Park *et al.*: Contrastive learning for unpaired image-to-image translation

A tutorial

Written by:
Domonkos VARGA

Table of contents

- What is contrastive learning?
  - Contrastive learning of visual representations
  - SimCLR
- What is image-to-image translation? What is unpaired image-to-image translation?
- How does Park et al.'s algorithm work?

Part 1. What is contrastive learning?

Contrastive learning of visual representations

- Motivation

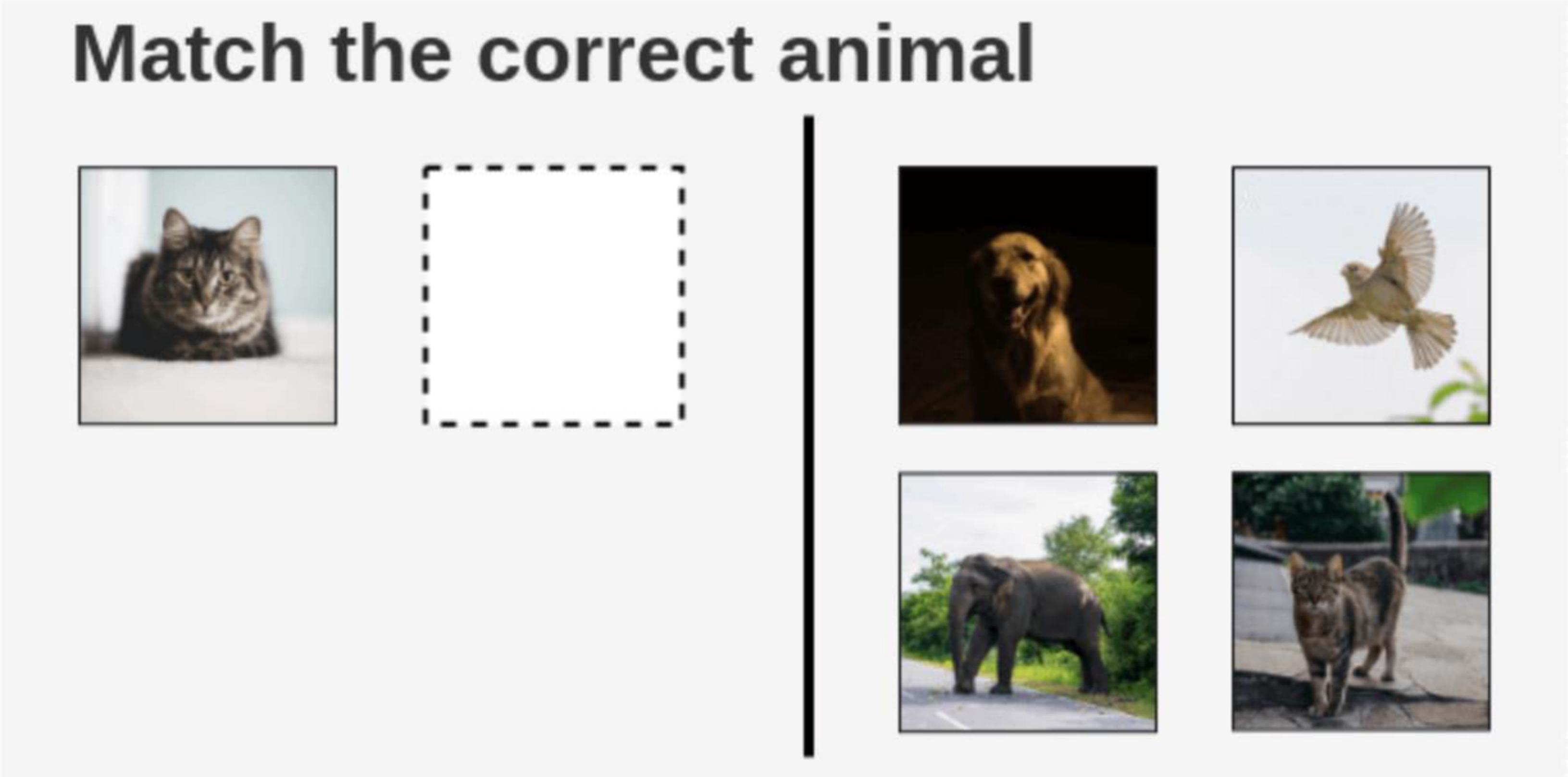
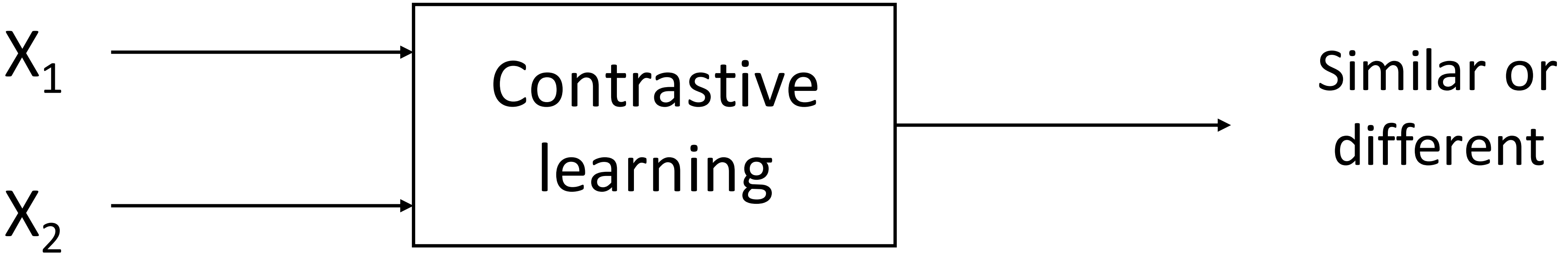


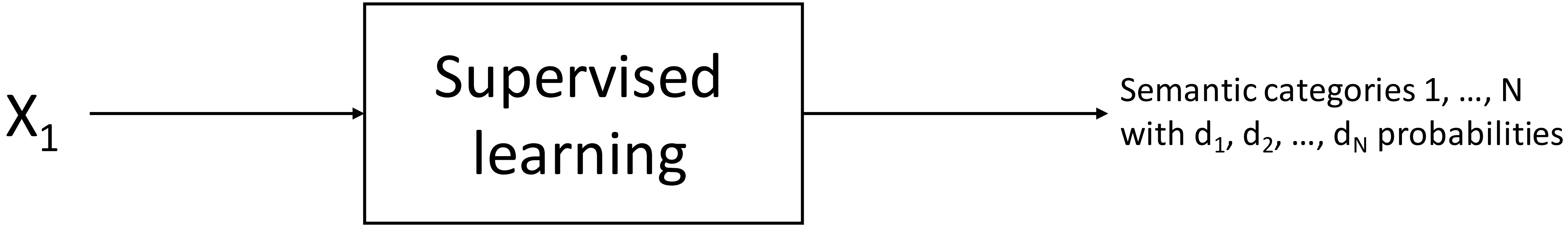
Image credit: Amit Chaudhary

Contrastive learning of visual representations

- Contrastive learning



- Supervised learning



Contrastive learning advantages and disadvantages

- Advantages:

1. Unsupervised, it does not need a labeled dataset.
2. Better results than supervised learning.
3. Far better results in comparison to other unsupervised techniques.
4. Promising results in other areas, such as audio and language.
5. Benefits stonger on data augmentation than supervised learning.

Contrastive learning advantages and disadvantages

- Disadvantages:

1. Larger batch size is required than in case of supervised methods.
2. More training steps are required than in supervised methods.
3. Deeper and wider networks are required.

Contrastive learning of visual representations

Formulation:

1.) Examples of similar and dissimilar images.

2.) Image representation: a mechanism is required that is able to obtain an image representation which is suitable for machines (on ImageNet database pretrained CNNs, such as AlexNet, VGG16, ResNet50, *etc.*)

3.) Defining similarity.

Contrastive learning for visual representations

- Chen *et al.* in A simple framework for contrastive learning of visual representations proposed a framework – codenamed "SimCLR" – for tackling the problems enumerated in the previous slide.
- Park *et al.* in Contrastive learning for unpaired image-to-image translation heavily rely on the work of Chen *et al.* so we have to take a closer look at SimCLR.

SimCLR

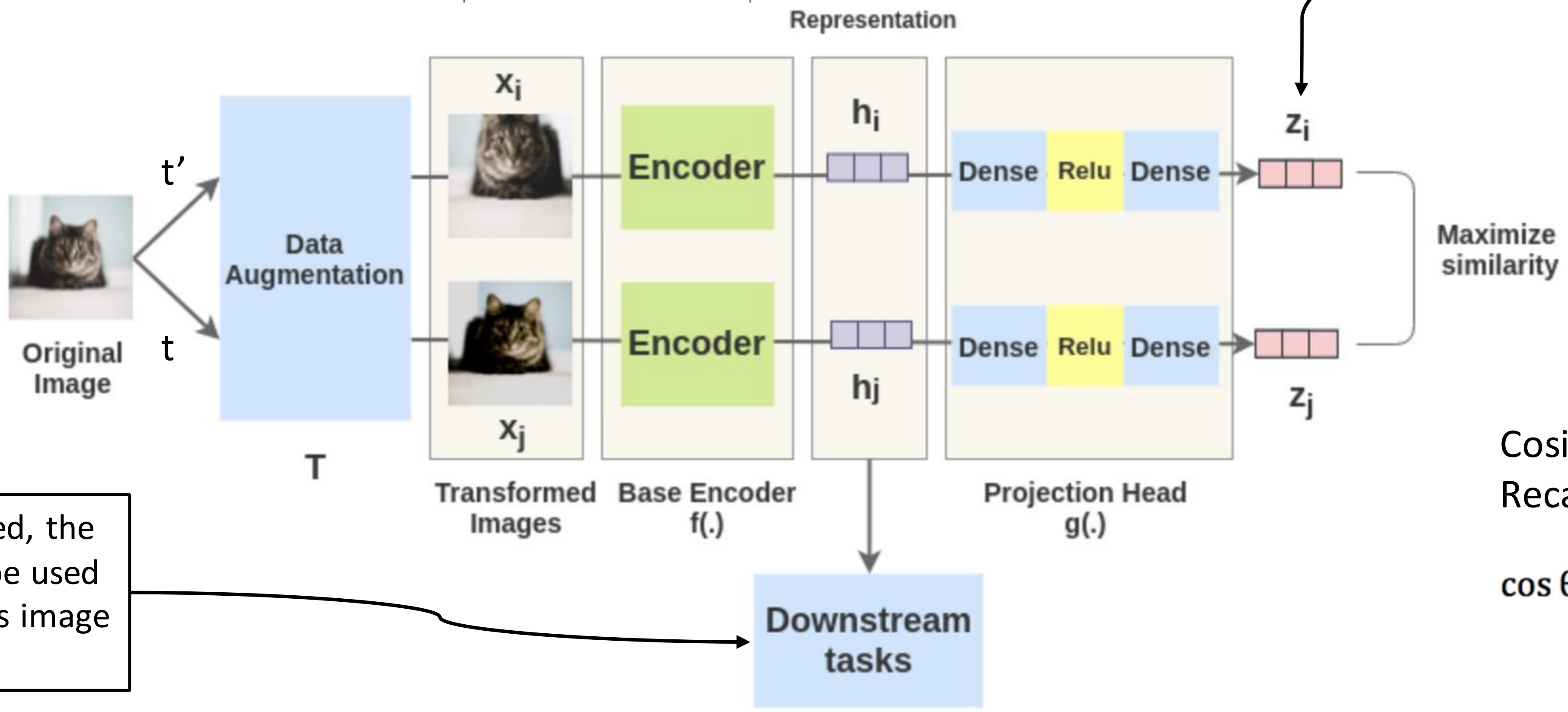
- Overview

Obtaining representation using a pretrained CNN

Projection head

2048-dimensional vector

128-dimensional vector



Once SimCLR is trained, the representations can be used for other tasks, such as image classification.

Cosine similarity!
Recall:

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\cdot\|\vec{b}\|}$$

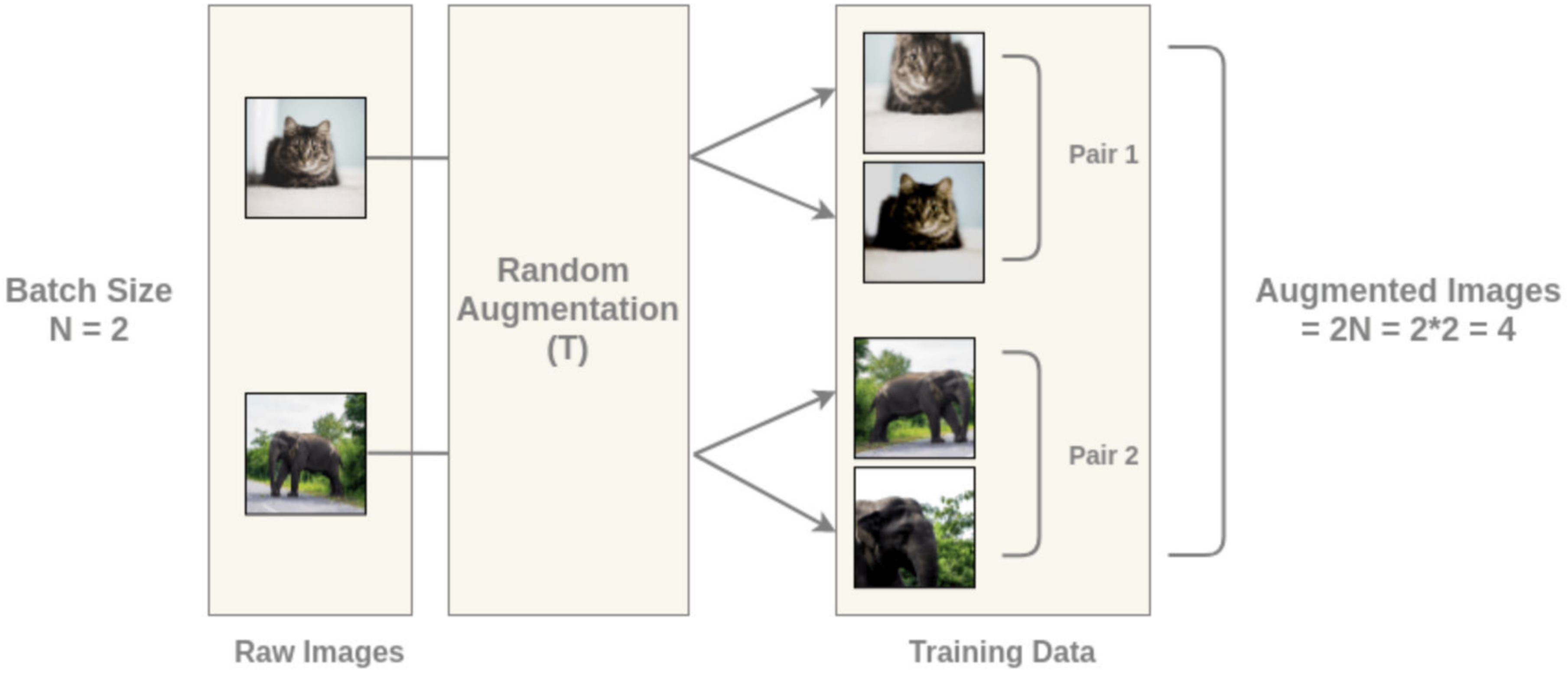
Image credit: Amit Chaudhary

SimCLR

- Step by Step: **Data augmentation**
- SimCLR selects transformation functions – denoted by *T(.)* – that applies a random combination of cropping, flipping, color jitter, and RGB to grayscale conversion.
- Parameter study in the paper justifies these operations.

SimCLR

- Step by step: **Data augmentation**



SimCLR

- Step by step: **Image representation**
- In the past decade, CNNs have gained a lot of attention. It have been shown that features extracted from on ImageNet database pretrained CNNs are able to provide powerful feature representations for a wide variety of tasks.
- In SimCLR, each augmented image is run through a ResNet-50 body which carries out all its defined operations. The image representation is given as 2048-dimensional vectors.

SimCLR

- Step by step: **Projection head**
- The image representations are passed through three layers (Dense - ReLU – Dense) to project them into z_i and z_j. These three layers are called projection head and denoted by function *g(.)* in the original paper.

SimCLR

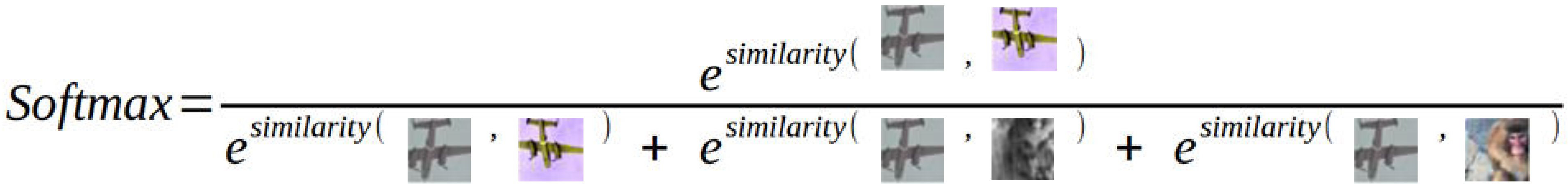
- Step by step: **Similarity**
- The similarity of z_i and z_j are determined by *cosine similarity* which is derived from the formula of Euclidean dot product.

Similarity = $\mathbf{z}_i^T \mathbf{z}_j / (||\mathbf{z}_i|| \, ||\mathbf{z}_j||)$

- Cosine similarity is divided by a temperature hyperparameter. It is left out for simplicity reasons.

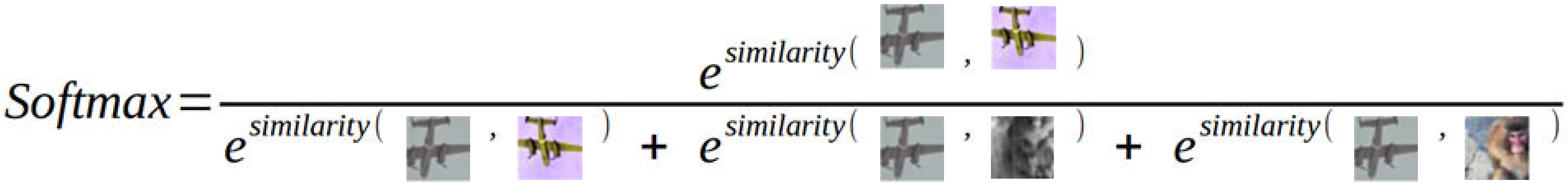
SimCLR

- Step by step: **loss function**
- **SimCLR** proposed the “NT-Xent loss” – Normalized Temperature-scaled Cross-entropy Loss
- **First,** the augmented pairs in the batch are taken one by one. Specifically, *softmax function* is applied to obtain the probability of two images being similar.



SimCLR

- Step by step: **loss function**



- The loss for a pair is determined by taking the negative log of the above calculation.

$$l(i, j) = -\log \frac{\exp\left(s_{i,j}\right)}{\sum_{k=1, k \neq i}^{2N} \exp\left(s_{i,k}\right)}$$

SimCLR

- Step by step: **loss function**
- Another loss for the same pair is also calculated.

$$l(i,j) = -\log \frac{\exp\left(s_{i,j}\right)}{\sum_{k=1,k\neq i}^{2N} \exp\left(s_{i,k}\right)}$$

Interchange images.

$$l(j,i) = -\log \frac{\exp\left(s_{j,i}\right)}{\sum_{k=1,k\neq j}^{2N} \exp\left(s_{j,k}\right)}$$

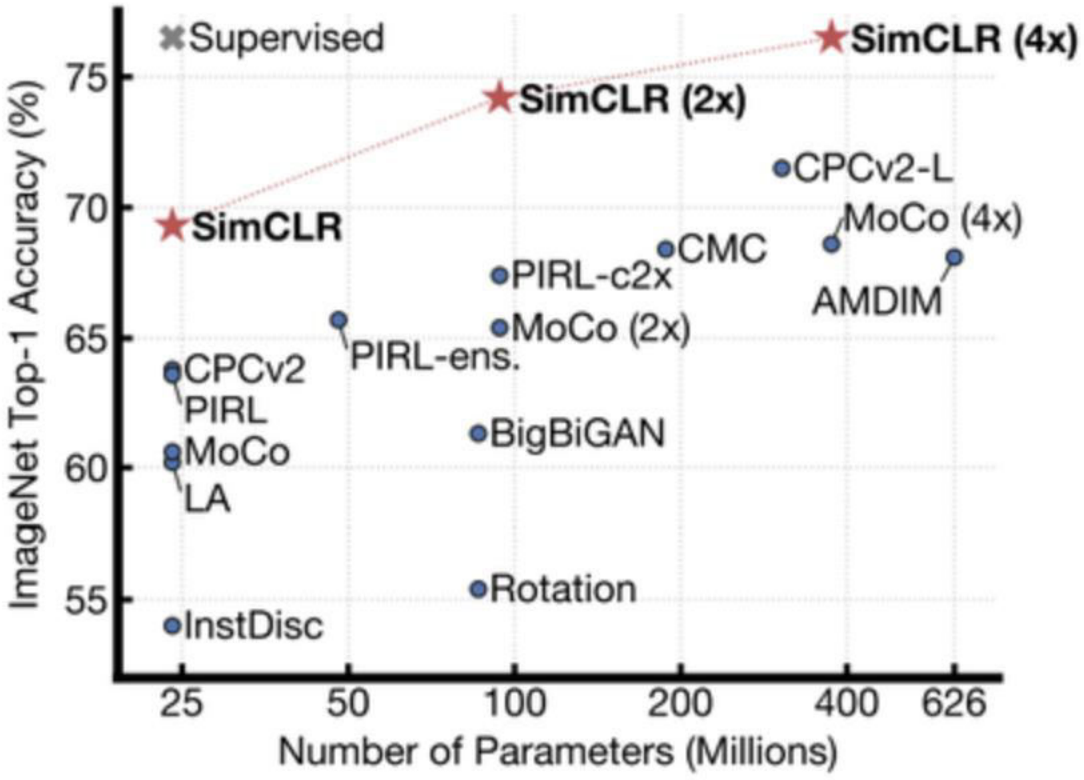
SimCLR

- Step by step: **loss function**
- The overall loss is computed by determining the loss for all pairs in the batch and taking the average

$$L = \frac{1}{2N} \sum_{k=1}^{N} [l(2k-1, 2k) + l(2k, 2k-1)]$$

SimCLR

- Results

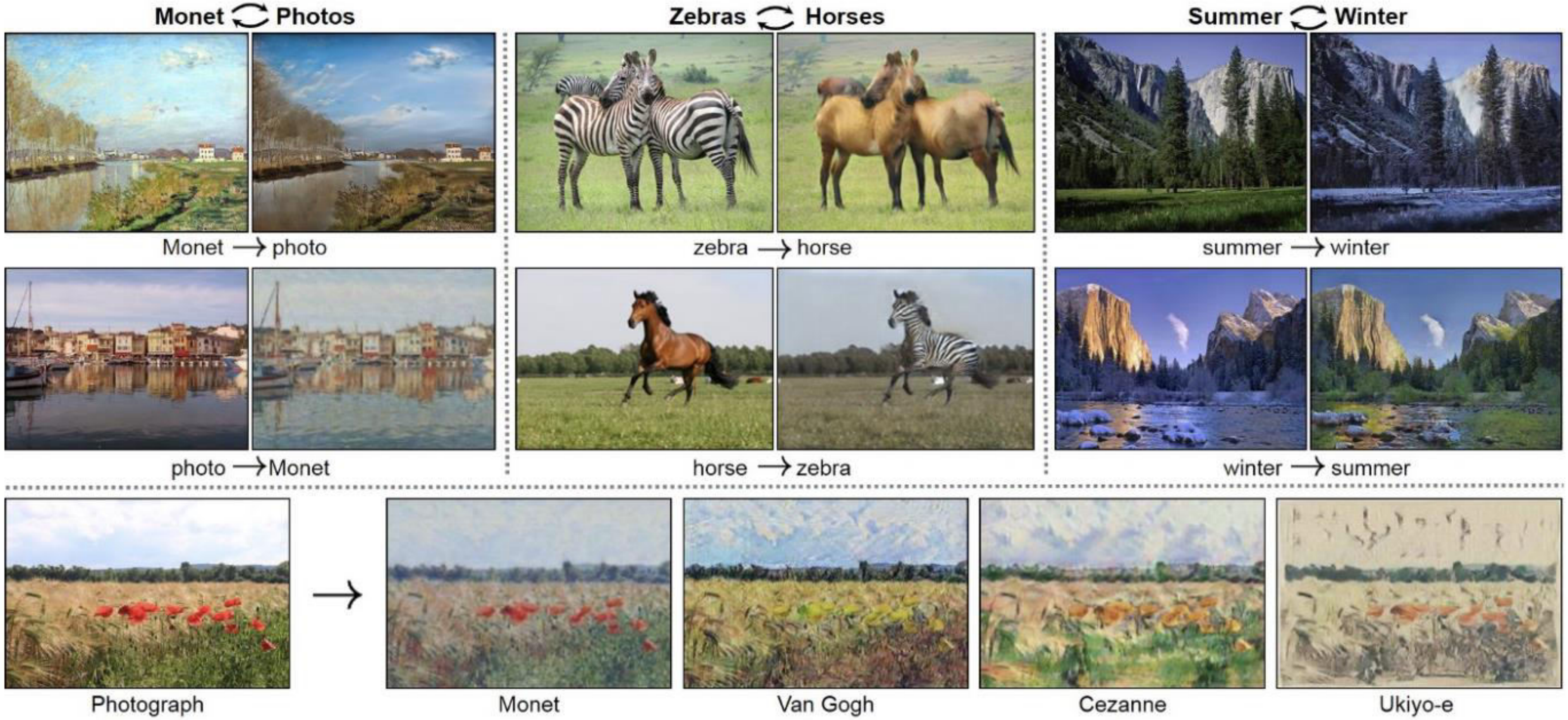


Part 2. What is (unpaired) image-to-image translation?

Image-to-image translation

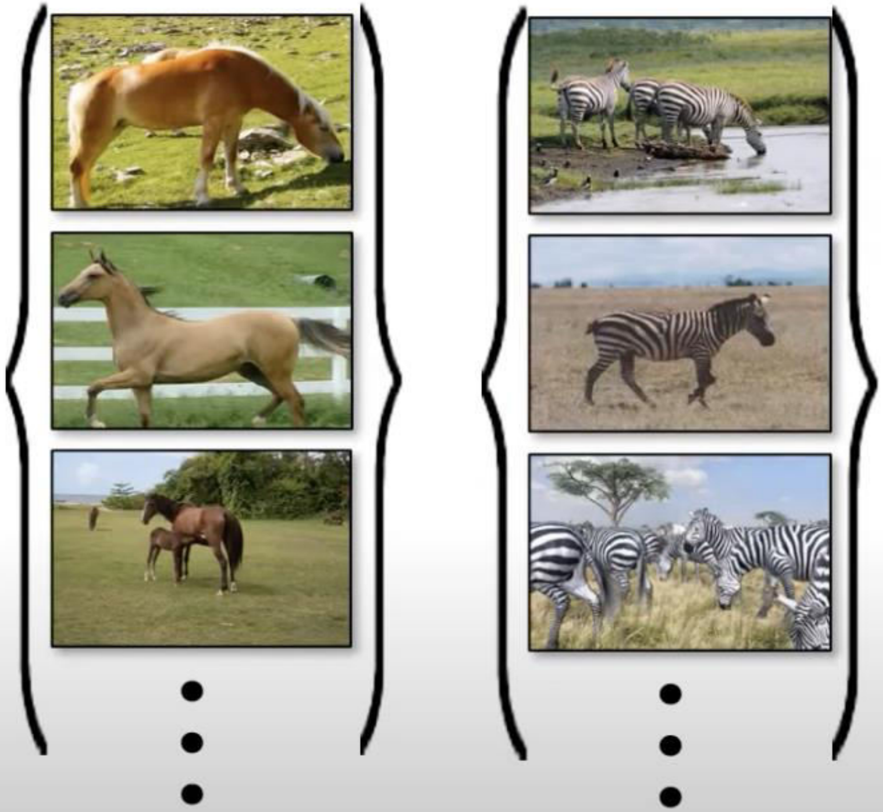
- **Image-to-image translation** is a computer vision and graphics problem where the goal is to transform an input image into another image given certain conditions.
- **Paired**: Full information is provided about the input and output image.
- **Unpaired**: Full information is provided only about the input image. High-level information is given about the desired output image.

Unpaired image-to-image translation

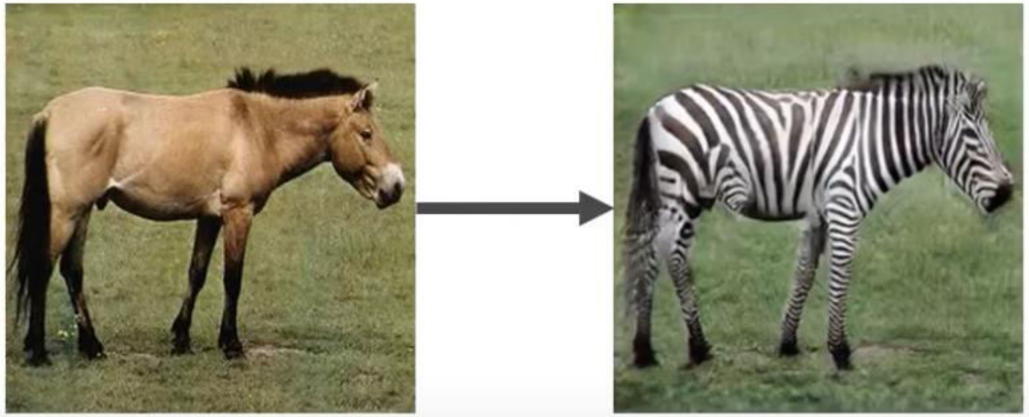


Unpaired image-to-image translation

Training database



Test-time behaviour



Part 3. How does Park et al.’s algorithm work?

Patchwise contrastive learning

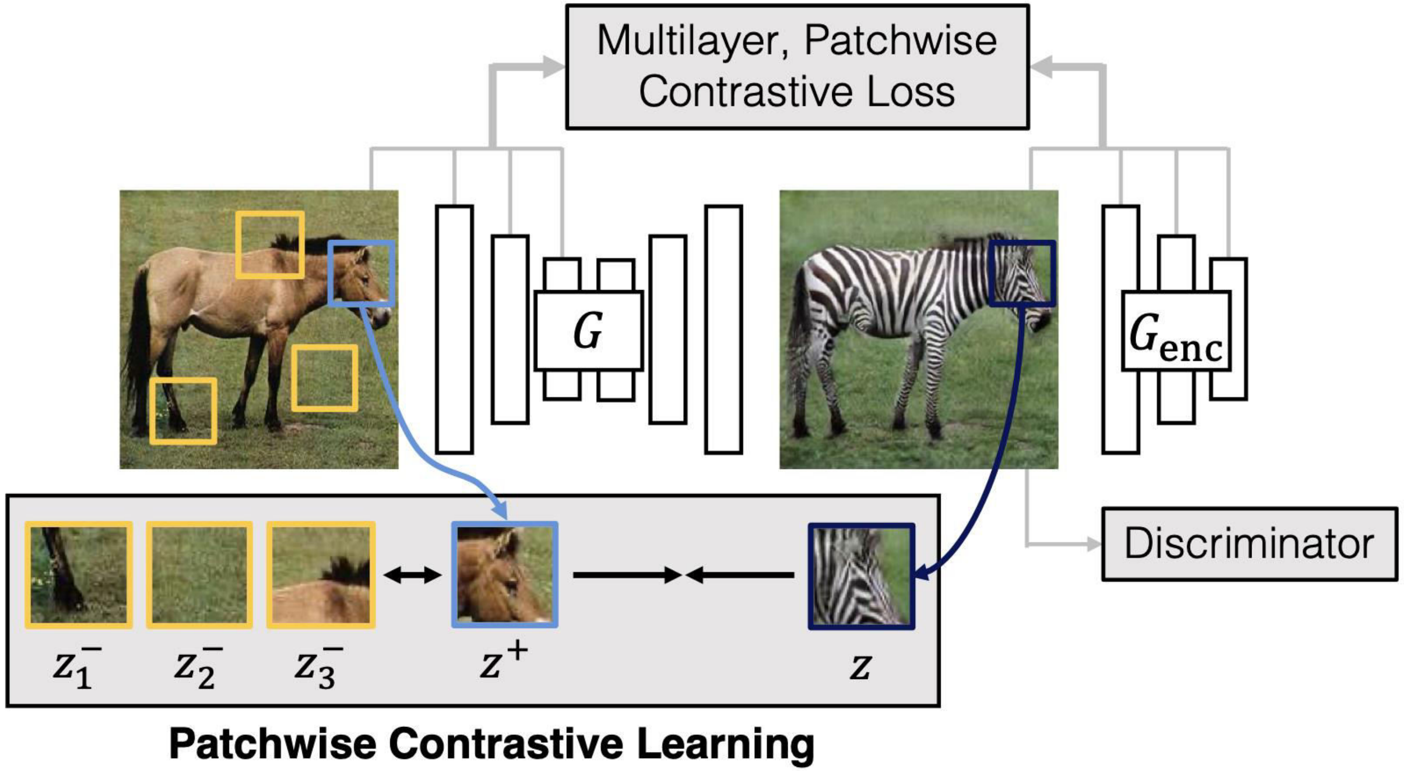
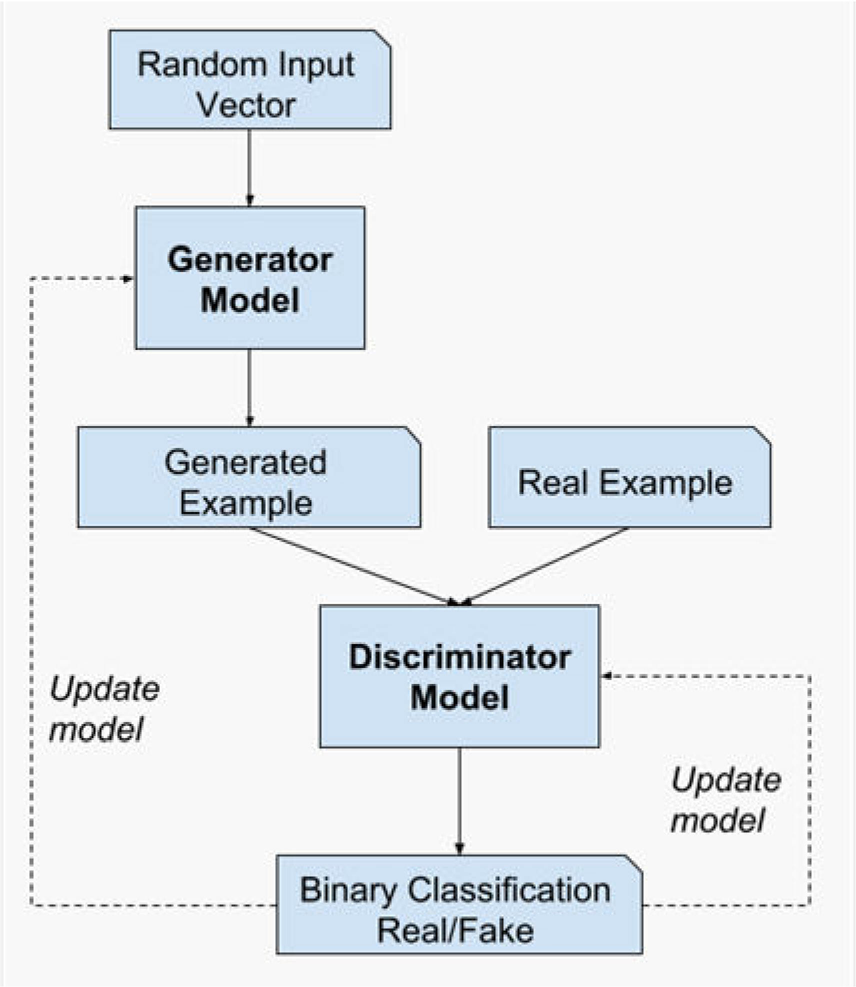


Fig. 1: **Patchwise Contrastive Learning for one-sided translation.** A generated **output patch** should appear closer to its **corresponding input patch**, in comparison to other **random patches**. We use a multilayer, patchwise contrastive loss, which maximizes *mutual information* between corresponding input and output patches. This enables one-sided translation in the unpaired setting.

Generative adversarial network

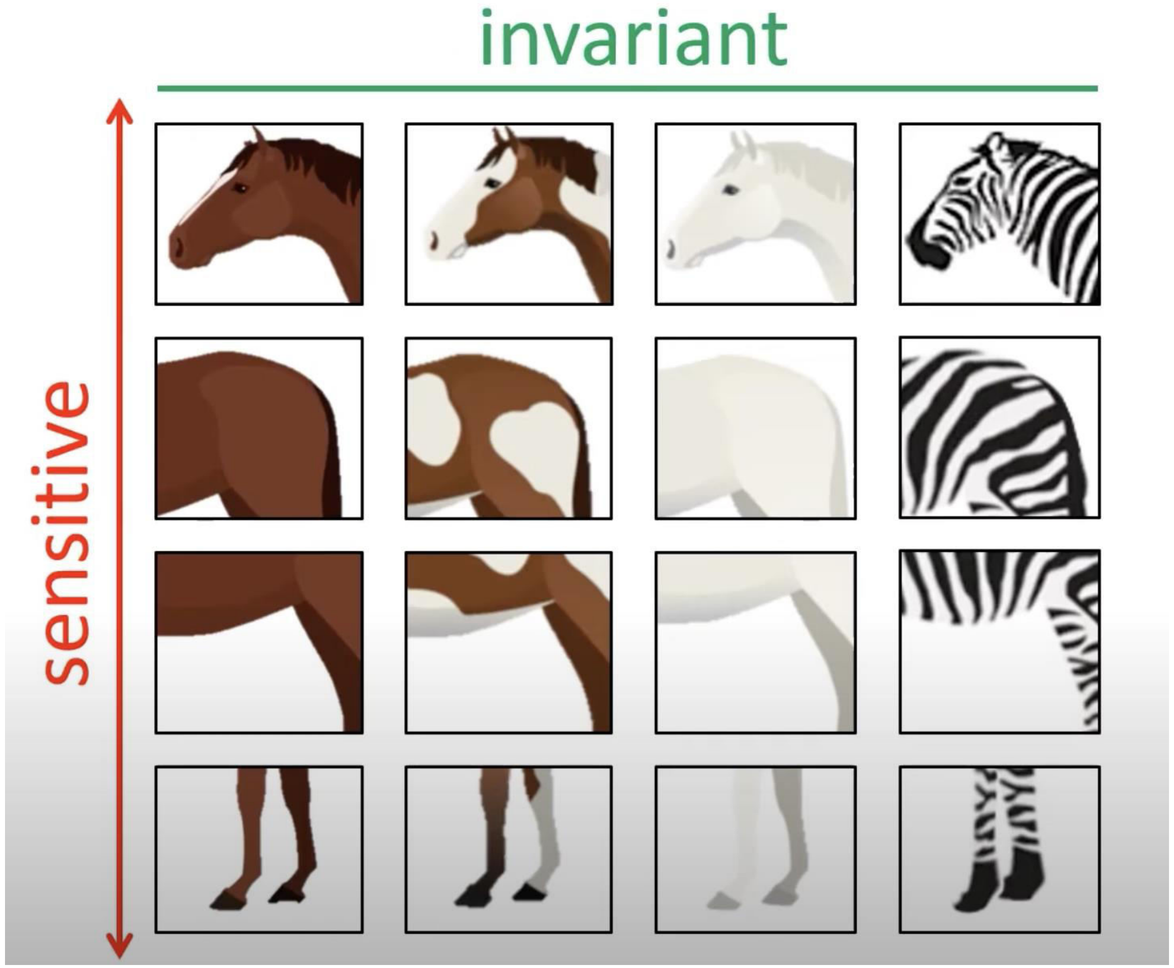


the problem is treated as supervised learning with two sub-models:

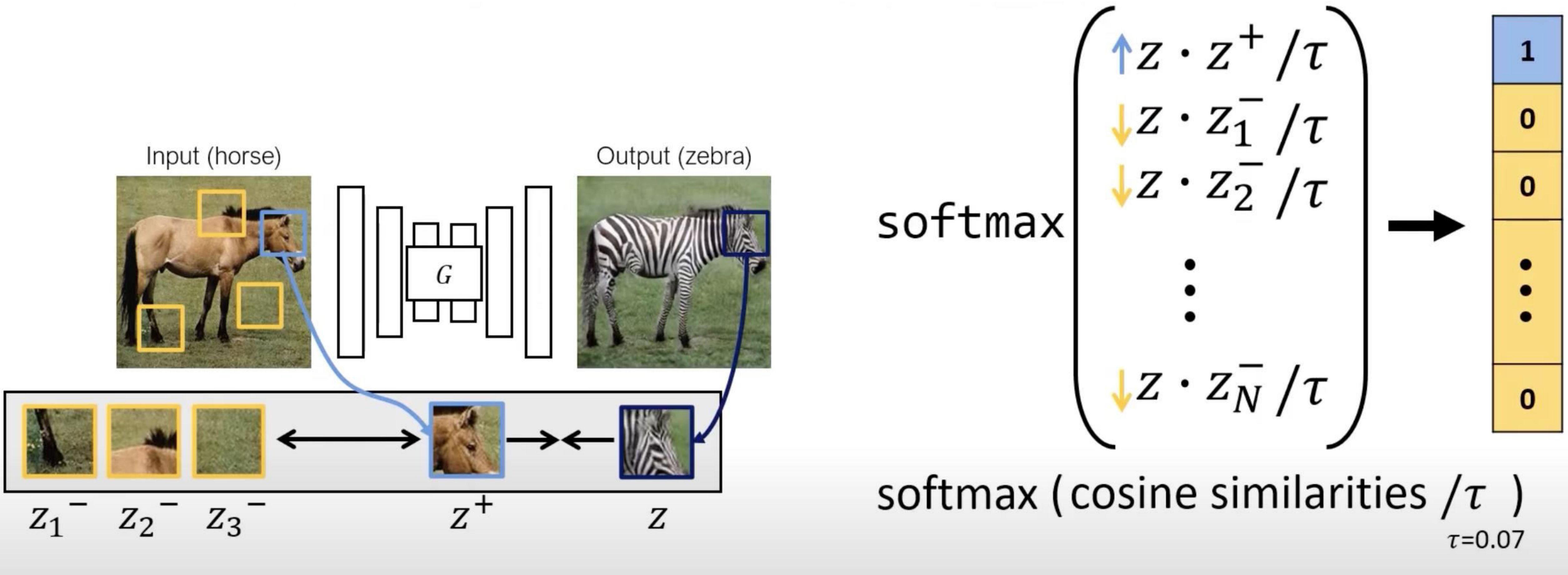
the **generator** model that we train to generate new examples,
and
the **discriminator** model that tries to classify examples
as either real (from your dataset) or fake (generated).

the generator and discriminator are trained interconnected using
gradient descent techniques

Patch-based contrastive loss

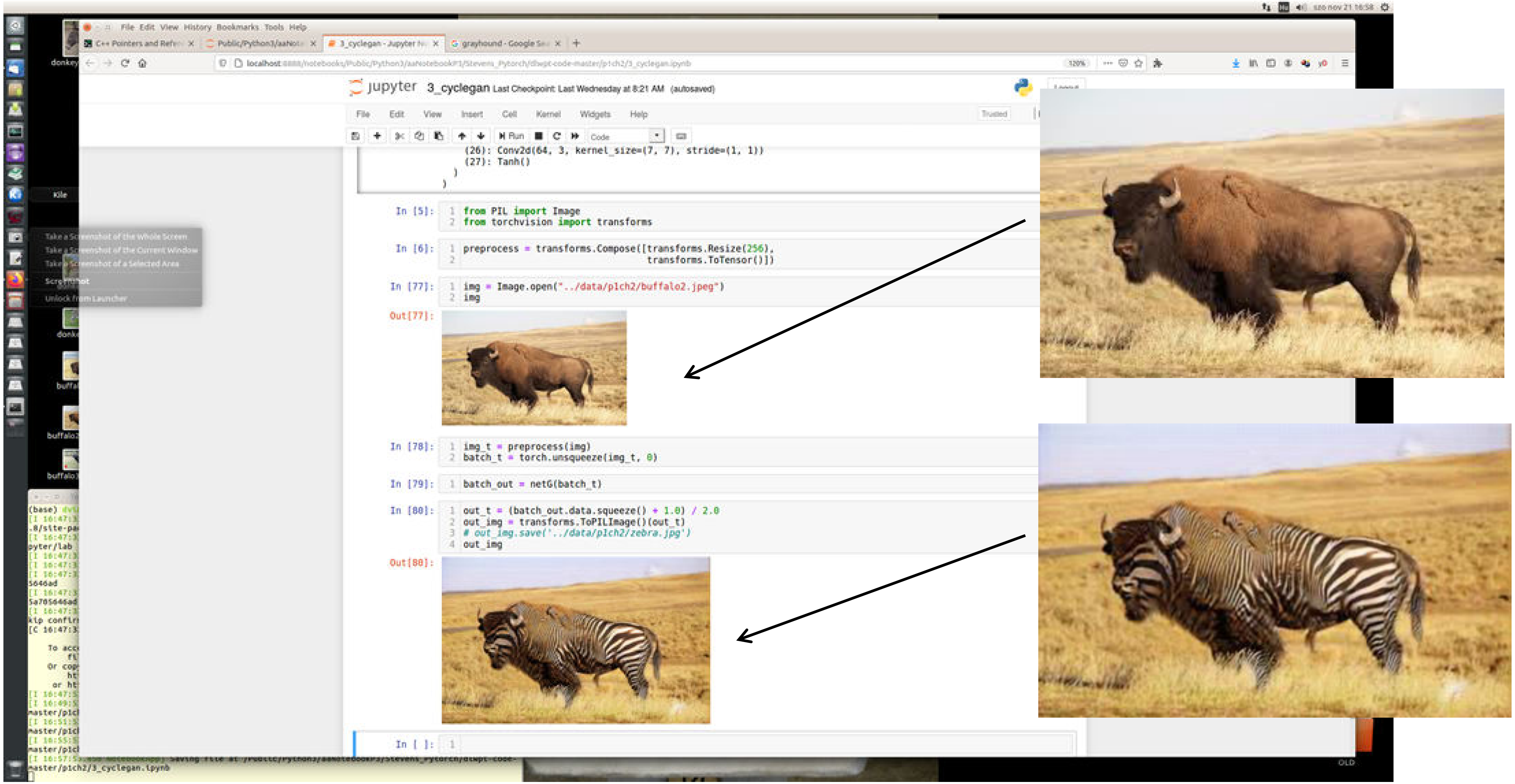


Patch-based contrastive loss

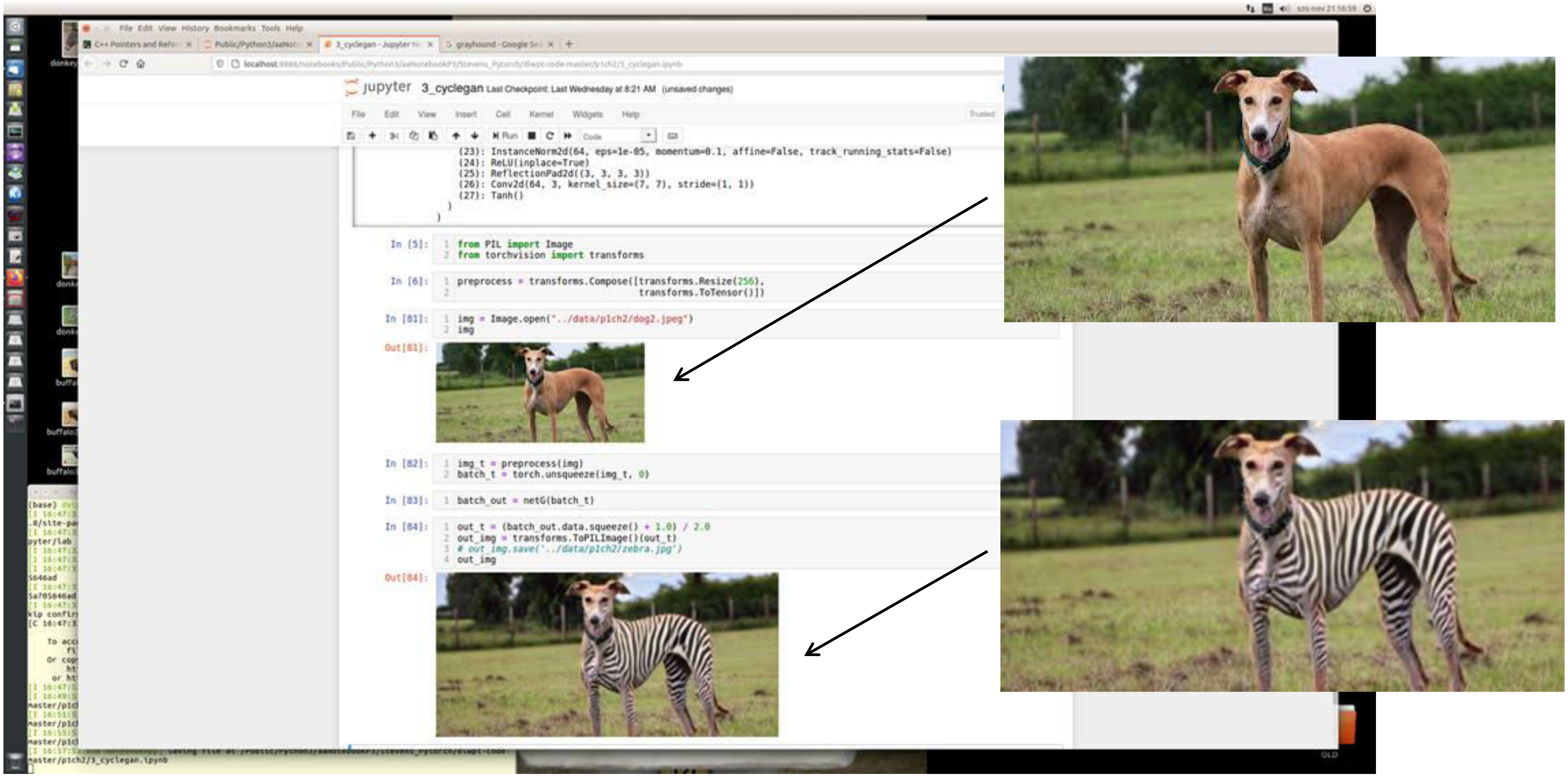


Corresponding patches should have high similarity!

Sample images of own runnings



Sample images of own runnings



Sample images of own runnings