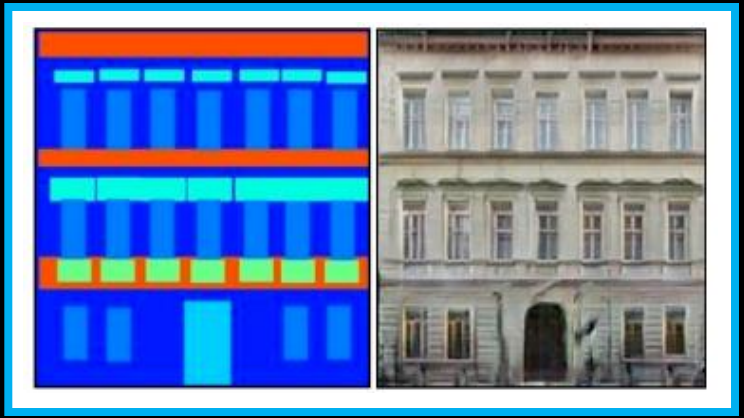
AI for the Media

Week 9, Domain 2 Domain

~ Vít Růžička

Motivation for today

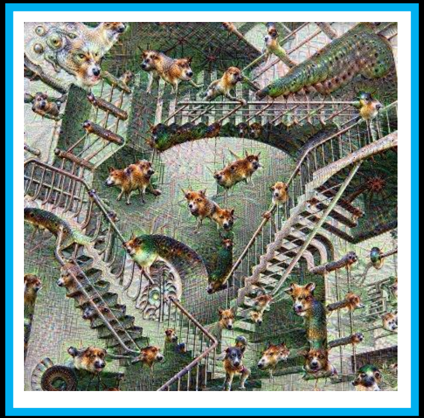
Overview of additional machine learning techniques:



pix2pix

& co

style transfer



deep dream

& adversarial attacks

Overview

Domain 2 Domain Models (*pre-recorded lecture*):

- **Pix2pix** and domain to domain transfer
- **Style transfer** technique
- **Deep Dream** technique

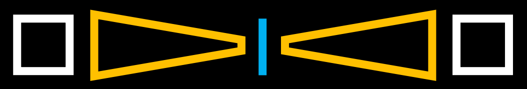
Practical session (*during the live session*):

- **Code**: Training a pix2pix model

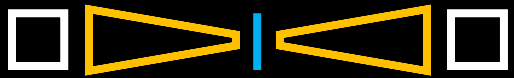
Note: Today we will at a lot of models and methods, don’t worry about understanding all of them, but focus on the concept of what is being learned by the models and how (by the special connectivity or by the special tricks in writing up the loss functions … etc.)

Domain to Domain

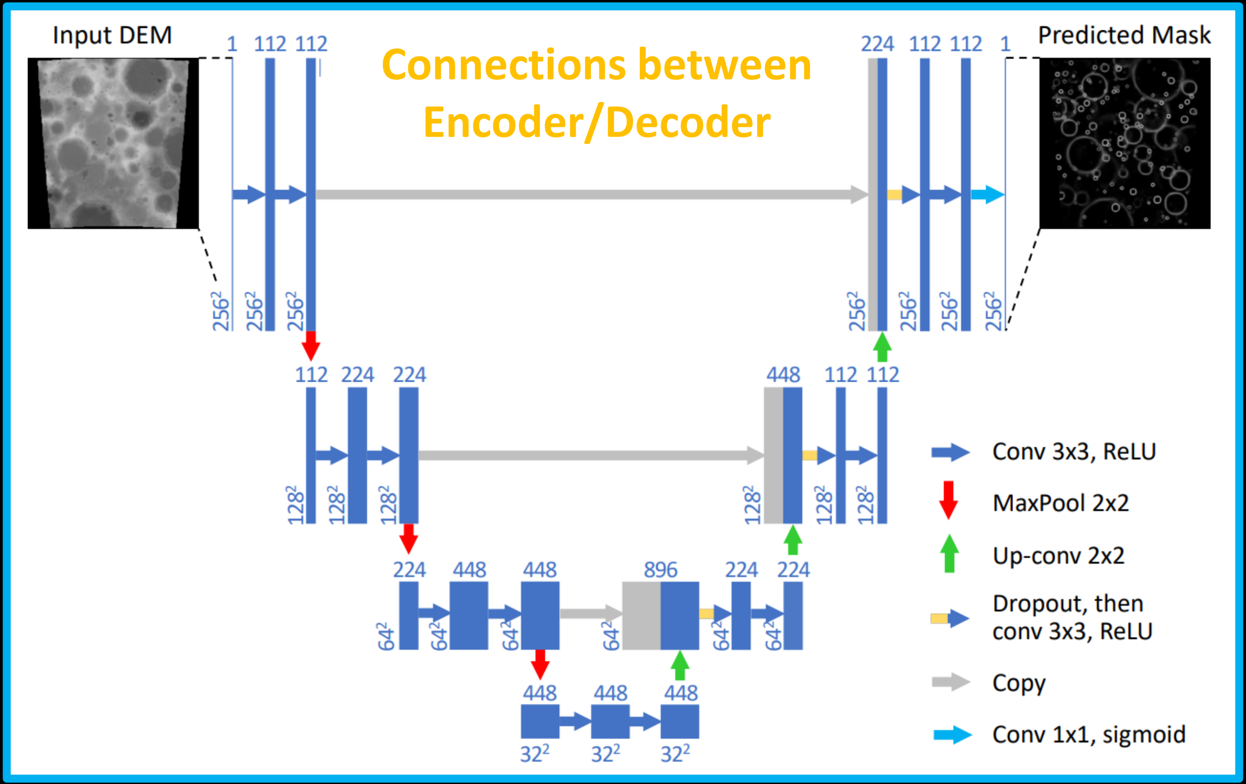
- We have seen few models working with images on outputs and inputs (for example AutoEncoders). Similar architectures can be also used to model **relation between two domains** of data.



Domain to Domain

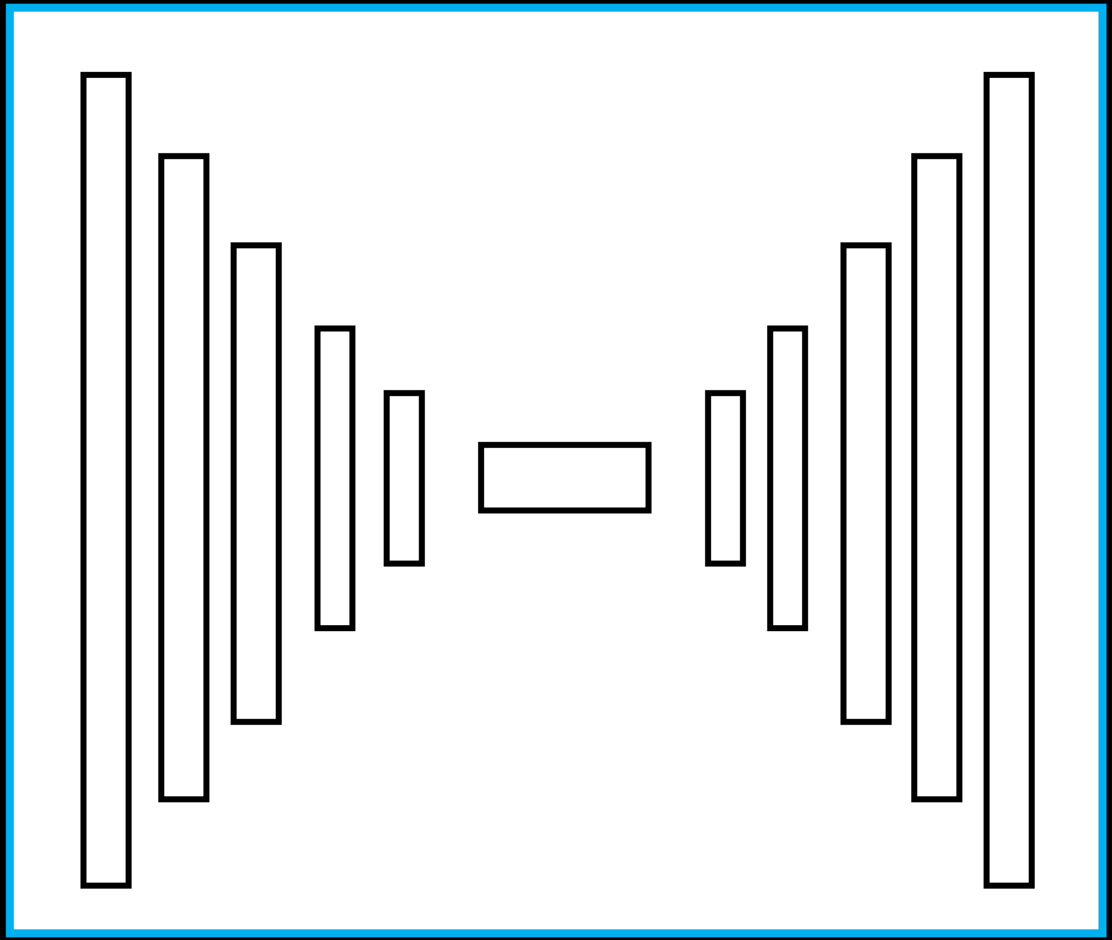
- We have seen few models working with images on outputs and inputs (for example AutoEncoders). Similar architectures can be also used to model **relation between two domains** of data. 

- Let's look at a predecessor of this idea, the **U-Net model**:
  - Paper with U-Net applied on the task of lunar crater identification

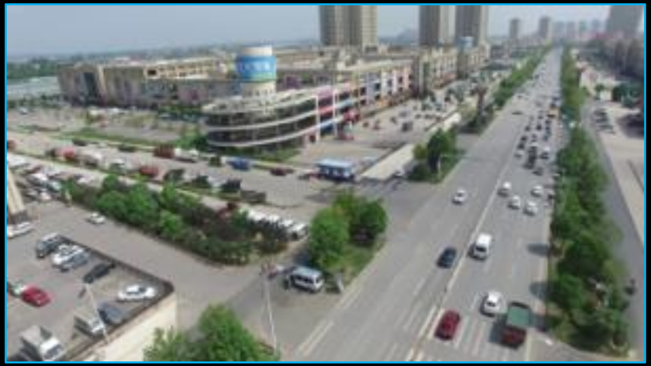


(PS: U-Net is from 2015, this paper from 2018)

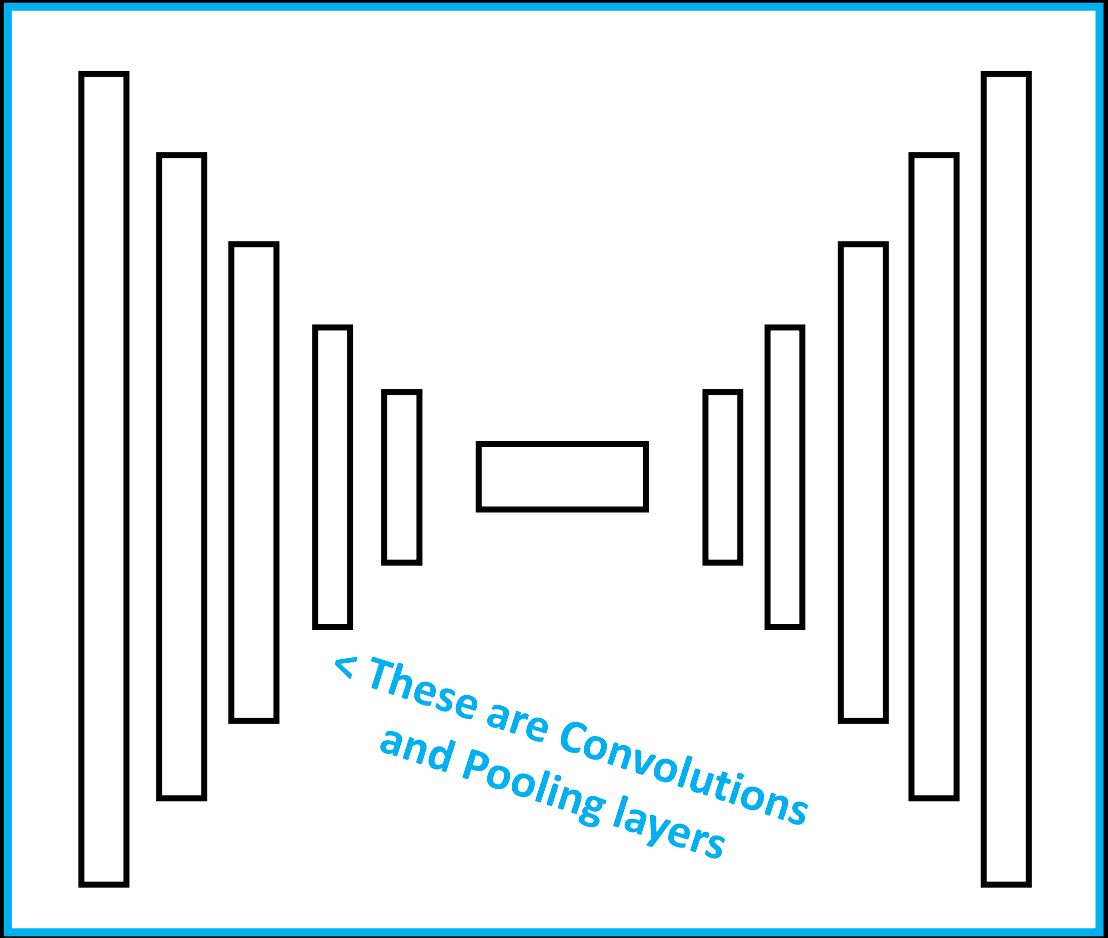
U-Net

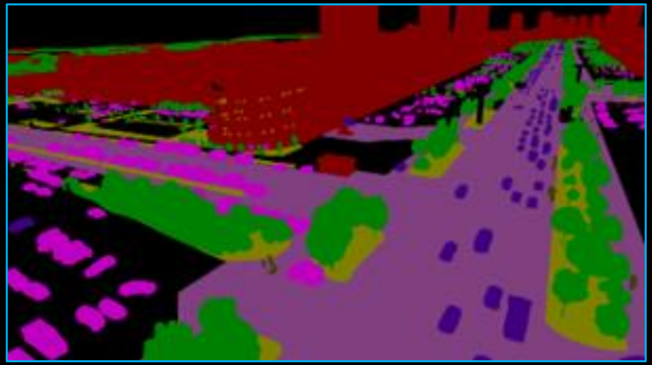


U-Net



Images from one "domain"

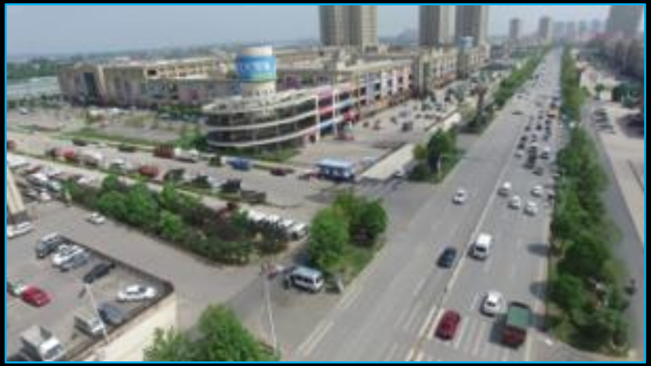




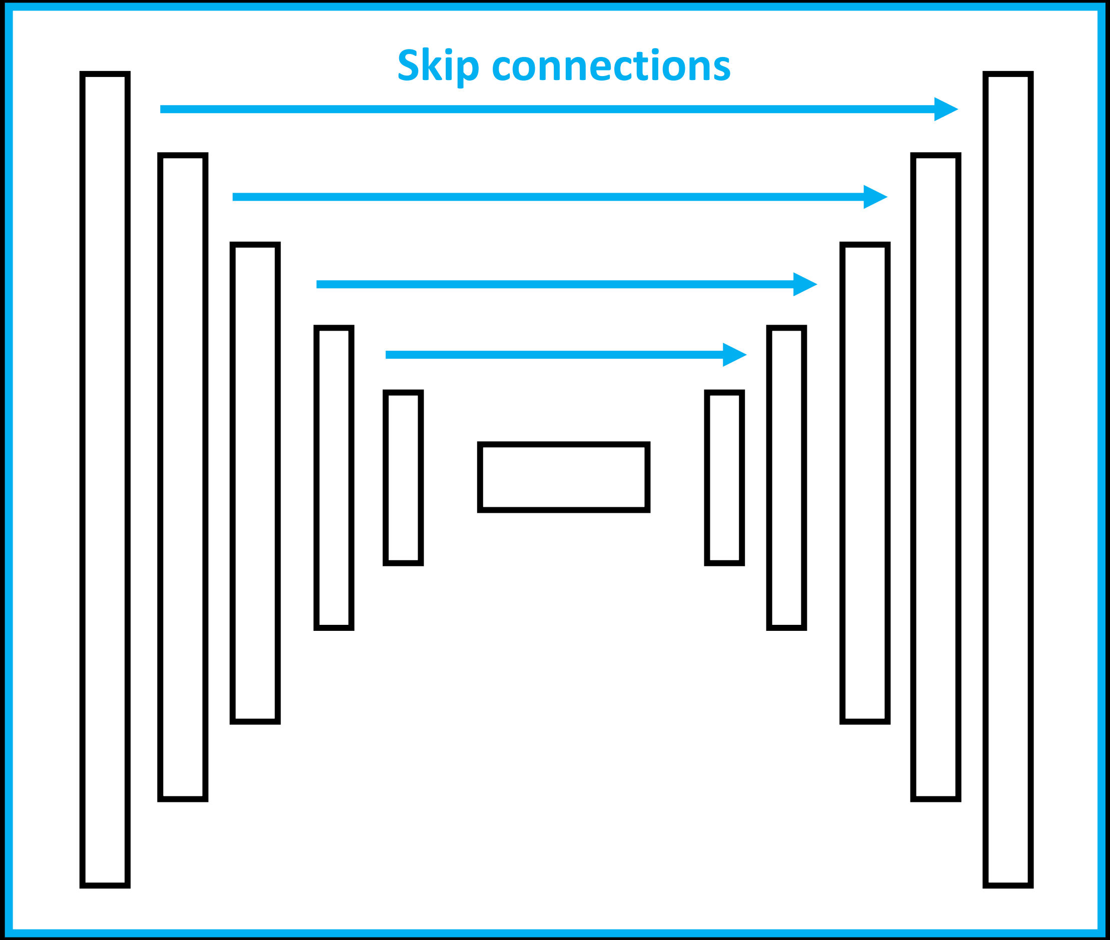
Paired images from another "domain"

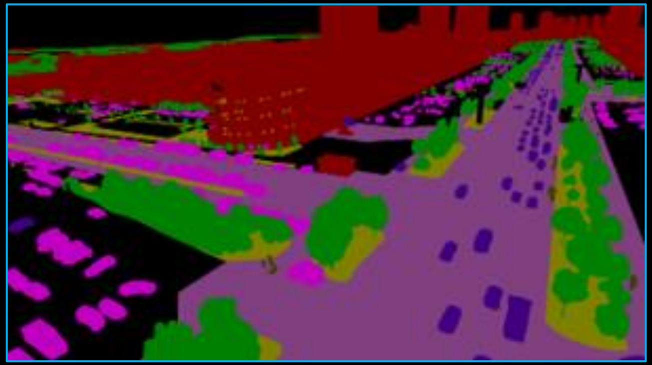
- Connects one image to another while going through a bottleneck

U-Net



Images from one "domain"

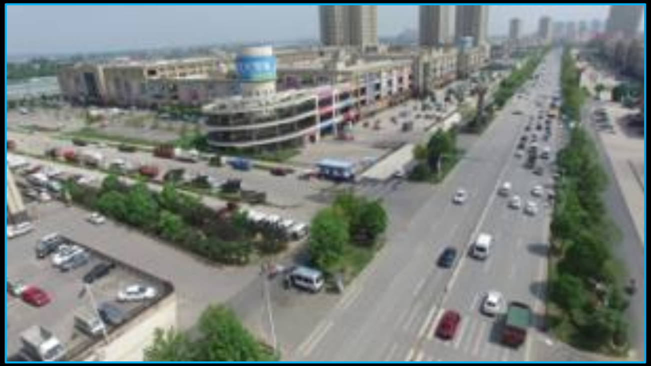




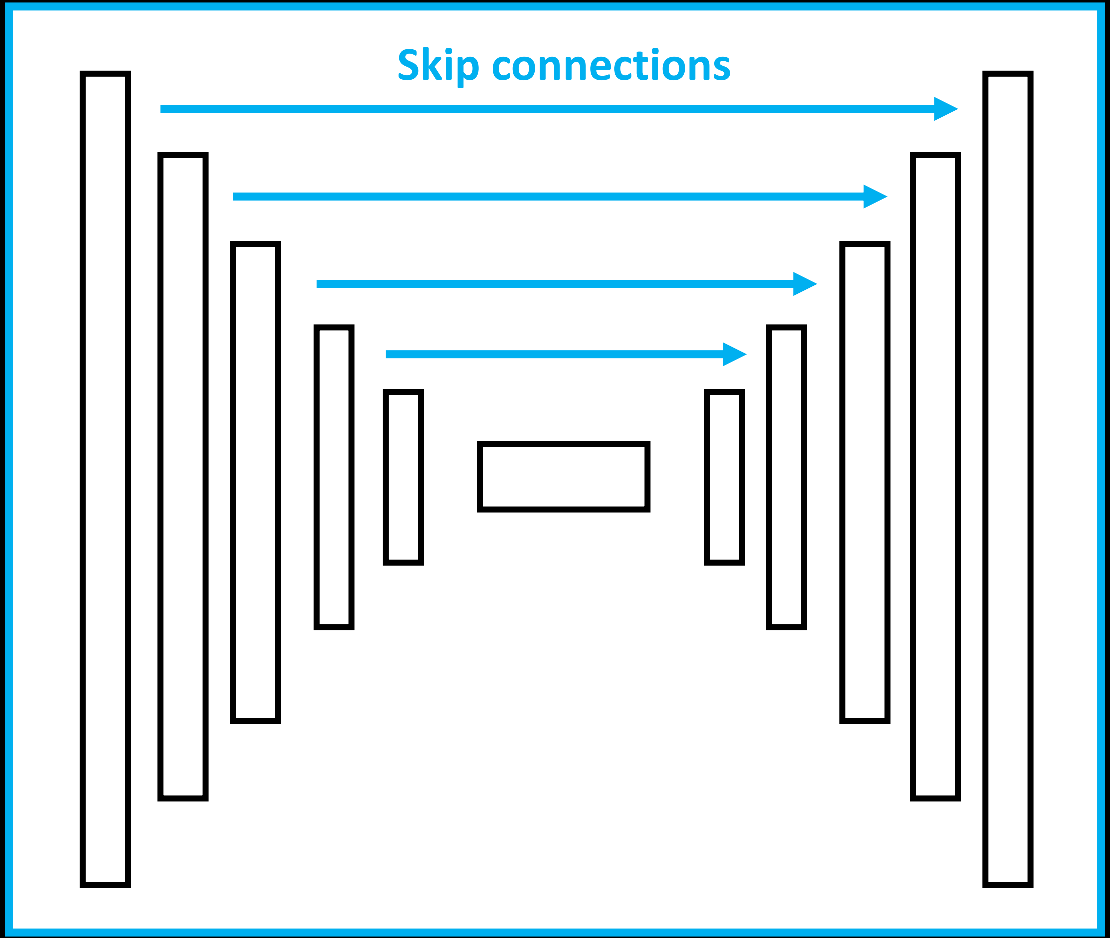
Paired images from another "domain"

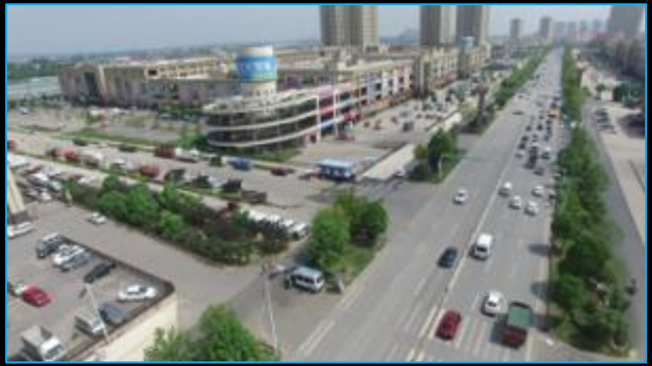
- Additionally: "*skip connections*" that allow the information to flow

U-Net



Images from one "domain"

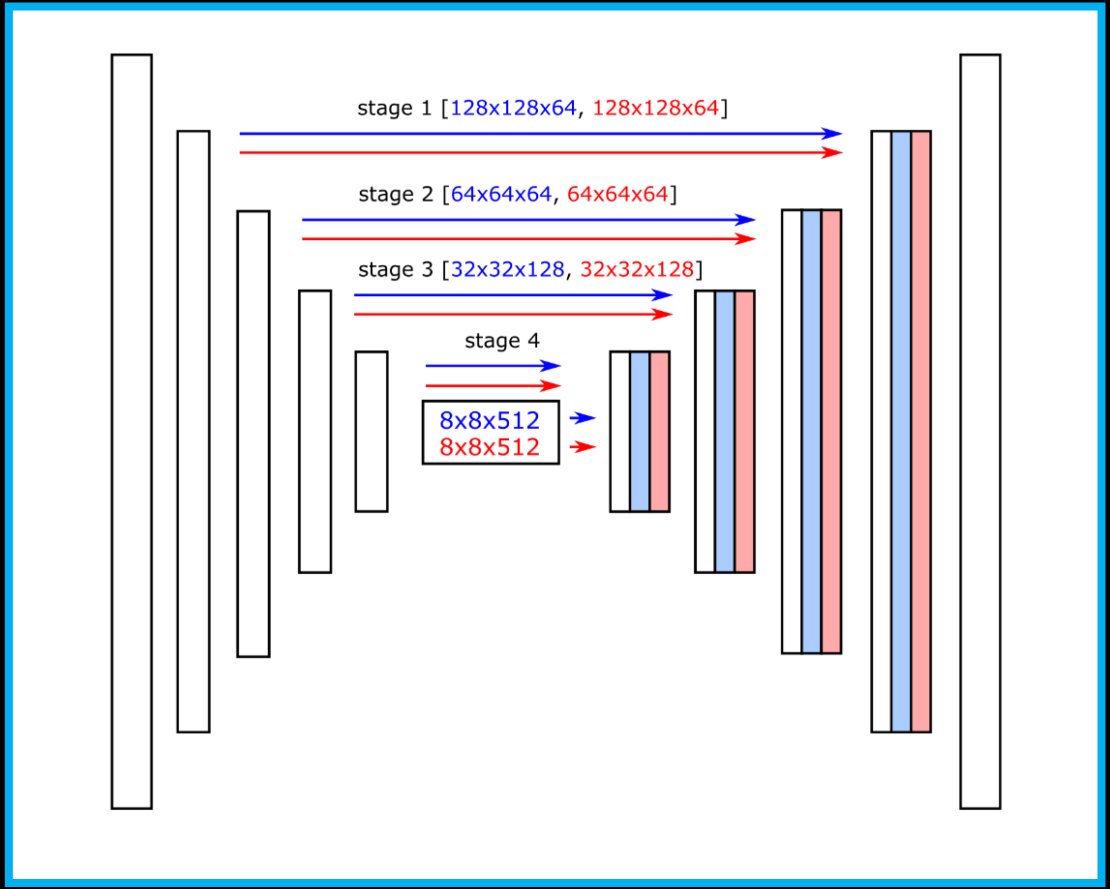




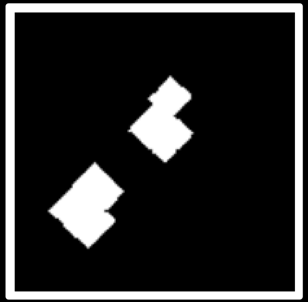
(if this was an autoencoder)

- PS: As an autoencoder, this model could very easily cheat …

U-Net in a real-world scenario



< Slightly more experimental setup – check Siamese U-Nets*

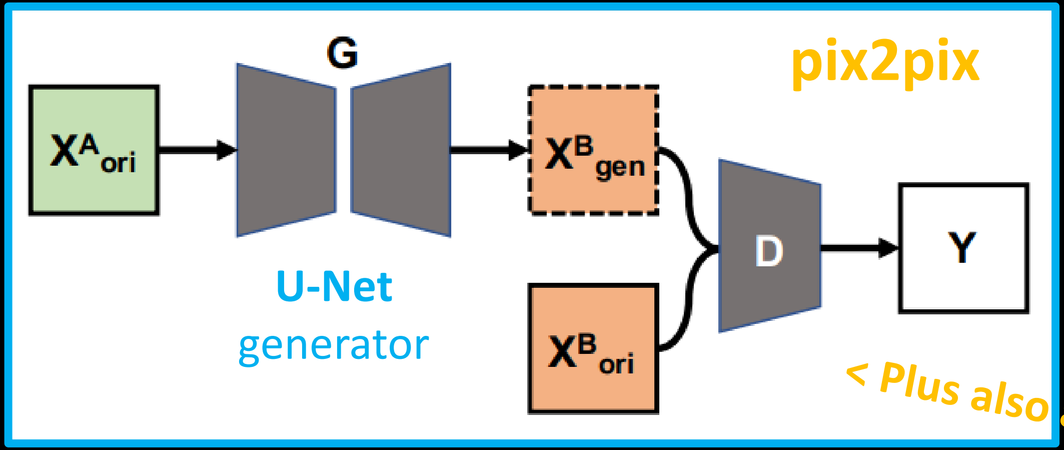


- Can be adapted to many tasks – for example this "change detection", where it needed to learn *which kind of change* we care about

* here: https://arxiv.org/abs/2008.11201

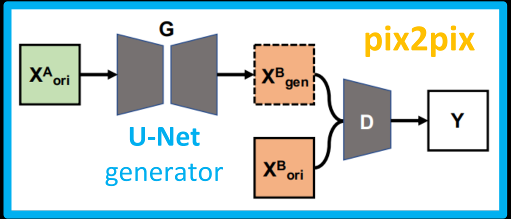
Pix2Pix

- General purpose **image-to-image translation:**
  - From their paper: *"Many problems in image processing, computer graphics, and computer vision can be posed as "translating" an input image into a corresponding output image."*

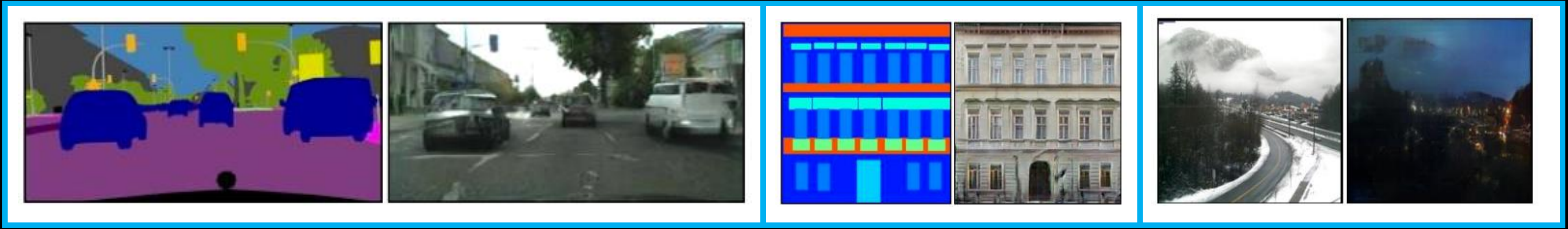


< Plus also adding adversarial training

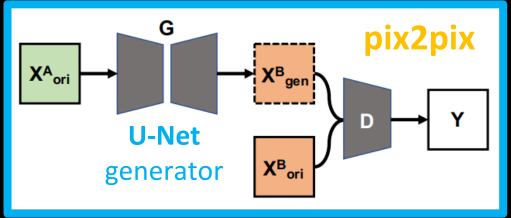
Pix2Pix



- General purpose **image-to-image translation:**
  - From their paper: *"Many problems in image processing, computer graphics, and computer vision can be posed as "translating" an input image into a corresponding output image."*
  - Translating without specifically defining the rules – data-driven translating between two domains by showing **paired** examples:
    - [Image from A, Corresponding Image from B] * N samples



Pix2Pix



- General purpose **image-to-image translation:**

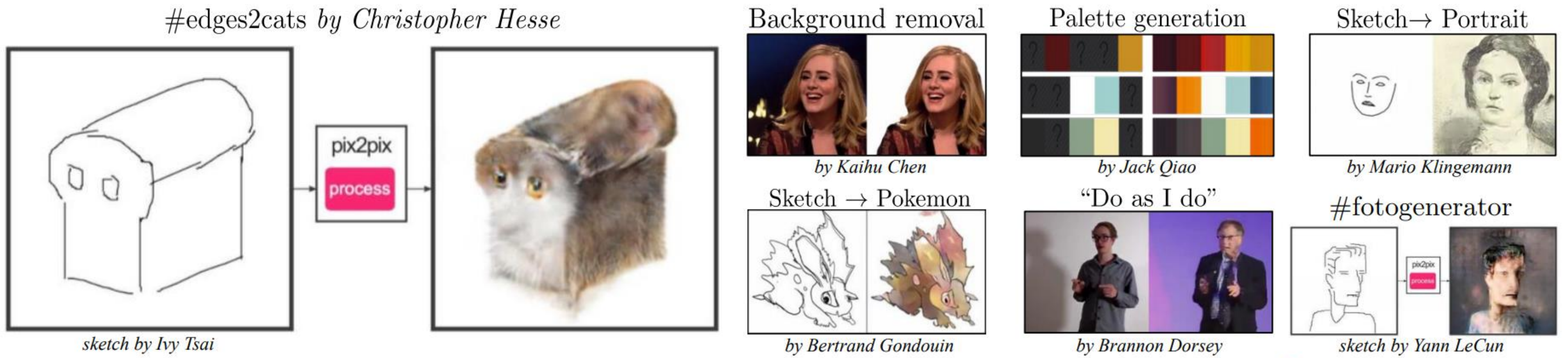
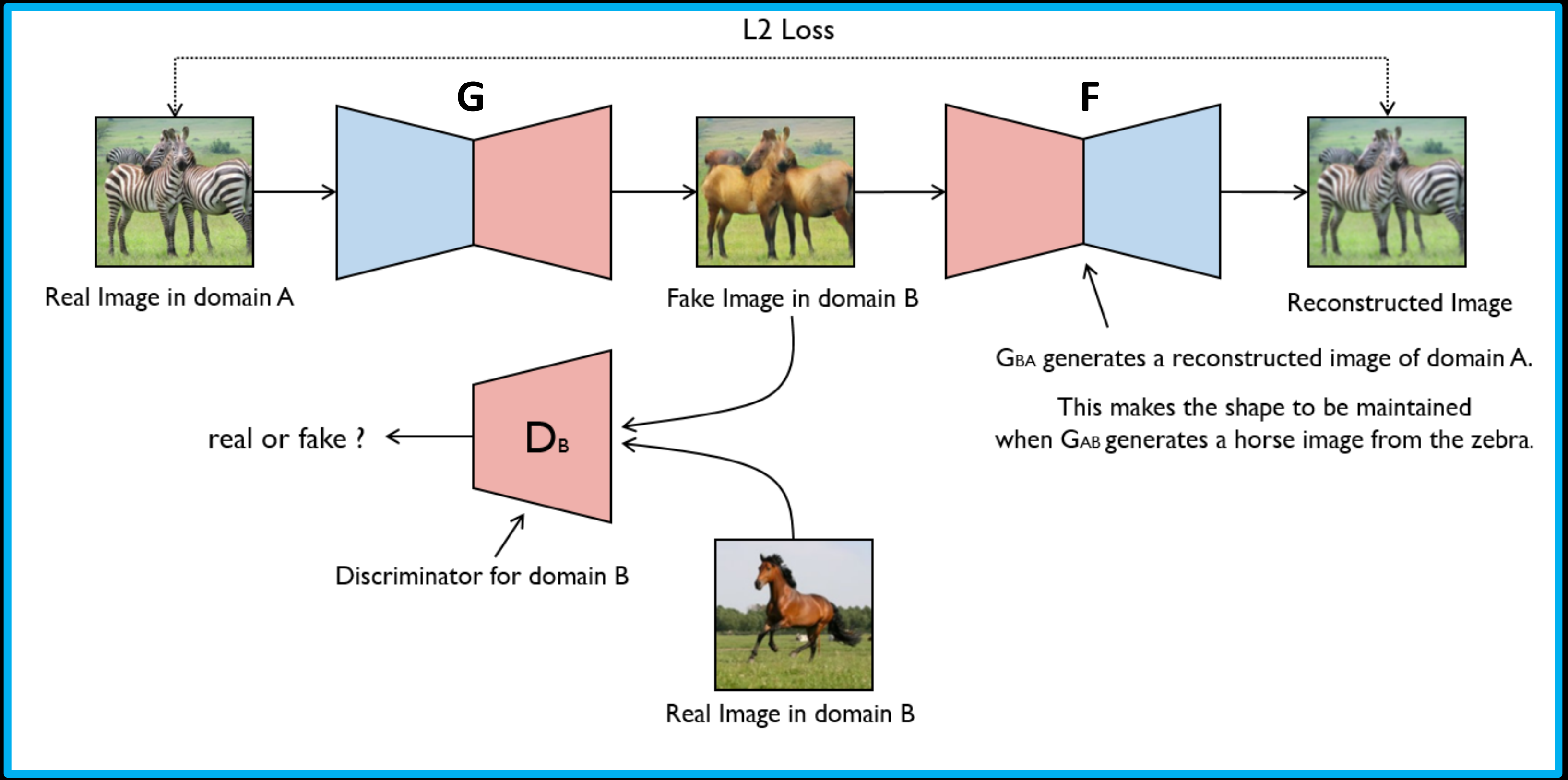


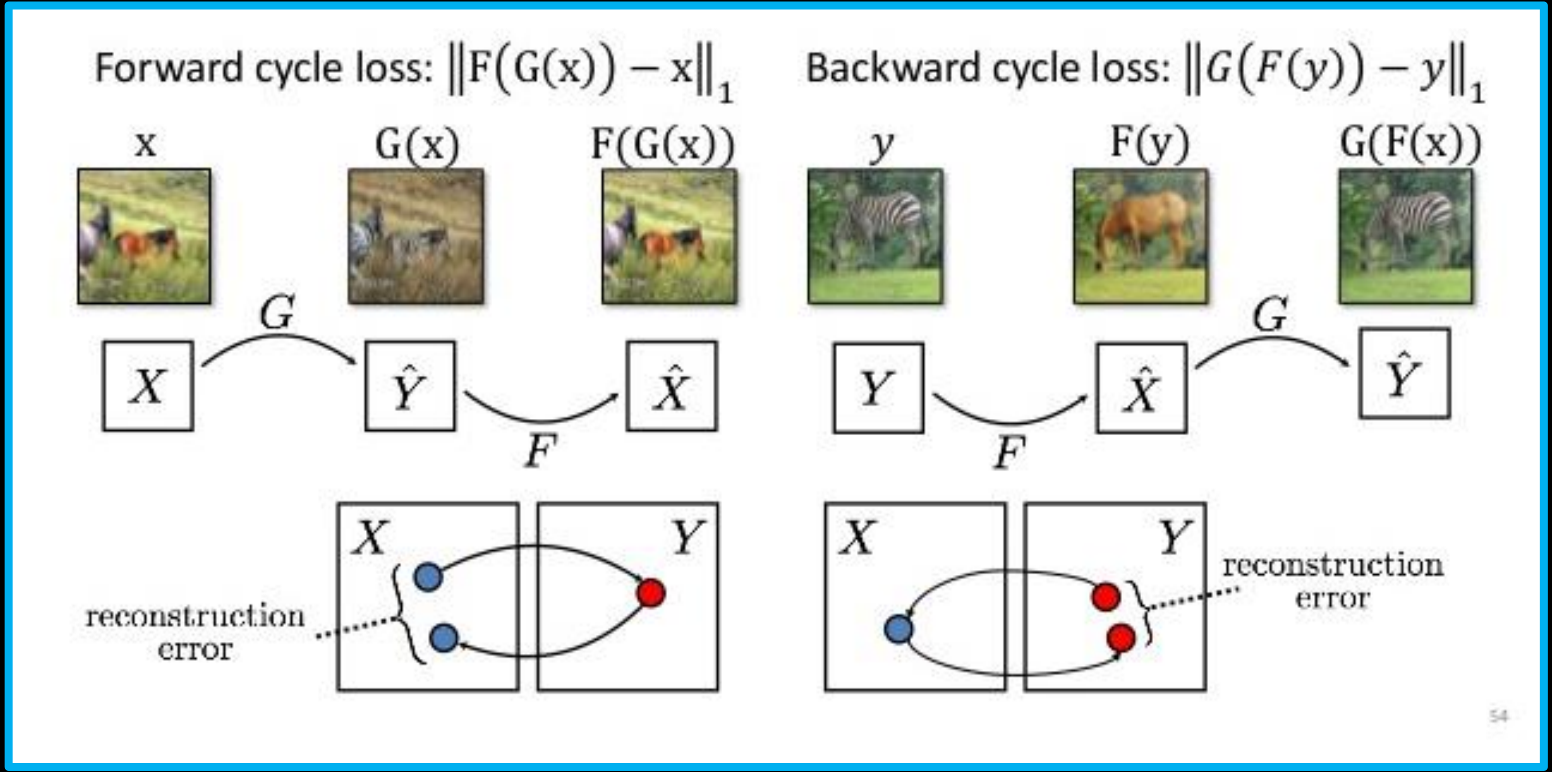
Figure 11: Example applications developed by online community based on our `pix2pix` codebase: *#edges2cats* [3] by Christopher Hesse, *Background removal* [6] by Kaihu Chen, *Palette generation* [5] by Jack Qiao, *Sketch → Portrait* [7] by Mario Klingemann, *Sketch→ Pokemon* [1] by Bertrand Gondouin, *"Do As I Do" pose transfer* [2] by Brannon Dorsey, and *#fotogenerator* by Bosman et al. [4].

CycleGAN, connectivity



- Setup with two generators going between two domains (**G** and **F**) and a special cyclic loss function. This model **doesn't require paired data** in the training datasets.

CycleGAN, loss functions



Check this video

- And a **cyclic loss function** which looks at two types of errors ^

CycleGAN examples

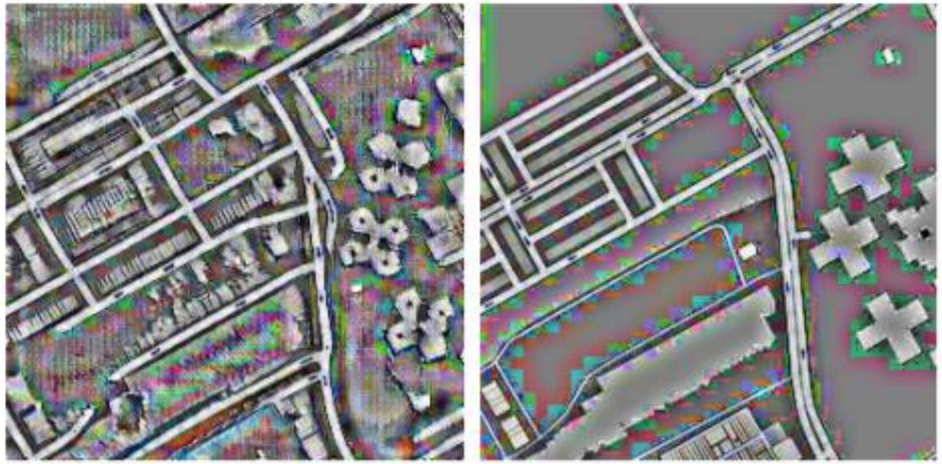
Monet 2 Real, and more examples at junyanz.github.io/CycleGAN/

Night 2 Day, followed by more tasks

Night-to-Day Image Translation for Retrieval-based Localization (arxiv.org/abs/1809.09767)

(a) Aerial photograph: x. (b) Generated map: Fx. (c) Aerial reconstruction: GFx.

Figure 1: Details in x are reconstructed in GFx, despite not appearing in the intermediate map Fx.



(a) Generated map. (b) Training map, for comparison.

Figure 2: Maps with details amplified by adaptive histogram equalization. Information is present in the generated map even in regions that appear empty to the naked eye.

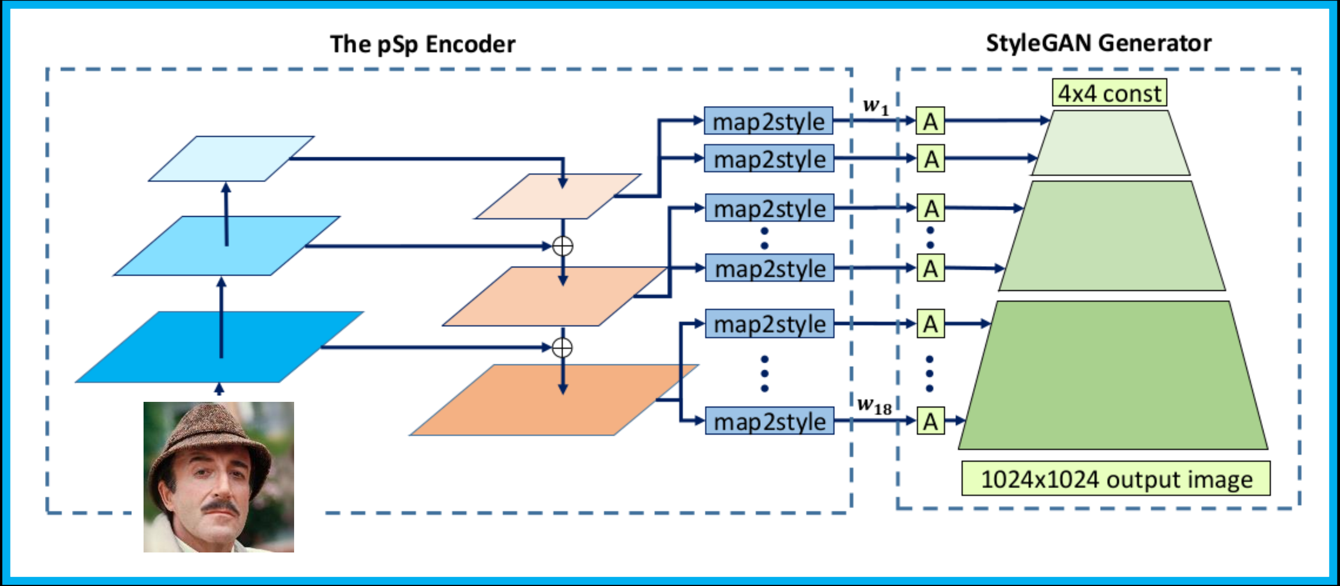
CycleGAN steganography

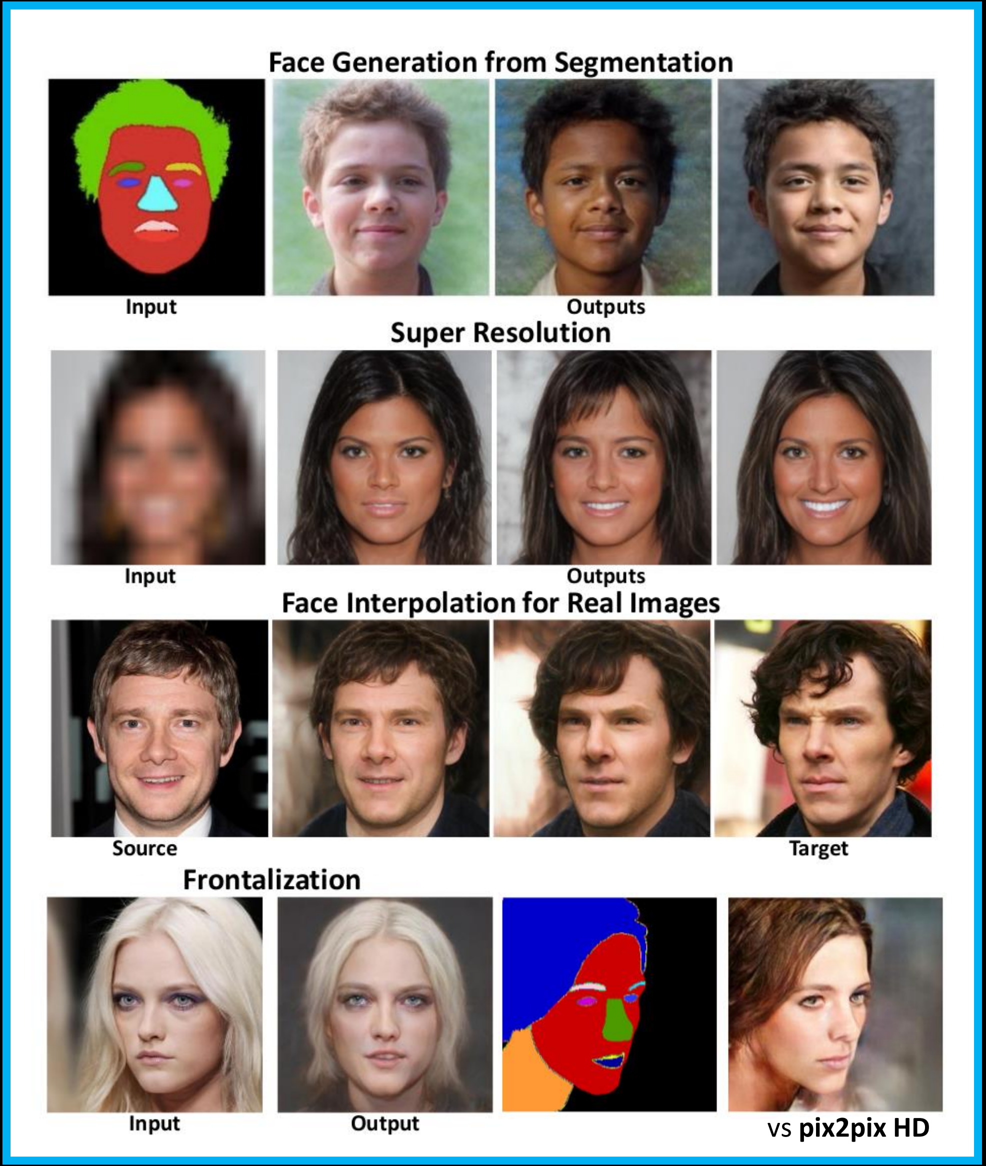
- Beware of what we are actually learning with the two generators …
  - **Steganography** = encoding secret information into an image (*such as the recipe of what to reconstruct*)

CycleGAN, a Master of Steganography - arxiv.org/abs/1712.02950

*Check repo & video

Follow-up works





- **Pixel2Style2Pixel**, new research (2020) which uses a frozen pre-trained StyleGAN2 generator and trains an encoder for many specialized tasks while getting much higher quality than a pix2pix would

***)** PS: now it’s a good time to take a pause

Style Transfer

- Previously we saw, how a Deep Convolutional network separates where it saves *high-level* and *low-level* representations (this is due to the *Conv->Pool* combo – each convolutional layer serves as an image-filter).

Style Transfer

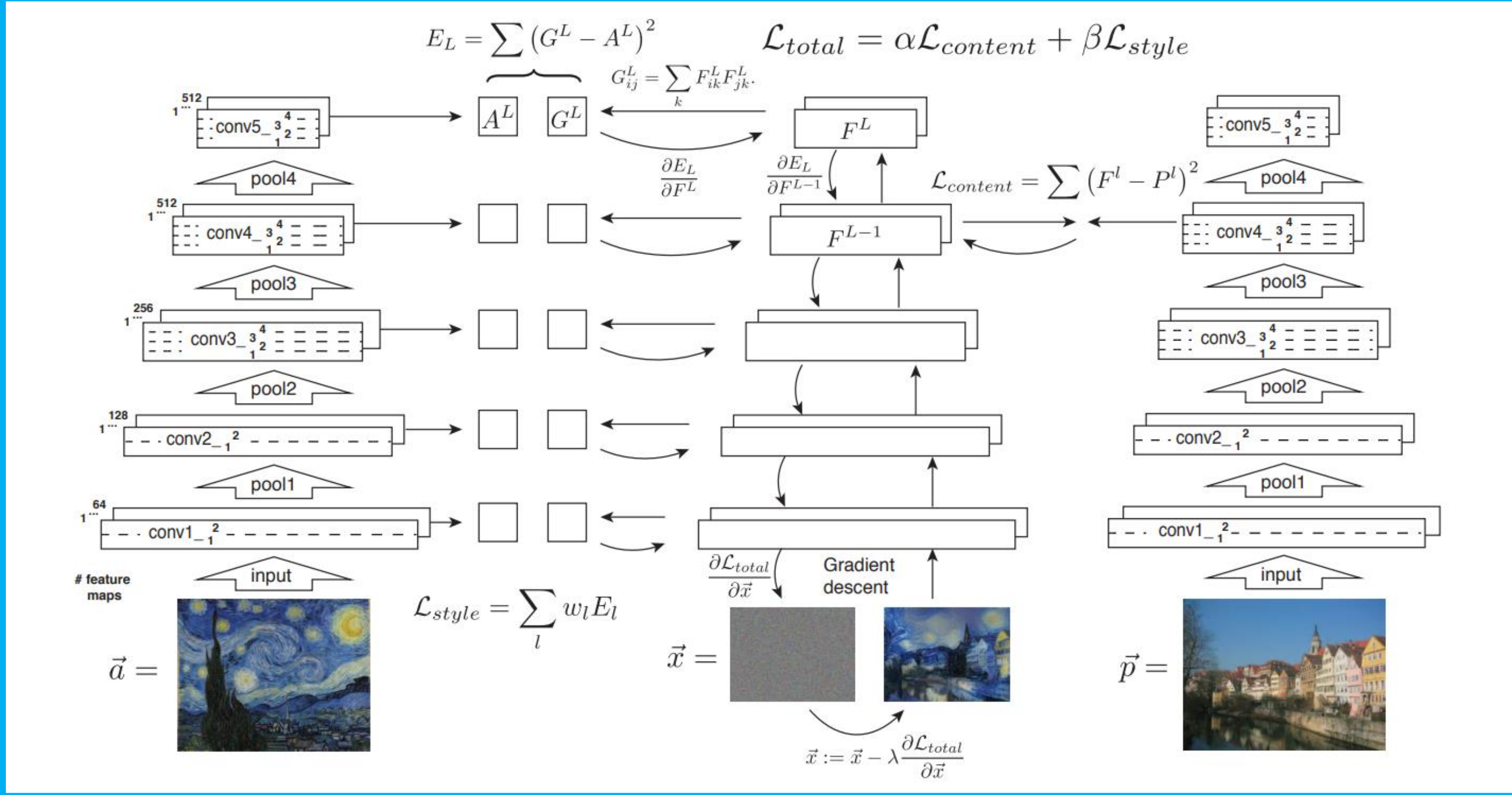
- Previously we saw, how a Deep Convolutional network separates where it saves *high-level* and *low-level* representations (this is due to the *Conv->Pool* combo – each convolutional layer serves as an image-filter).
  - **High-level information** – contents of the whole image – **image content**
    - Encode one image extracting its content information, the feature responses in deeper layers of the network

Style Transfer

- Previously we saw, how a Deep Convolutional network separates where it saves *high-level* and *low-level* representations (this is due to the *Conv->Pool* combo – each convolutional layer serves as an image-filter).
  - **High-level information** – contents of the whole image – **image content**
    - Encode one image extracting its content information, the feature responses in deeper layers of the network
  - **Low-level information** – details used inside the image, texture – **image style**
    - Encode another image extracting the style information, feature response alongside a selection of layers

(PS: One possible similarity if the content of low and high frequencies in music converted to spectrograms)

Style Transfer



Style Transfer

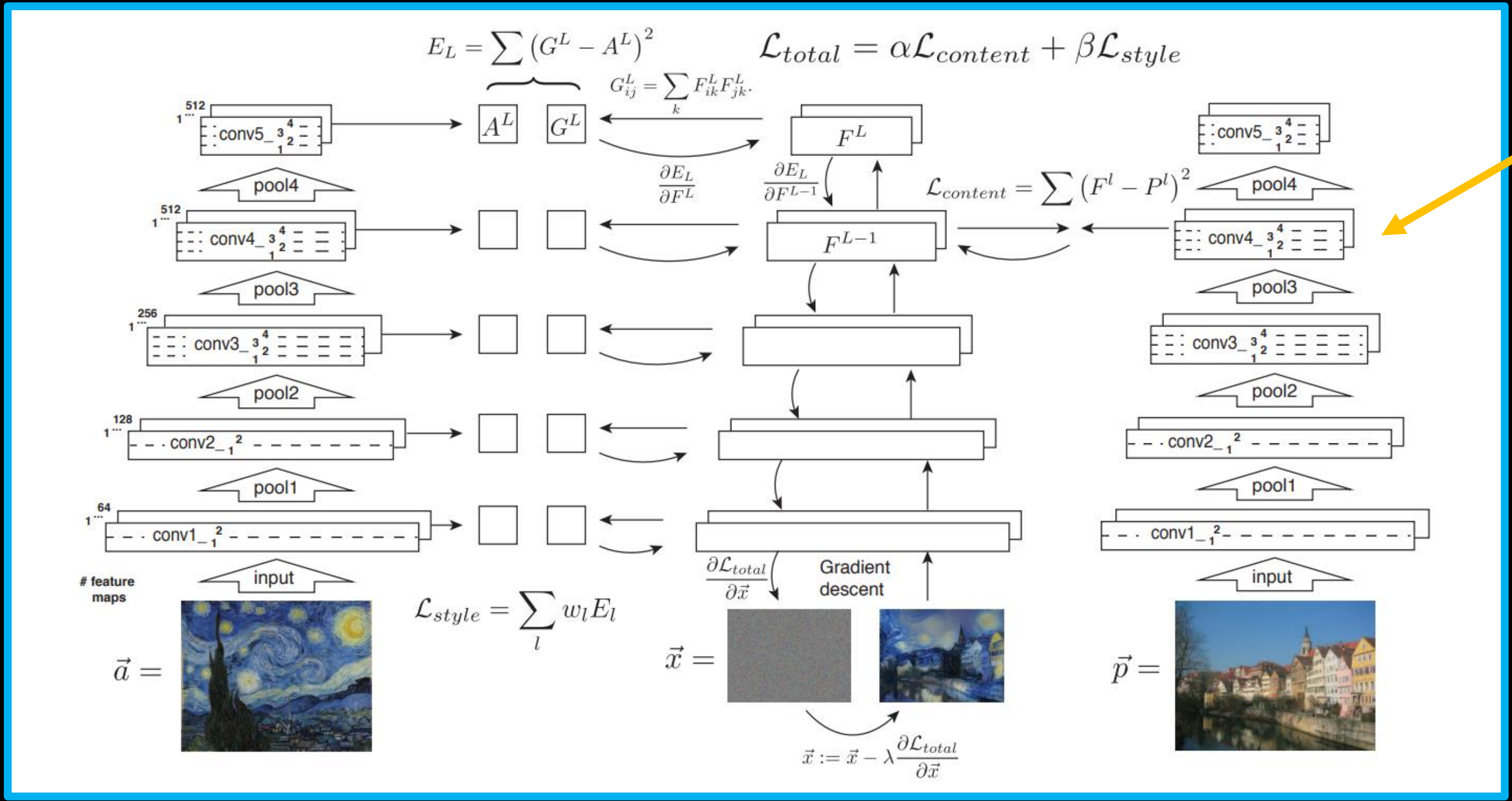


Image content = the feature responses in deeper layers of the VGG network

Style Transfer

Image style =
feature response alongside a selection of layers

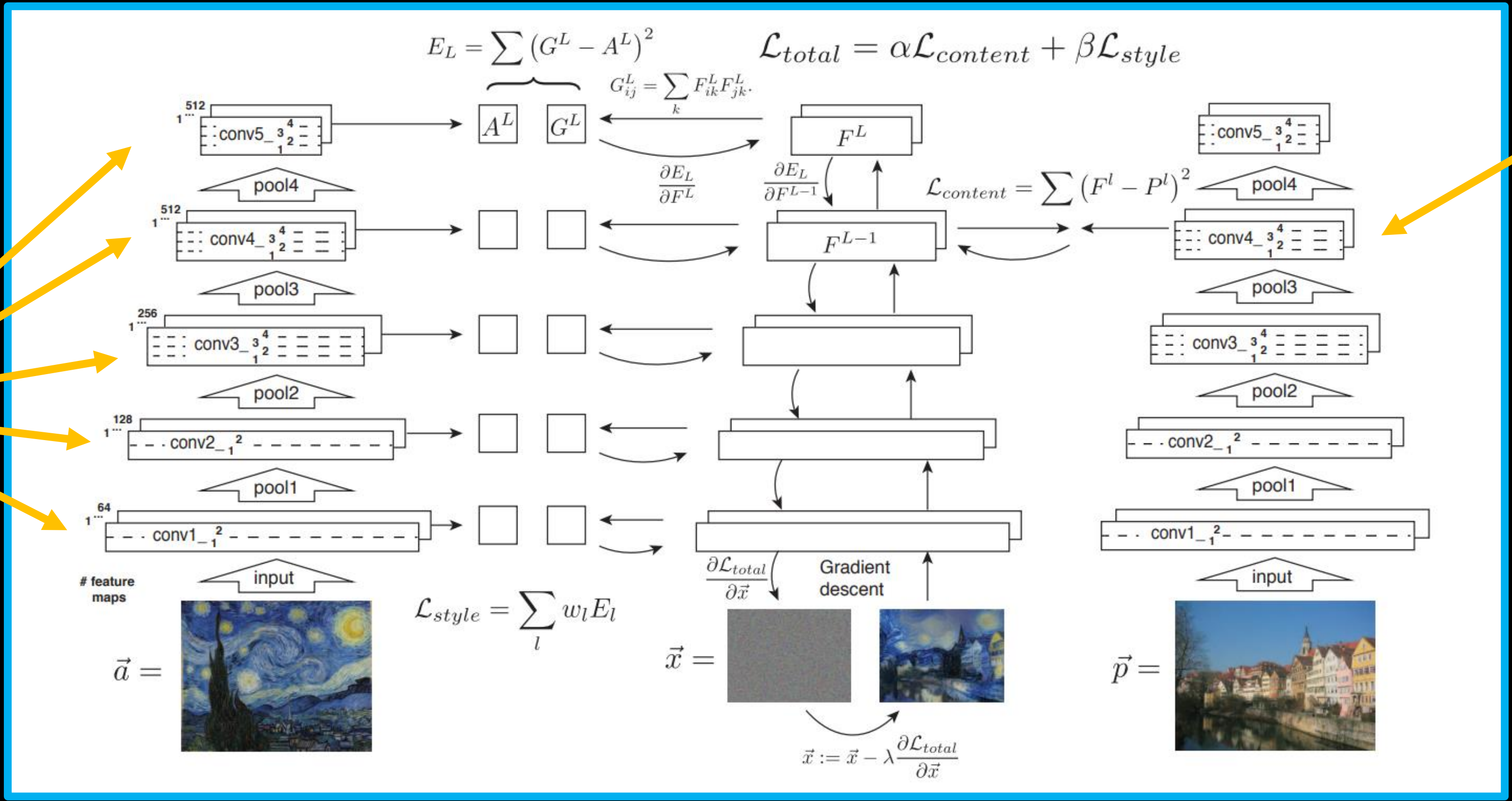


Image content =
the feature responses in deeper layers of the VGG network

Style Transfer

Image style = feature response alongside a selection of layers

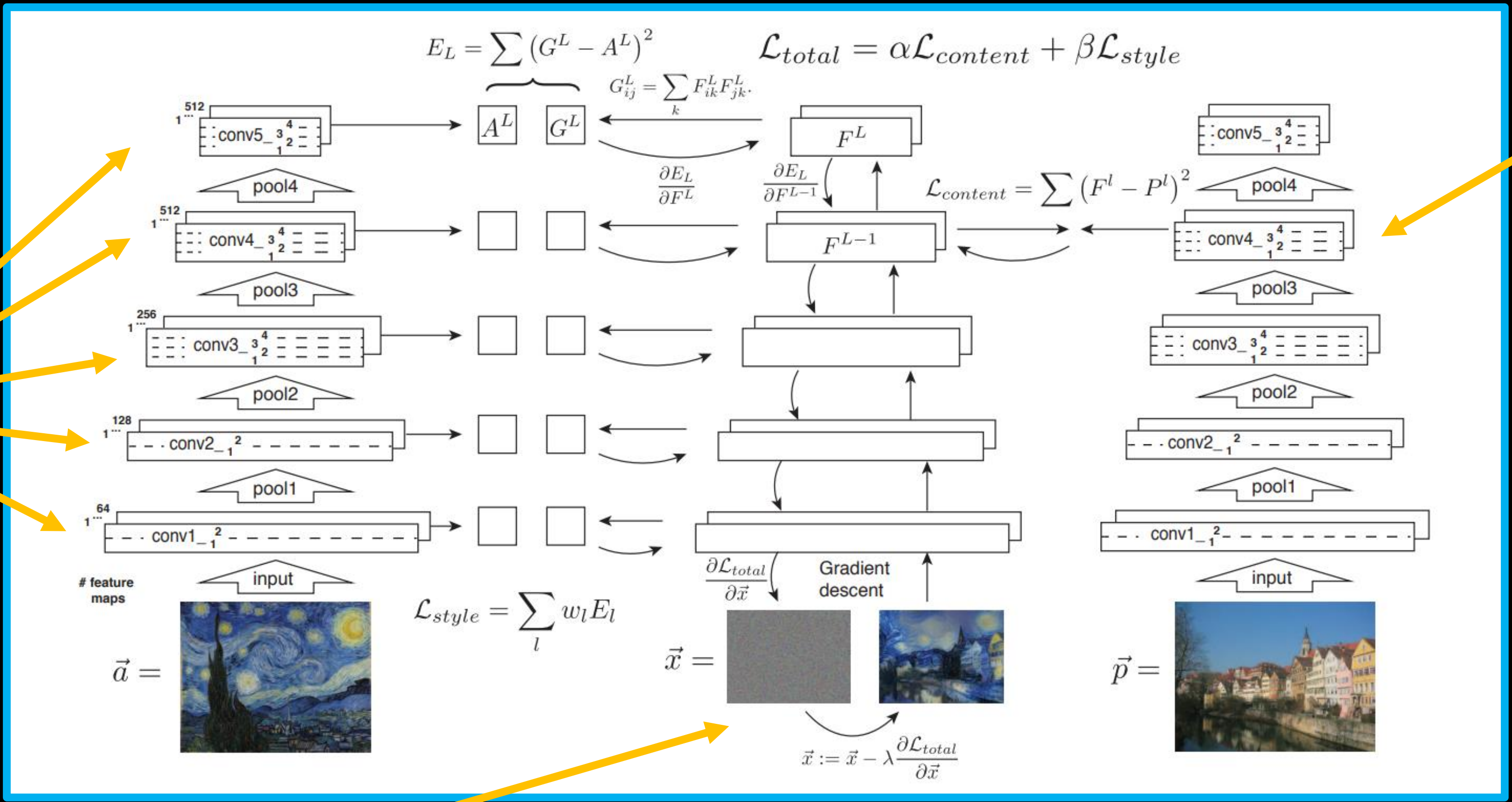


Image content = the feature responses in deeper layers of the VGG network

Optimizing random noise image to have the same responses as those features we saved.

- Iteratively we will create a new image which has the style responses similar to our encoded style features + content responses similar to our encoded content features

Style Transfer

Image style = feature response alongside a selection of layers

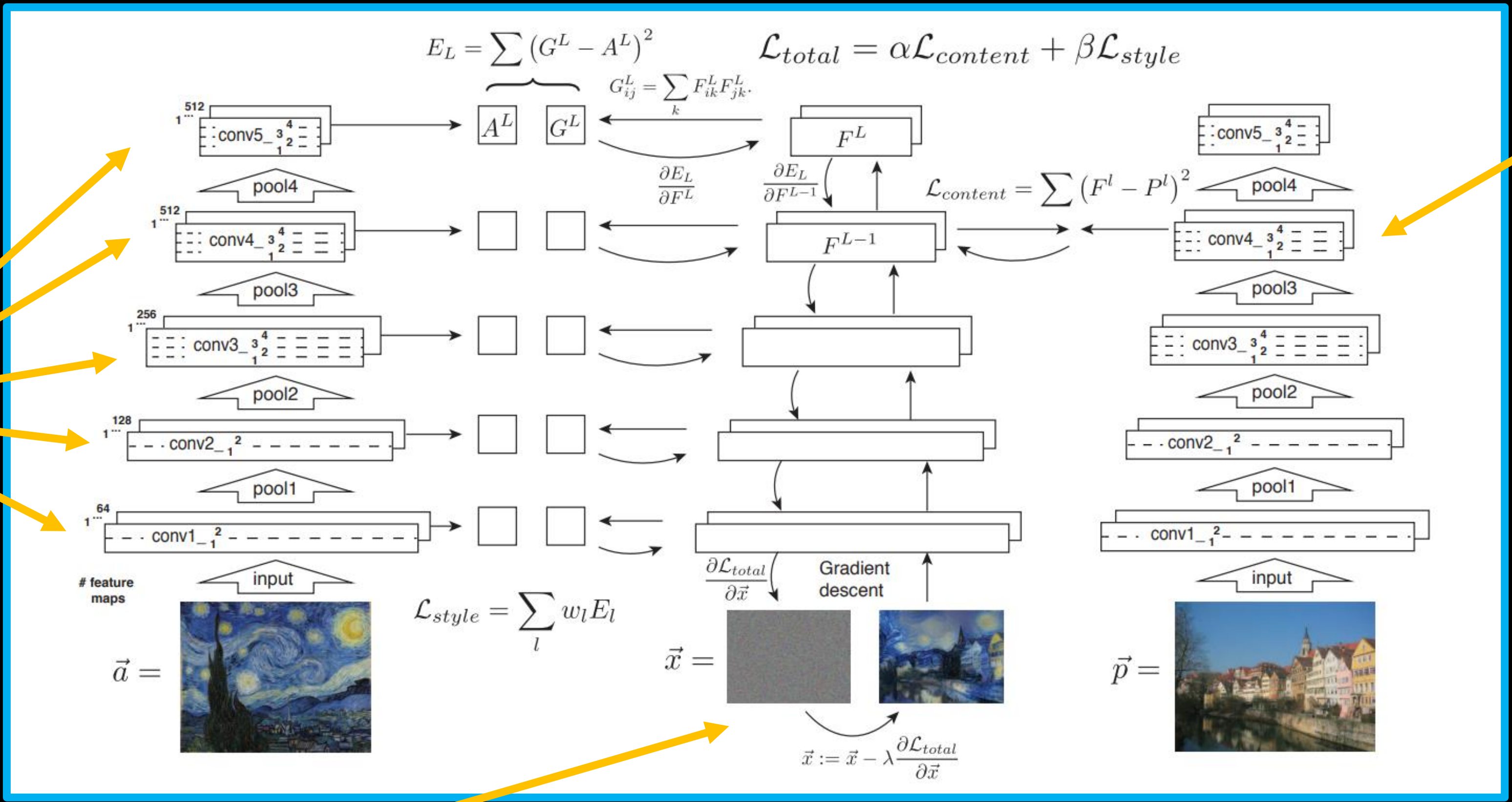


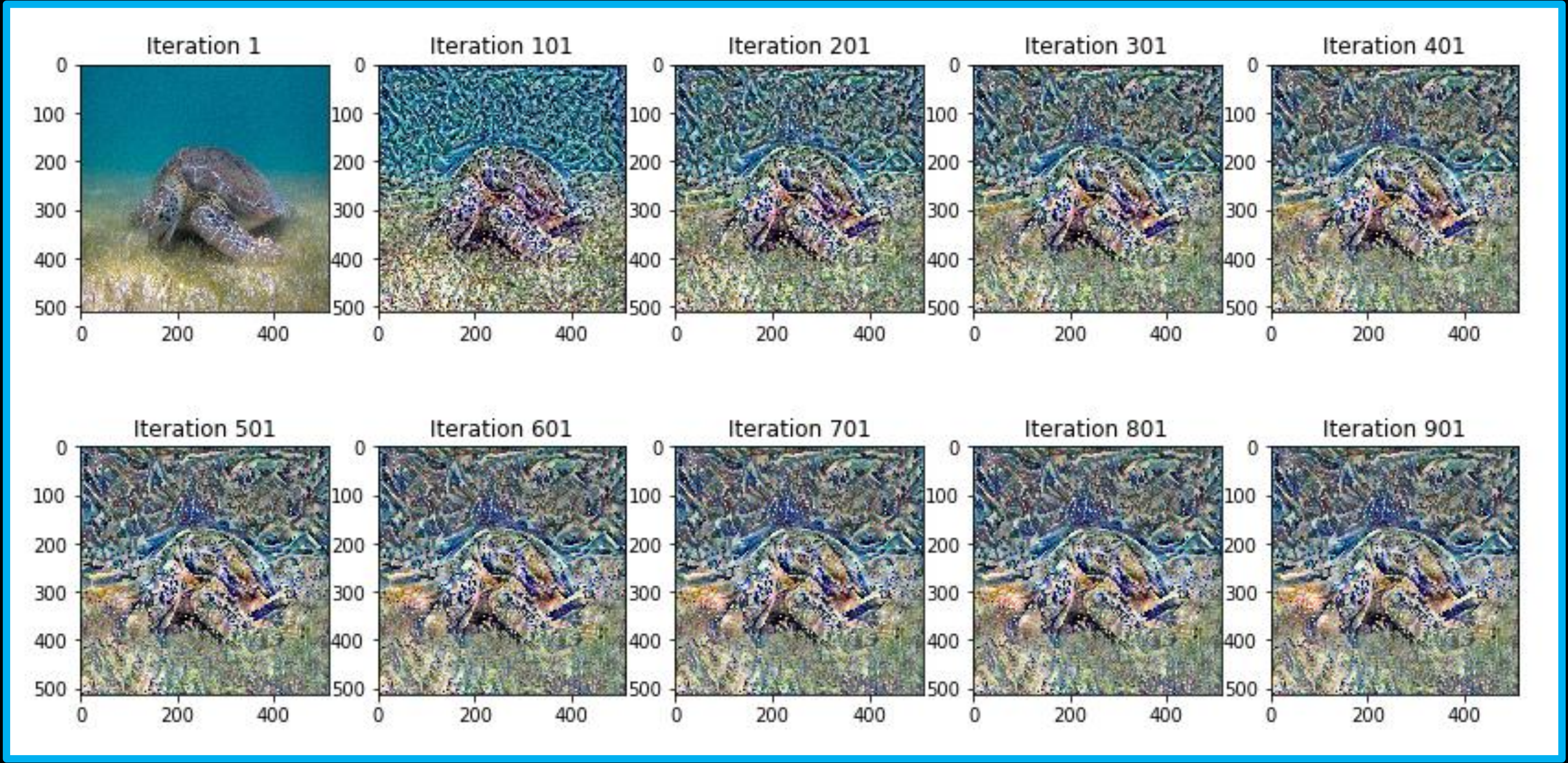
Image content = the feature responses in deeper layers of the VGG network

Optimizing random noise image to have the same responses as those features we saved.

- Iteratively we will create a new image which has the style responses similar to our encoded style features + content responses similar to our encoded content features

This is relatively slow (iterative optimization of the input image), later papers sped it up by using an image2image translation method with feed-forward networks.

Style Transfer

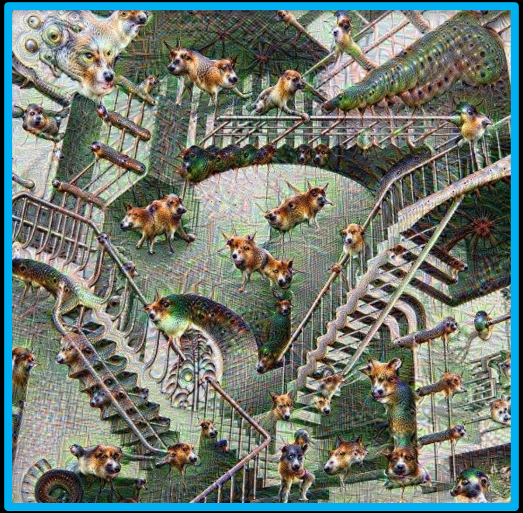


Style Transfer

Style Transfer

- **Online demos**: deepart.io/latest/
- **Colab notebooks**:
  - Basic style transfer (with arbitrary images): ArtML / style_transfer_keras.ipynb
  - Fast style transfer (with pretrained styles): ArtML / fast-style-transfer
- **Papers**: style transfer (2015), fast style transfer (2016)

Deep Dream

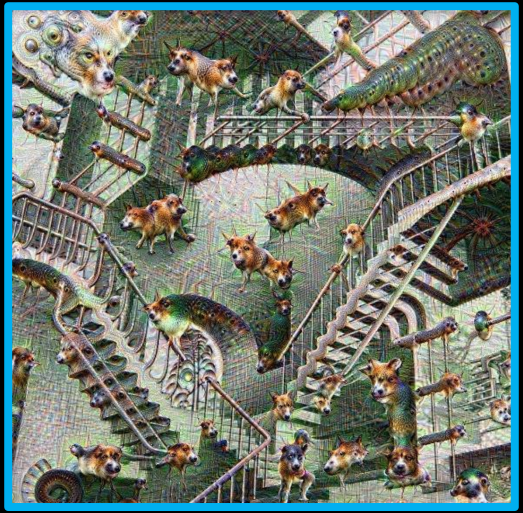


- Deep dream is a method which was originally used to visualize what a network has learned. It works on image optimization principle (as did the first version of Style Transfer).

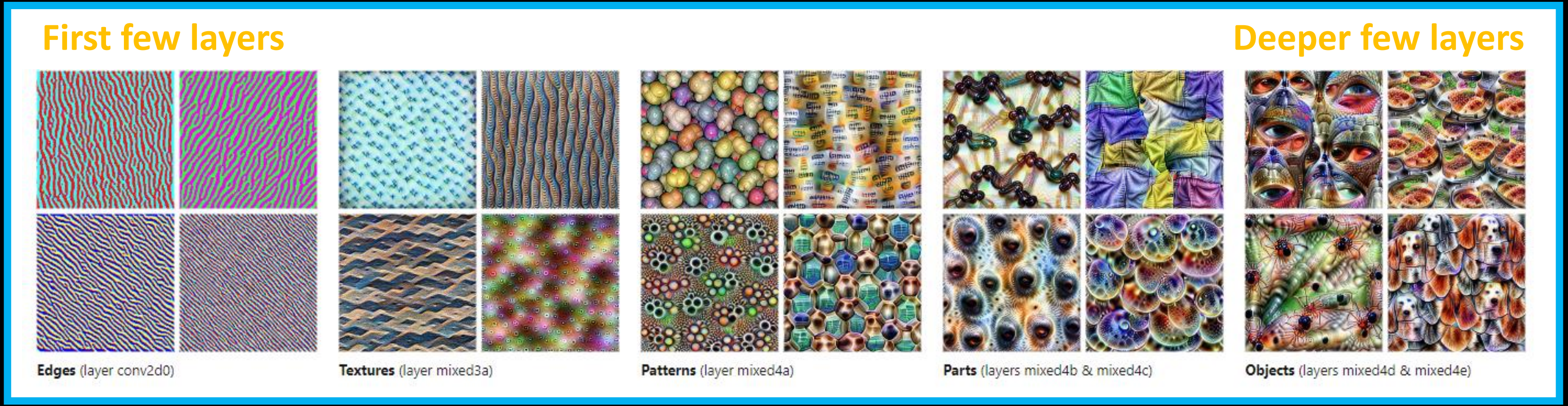
(PS: image source, features viz., illustrational video)

Deep Dream

- Deep dream is a method which was originally used to visualize what a network has learned. It works on image optimization principle (as did the first version of Style Transfer).



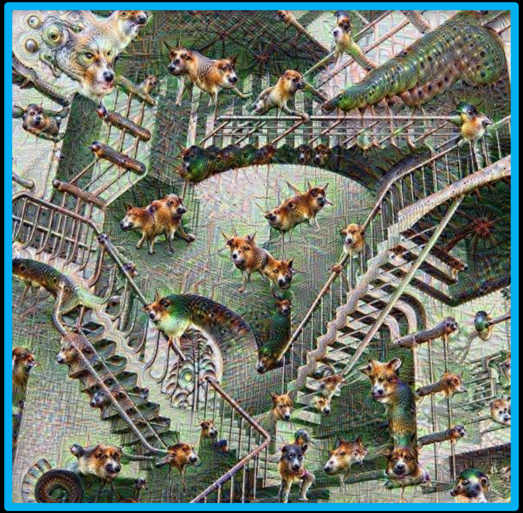
Using **Convolutional network** (GoogLeNet) trained for classification on ImageNet:



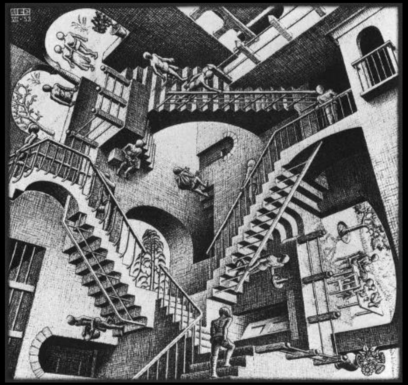
(PS: image source, features viz., illustrational video)

Deep Dream

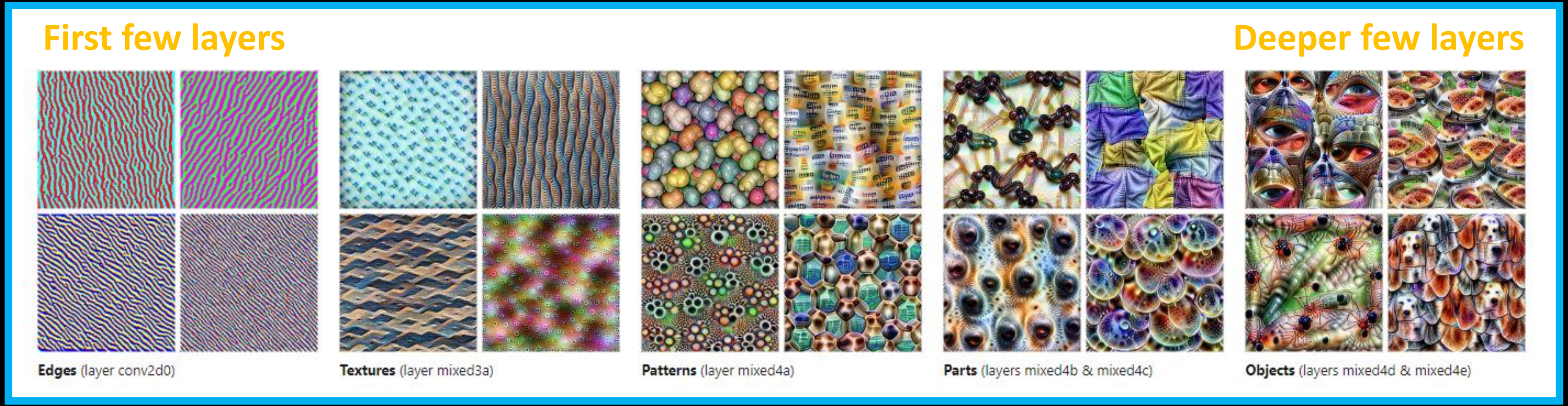
- Deep dream is a method which was originally used to visualize what a network has learned. It works on image optimization principle (as did the first version of Style Transfer).



Change this **original image** so that it activates a **selected feature** with the highest possible force!



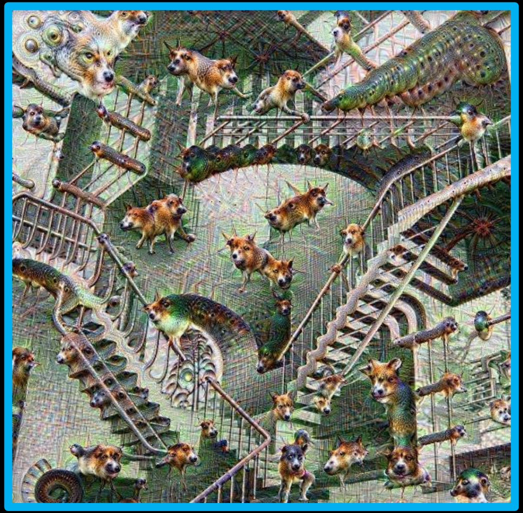
Using **Convolutional network** (GoogLeNet) trained for classification on ImageNet:



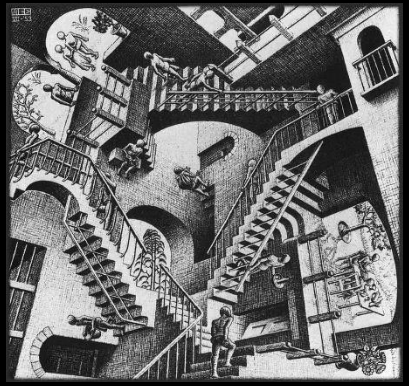
(PS: image source, features viz., illustrational video)

Deep Dream

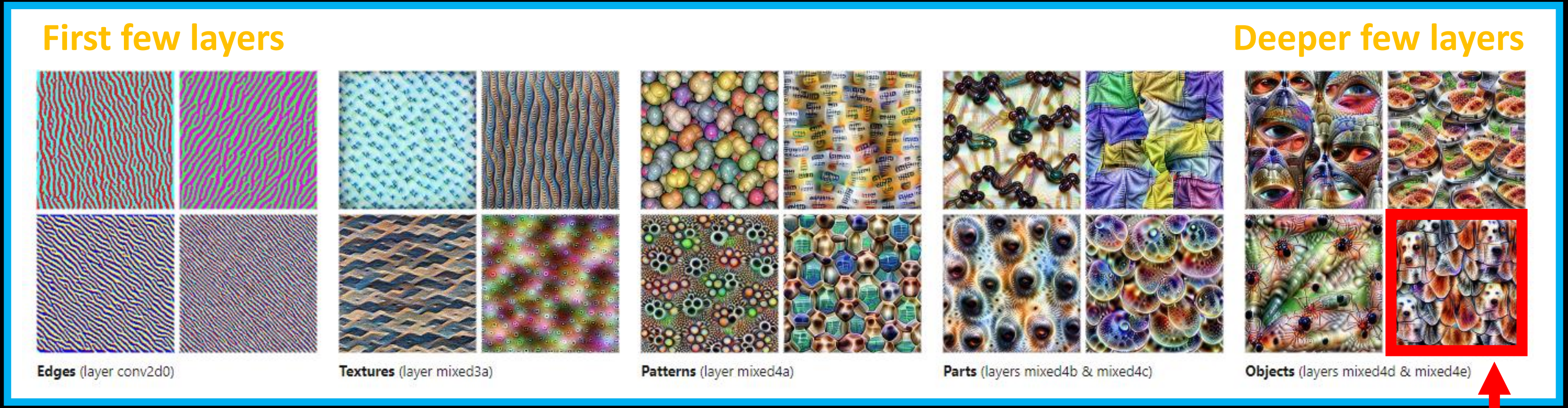
- Deep dream is a method which was originally used to visualize what a network has learned. It works on image optimization principle (as did the first version of Style Transfer).



Change this **original image** so that it activates a **selected feature** with the highest possible force!



Using **Convolutional network** (GoogLeNet) trained for classification on ImageNet:



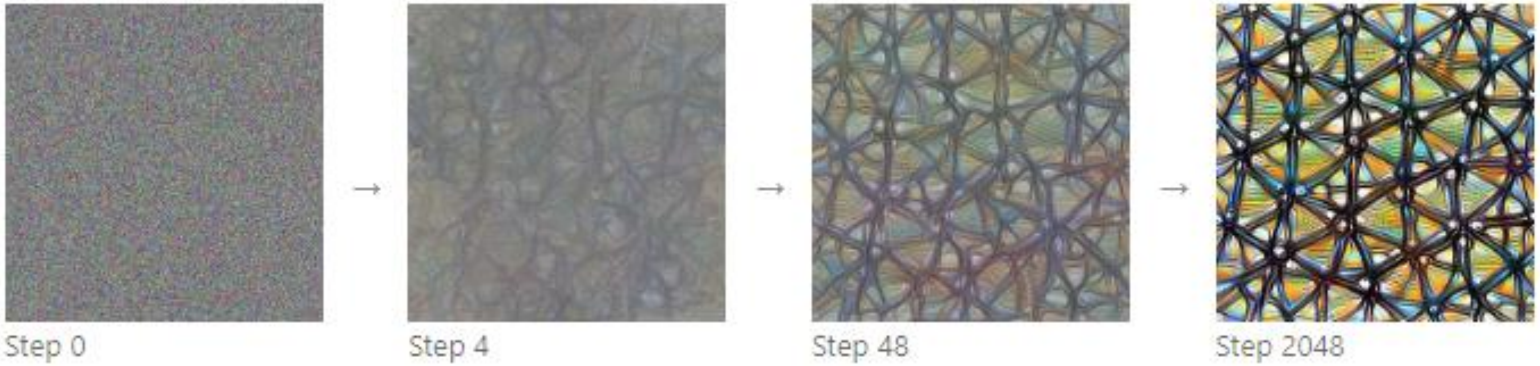
selected feature

(PS: image source, features viz., illustrational video)

Deep Dream

Iterations:

Starting from random noise, we optimize an image to activate a particular neuron (layer mixed4a, unit 11).

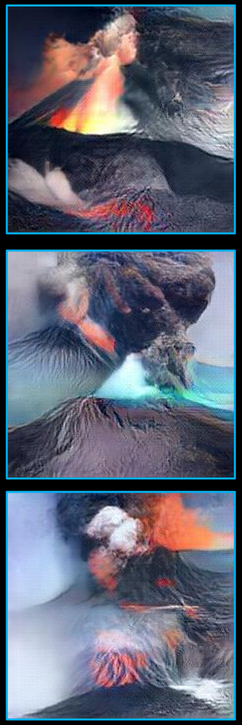


(PS: image source, features viz., illustrational video)

Deep Dream

As a network visualization technique …

Deep dreaming can reveal information stored inside the network …
Recall shapes and content information from the originally used dataset.



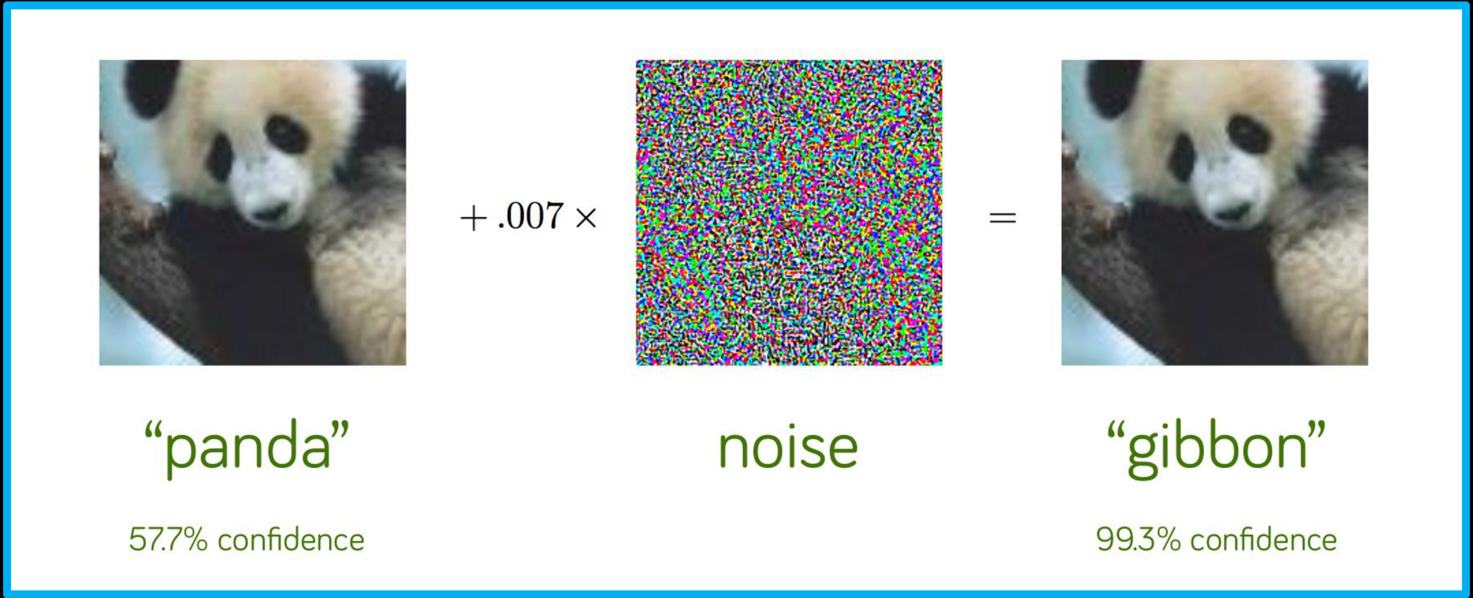
- Model released by Yahoo to identify NSFW content (without releasing the sensitive information about the dataset): github.com/yahoo/open_nsfw
- Which was (*of course …*) soon followed by using a deep-dream–like technique: open_nsfw.gitlab.io/
  - Interesting usage of the network to generate *suggestive* imagery (changing shapes of landscapes imagery, etc.)

(PS: link about this in media)

Deep Dream

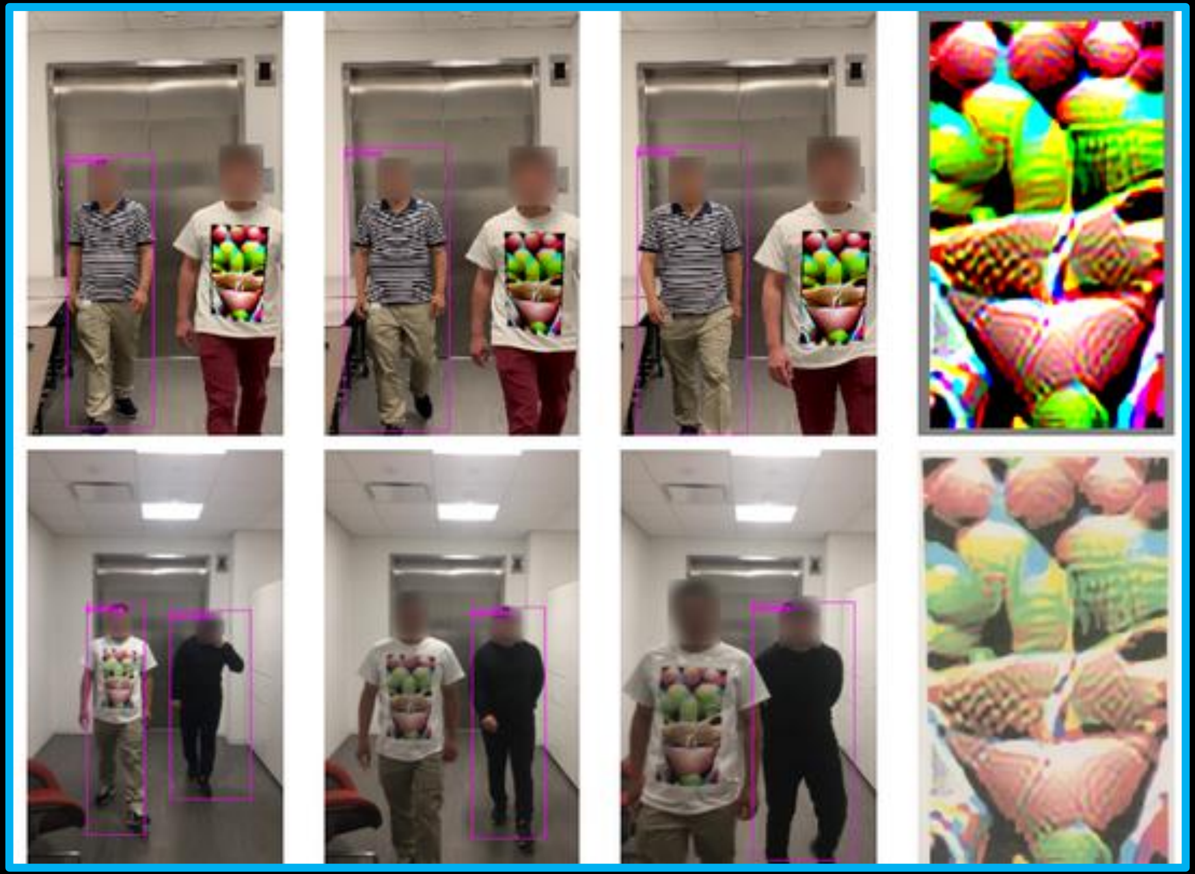
- **Online demos**: dreamscopeapp.com (*not 100% sure if it's not a style transfer of deep dream like effect*)
- **Colab notebooks**:
  - Deep dream a photo: as a ML4A guide
  - Alternative code: ArtML / neural-synth-clustering-v2.ipynb
- **Reading**: distill.pub/2017/feature-visualization/

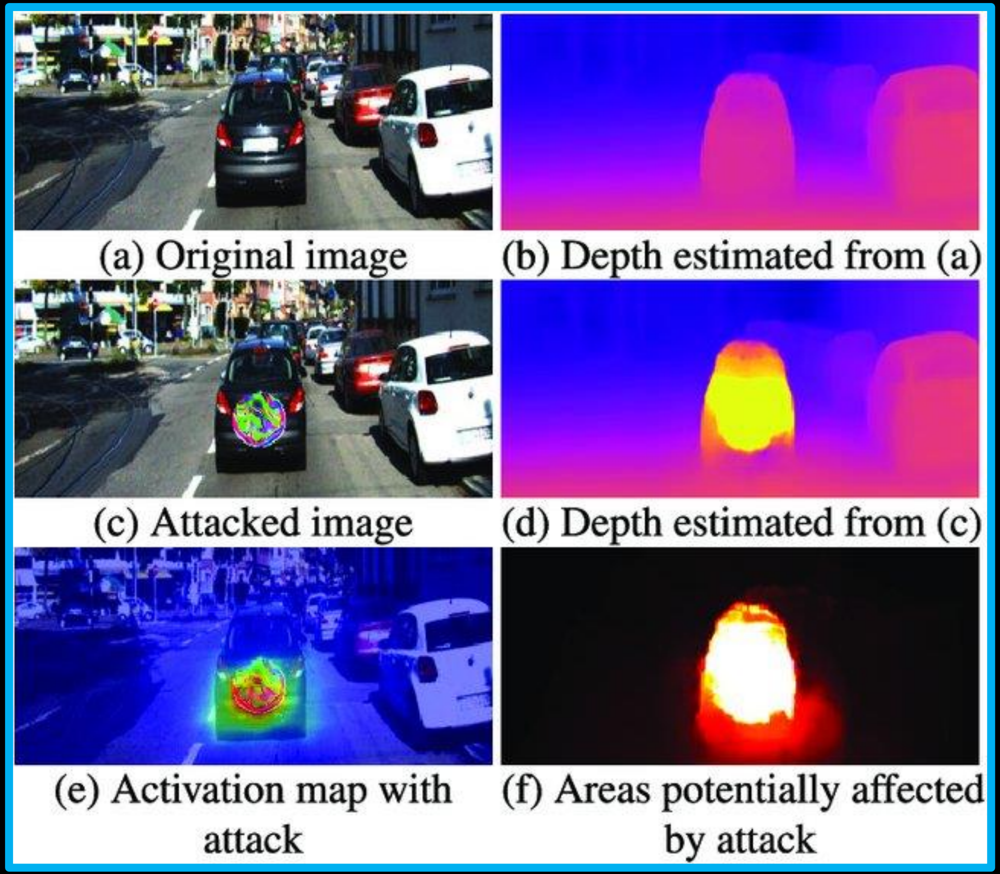
Sidenote: Adversarial attacks



- Related to input image optimization techniques (such as Deep Dream) – **optimize the "noise"** image so that a target network classifies the combined image as a "gibbon" (*selecting the target feature which responds to "gibbons"* in classification)
- *Note this can be further used to test how robust/fragile some networks are*

Physical Adversarial attacks





- Against YOLOv2 model
- Against depth estimation nets

Adversarial T-shirt! Evading Person Detectors in A Physical World - arxiv.org/abs/1910.11099

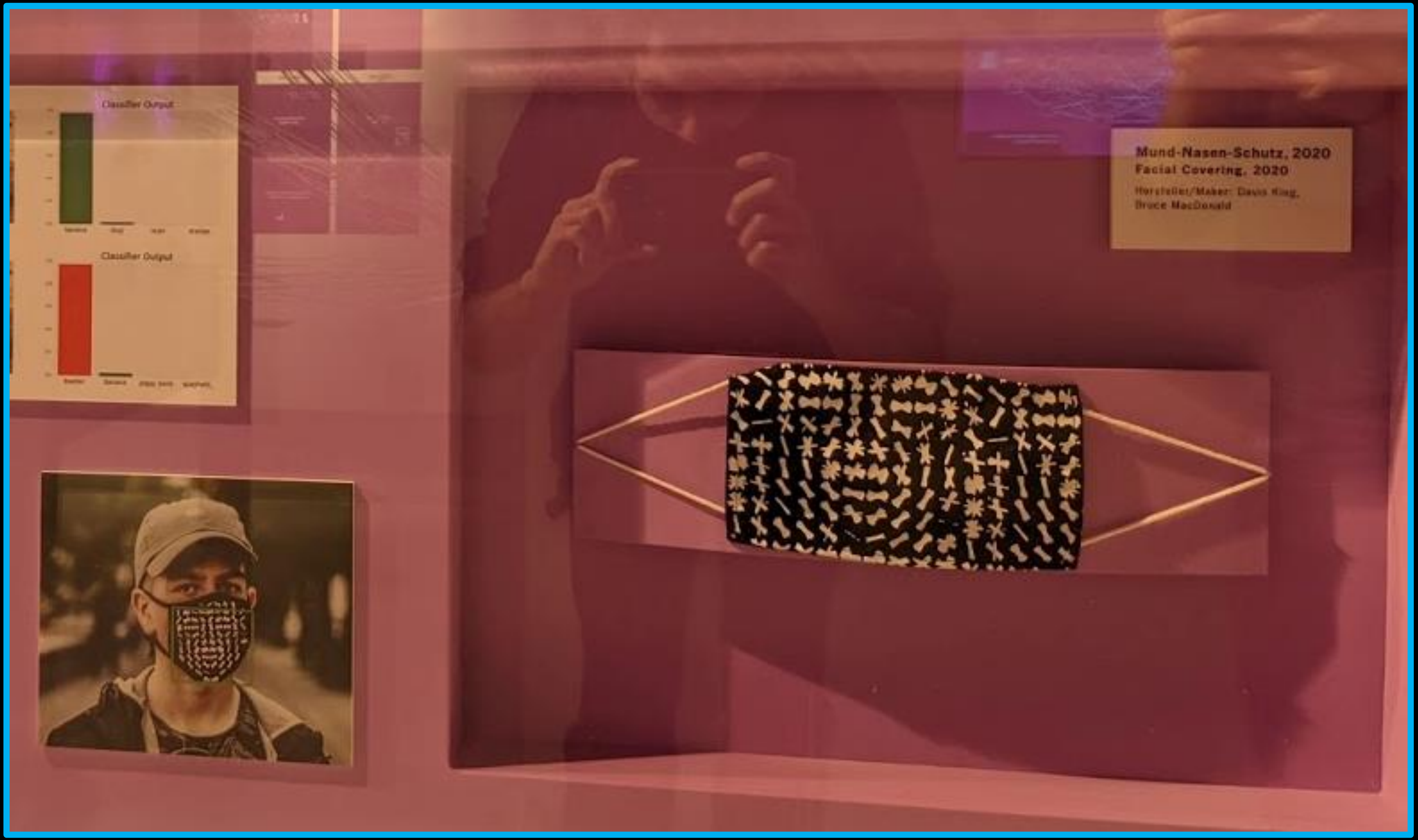
Adversarial Patch Attacks on Monocular Depth Estimation Networks (2020)

Physical Adversarial attacks

Adversarial turtle:

www.youtube.com/watch?v=YXy6oX1iNoA

Physical Adversarial attacks



Today we learned …

- **Image 2 image models** and what all can be learned as a translation problem – and how this can work with paired or even unpaired datasets
- **Optimization of the input image** in terms of optimizing image to have a certain style, or to excite a certain target neuron (including ones that mistake the original class of the image)

Links and additional readings:

- **Online pix2pix demos**: affinelayer.com/pixsrv/
  - **Follow-up papers**: pix2pixHD (with high. res.), vid2vid (with frame to frame consistency)

- **Bonus readings:**
  - **Feature Visualization**, Distill – blog
  - **Sensory Optimization**: Neural Networks as a Model for Understanding and Creating Art – paper
  - About **Pix2Pix** on ML4A – blog with code

End of the lecture

***)** PS: follows material for the practical session …

AI for the Media

Week 9, Domain 2 Domain

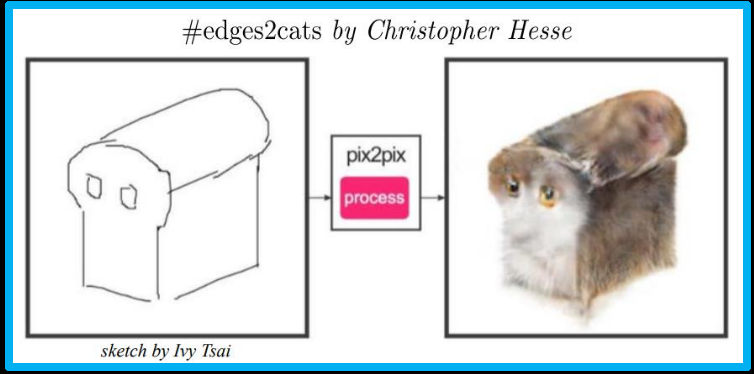
Practical: **pix2pix training**

~ Vít Růžička

Practical: Domain 2 Domain

Training Pix2Pix models

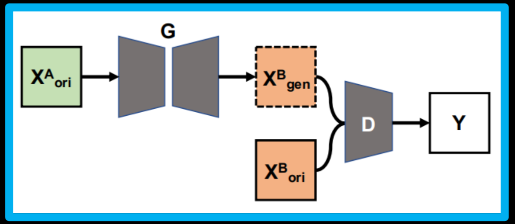
Namely using the sketch2image with our own data.



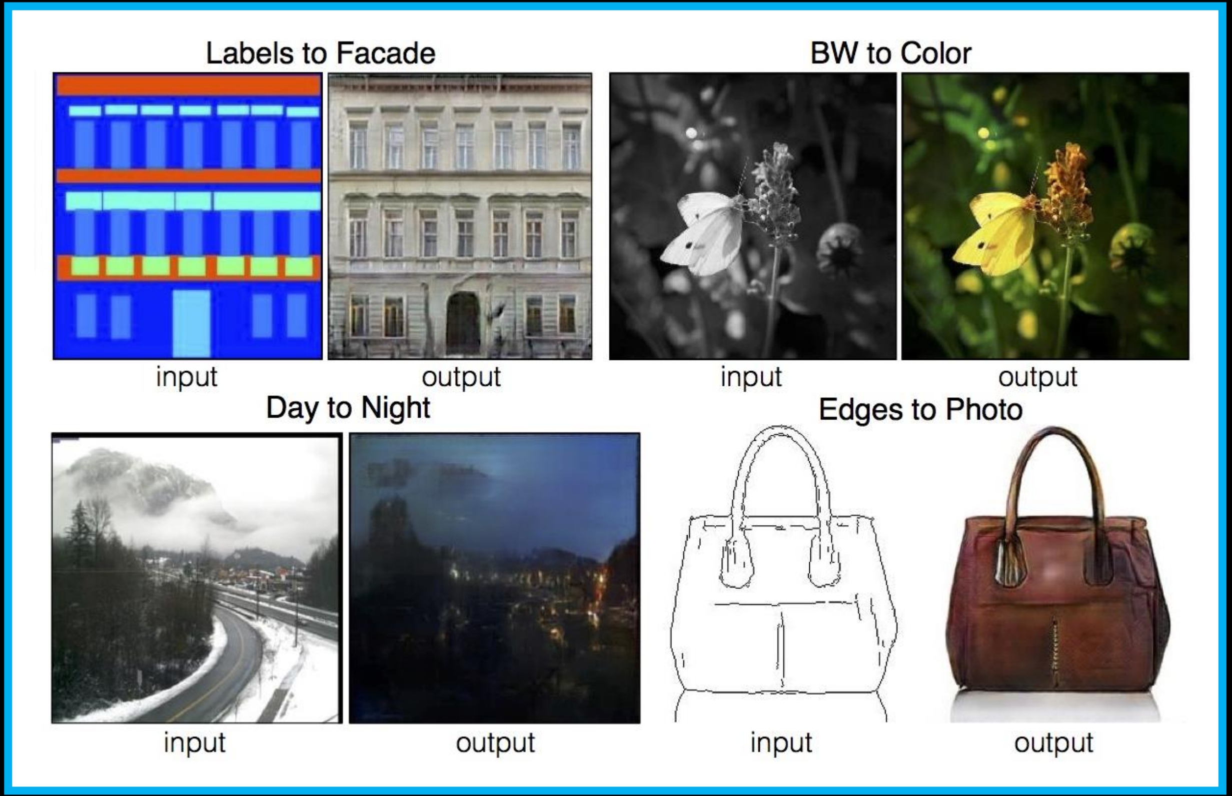
Continue with code on Github:

- Repo: github.com/previtus/cci_AI_for_the_Media
- **Notebook** directly: week09_domain-to-domain/aim09_pix2pix_keras.ipynb

Pix2Pix



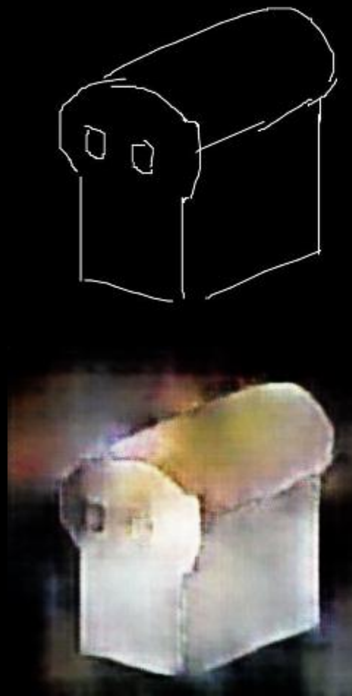
- You might recall that we will need **paired data** in our dataset:
  - Data from domain **A**
  - Data from domain **B**



- And we will be learning the **translation between A to B**

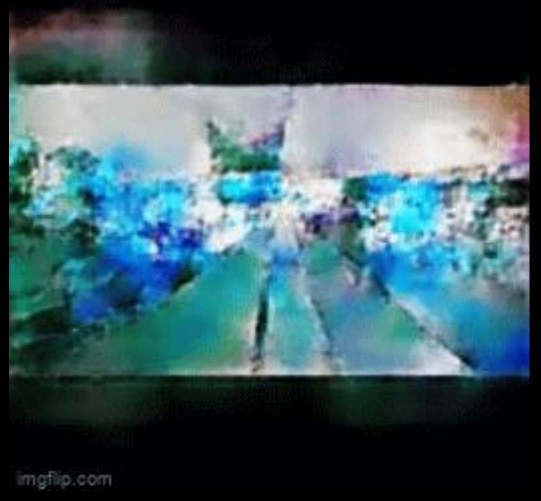
Examples from training:

- **ps:** remember that the model saw only the edges/sketch when reconstructing these images
- **(note)** the pix2pix demo's have much better quality of the trained models



Demo with a video ->

- Note that there is almost none temporal consistency between these frames …
- Longer **video:** [here]



The end