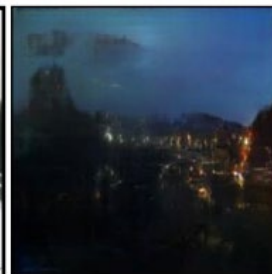
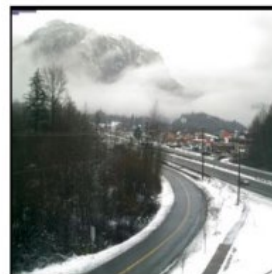


AI for the Media

Week 6, Domain 2 Domain



Overview

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

- **What can we do** with these models?
- **What does the model need?**
 - Combining what we already learned about to make this work!
- **Limitations and follow-up models**

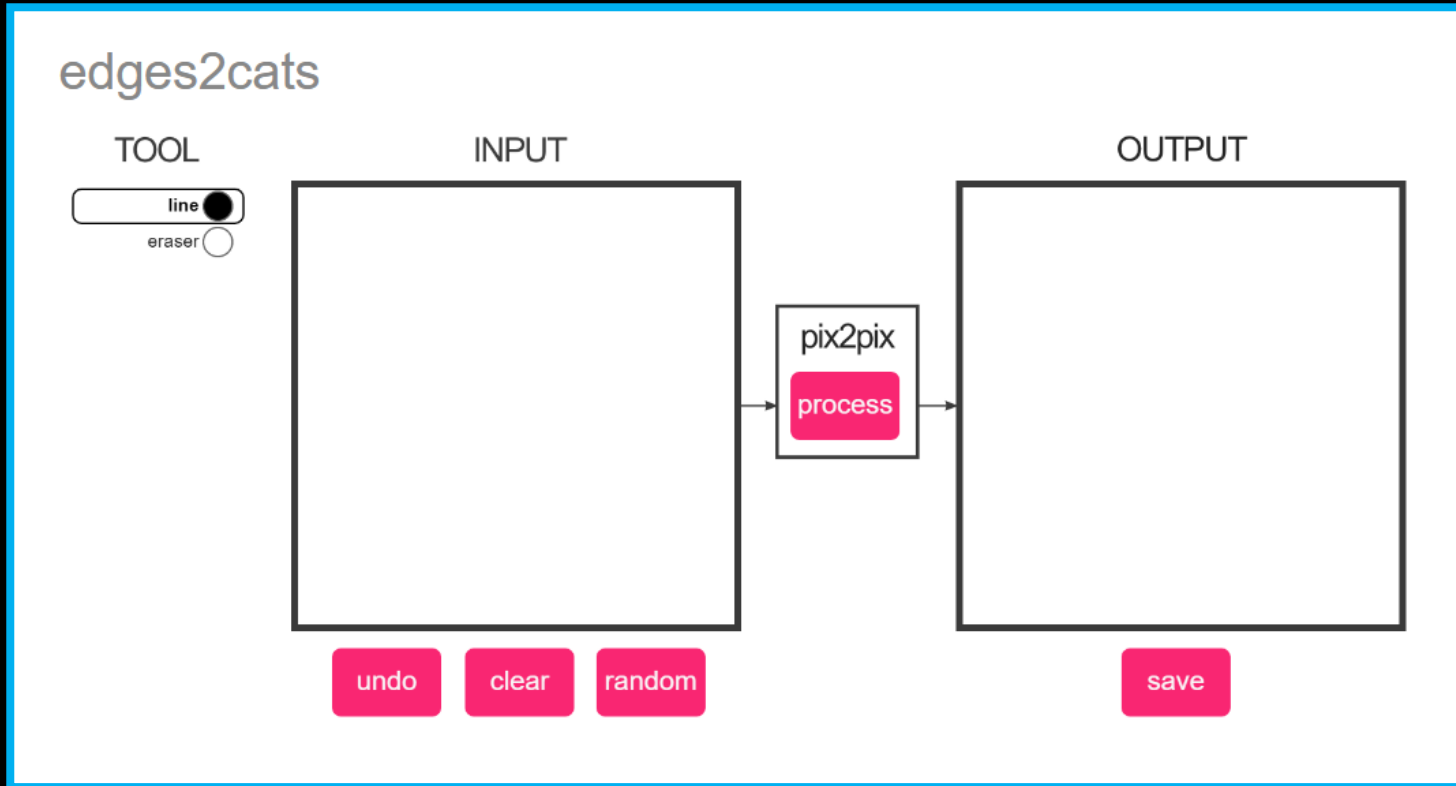
Practical session (*during the live session*):

- **Code**: Training a pix2pix model

I Motivation

Motivation for today

Domain to domain (or image to image) models



pix2pix (2017)

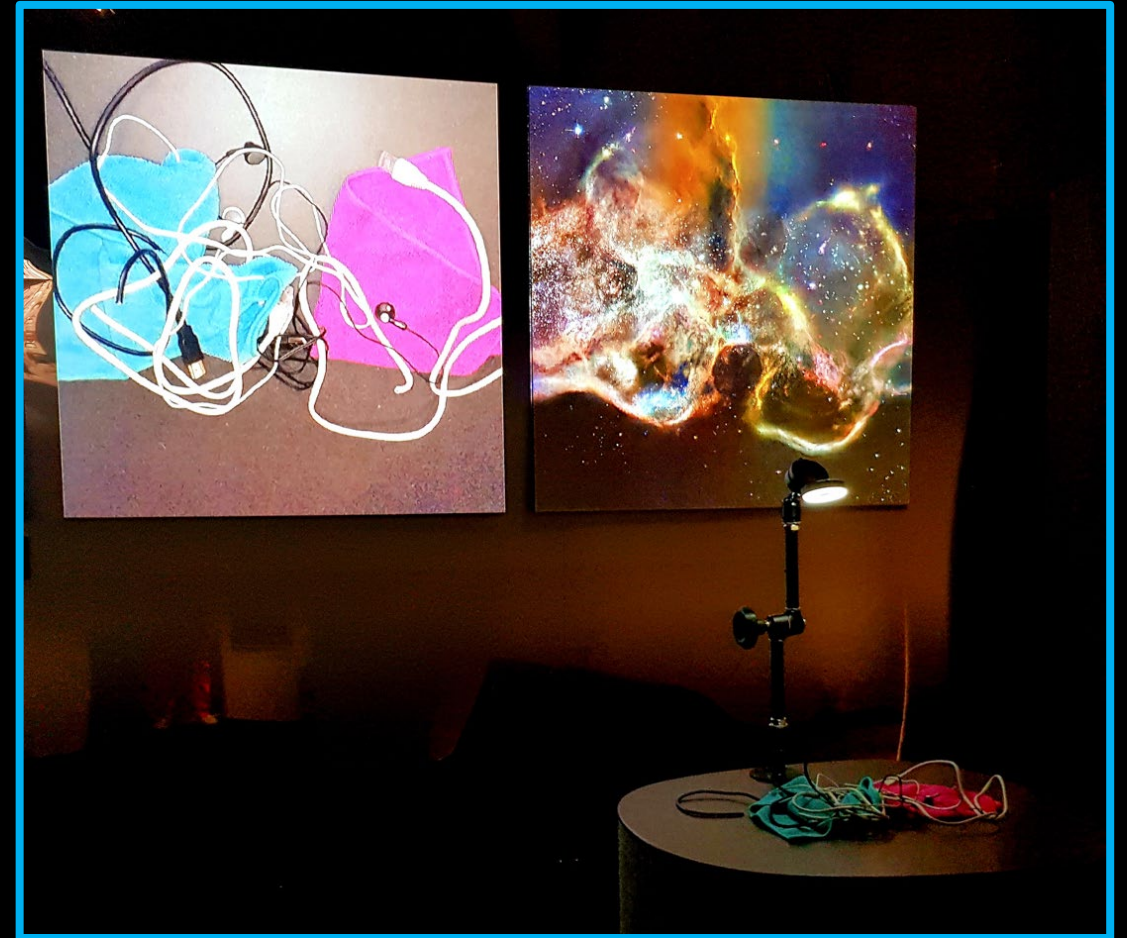
Online demo:
[affinelayer.com/
pixsrv/](https://affinelayer.com/pixsrv/)

Example 1: Learning to see...

- Custom **pix2pix** running in real-time and interactively at the “AI: More than Human” exhibition at Barbican

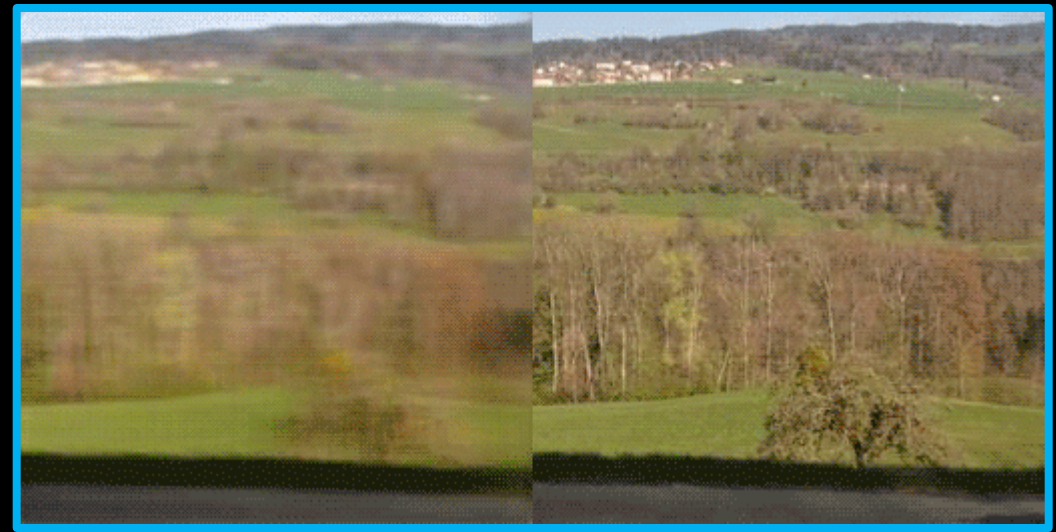
Video: vimeo.com/260612034

Info: memo.tv/works/learning-to-see/



Example 2: Predicting the future ...

- Uses **pix2pix** trained on a sequence of frames from a video
 - Also shows the progress of the trained network (rubbish at the beginning, but gets better)



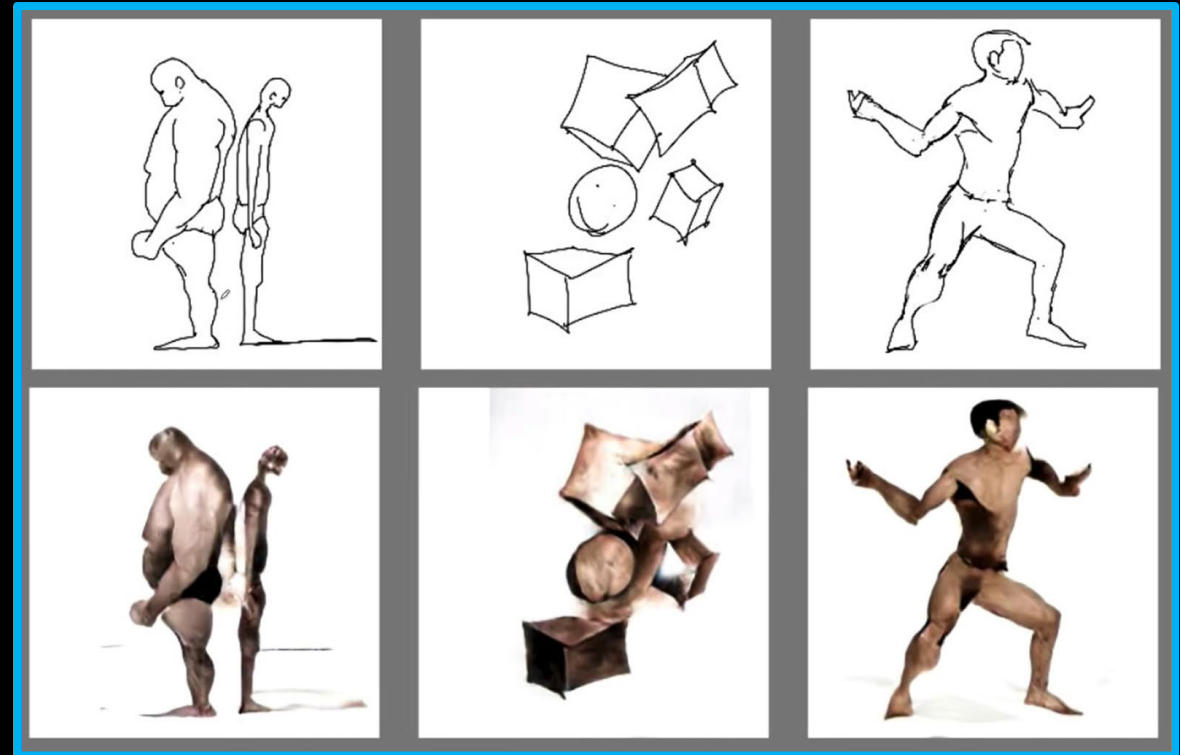
Video: [youtube.com/watch?v=lr59AhOPgWQ](https://www.youtube.com/watch?v=lr59AhOPgWQ)
Blog: https://magenta.tensorflow.org/nfp_p2p

Example 3: Sculpting with AI ...

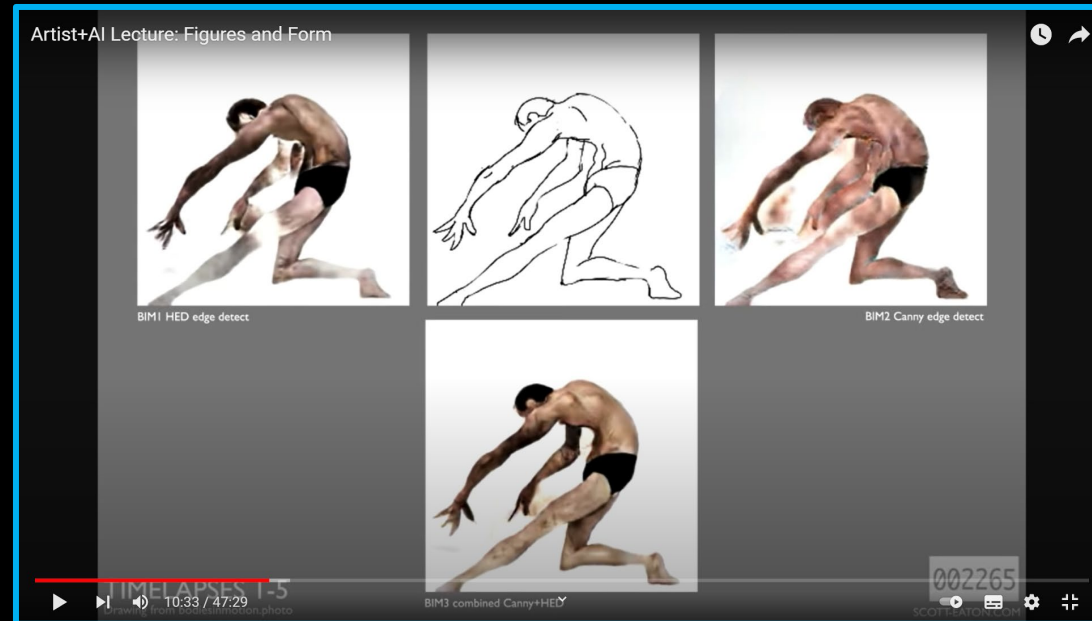
- Improved versions of **pix2pix** trained with custom dataset of human bodies

Video: youtube.com/watch?v=TN7Ydx9ygPo

Info: nvidia.com/en-us/research/ai-art-gallery/artists/scott-eaton/



Scott Eaton's description



<https://youtu.be/TN7Ydx9ygPo?t=480>
(around 2 mins)

Motivation for today

Learning how these artworks were made!

Dataset:



Motivation for today

Learning how these artworks were made!

Dataset:



To edges:



Motivation for today

Learning how these artworks were made!

Dataset:



To edges:



Train a model on
this translation

Motivation for today

Learning how these artworks were made!

Dataset:

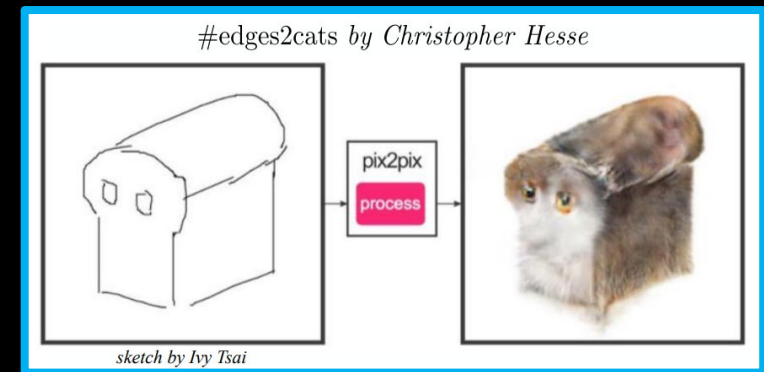


To edges:



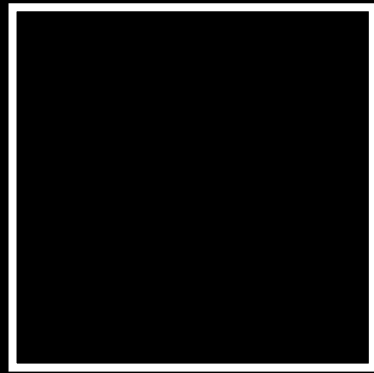
**Train a model on
this translation**

**Then use the model
on new kinds of data:**

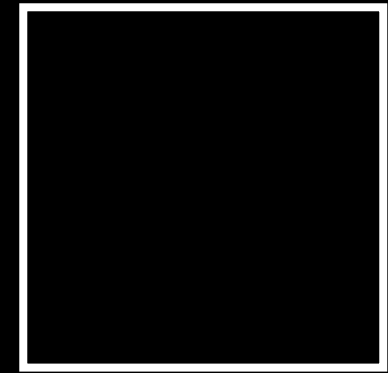
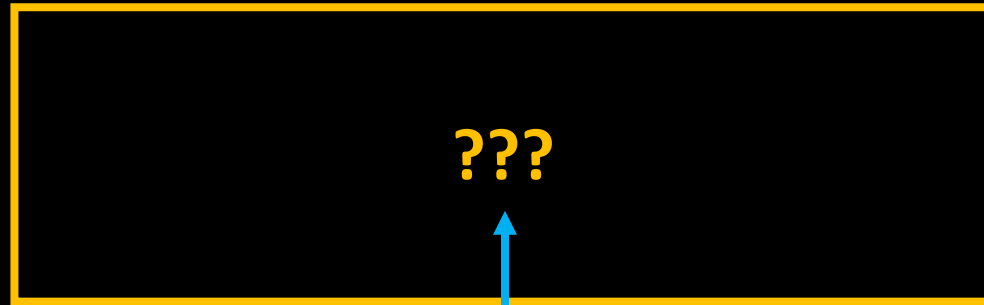


II How to get there?

Domain to Domain



Images from
domain A

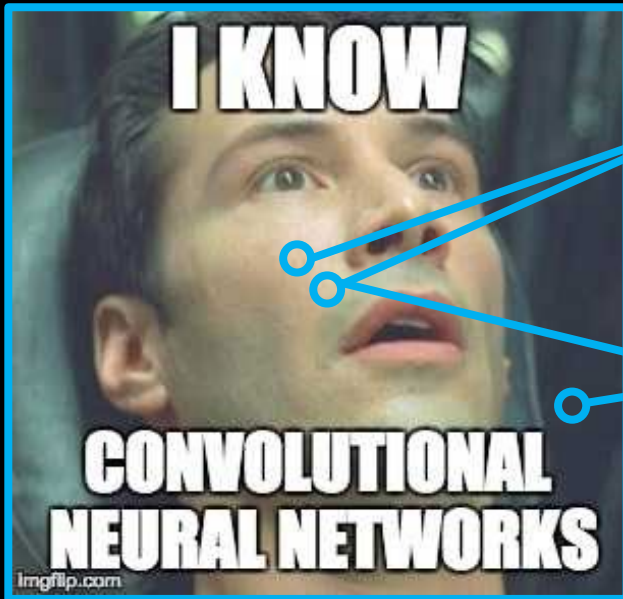


Images from
domain B

What kind of model can we use here?

Convolutional layers (recap)

Intuition: working
with images

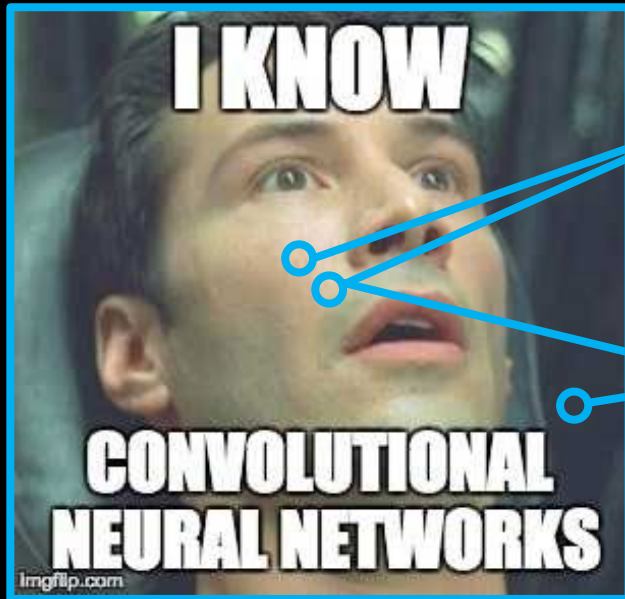


Nearby pixels
correspond.

Far away
pixels don't.

Convolutional layers (recap)

Intuition: working with images



Nearby pixels correspond.

Far away pixels don't.

Manual Convolutional filter:

emboss

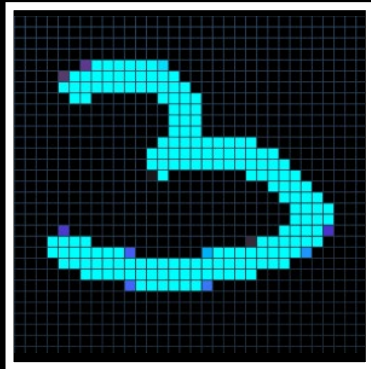
-2	-1	0
-1	1	1
0	1	2

3x3 conv



Convolutional layers (recap)

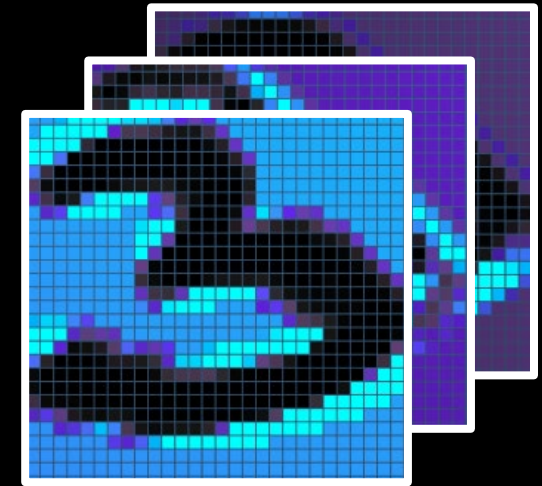
Learned Convolutional filters:



Input to the
layer



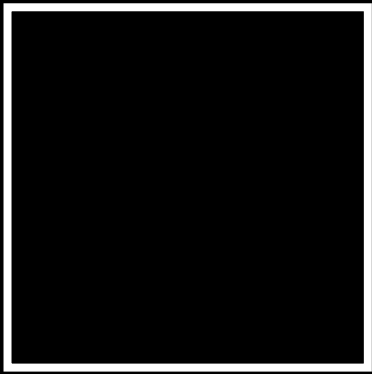
**Layer with 5x5
convolution**



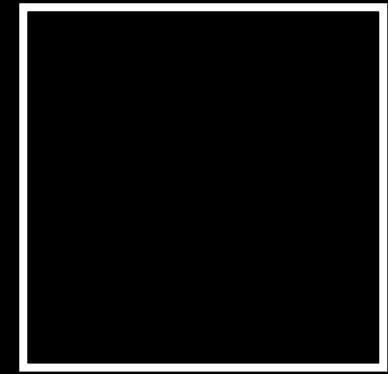
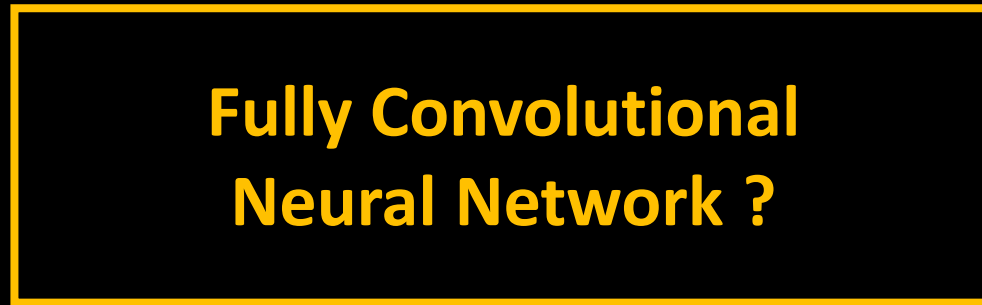
Outputs of the
convolution

- Interactive Convolutional Neural Network (runs in the browser and has good visualization): cs.cmu.edu/~aharley/vis/conv/flat.html

First idea: Convolutional NN



Images from
domain A

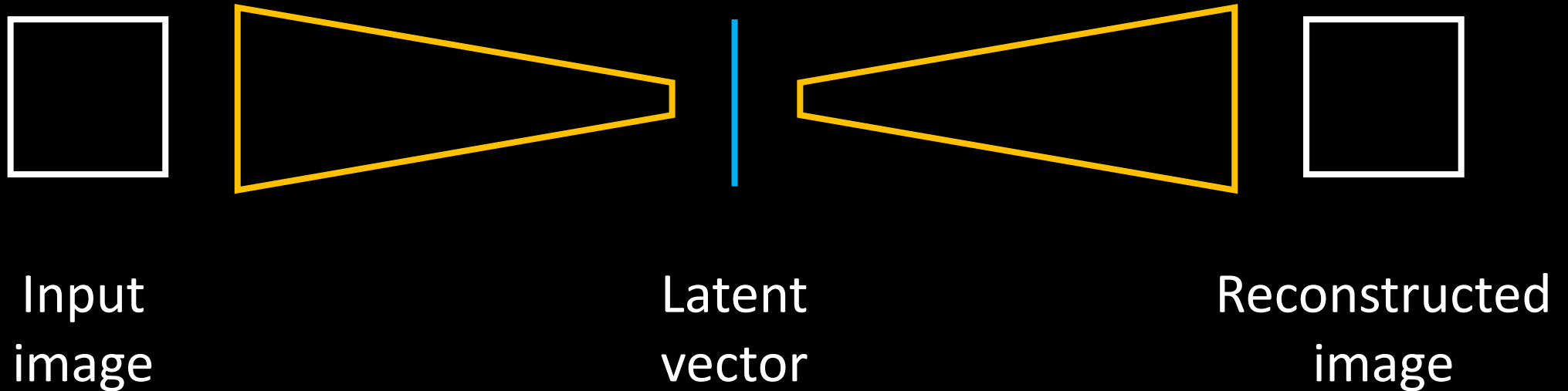


Images from
domain B

- “Fully Convolutional” with only convolutional layers
- Problem: Too complicated task, no separation for different object sizes

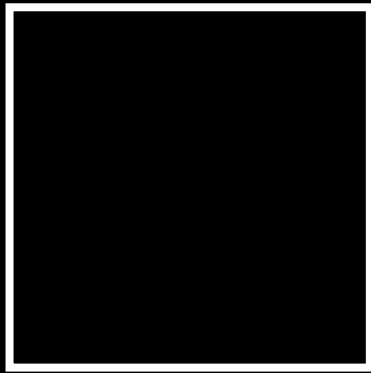
Encoder-Decoder models (recap)

Auto-encoder models

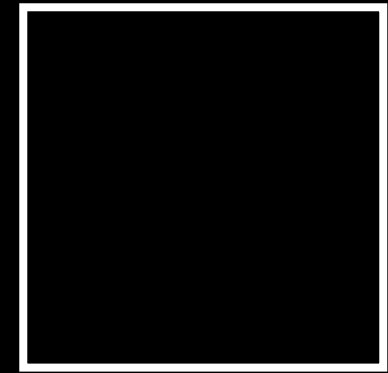
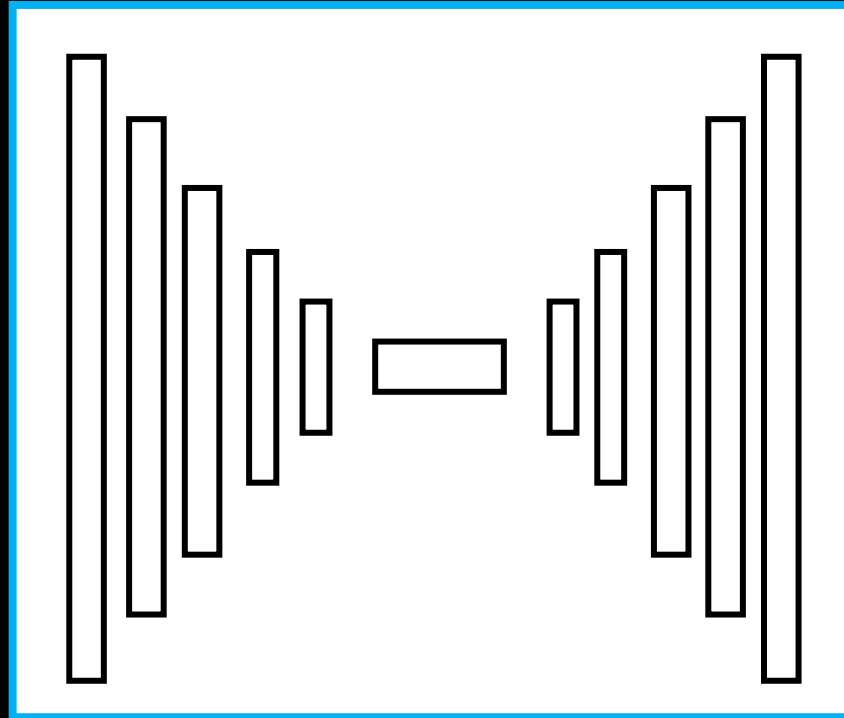


- AE models are trained to reconstruct images as best as they can, while forcing the image through a lower dimensional **latent vector**

Second idea: Auto-Encoders



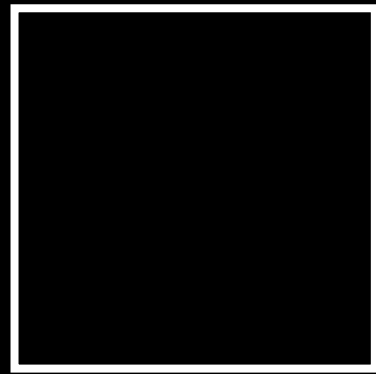
Images from
domain A



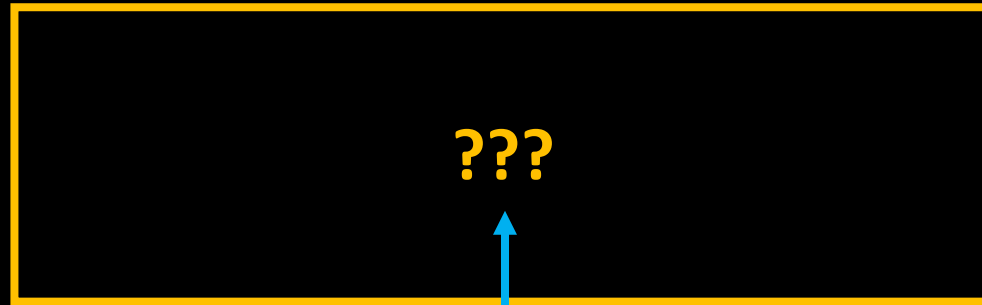
Images from
domain B

- Encoder-decoder architecture to learn the translation?
- **Problem: AE models force the reconstruction through a small bottleneck**
 - We loose a lot of information in the process and the results would be *blurry*

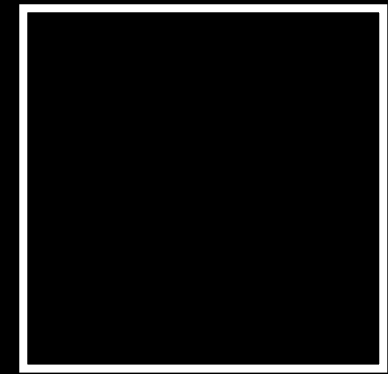
Domain to Domain



Images from
domain A



???



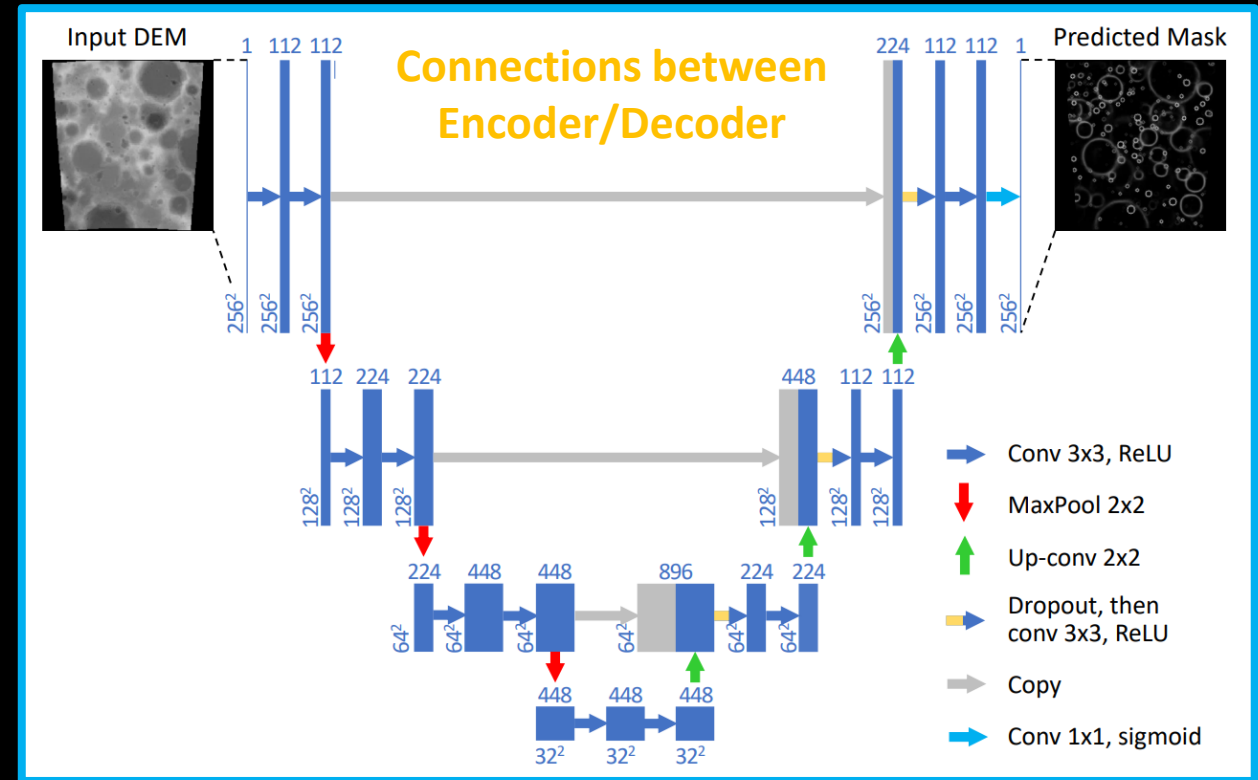
Images from
domain B

What kind of model can we use here?

Not: Fully Conv NNs, AutoEncoders, ...

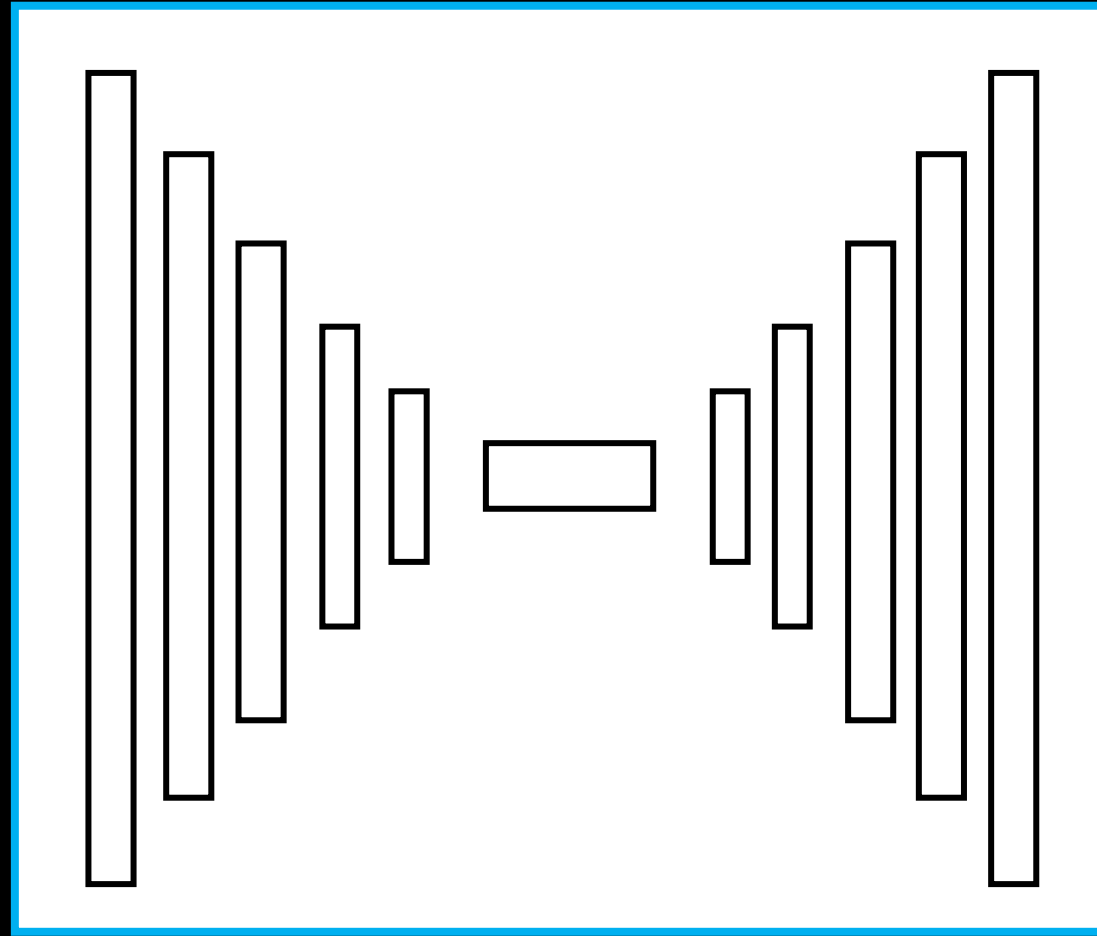
Adding skip connections

- Idea for the **U-Net model**:
 - Add **skip connections** between the encoder and decoder networks
 - Paper with U-Net applied on the task of lunar crater identification

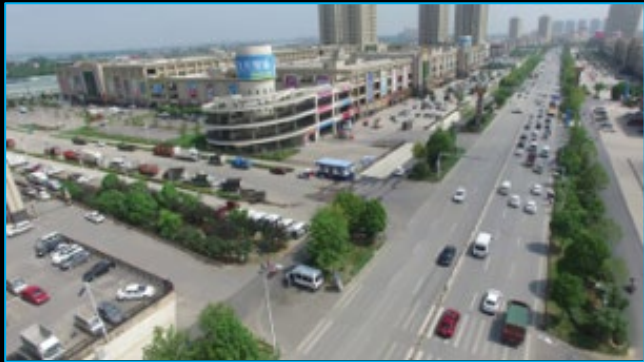


(PS: U-Net is from 2015, this [paper](#) from 2018)

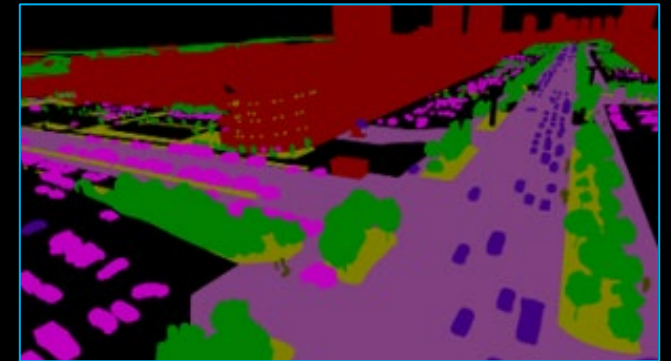
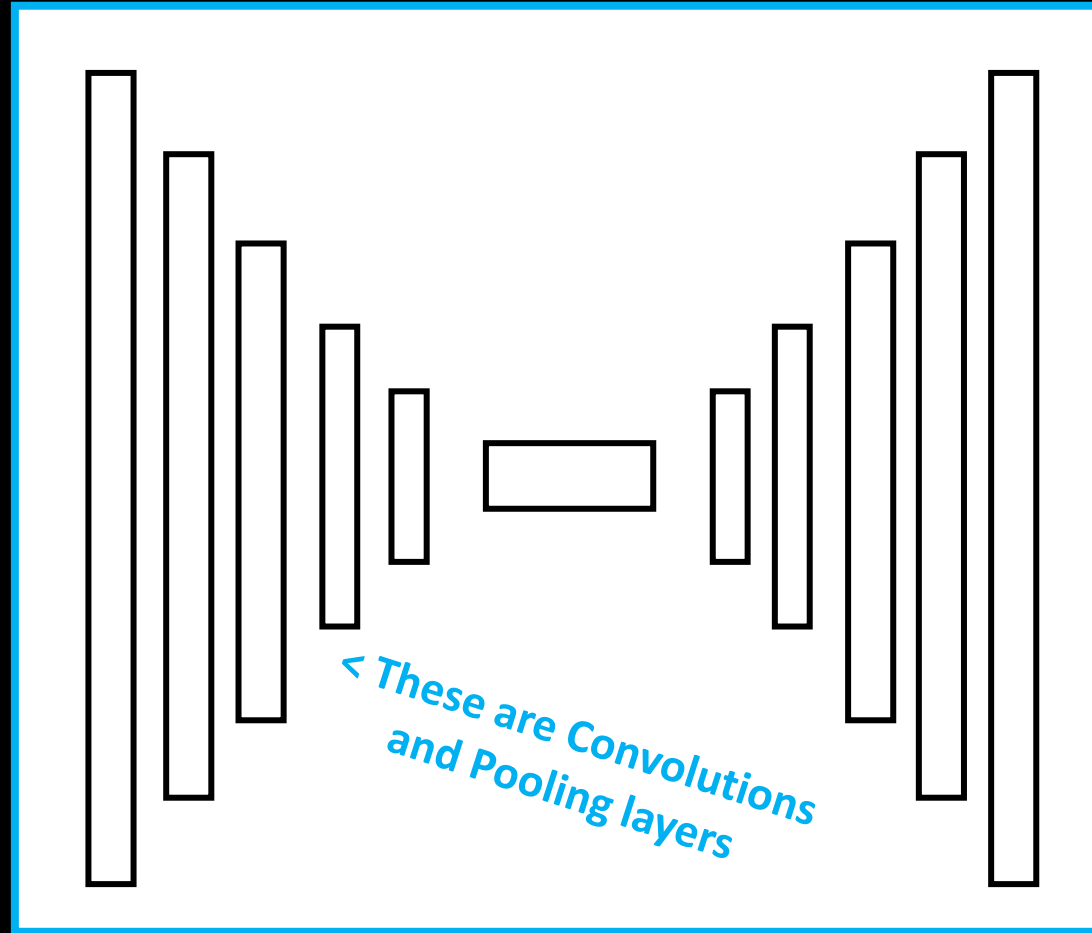
U-Net



U-Net



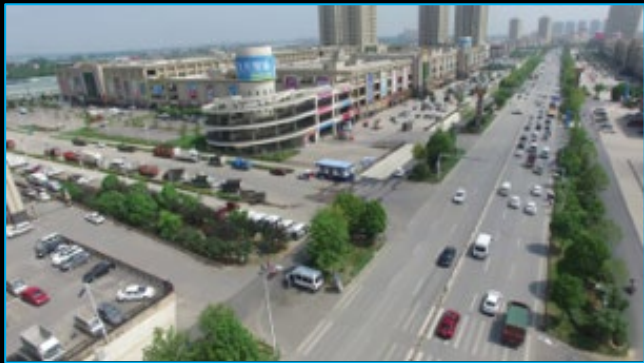
Images from one
“domain”



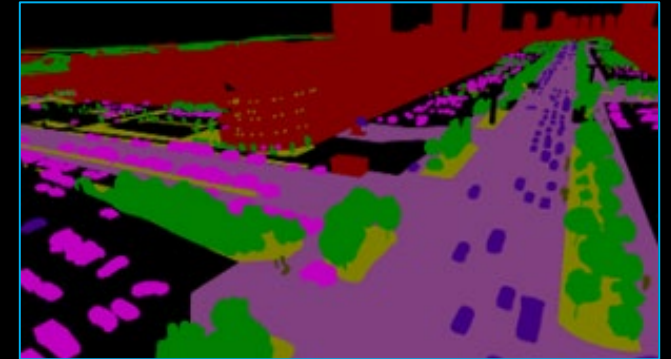
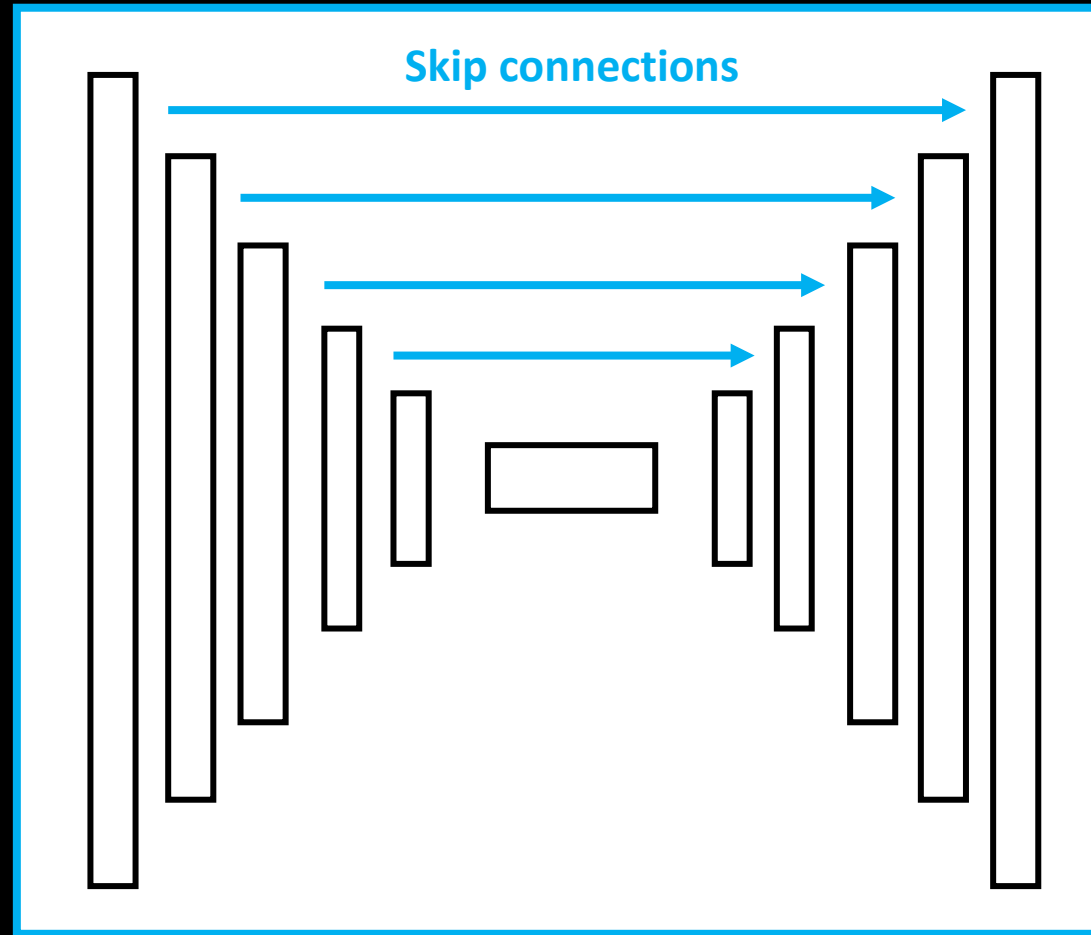
Paired images from
another “domain”

- Connects one image to another while going through a bottleneck

U-Net



Images from one
“domain”

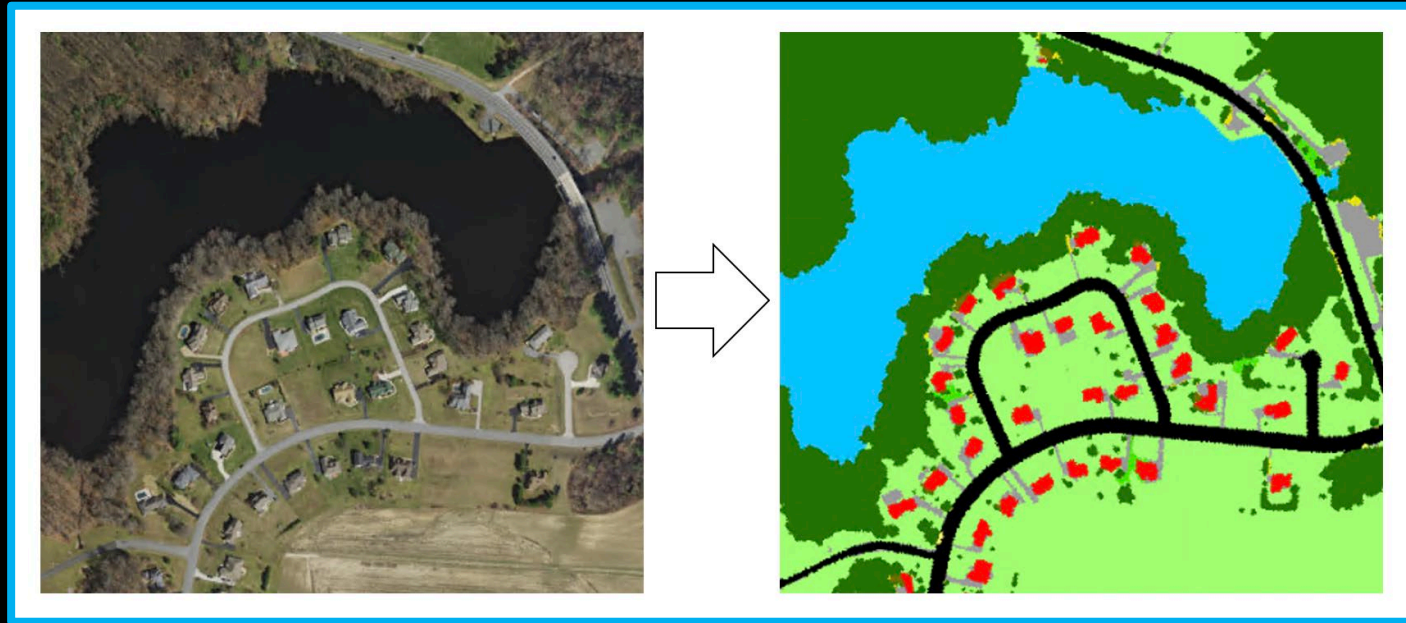


Paired images from
another “domain”

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

U-Net in a real-world scenario

- Usually used for the task of **semantic segmentation**

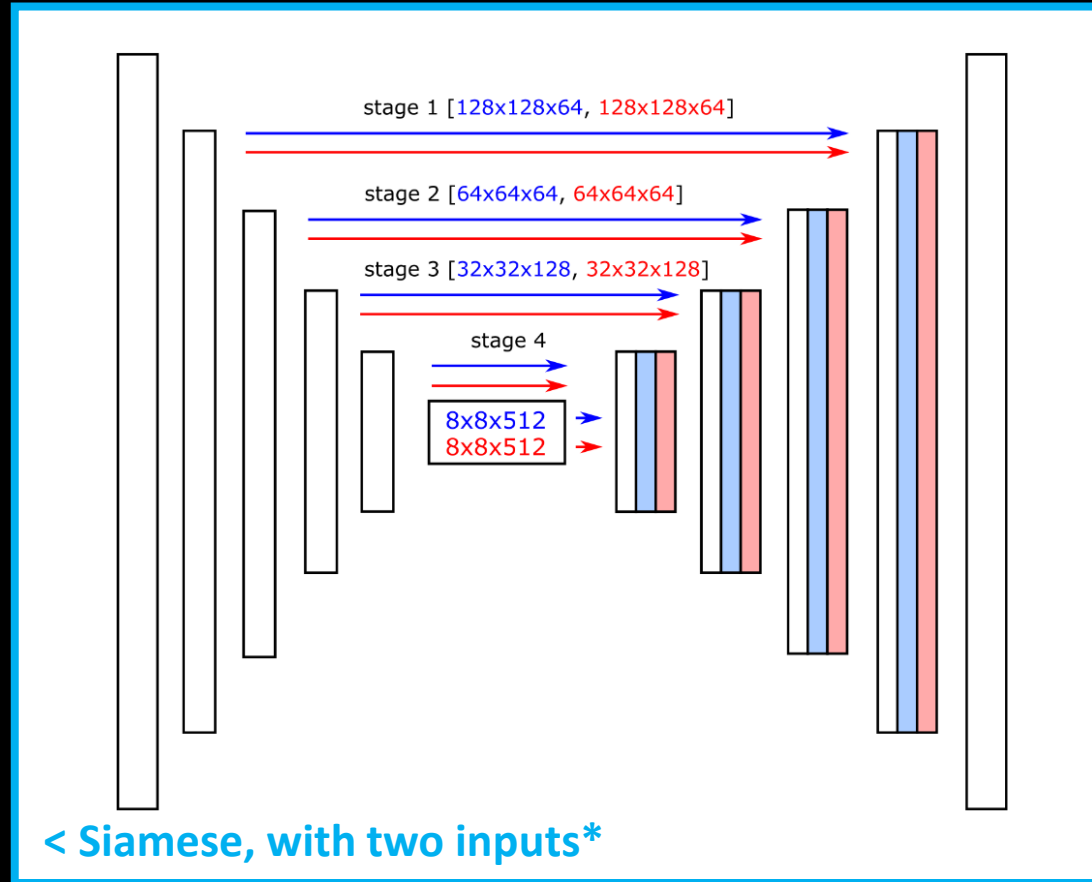


From real
world data to
semantic
labels

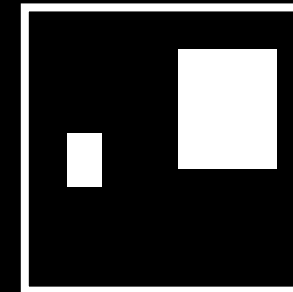
Example code (cat segmentation demo):

github.com/zaidalyafeai/Notebooks/blob/master/unet.ipynb

U-Net in a real-world scenario



Label

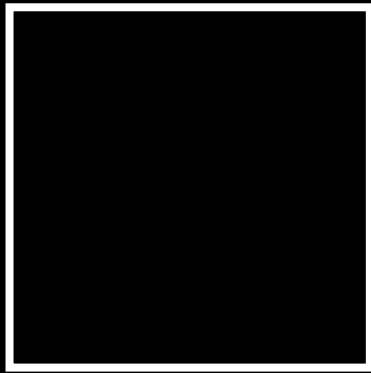


Prediction

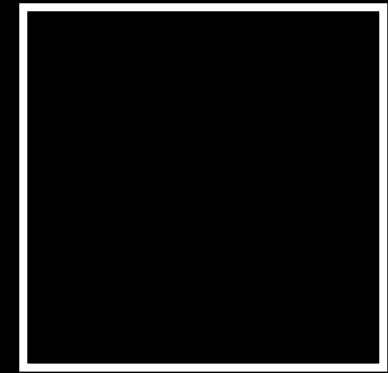
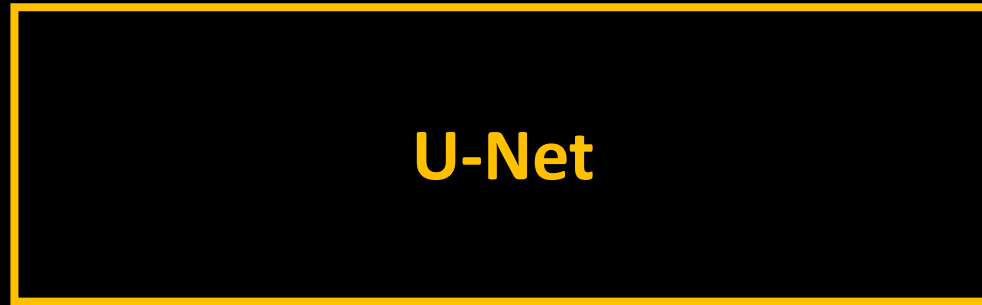


- Can be adapted to many tasks – for example this “**change detection**”, where it needed to learn *which kind of change* we care about

Third idea: U-Net



Images from
domain A



Images from
domain B

- Encoder-decoder model with skip connections

Third idea: U-Net



Images from
domain A



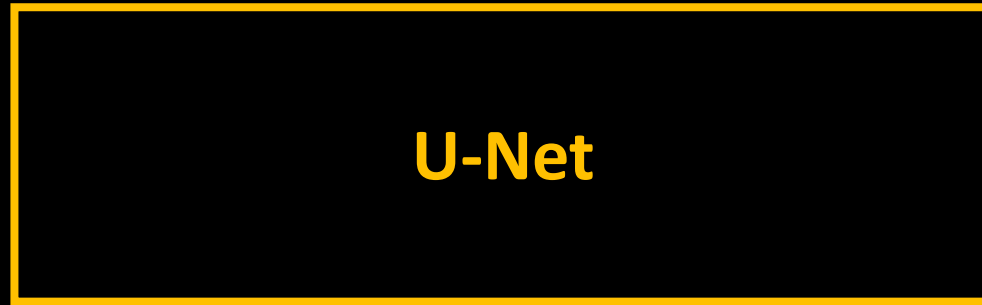
Images from
domain B

- Encoder-decoder model with skip connections

Third idea: U-Net



Images from
domain A



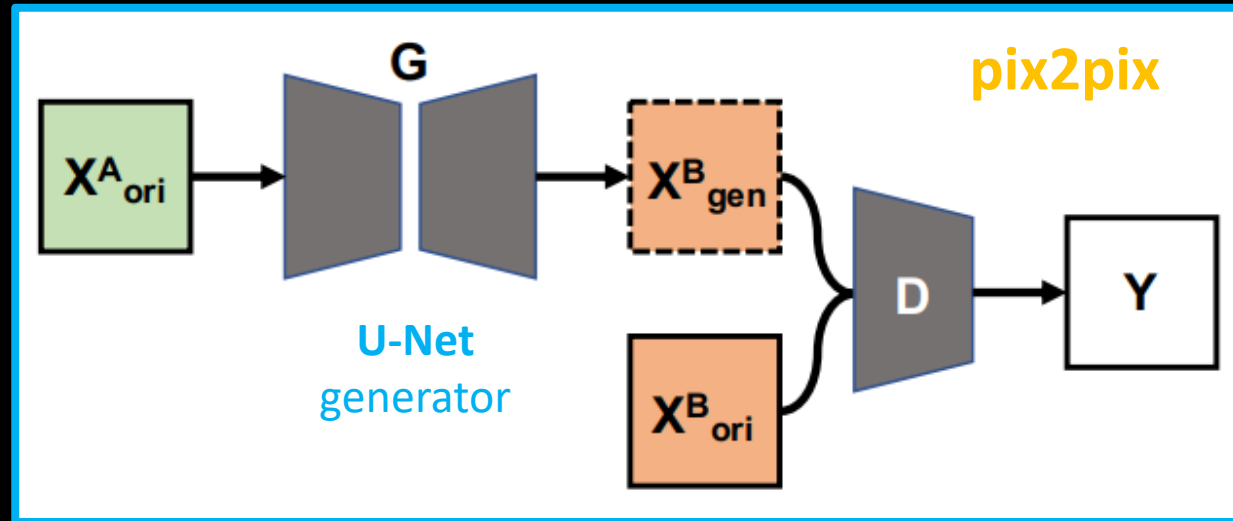
Images from
domain B

- Encoder-decoder model with skip connections
- Almost there! We can do even better ... *< Add adversarial training*

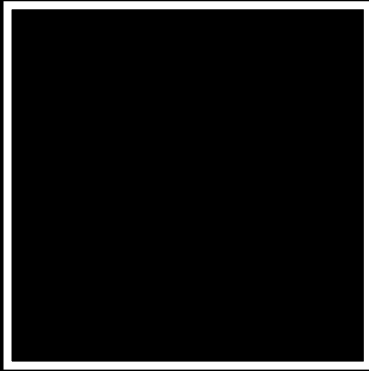
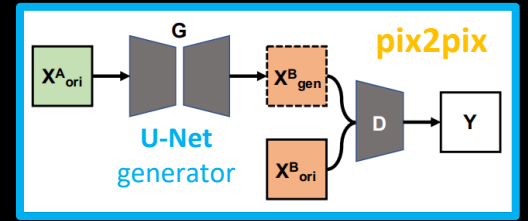
III pix2pix model

Pix2Pix

- Uses ideas from the Generative Adversarial Network (GAN) models
 - Generator is a U-Net network
 - With the **addition of the discriminator network and adversarial training**

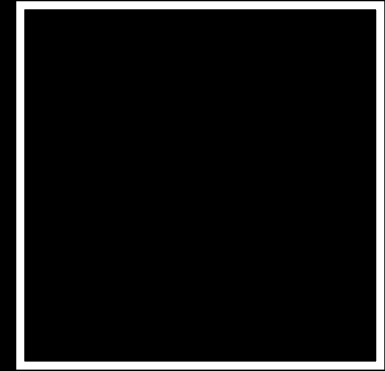


Final idea: pix2pix



Images from
domain A

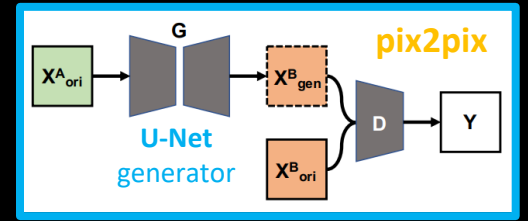
pix2pix (*the trained
generator network bit*)



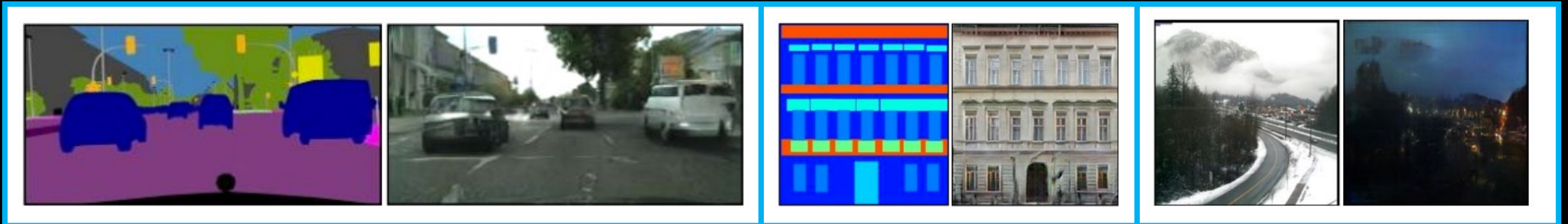
Images from
domain B

- Encoder-decoder model with skip connections
with adversarial training helping the quality of the generative model

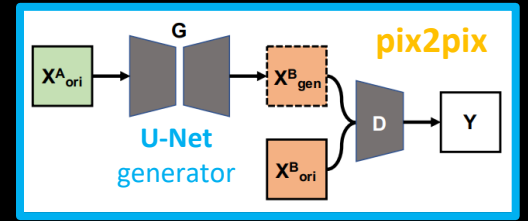
Pix2Pix



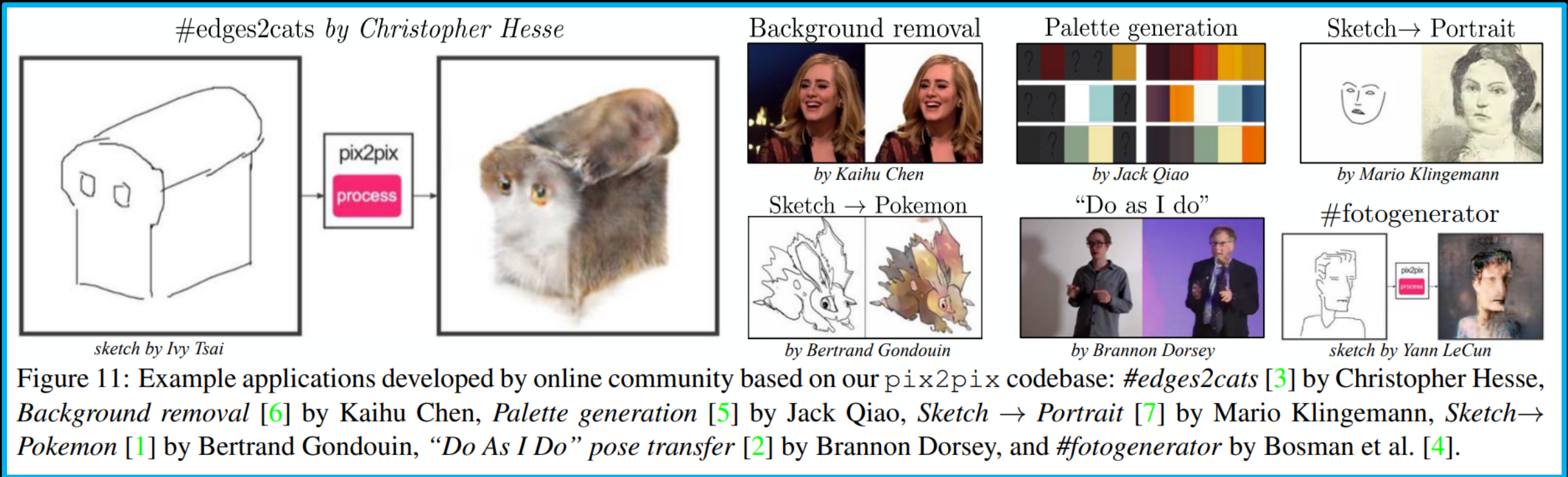
- General purpose **image-to-image translation**:
 - From their 2017 paper: “Many problems in image processing, computer graphics, and computer vision can be posed as “*translating*” an input image into a corresponding output image.”
 - Translating without specifically defining the rules – data-driven translating between two domains by showing **paired** examples:
 - [*Image* from A, Corresponding *Image* from B] * N samples



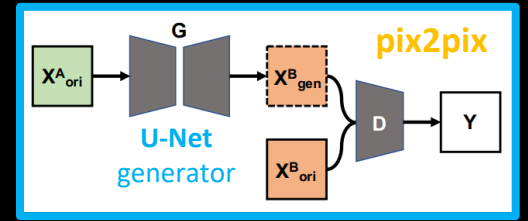
Pix2Pix



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

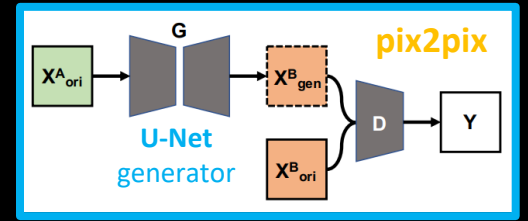


Pix2Pix limitations



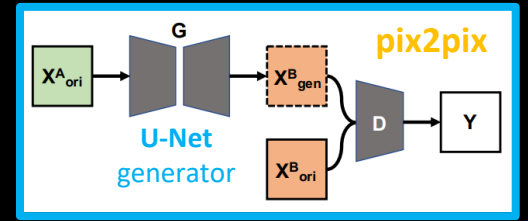
- For training it needs **paired data**
 - dataset of images from domain A, and dataset of corresponding images from domain B
 - Simple example “*horse 2 zebra*” model, would require careful arranging of animals

Pix2Pix limitations



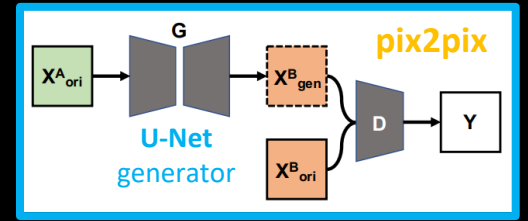
- For training it needs **paired data** \Rightarrow **CycleGAN** (next week)
 - dataset of images from domain A, and dataset of corresponding images from domain B
 - Simple example “*horse 2 zebra*” model, would require careful arranging of animals

Pix2Pix limitations



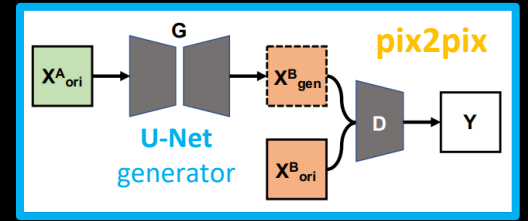
- For training it needs **paired data** \Rightarrow **CycleGAN** (next week)
 - dataset of images from domain A, and dataset of corresponding images from domain B
 - Simple example “*horse 2 zebra*” model, would require careful arranging of animals
- Basic pix2pix has relatively **low resolution**

Pix2Pix limitations



- For training it needs **paired data** \Rightarrow **CycleGAN** (next week)
 - dataset of images from domain A, and dataset of corresponding images from domain B
 - Simple example “*horse 2 zebra*” model, would require careful arranging of animals
- Basic pix2pix has relatively **low resolution** \Rightarrow **pix2pixHD**

Pix2Pix limitations



- For training it needs **paired data** \Rightarrow **CycleGAN** (next week)
 - dataset of images from domain A, and dataset of corresponding images from domain B
 - Simple example “*horse 2 zebra*” model, would require careful arranging of animals
- Basic pix2pix has relatively **low resolution** \Rightarrow **pix2pixHD**
- For animation, we want something to enforce **temporal consistency** between consecutive frames

IV uses and follow-up work

Follow-up papers?

- **Pix2Pix HD (2018)**

- Allows **higher resolution** of the trained pictures (from 256x256px up to 2048x1024px!)

Code: github.com/NVIDIA/pix2pixHD

Example project:

GAN Theft Auto: Autonomous Texturing of Procedurally Generated Interactive Cities

Oscar Dadfar
Carnegie Mellon University
Pittsburgh, USA
odadfar@andrew.cmu.edu

Lingdong Huang
Carnegie Mellon University
Pittsburgh, USA
lingdonh@andrew.cmu.edu

Hizal Çelik
Carnegie Mellon University
Pittsburgh, USA
hizalc@andrew.cmu.edu



Figure 1: Cross-views of Cityscape [Left] and GTA V [Right] texturings using Pix2PixHD.

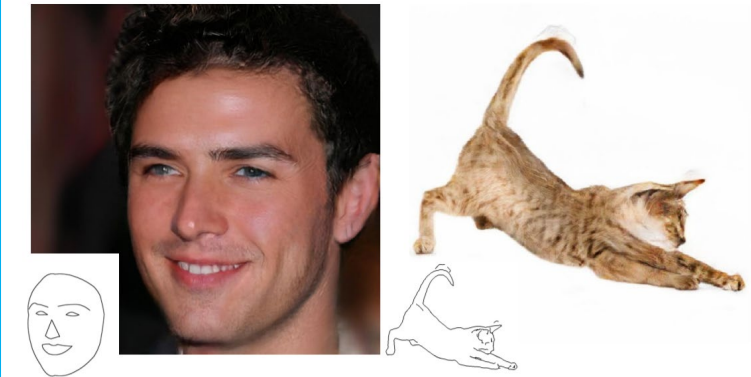
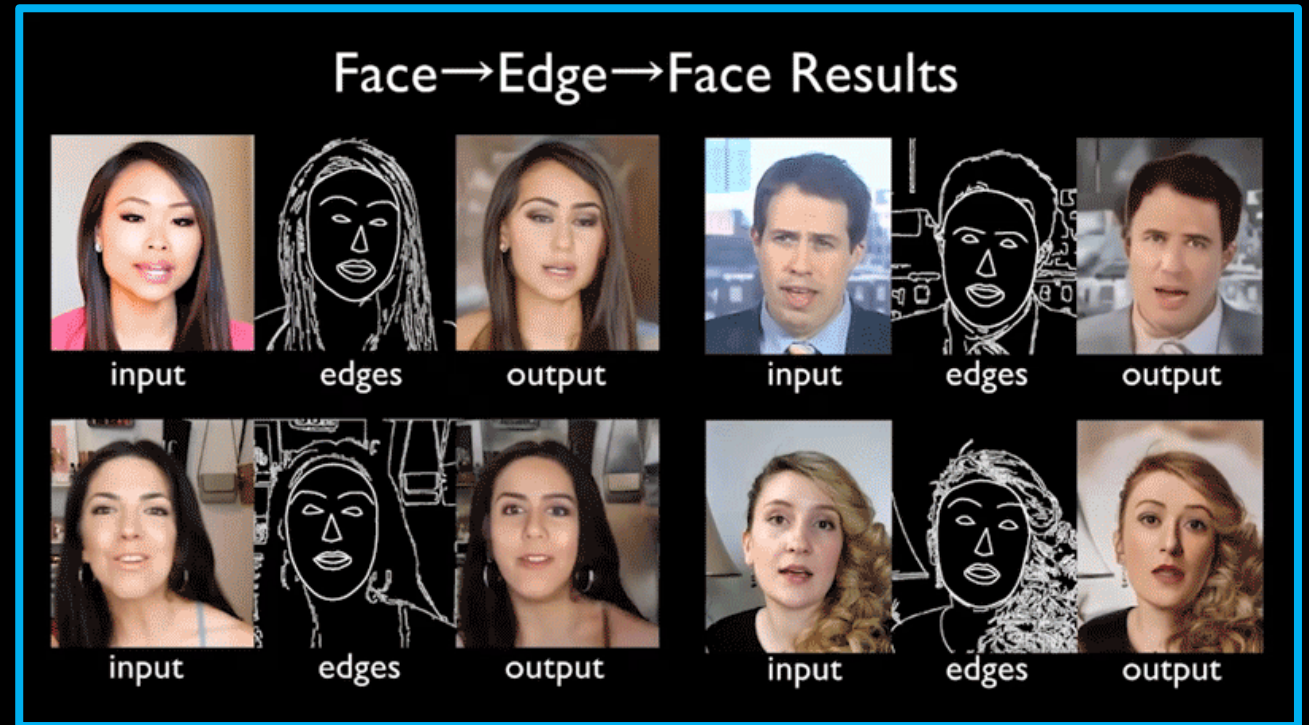


Figure 2: Example results of using our framework for translating edges to high-resolution natural photos, using CelebA-HQ [26] and internet cat images.

Follow-up papers?

- **Vid2Vid (2018)**

- Keep **temporal consistency** between frames



Code: github.com/NVIDIA/vid2vid

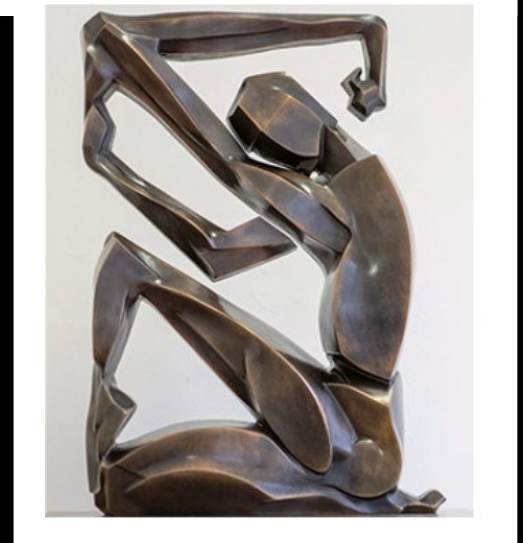
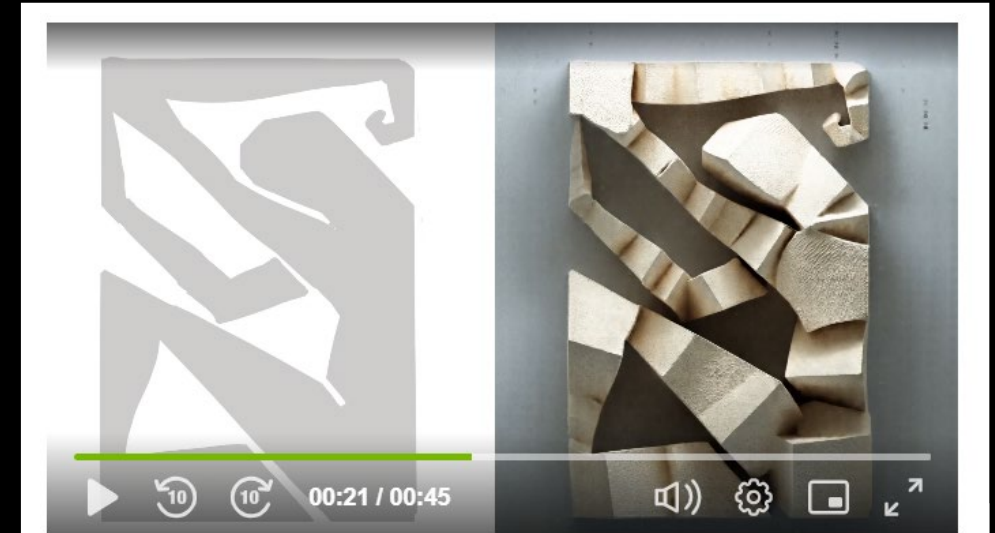
Follow-up papers?

- **SPADE** and the interactive demo **GauGAN (2019)**
 - **Interactive** painting tool based on pix2pix like models



Code: github.com/NVlabs/SPADE

More high-quality examples:



Summary from the lecture

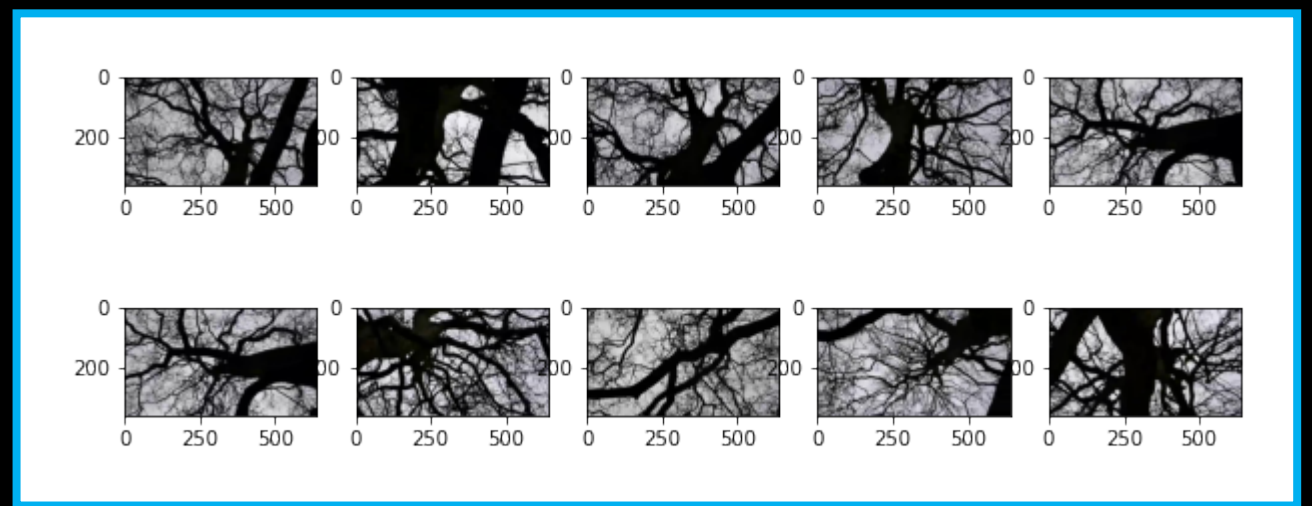
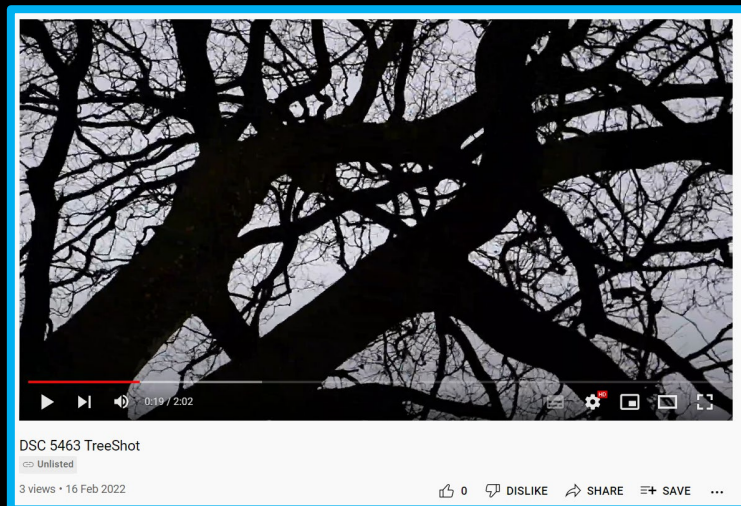
- **Combining ideas from last classes**, we can achieve image2image models:
 - **Convolutional layers** (working with images)
 - **Encoder decoder models** (split small details and large context)
 - **U-Nets: Skip connections** (allow information to keep at the original resolution)
 - **Adversarial training** (more realistic images)

Links and additional readings:

- **Online pix2pix demos:** affinelayer.com/pixsrv/
 - **Follow-up papers:** [pix2pixHD](#) (with high. res.), [vid2vid](#) (with frame to frame consistency), [SPADE](#) (interactive), ... (*and many more...*)
- **Bonus readings:**
 - About **Pix2Pix** on ML4A – [blog with code](#)

Preparation for the practical

- Think about a **dataset of images** you'd like to use when training pix2pix model.
- We will use a Colab code which will download a YouTube video and extract frames from it – this will serve as our dataset
- **Task: choose a video!**



End of the lecture

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

AI for the Media

Week 6, Domain 2 Domain

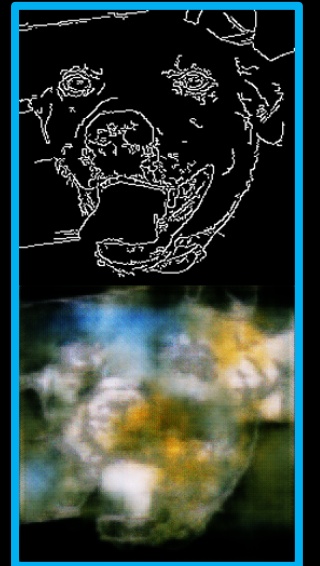
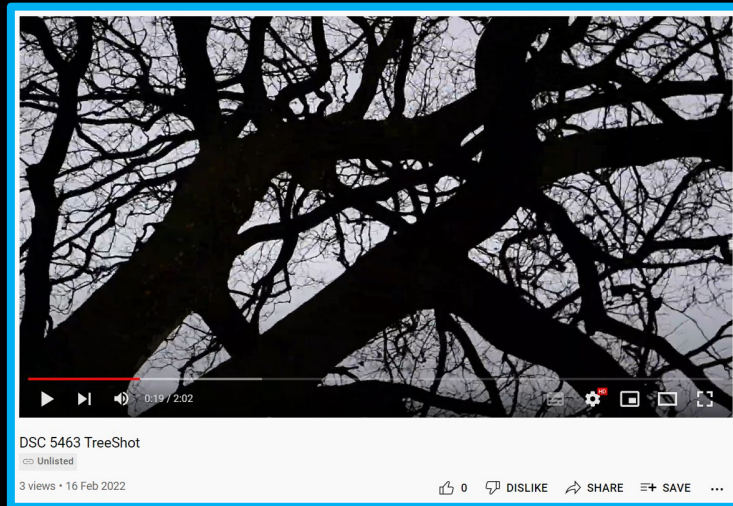
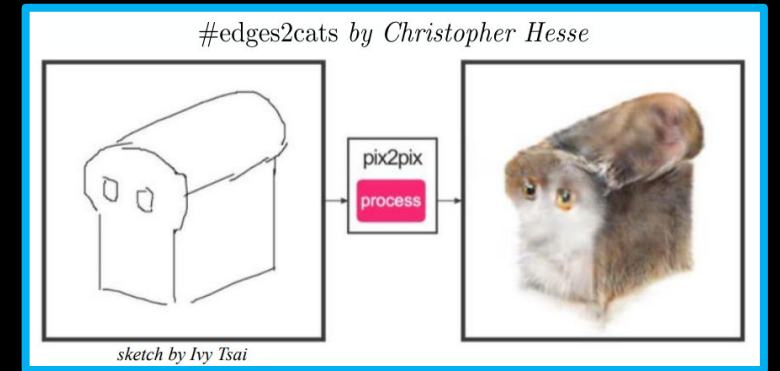


Practical: pix2pix training

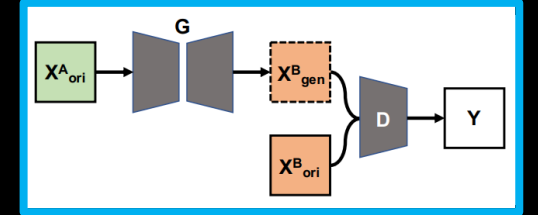
Practical: Domain 2 Domain

Training Pix2Pix models

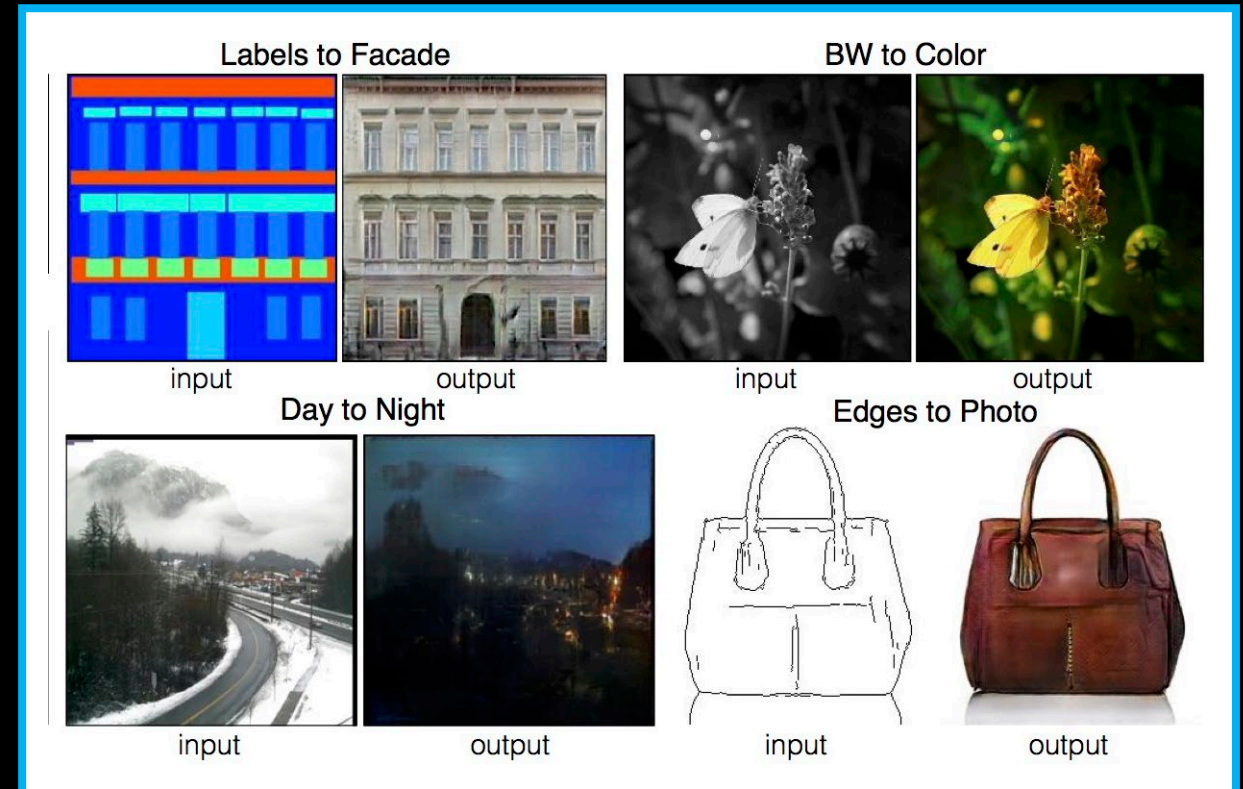
Namely using the sketch2image with our own data.



Pix2Pix

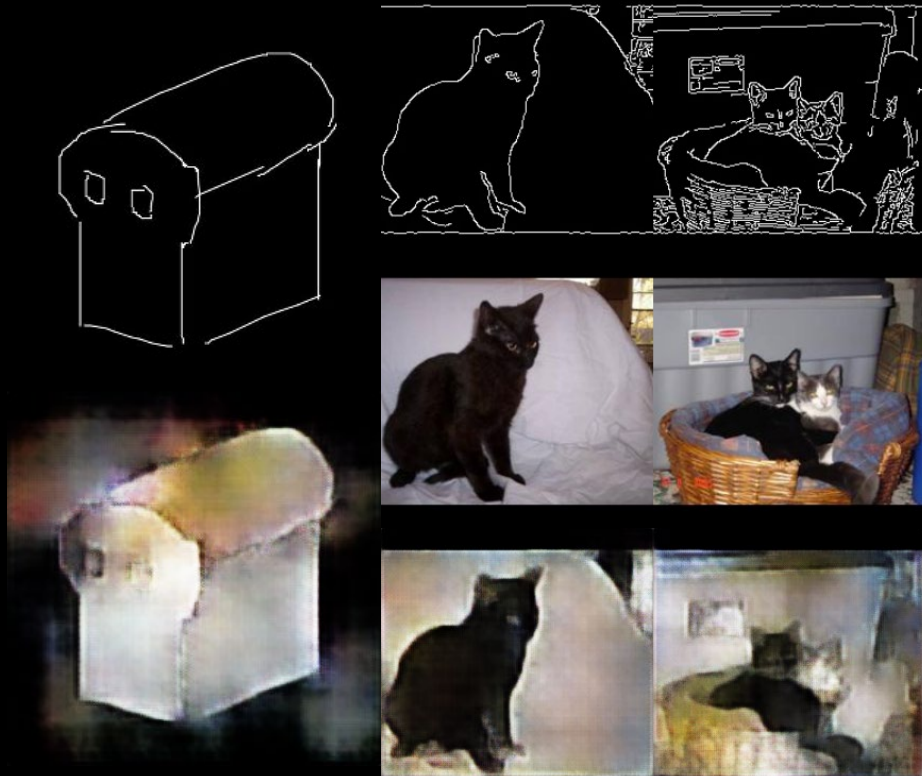


- You might recall that we will need **paired data** in our dataset:
 - Data from domain **A**
 - Data from domain **B**
- And we will be learning the **translation between A to B**



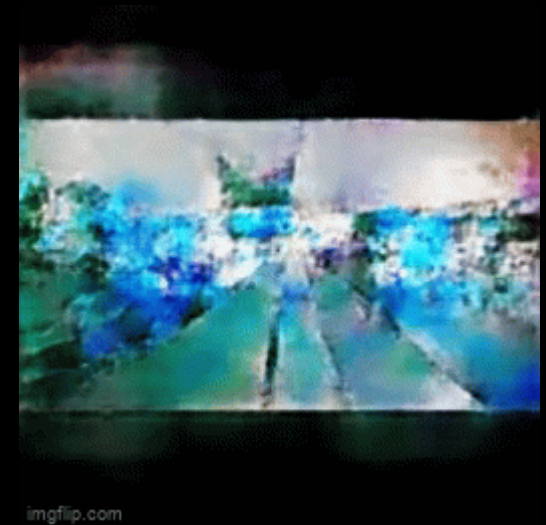
Examples from training:

- **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 no temporal consistency between these frames ...
- Longer video: [here](#)



Three links

Starter code:

Opens directly in Colab

[w06_pix2pix_keras_student_starter_code.ipynb](#)

Interaction

Copy your results / intermediate images here:

docs.google.com/document/d/1lF8Q2rgNjUg5Mq_6jKQRJBISP7Cm6o_lDrYdJeAuLH8/edit?usp=sharing

Feedback In like 1 min forms.gle/EduNP2KJnuVRhhih9

The end