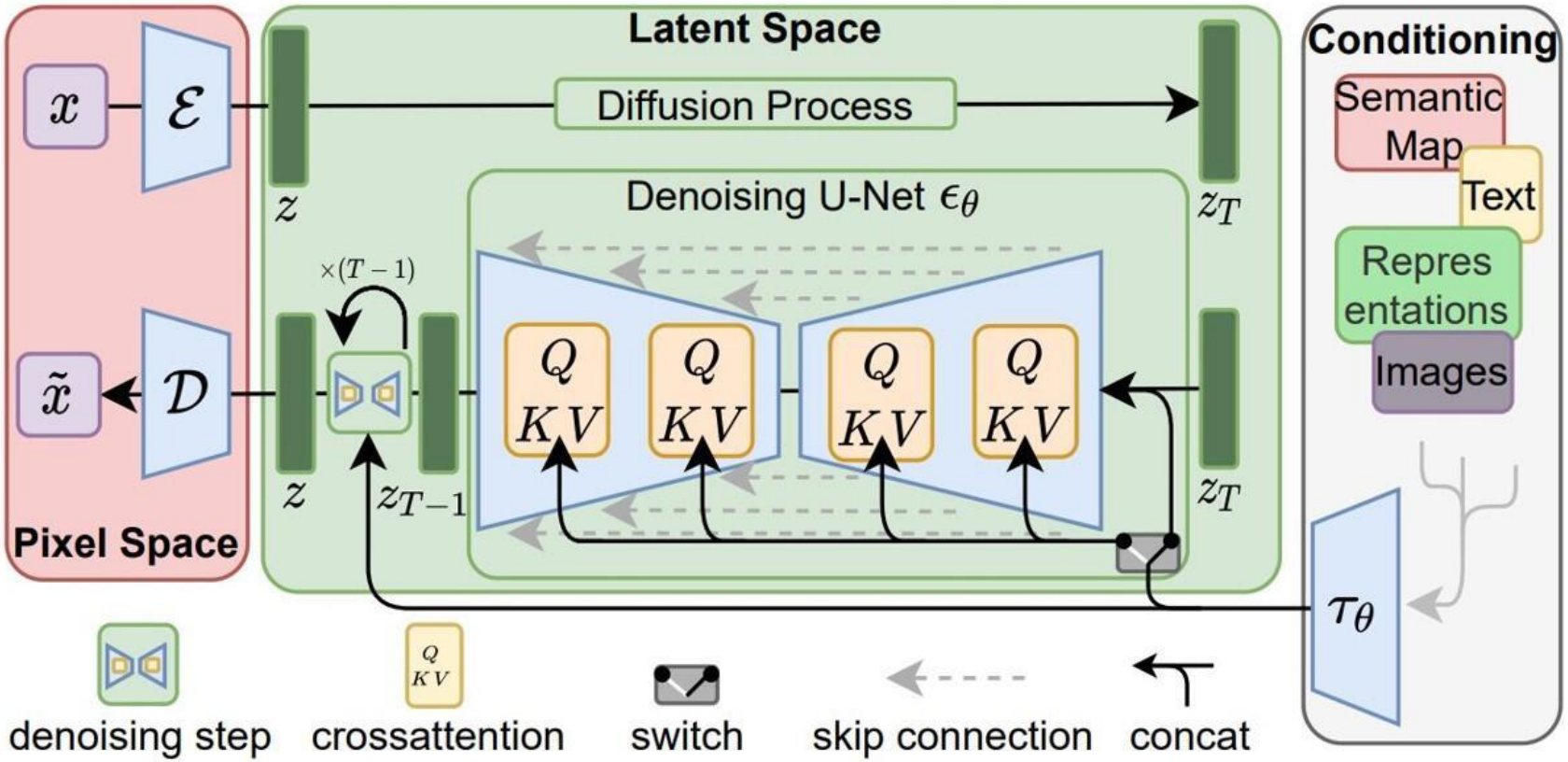
until converged

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

Building Blocks

Perhaps this makes more sense…



ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

SCHMIDT FUTURES

Building Blocks

Aims of this section:

- By the end of the session we want to be able to understand the architecture.
- We start with the architecture:
  - Variational Autoencoders
  - UNet
  - CLIP
- And some common operations in deep learning:
  - Convolution
  - Normalization
  - Pooling
  - Residual connections
  - Upsampling

Variational Autoencoders

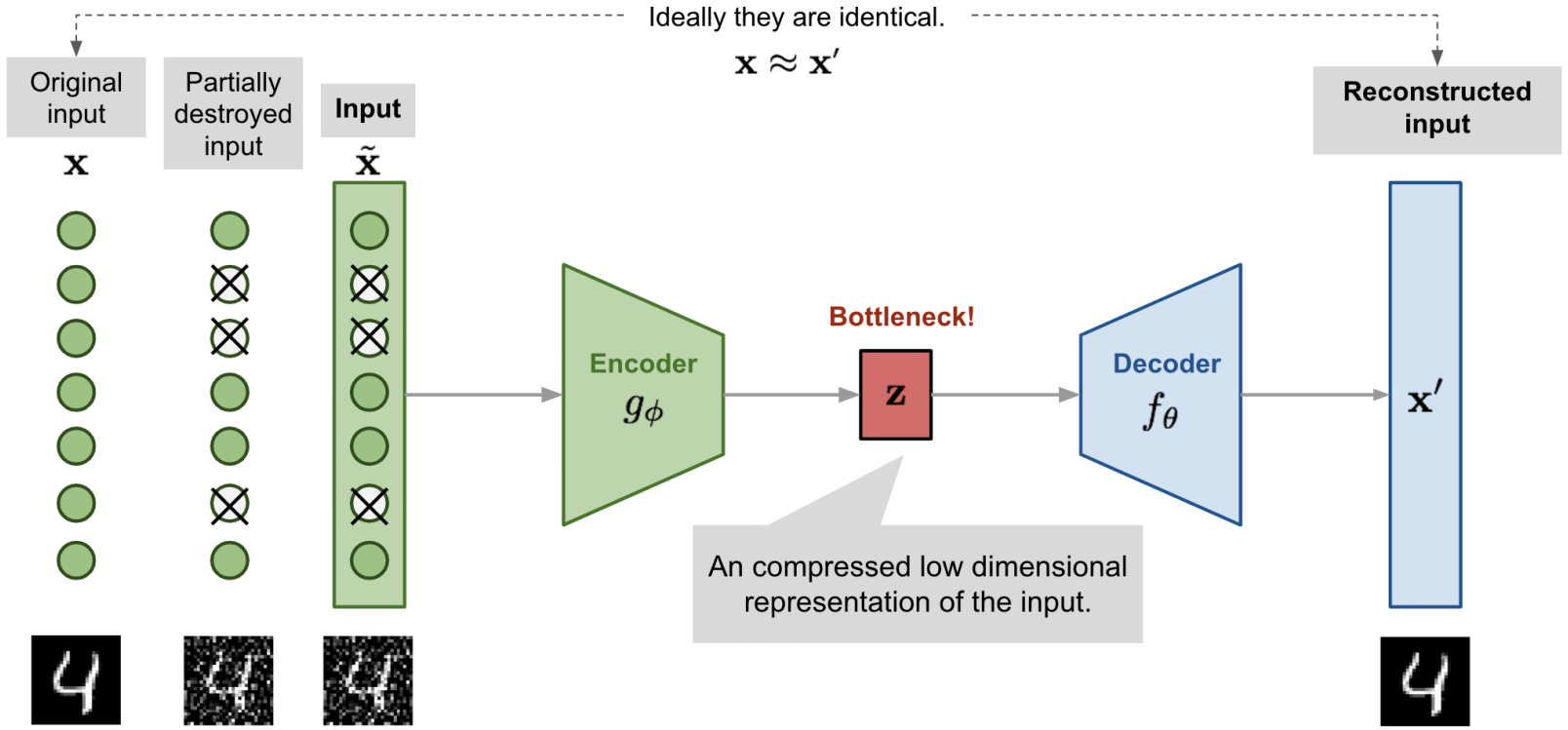
ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

SCHMIDT FUTURES

VAEs

A simple example…

- Suppose you want to create a neural network that denoises images.
- One way of doing this is by using an autoencoder…



ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

SCHMIDT FUTURES

VAEs

But there are some issues with using this kind of autoencoder…

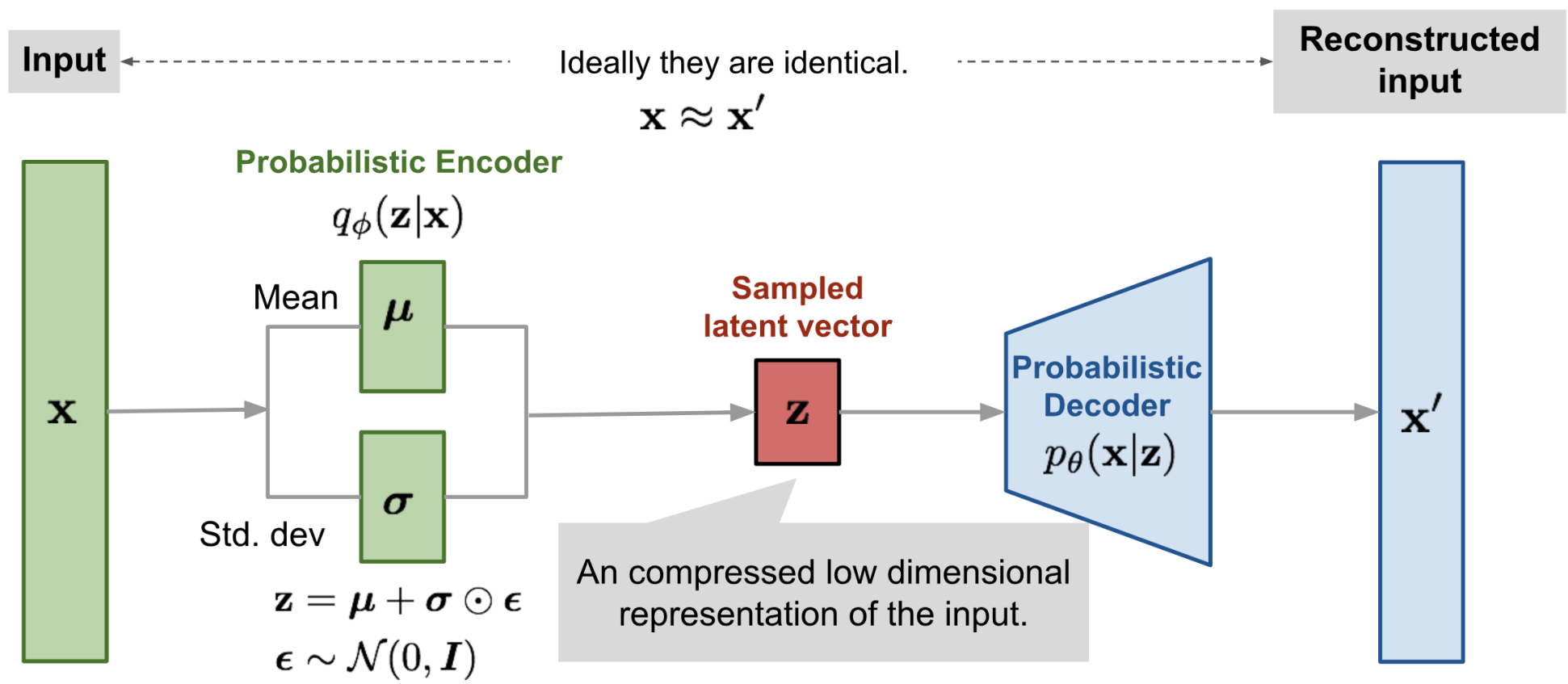
- Gaps between latent points, where the model doesn't really know what is supposed to be there
- It is deterministic – the encoder will map input data to a single point in latent space and then decode it back to the original space.
- This mapping limits the ability to generate samples
- They are not really trying to model the underlying data distribution

VAEs

Instead of trying to build a latent space encoding, why not try and model a distribution…

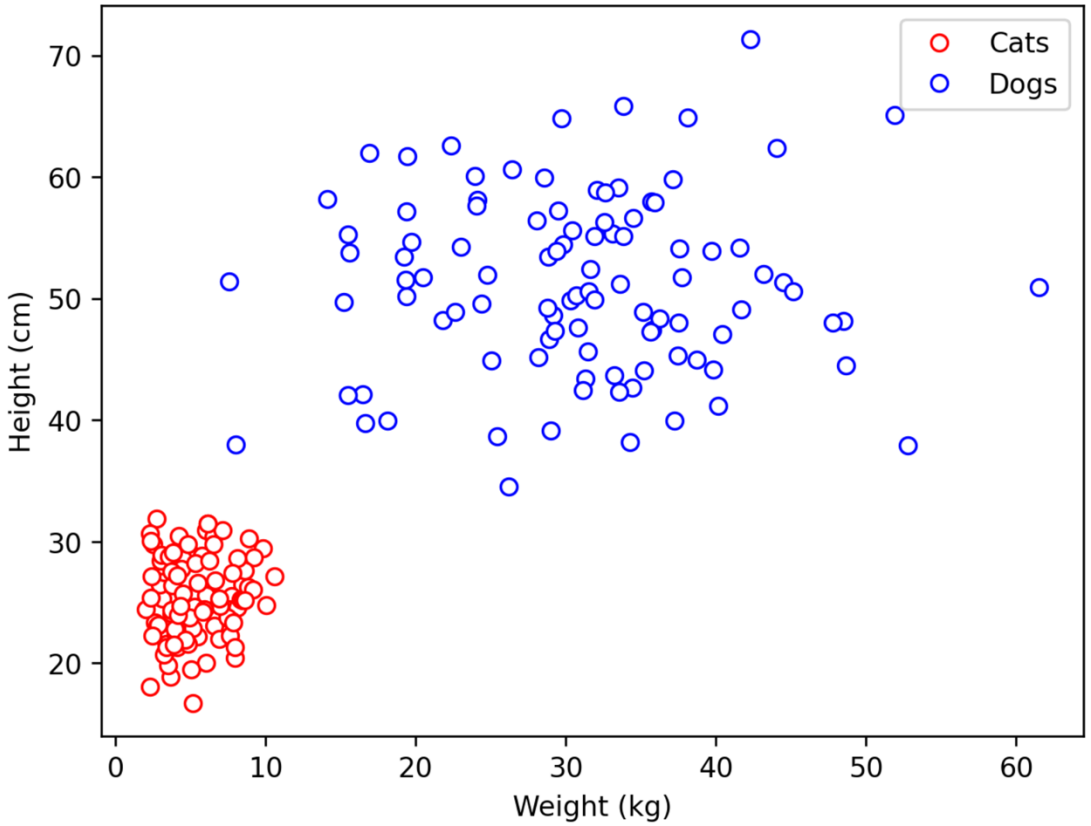
- We should try and map the input space to a latent *distribution* rather than a single point in space
- We should therefore have a continuous and dense latent space
- This will enable smooth interpolation between points
- We impose restrictions on the latent space to force it to map the input data to the distribution that we want.
- This distribution is the normal distribution. This is because we can draw from it easily, and computing gradients is also easy.

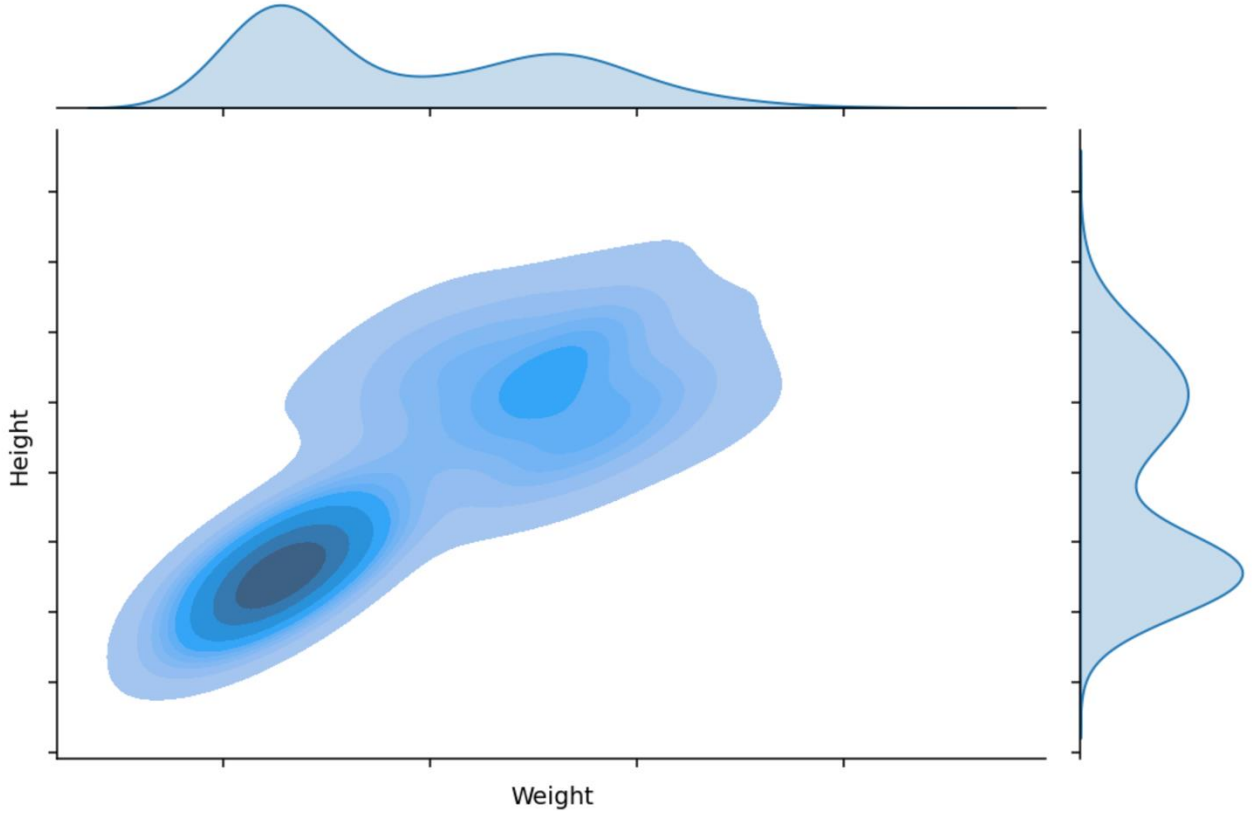
VAEs


Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
Probabilistic Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
Mean
μ
σ
Std. dev
$\mathbf{x}$
$\mathbf{z} = \mu + \sigma \odot \epsilon$
$\epsilon \sim \mathcal{N}(0, I)$
Sampled latent vector
$\mathbf{z}$
An compressed low dimensional representation of the input.
Probabilistic Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
$\mathbf{x}'$

VAEs

Back to cats and dogs…





VAEs

Latent variables

- Latent variables represent the underlying factors that generate our observed data x (height and weight).
- The latent variable z can represent whether the animal is a cat or dog – the data example has small height and weight **because** it is a cat, and vice versa
- In a VAE, we assume that these latent variables follow a standard Gaussian:

$$p_\theta(z) = \mathcal{N}(0, I)$$

- We initially do not distinguish between cats and dogs, and assume that all latent variables are drawn from the same distribution.
- This distribution is the **prior** – it is an assumption we make about how the values in our latent variables are distributed.

VAEs

Decoder

- The decoder says given some latent z, what should my height and weight (i.e. x) be. This is given by:

$$p_\theta(x|z)$$

- If z is a cat, then the decoder will give measurements that (hopefully) fall within the cat cluster.
- In other words:

  "Given that z is a cat, generate a height and weight that fits within the cat Gaussian cluster."
- This is the **Likelihood**.

VAEs

Posterior

- The true posterior $p_\theta(z|x)$ gives the probability of the example being a cat (or dog), given the height and weight that you have shown me.
- To calculate this value, we can use Bayes' Theorem:

$$p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}$$ Problem!

- This bottom term asks: what's the probability of this particular height and weight…but this is a problem:

$$p_\theta(x) = \int p_\theta(x|z)p_\theta(z)\,dz$$ We cannot solve this!

VAEs

So what do we do?

- We approximate it.
- Use a neural network to find $q_\phi(z|x) \approx p_\theta(z|x)$
- But remember, we want to force the latent space distribution to be as close as possible to Normal. In other words we want to learn:

$$q_\phi(z|x) \sim \mathcal{N}(E_\phi, \sigma_\phi(x)^2 I)$$

- Now we want to do 2 things:
  - Enforce the latent space to take on the same structure as our prior distribution
  - Ensure that the output of the decoder is as close as possible to the original input.
- But we also wanted to force this **learned** distribution to be as close to the **true** distribution as possible.

$$L = -\mathbb{E}_{q(Z|X)}[\log p_\theta(x|z)] + \beta D_{KL}\big(q_\phi(z|x) \| p_\theta(z)\big)$$

U-Net

ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

SCHMIDT FUTURES

U-Net

U-Net is a classic neural network architecture

- U-Net was originally developed for biomedical image segmentation.
- The original paper *U-Net: Convolutional Networks for Biomedical Image Segmentation* was presented at MICCAI in 2015, and now has nearly 85k citations!

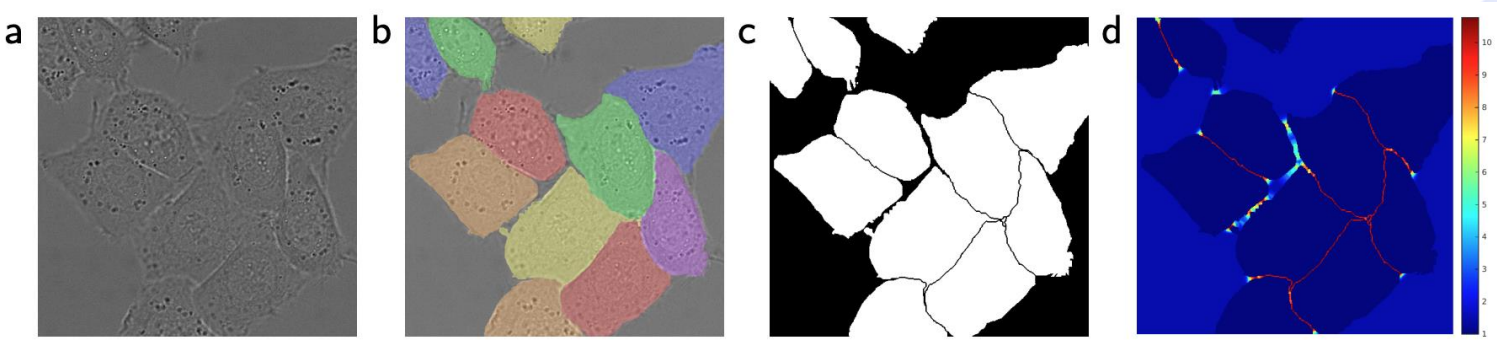


Fig. 3. HeLa cells on glass recorded with DIC (differential interference contrast) microscopy. (**a**) raw image. (**b**) overlay with ground truth segmentation. Different colors indicate different instances of the HeLa cells. (**c**) generated segmentation mask (white: foreground, black: background). (**d**) map with a pixel-wise loss weight to force the network to learn the border pixels.

U-Net

U-Net was originally designed for semantic segmentation

- The input data consists of the original image
- For the label data, each corresponding pixel is given a label.

0.54	0.64	0.36	0.58	0.48
0.90	0.89	0.91	0.87	0.89
0.91	0.90	0.85	0.89	0.90
0.46	0.48	0.59	0.56	0.47
0.54	0.64	0.36	0.58	0.48

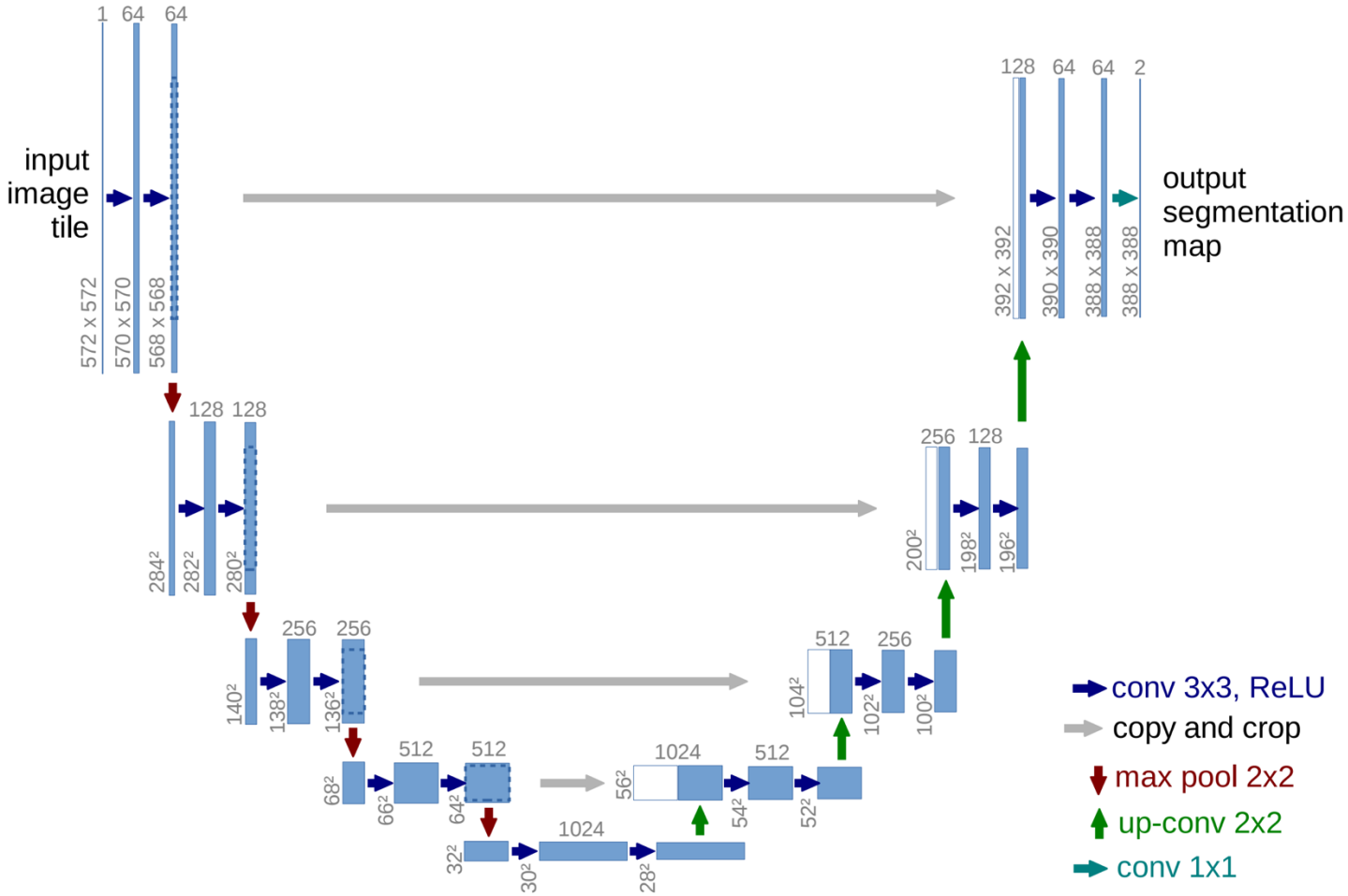
grass	grass	grass	grass	grass
road	road	road	road	road
road	road	road	road	road
grass	grass	grass	grass	grass
grass	grass	grass	grass	grass

ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

SCHMIDT FUTURES

U-Net

Perhaps unsurprisingly, U-Net has a distinctive U-shaped architecture…



ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

SCHMIDT FUTURES

U-Net

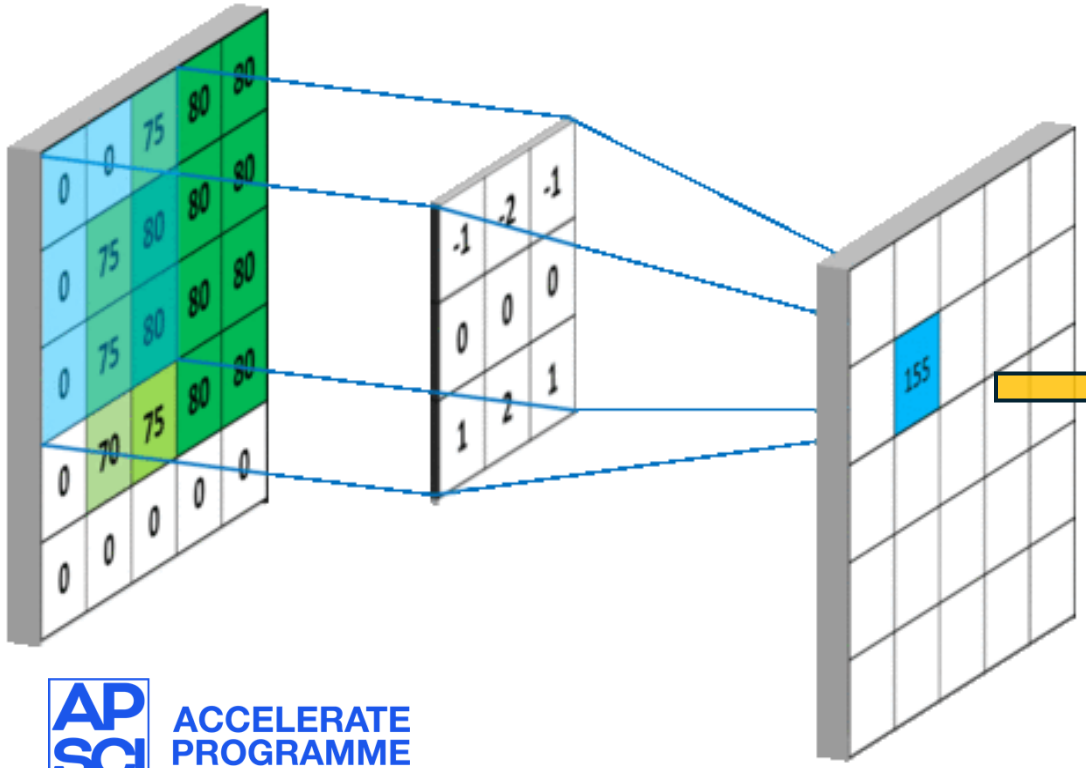
There are some key components of U-Net that are worth going over…

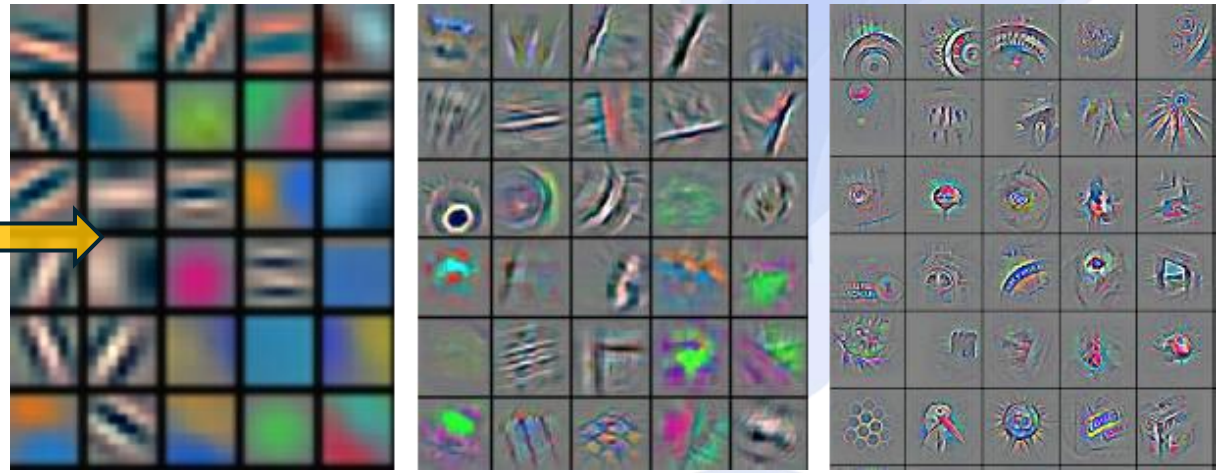
- Convolution
- Residuals
- Upsampling
- Pooling
- Normalization

U-Net

Convolution was a major advancement in machine vision

- *Gradient-based Learning Applied to Document Recognition*, 1998, LeCun et al, ~65k citations
- *ImageNet Classification with Deep Convolutional Neural Networks*, 2012, Krizhevsky et al, ~130k citations





Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13. Springer International Publishing, 2014.

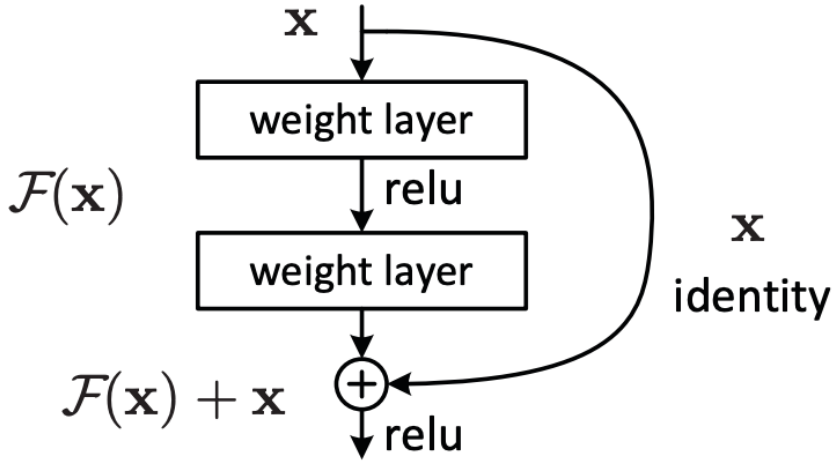
ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

SCHMIDT FUTURES

U-Net

Like CNNs, residual neural networks were a major step forward for deep learning.

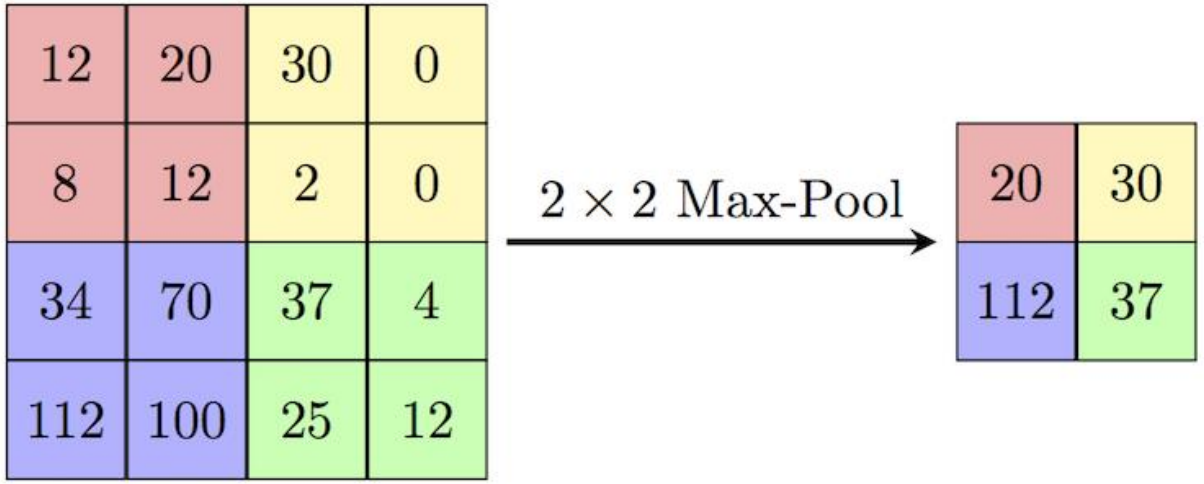
- *Deep Residual Learning for Image Recognition*, He et al, Microsoft Research, 2015 ~210k citations
- Most neural networks now use residual connections.
- After AlexNet, the main idea was simply to make the neural network bigger.
- However, after a certain point, error appeared to increase that was not attributable to overfitting.
- Instead of trying to learn a complicated function to transform the inputs directly to the outputs, we learn: “what do we need to change in order to transform the inputs to the outputs”
- The trick is to add skip connections:



U-Net

Pooling

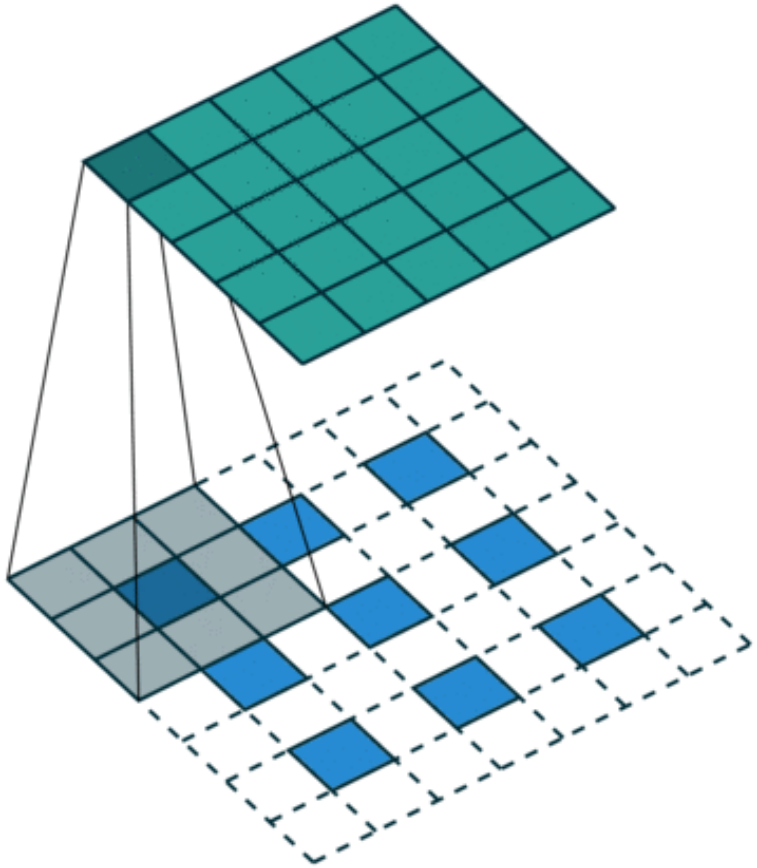
- Yamaguchi et al. "A neural network for speaker-independent isolated word recognition." 1990
- It’s a very simple concept:



- Pass a window across the filter and take the maximum value
- The maxpool operation is defined by the stride and the filter size
- It preserves important features and downscales to make computation less intense
- Makes the networks somewhat equivariant.

U-Net

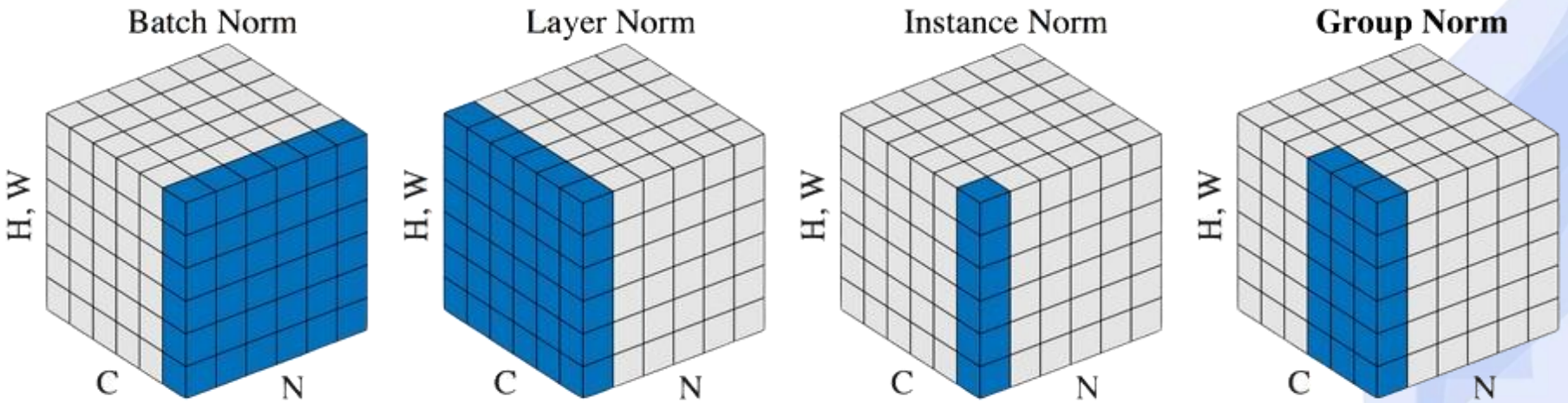
Upsampling



U-Net

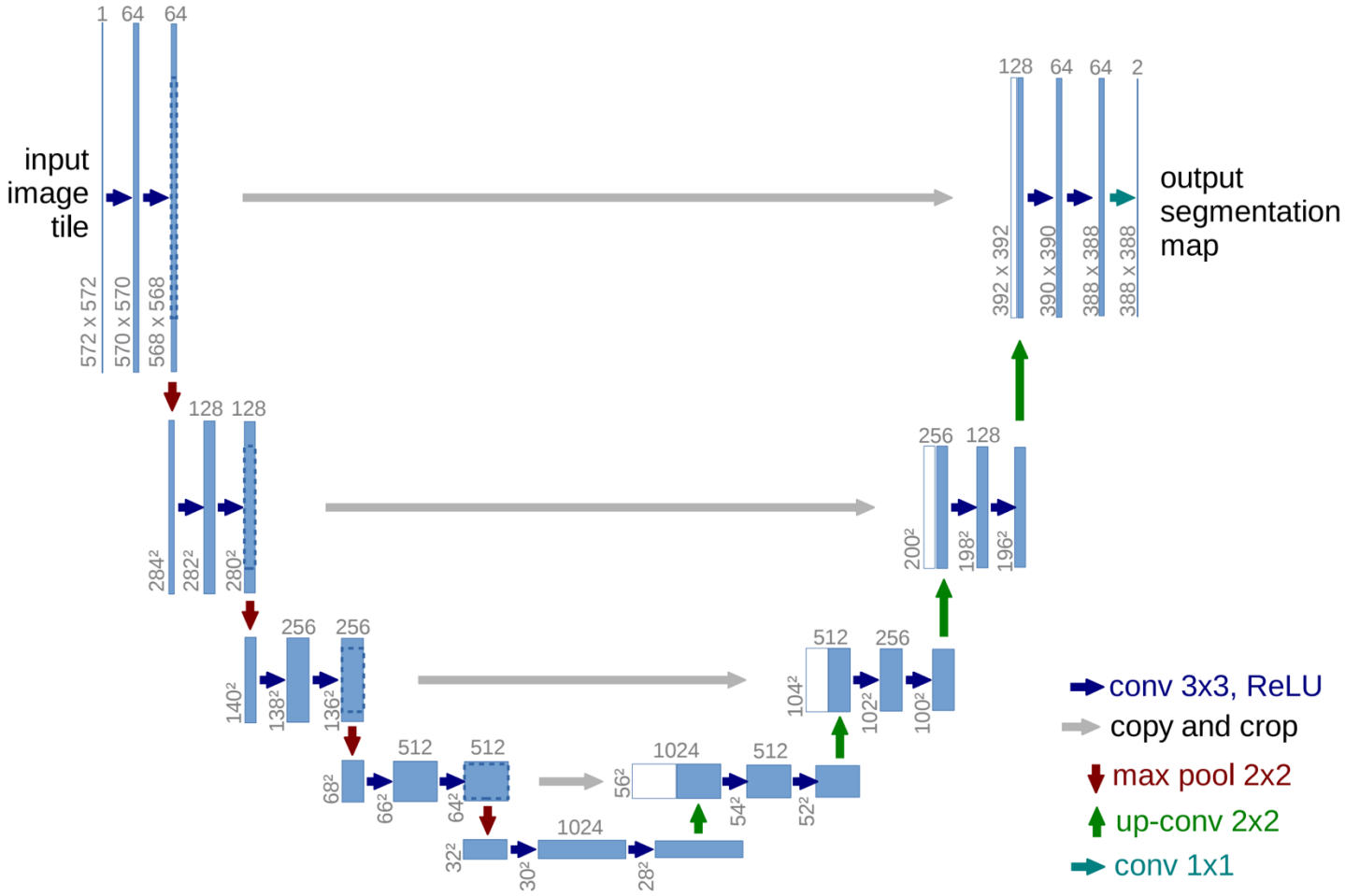
Normalization

- There are a few different types of normalization:



U-Net

Let’s look at the U-Net architecture again…



ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

SCHMIDT FUTURES

CLIP

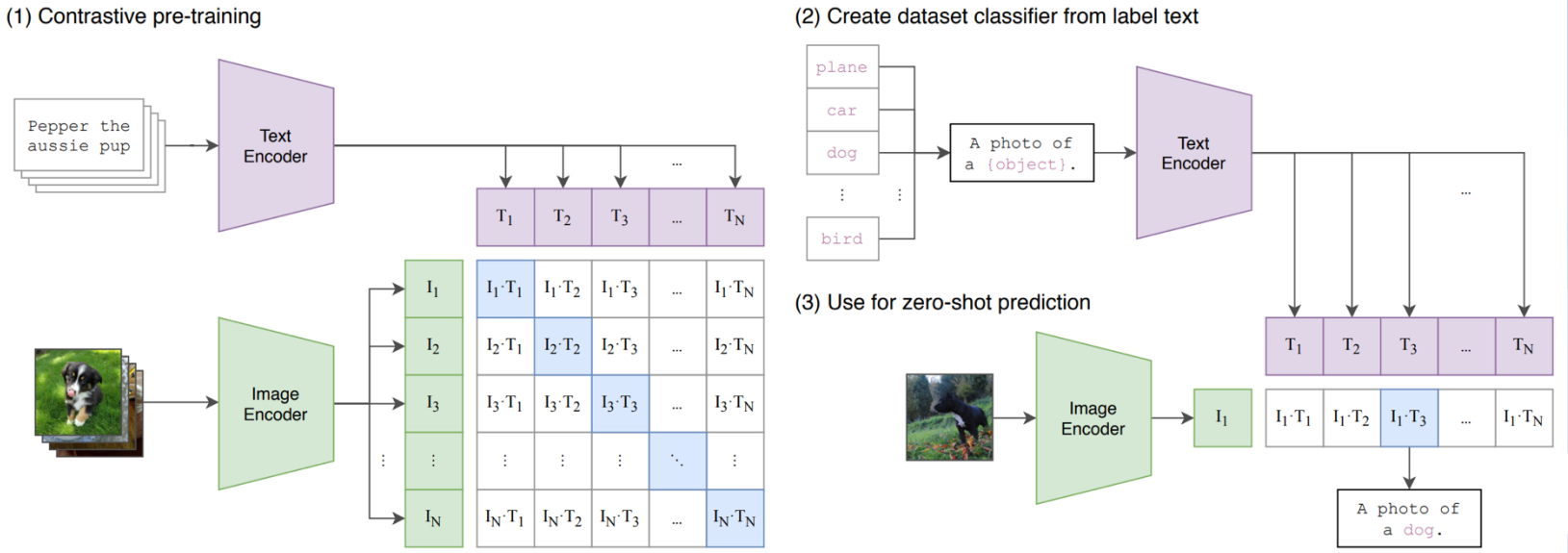
ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

SCHMIDT FUTURES

CLIP

Contrastive Language-Image Pre-training

- *Learning Transferable Visual Models for Natural Language Supervision*, Radford et al, 2021, ~17k citations
- Takes in images and texts and connects them together



Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PMLR, 2021.