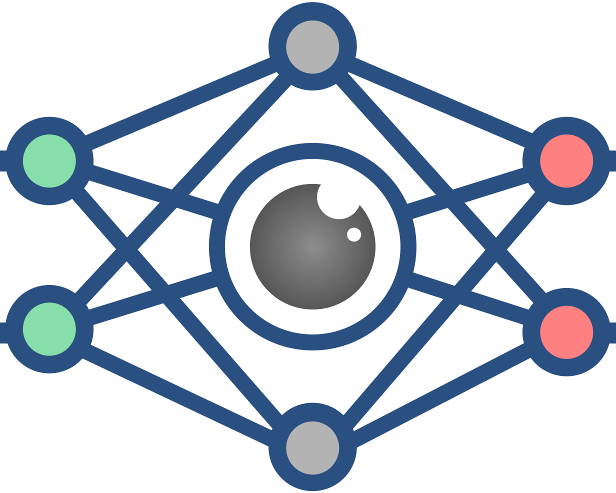
CS3485

Deep Learning for Computer Vision



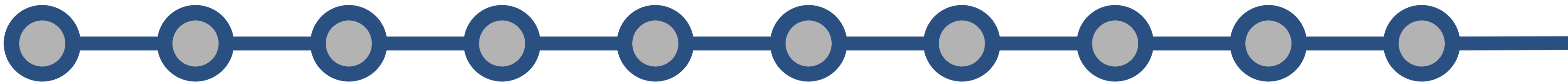
Lec 15: Autoencoders

Announcements

- Lab 5 due tomorrow at midnight.
- Lab 6 will be posted tomorrow and will be due next Tuesday.

(Tentative) Lecture Roadmap

Basics of Deep Learning



Intro to Computer Vision

Linear Classifiers and Perceptron

Multilayer Perceptron and Intro to Deep Learning

Optimization and Regularization

Pytorch

Convolutional Neural Networks

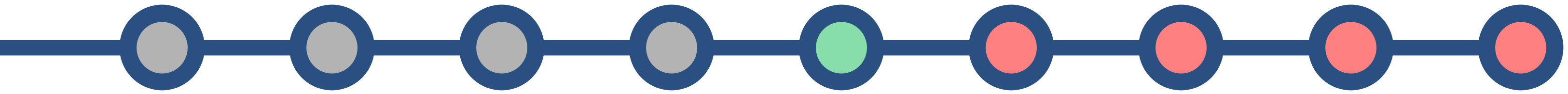
Data Augmentation and Deep CNNs

Transfer Learning and Residual Nets

Inception Net and what CNNs learn

Adversarial Examples and Self-supervision

Computer Vision Tasks



Intro to Object Detection

Fast Object Detection

Intro to Image Segmentation

Applications of Detection and Segmentation

Autoencoders

Image Generation with GANs

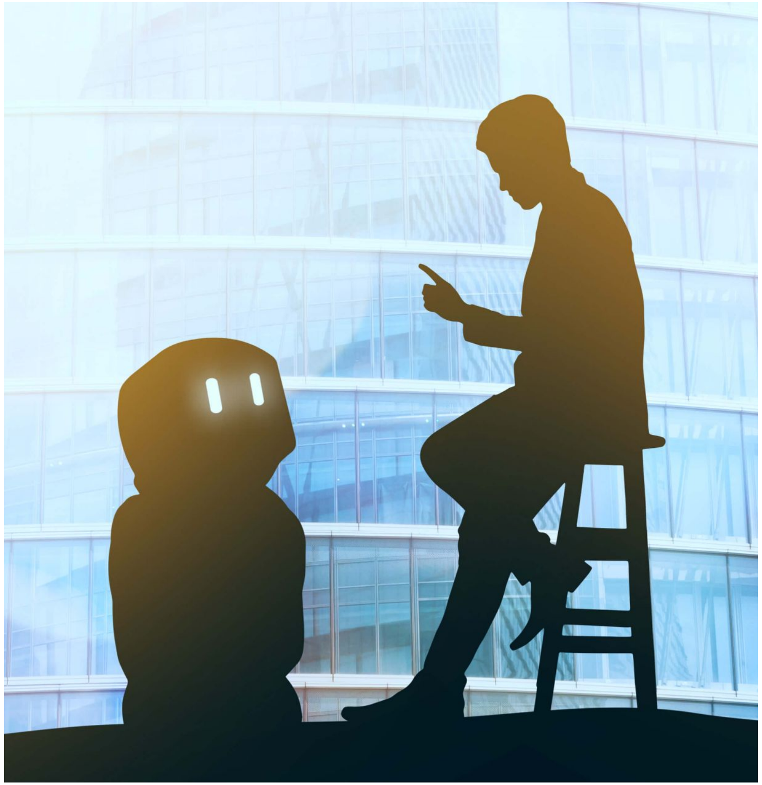
Advanced GANs

Transformers and ChatGPT

Image Generation by Prompt

Unsupervised Problems

- So far, we saw many applications of **supervised learning** in Deep Learning to solve many Computer Vision problems.
- It means that if we needed ground truth data in that pipeline, i.e.:
  - True classification labels of an image,
  - Bounding Boxes for each object in an image,
  - Segmentation masks,
  - Annotated Keypoints,
  - Eye tracking data, etc.
- Now, in the next lectures, we'll turn to **unsupervised learning** tasks and the Deep Learning approaches to them.



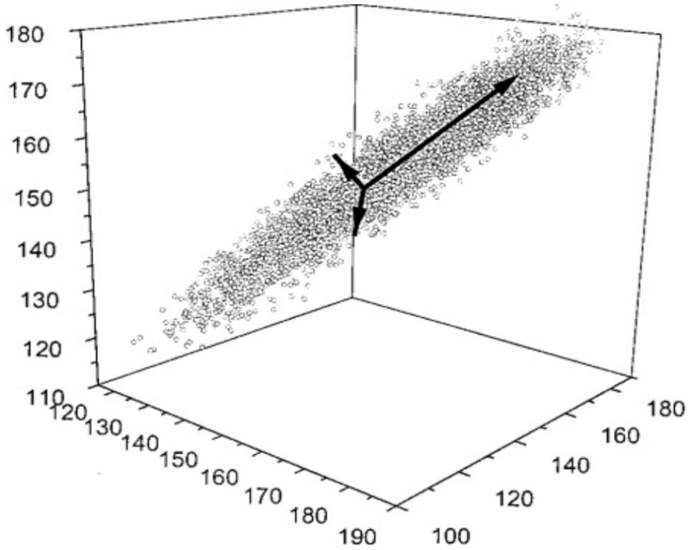
Dimensionality Reduction and PCA

- The first unsupervised task we'll see is **Dimensionality Reduction**:

  Dimensionality reduction is the task of finding a transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data.

- In sum, you want to bring a high dimensional complex data into a low dimensional space, losing the **least possible amount of information**.
- This is the goal of Principal Component Analysis (PCA), for example.

Principal Components of a Dataset

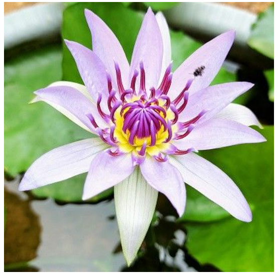


Notice that we don't need 3 dimensions for the dataset above, 2 would suffice.

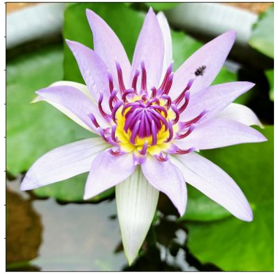
Applications of Dimensionality Reduction

- Dimensionality reduction is core to Machine Learning because we can use it in many applications, such as:
  - **Data compression**: compressing images is very important in order to send them via environments with limited connection/bandwith.
  - **Visualization of high-dimensional data**: Humans can only fully visualize data in at most 3 dimensions, hence the need to reduce its dimensionality for visualization.
  - **Data simplification for before training**: the higher the data dimensionals, the more computation it will demand for learning it. Lowering its dimension may help.
  - **Noise or spurious information removal**: Since the reduction only removes the useless information in the data, it then also removes noise from it.

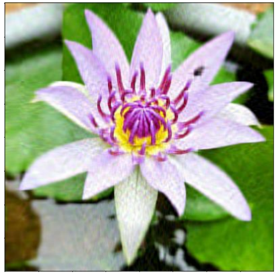
Original Image



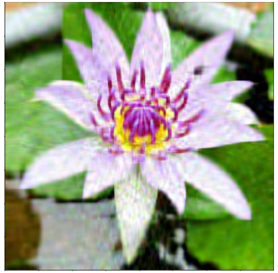
400 PCA Dim.



40 PCA Dim.

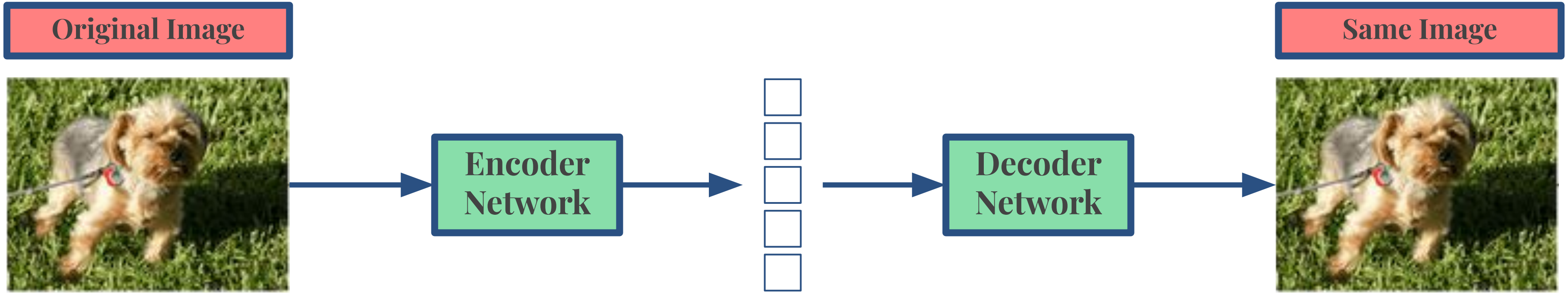


25 PCA Dim.



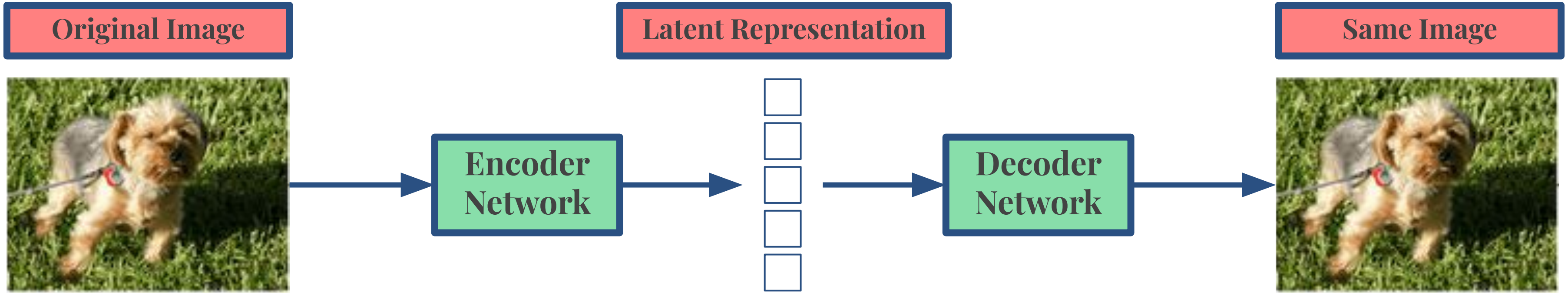
Dimensionality Reduction and Autoencoders

- A simple way to do dimensionality reduction using Deep Learning is via **Autoencoders**.
- All autoencoders are composed of two neural networks:
  - **Encoder network**: it translates the image into the lower-dimensional vector.
  - **Decoder network**: reconstructs the original image from the encoded lower-dimensional vector.
- To train an autoencoder, we pass each image in our dataset as input to the encoder, and try to reproduce the same image as output from the decoder.



Dimensionality Reduction and Autoencoders

- The lower dimensional vector is usually called **latent representation** (“*lantent*” = “*not visible, hidden*”) of the input image.
- Through this training process we hope that images that are **visually similar** will have similar encoding/latent representation.
- This way, we hope that the latent representation will only contain the visual features necessary to recreate the original image in the most faithful way.



Original Image

Latent Representation

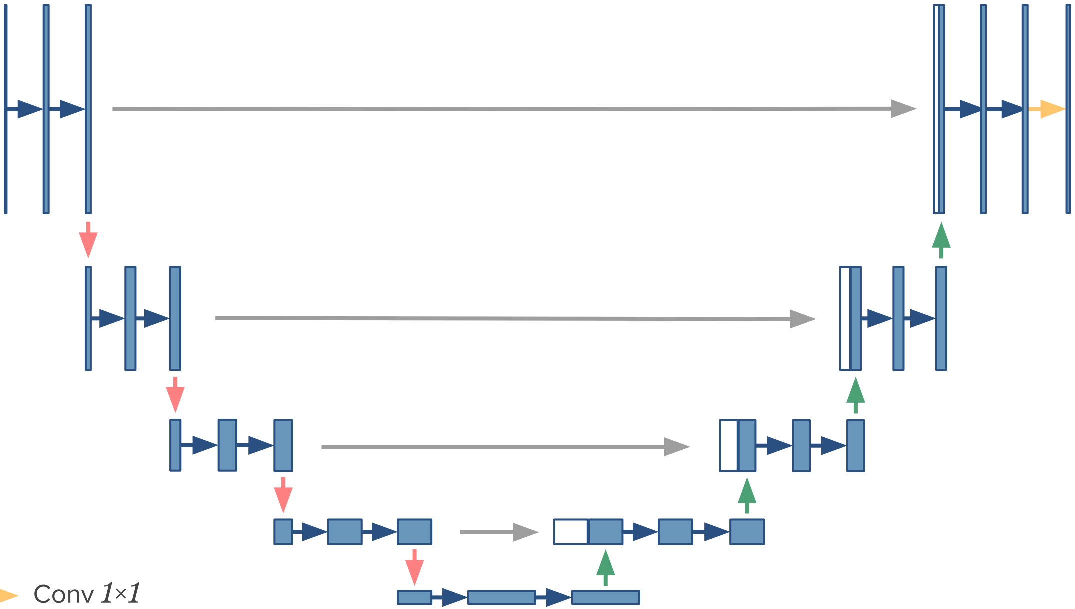
Same Image

Encoder Network

Decoder Network

Autoencoders and the U-Net

- We already saw a similar process taking place with the U-Net, where an image has its size reduced to a latent feature map and then it its enlarged to a segmentation map.
- Notice, however, that the latent feature map is not necessarily smaller in overall dimension, as it has many channels.
- It also presents the skip connections that are usually absent in auto encoders.



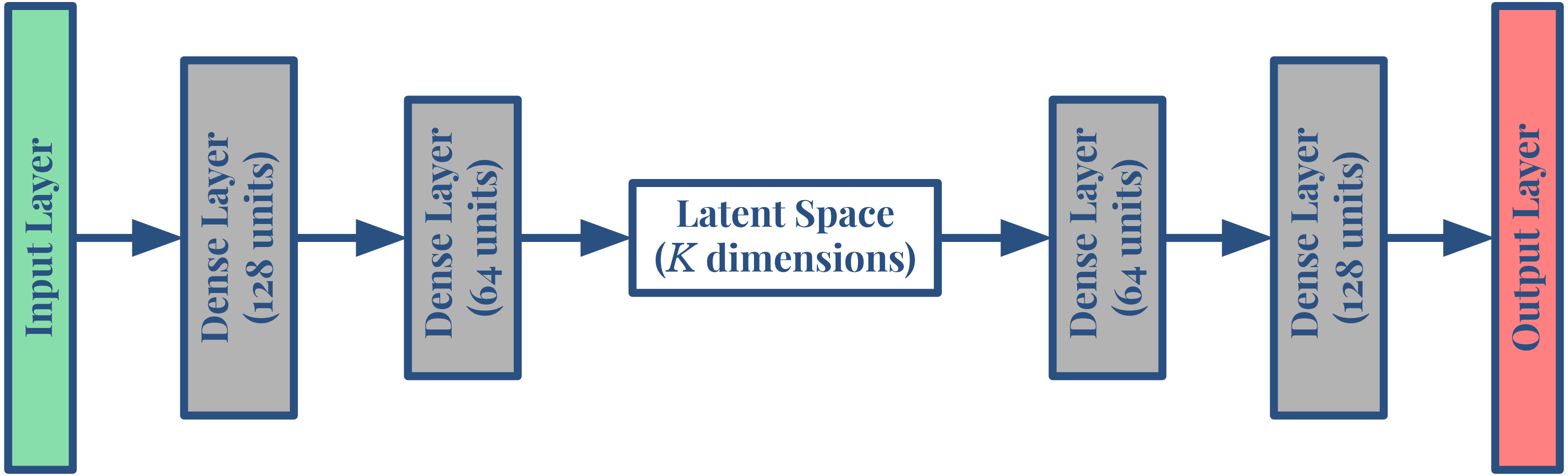
Legend:

→ Conv 3×3 + ReLU → Max-pool 2×2

→ Concat. → Transp. Conv 2×2 + ReLU → Conv 1×1

Autoencoders

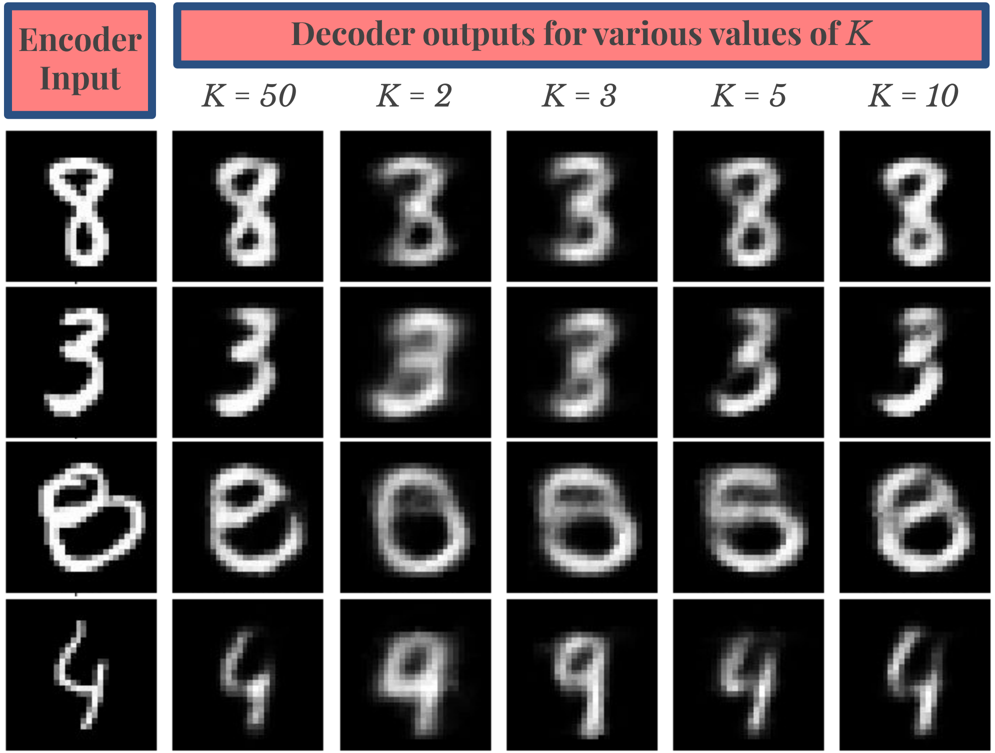
- Although we are not using the U-Net model here, we'll keep some of its principles.
- For now, we'll only use Multilayer Perceptrons for the Encoder and Decoder Networks:



- As you can picture from the above diagram, the latent representation block is also called the **bottleneck** of the Autoencoder.

Autoencoders on MNIST

- We train this network on MNIST images for different values latent space dimensions K.
- Then we see the output of that trained network on the test data (which was not seen in training).
- Note that we fairly recover the original image even with low K, demonstrating the **compressing power** of autoencoders.
- But as K increases, the clarity of the decoded image improves.



Exercise (*in pairs*)

- An important task in Computer Vision is that of **Image Search**, which given a query image, one wants to find the most similar image in a dataset of images. How can Autoencoders be applied in that context? What would be the advantage(s) of using them?
- How can you make the previous MLP-based autoencoder using CNNs instead? Which types of layers you'd use in that case?
- How can you use CNN-based autoencoders as in a self supervised framework?

Image Manipulation

- We learned that Autoencoders are helpful to the problem of Dimensionality Reduction.
- Now, we'll see how Autoencoders can be used to perform **Image Manipulation**:

In Image Manipulation we aim at change the color content of the input image in order to solve specific visual problems with it.

- The board definition of image manipulation will give rise to important tasks in computer vision, such as image denoising and inpainting.

Image Denoising

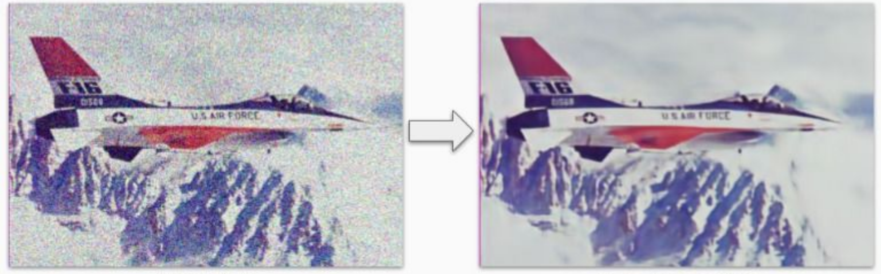
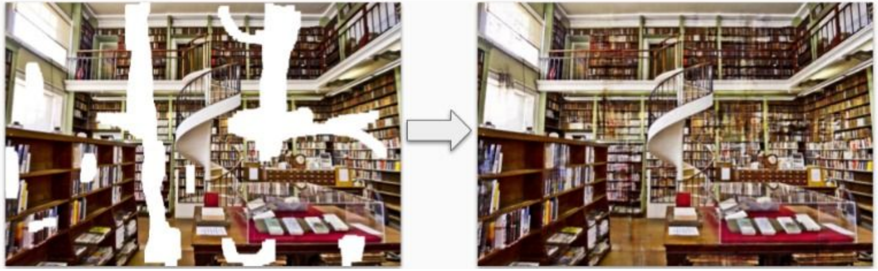


Image Inpainting



Main template for solvers

- We saw that, just like the U-Net, autoencoders transform an image sized input into an image-sized output, and they do that using two (**convolutional,** in practice) nets:
  - An encoder that transforms an input image into a low dimensional vector (the latent representation).
  - A decoder that transforms the low dimensional vector back to the image size.
- The beauty of using this strategy is that all the following computer vision tasks can be solved using this same template, but for different data formats.
- Note that th this same process could also be accomplished using a U-Net, for example.

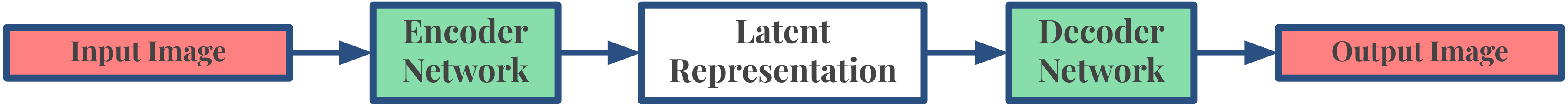
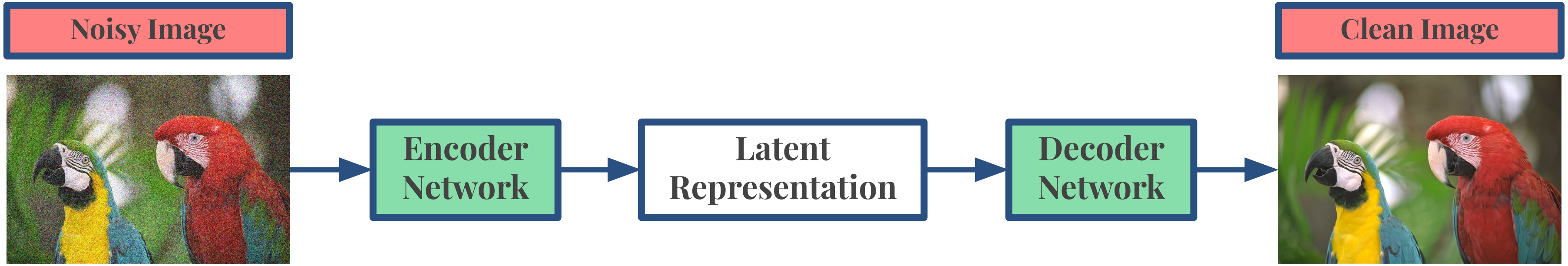


Image Denoising

- A task that is usually solved using autoencoders is **Image Denoising**:

  Image Denoising is the task of removing random color artifacts, called noise, from an image in order to make it clearer.

- The image noise is random variation of color information in images and it is an undesirable by-product of image capture that obscures the desired information.



Noisy Image

Encoder Network

Latent Representation

Decoder Network

Clean Image

Image Inpainting

- The next task that can be solved using the autoencoder strategy is **Image Inpainting**:

  Image Inpainting is a task of reconstructing missing regions in an image.

- The missing pixel values can be due to:
  - Defects in the image (maybe it was accidentally cropped or ripped),
  - Are the result of occlusions caused by an object,
  - Watermarks, legends and other texts that are placed on the image.

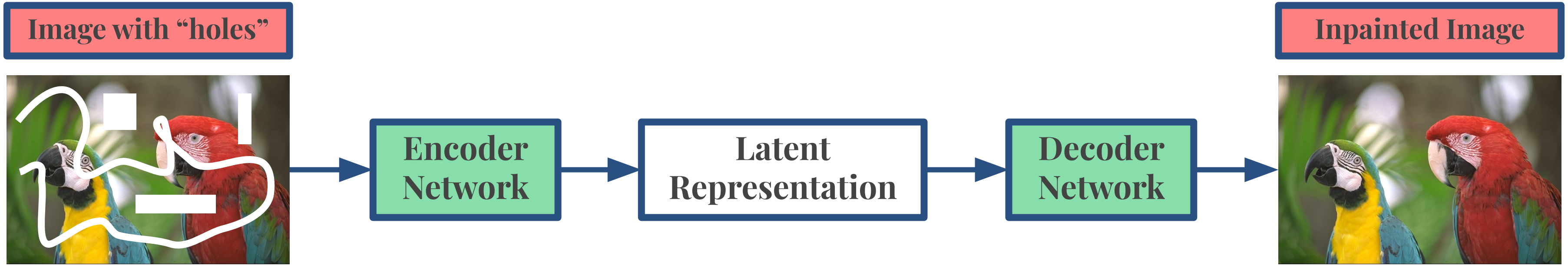


Image with "holes"

Encoder Network

Latent Representation

Decoder Network

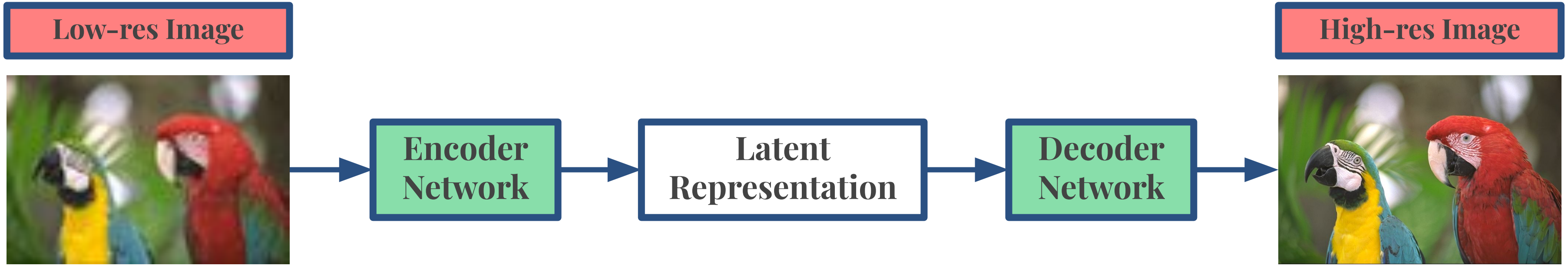
Inpainted Image

Image Super Resolution

- Another one is **Image Super Resolution**:

  Image Super Resolution is the task that aims to turn a small image (low resolution) to a large (high resolution) image by adding details that were missing.

- Many cameras (specially the old ones) can one capture images a relatively low amount of pixels and that can be visually unpleasant in modern high resolution screens.



Low-res Image

Encoder Network

Latent Representation

Decoder Network

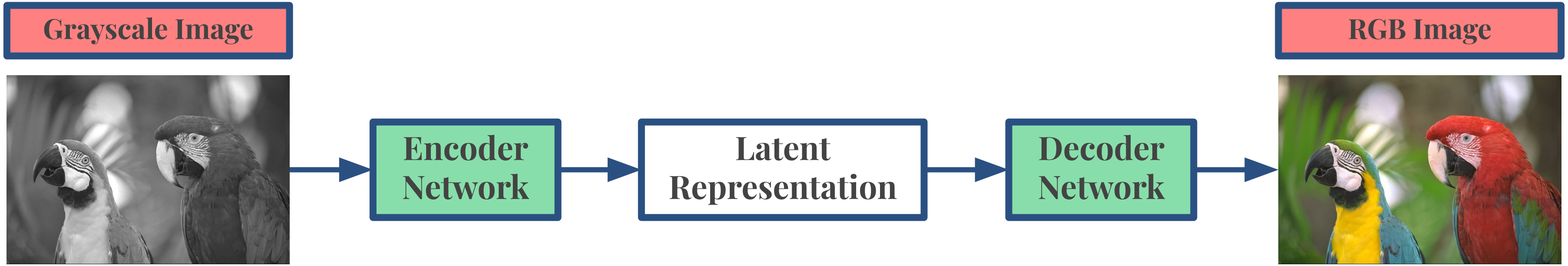
High-res Image

Image Colorization

- The final example here is **Image Colorization**:

  Image Colorization is the task that is concerned in automatically colorizing grayscale images.

- Images (and videos) used to be captured using a technology that wasn't able to capture colors. Just like in super resolution, colorizing images also makes them more pleasing.



Grayscale Image

Encoder Network

Latent Representation

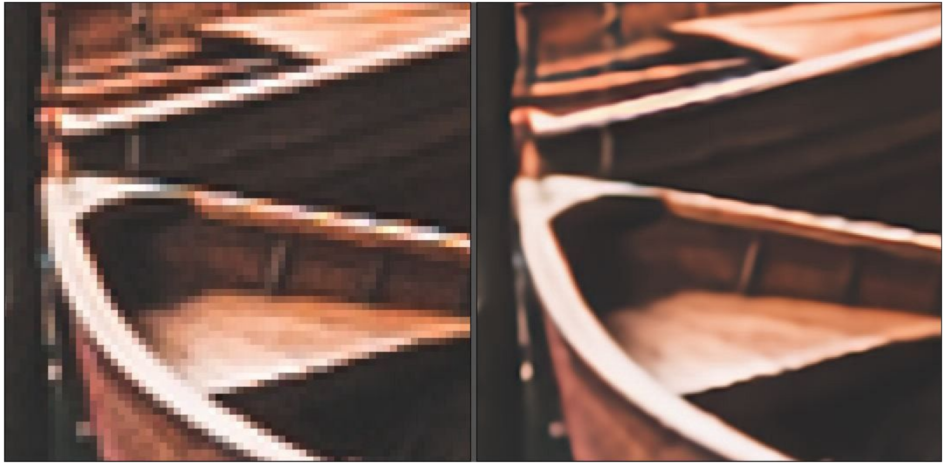
Decoder Network

RGB Image

Applications of denoising and super resolution

- Super resolution and image denoising are usually paired up in many industrial applications that involve the broader task of **image enhancement***

Image Depixelation



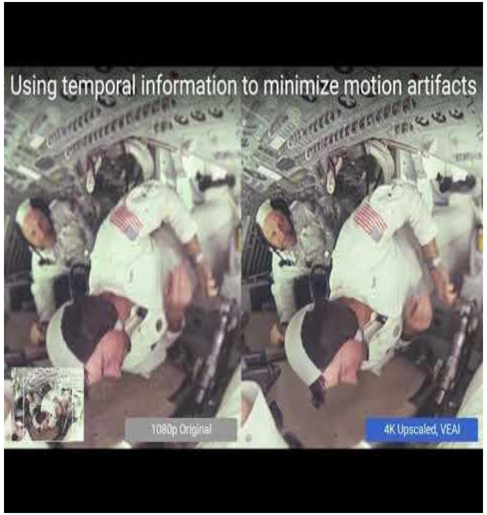
(True) Image and Video Zooming**

* Other deep learning models other than autoencoders can used to solve these tasks. ** Maybe one day like this, but we are not there yet.

Applications of denoising and super resolution

- Super resolution and image denoising are usually paired up in many industrial applications that involve the broader task of **image enhancement**.

Video Remasteration



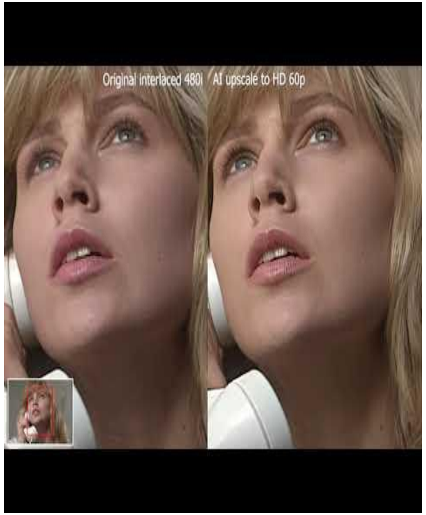
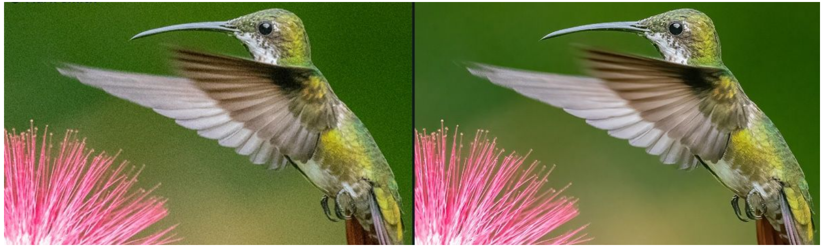


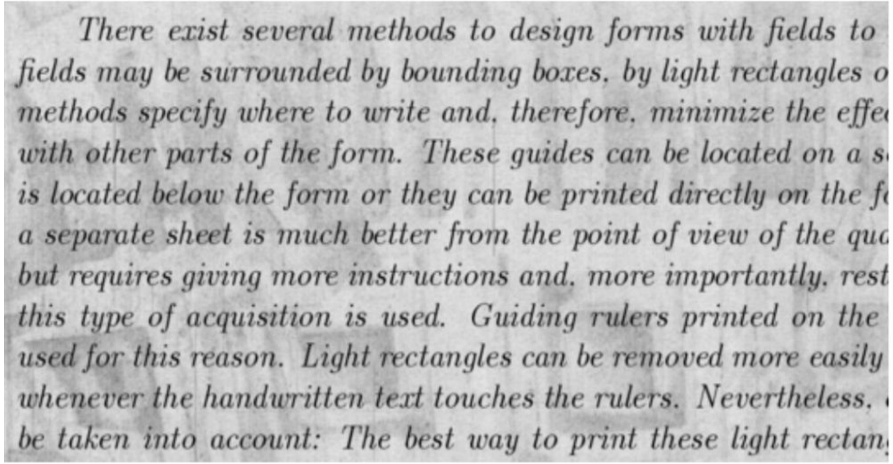
Image and Video Noise Removal



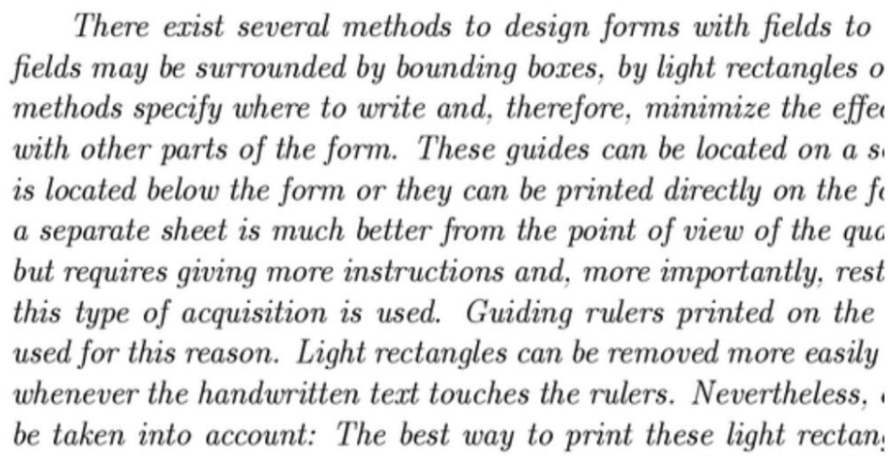
Applications of denoising and super resolution

- Super resolution and image denoising are usually paired up in many industrial applications that involve the broader task of **image enhancement**.

Background removal for scanned for Text recognition



There exist several methods to design forms with fields to
fields may be surrounded by bounding boxes, by light rectangles o
methods specify where to write and, therefore, minimize the effe
with other parts of the form. These guides can be located on a s
is located below the form or they can be printed directly on the f
a separate sheet is much better from the point of view of the qua
but requires giving more instructions and, more importantly, rest
this type of acquisition is used. Guiding rulers printed on the
used for this reason. Light rectangles can be removed more easily
whenever the handwritten text touches the rulers. Nevertheless,
be taken into account: The best way to print these light rectan



There exist several methods to design forms with fields to
fields may be surrounded by bounding boxes, by light rectangles o
methods specify where to write and, therefore, minimize the effe
with other parts of the form. These guides can be located on a s
is located below the form or they can be printed directly on the f
a separate sheet is much better from the point of view of the qua
but requires giving more instructions and, more importantly, rest
this type of acquisition is used. Guiding rulers printed on the
used for this reason. Light rectangles can be removed more easily
whenever the handwritten text touches the rulers. Nevertheless,
be taken into account: The best way to print these light rectan

Applications of image inpainting

- Image inpainting also plays an important role in industrial image manipulation, especially when linked to photography.

Photo and art restoration

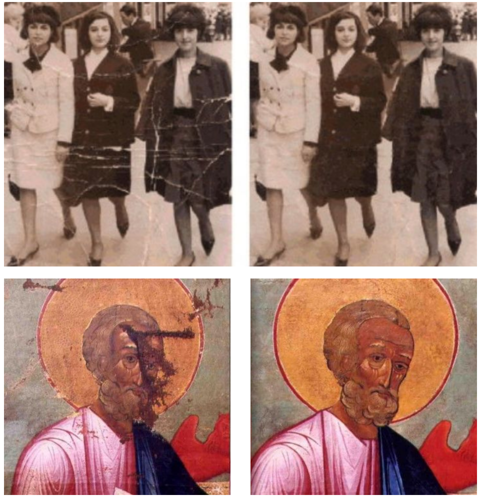
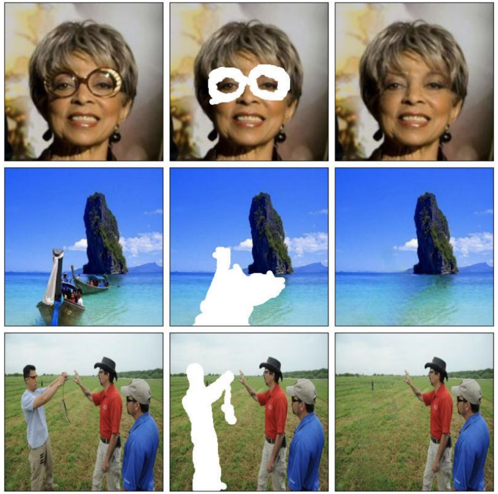


Photo Editing



Exercises (*in pairs*)

- How can we frame all these methods as self-supervised? How can one create the data for training each one of these architectures? *For example*: for image denoising, once can artificially add artifacts to the image in their dataset and then try to predict the clean ones from the noisy ones.

Video: DeepFakes