DRAG
Drone and Robotics Aziz Group

Generative Deep Learning

Naeemullah Khan
naeemullah.khan@kaust.edu.sa

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

KAUST Academy
King Abdullah University of Science and Technology

November 15, 2022

Intro duction

- Can you imagine an image of dog?
- Now, can you imagine an image of that dog driving a car?
- How are we able to this?
- We excel at extracting knowledge from the data we observe and perform complex reasoning based on it

Introduction (cont.)

How can we generate such images/data through AI

What we would need? Let's think about it step by step.

- Some prior data
- A model which will "learn" the data
- A method through which the model will learn
- A method to generate new data from the trained model

This what generative deep learning is all about and what we will be covering in this course.

Introduction (cont.)

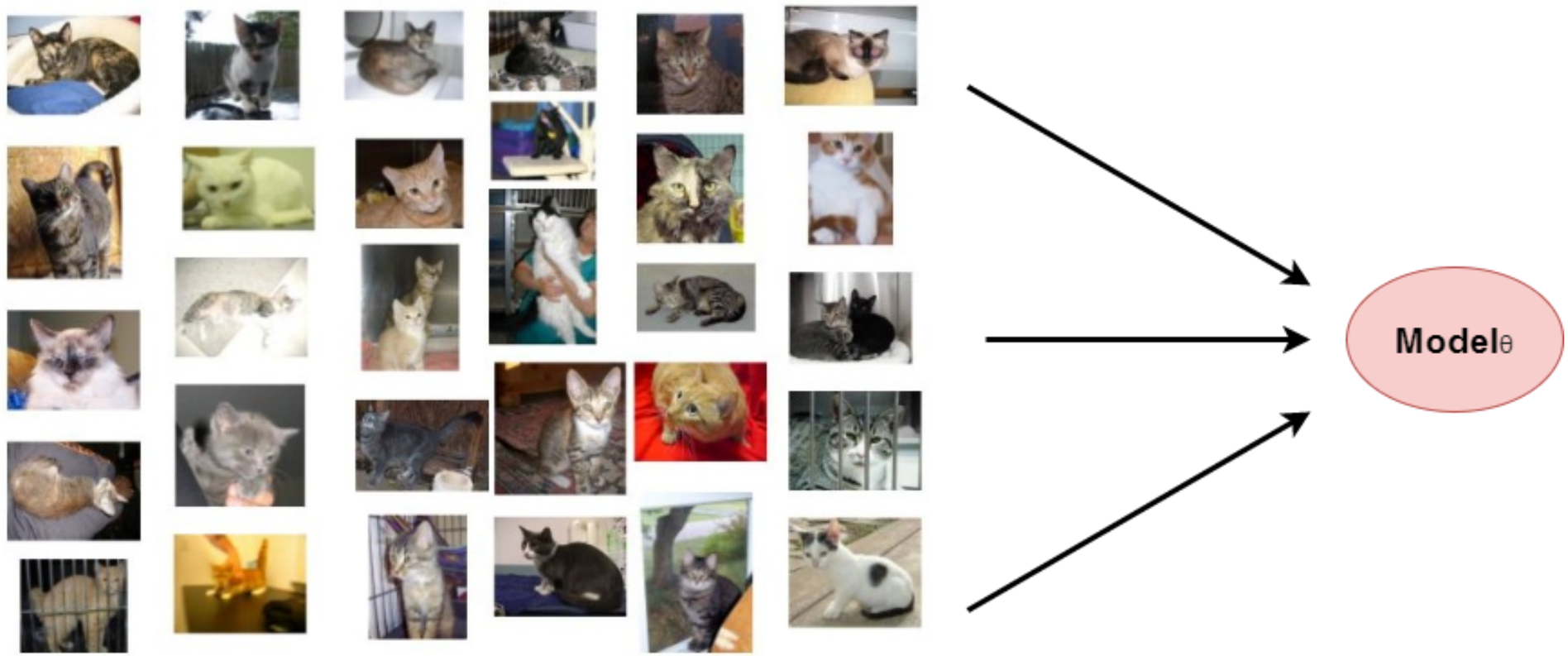

Figure 2: The model learns the cat images data

Introduction (cont.)

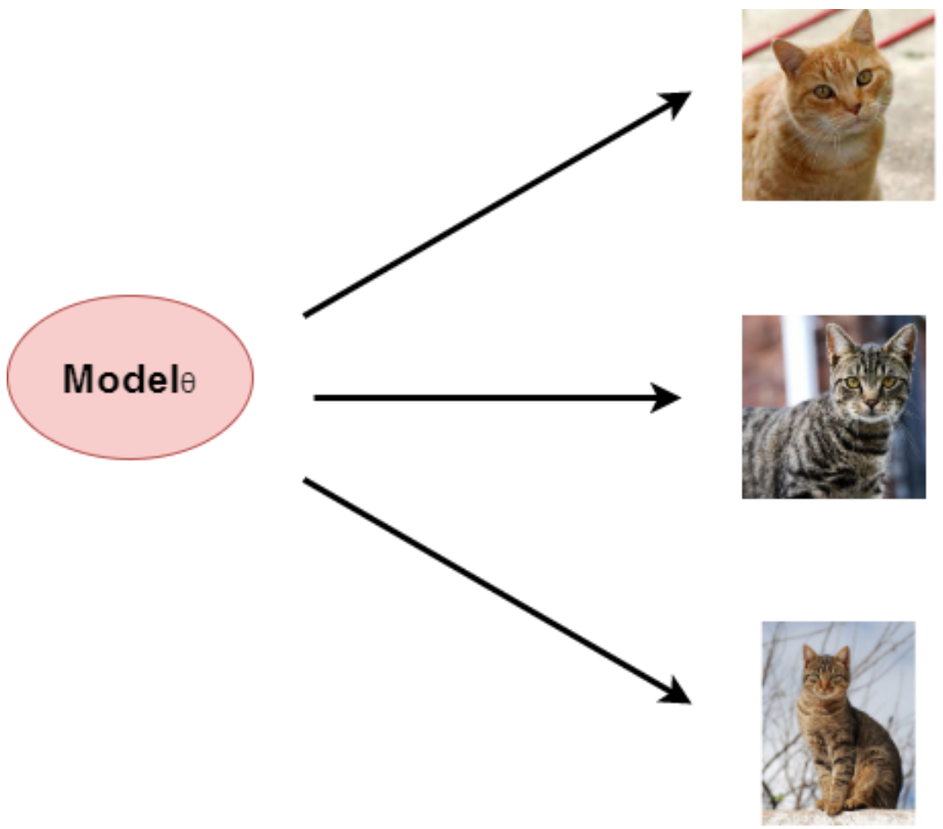

Figure 3: The trained model generates new cat images

Progress on Face Generation

Figure 4: Progress in face generation over the years1

[1] https://twitter.com/tamaybes/status/1450873331054383104

Completing Incomplete Content

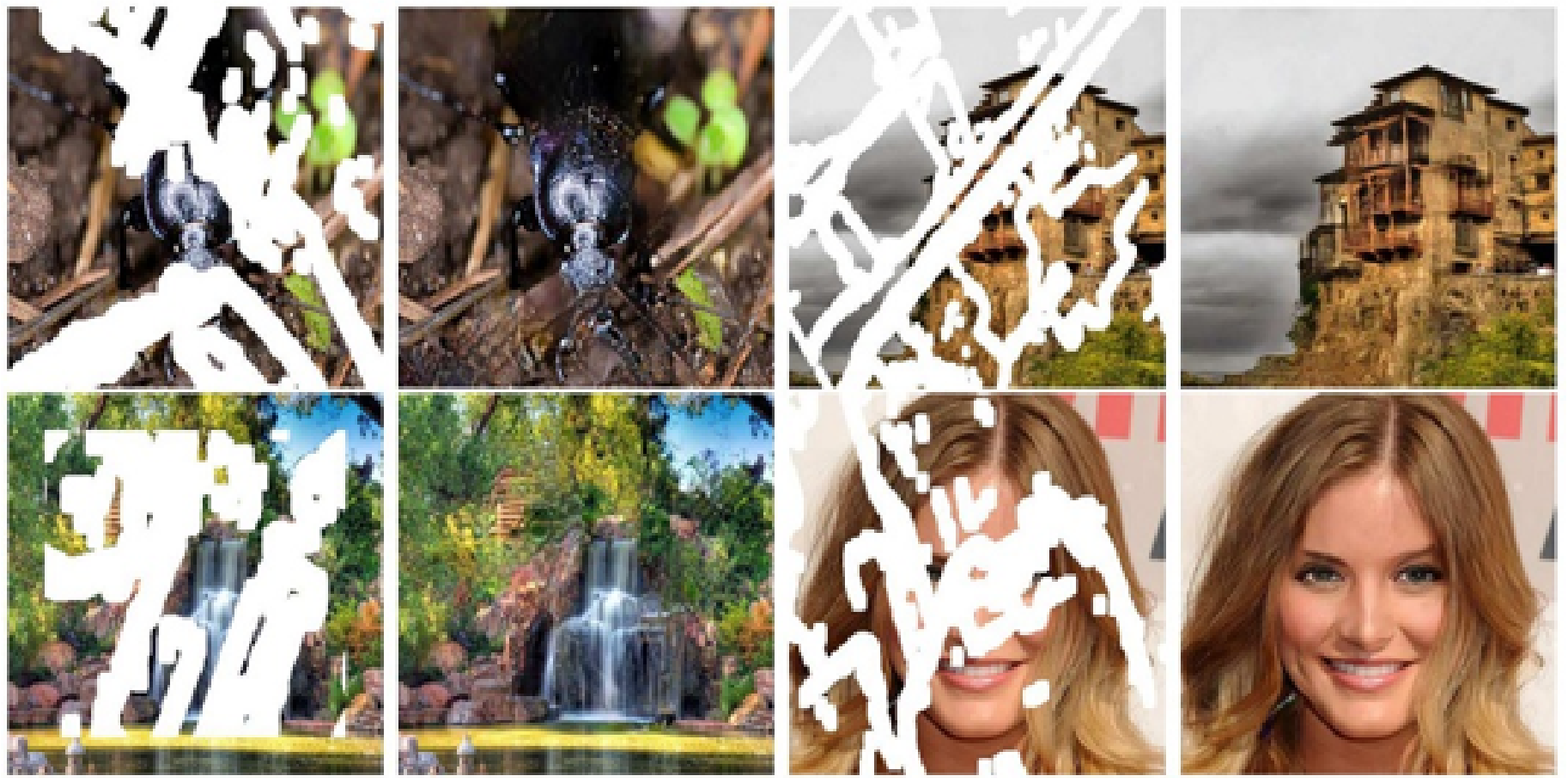

Figure 5: Complete missing patches in image

Completing Incomplete Content (cont.)

Figure 6: Improve image quality

Completing Incomplete Content (cont.)

Figure 7: Color b/w images

Completing Incomplete Content (cont.)

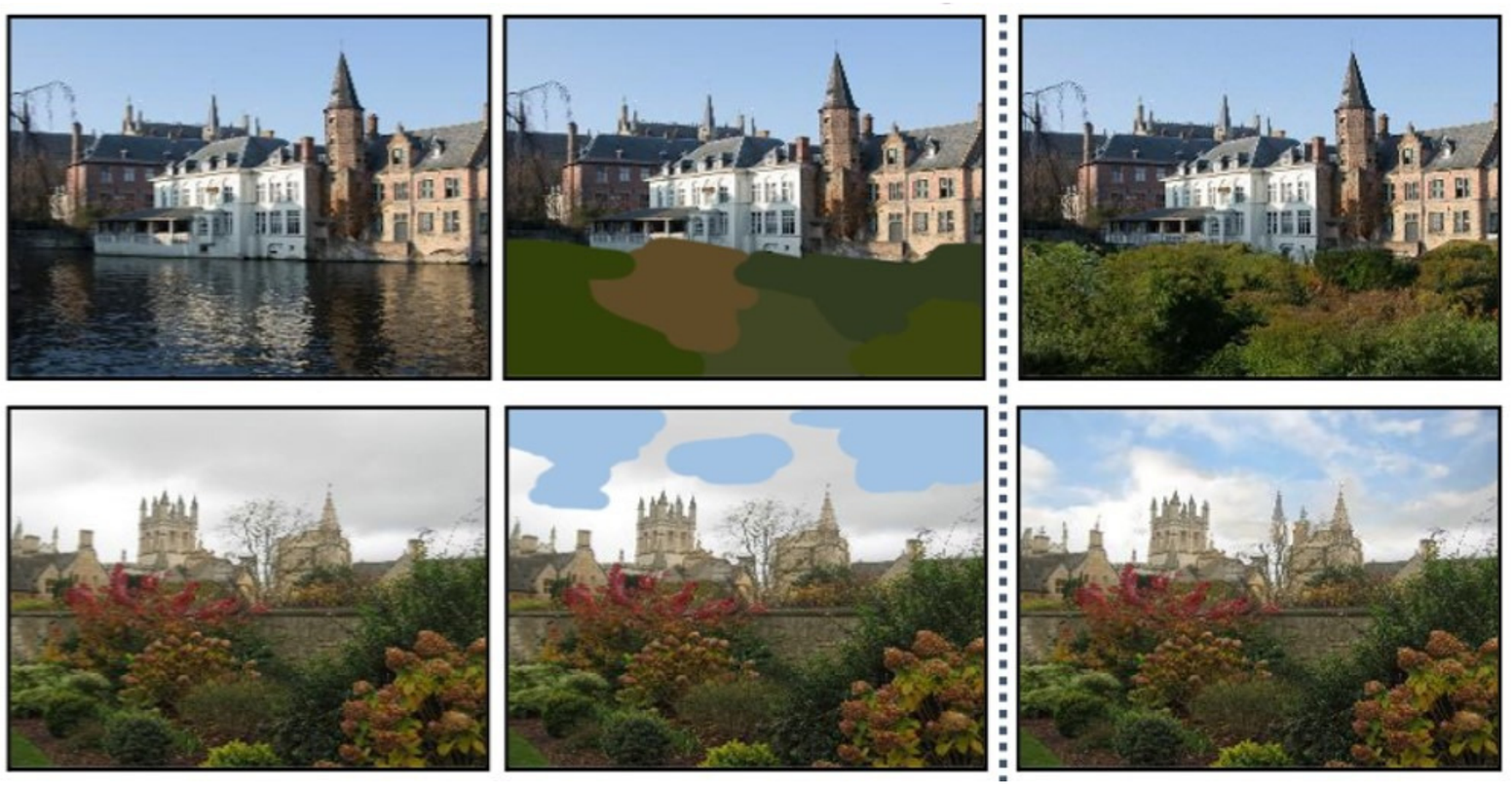

Figure 8: Editing based on strokes in image

Completing Incomplete Content (cont.)

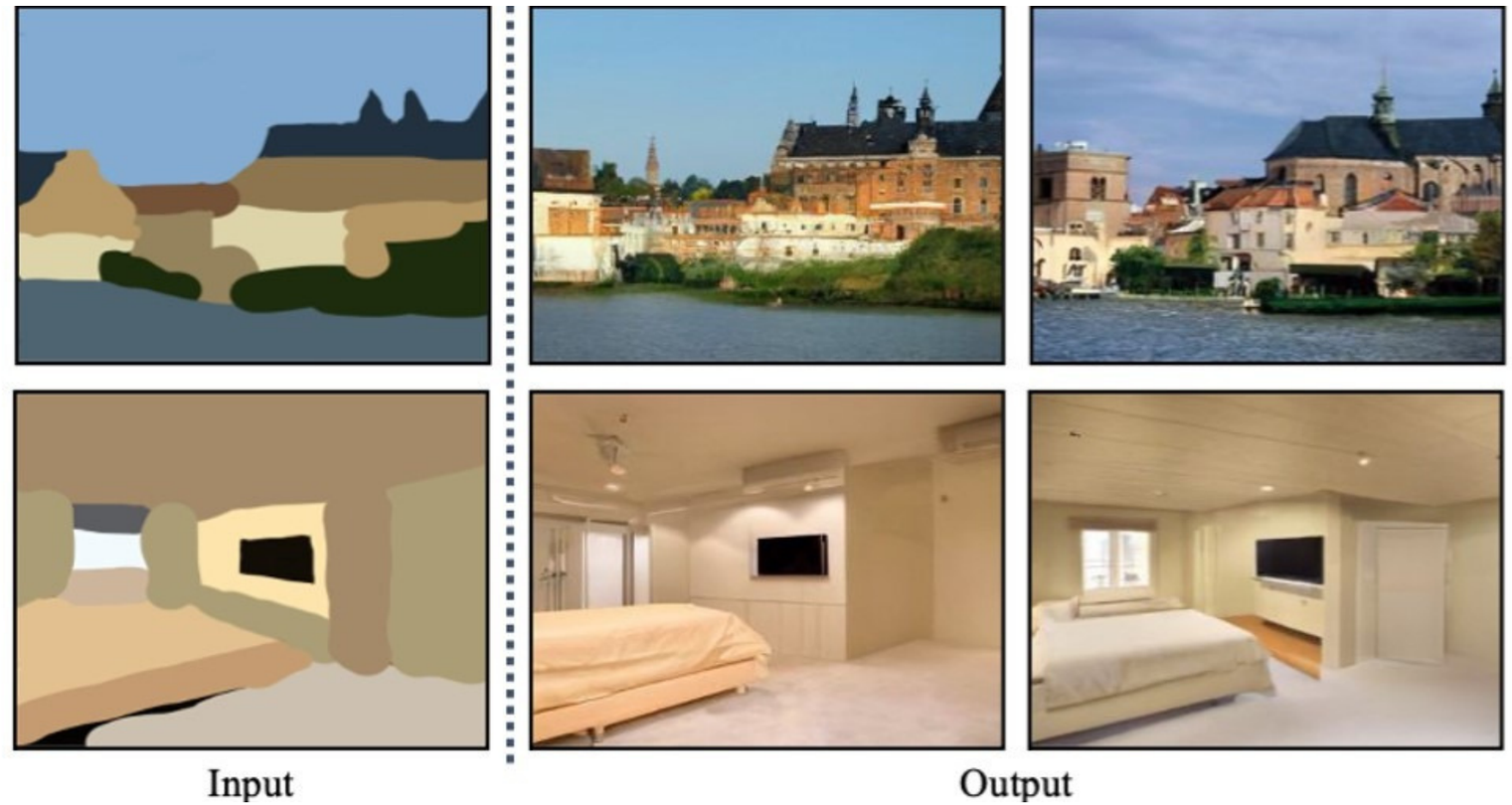

Figure 9: Converting strokes to proper images

Text Generation

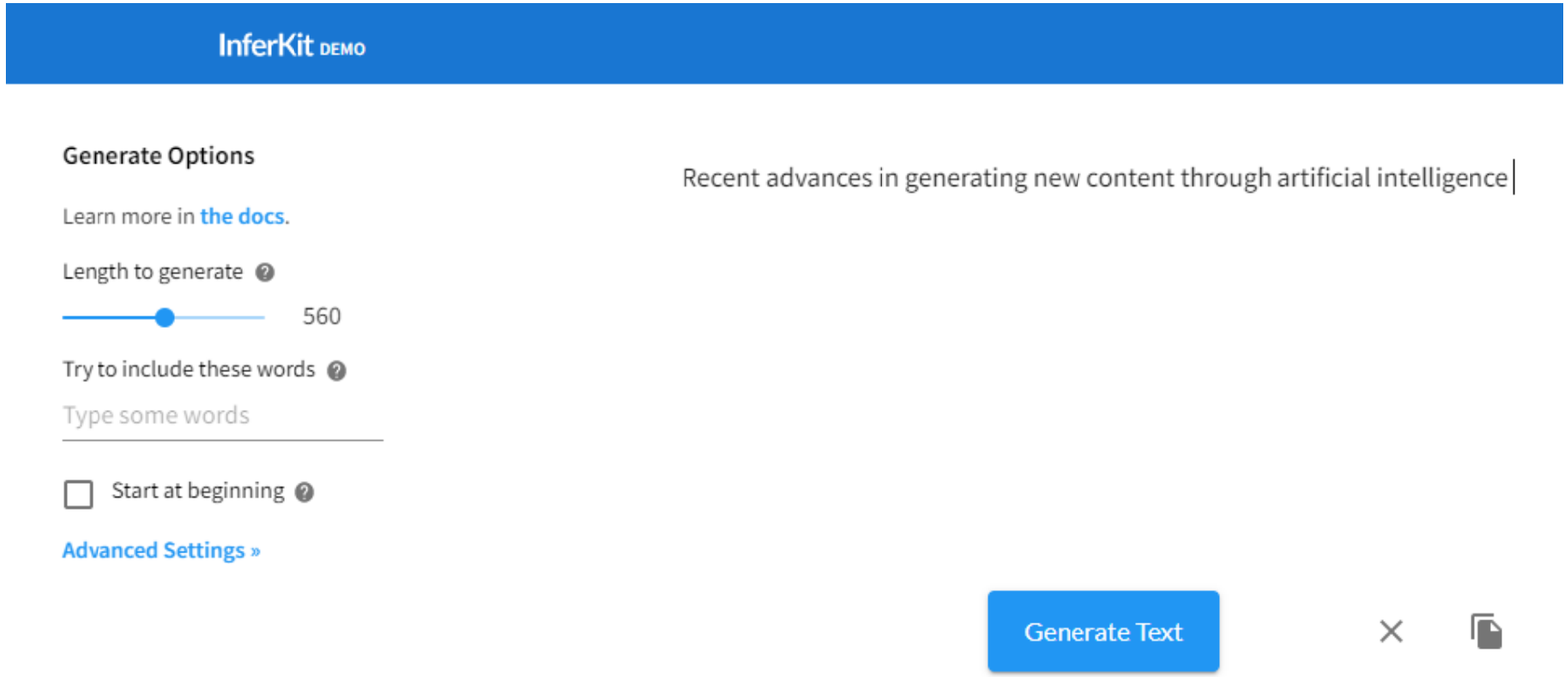

Figure 10: Completing Text. https://app.inferkit.com/demo

Text Generation (cont.)

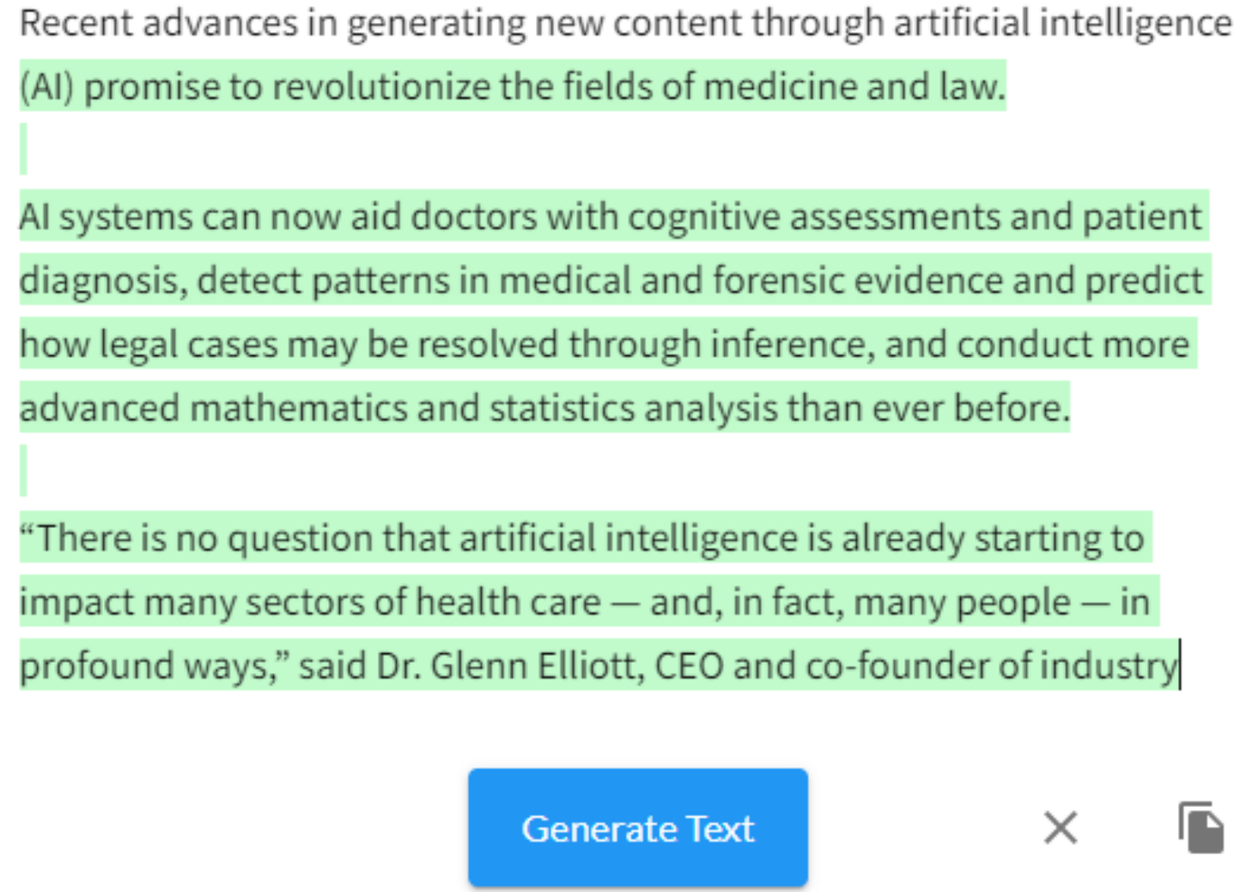

Figure 11: Completing Text. https://app.inferkit.com/demo

Machine Translation

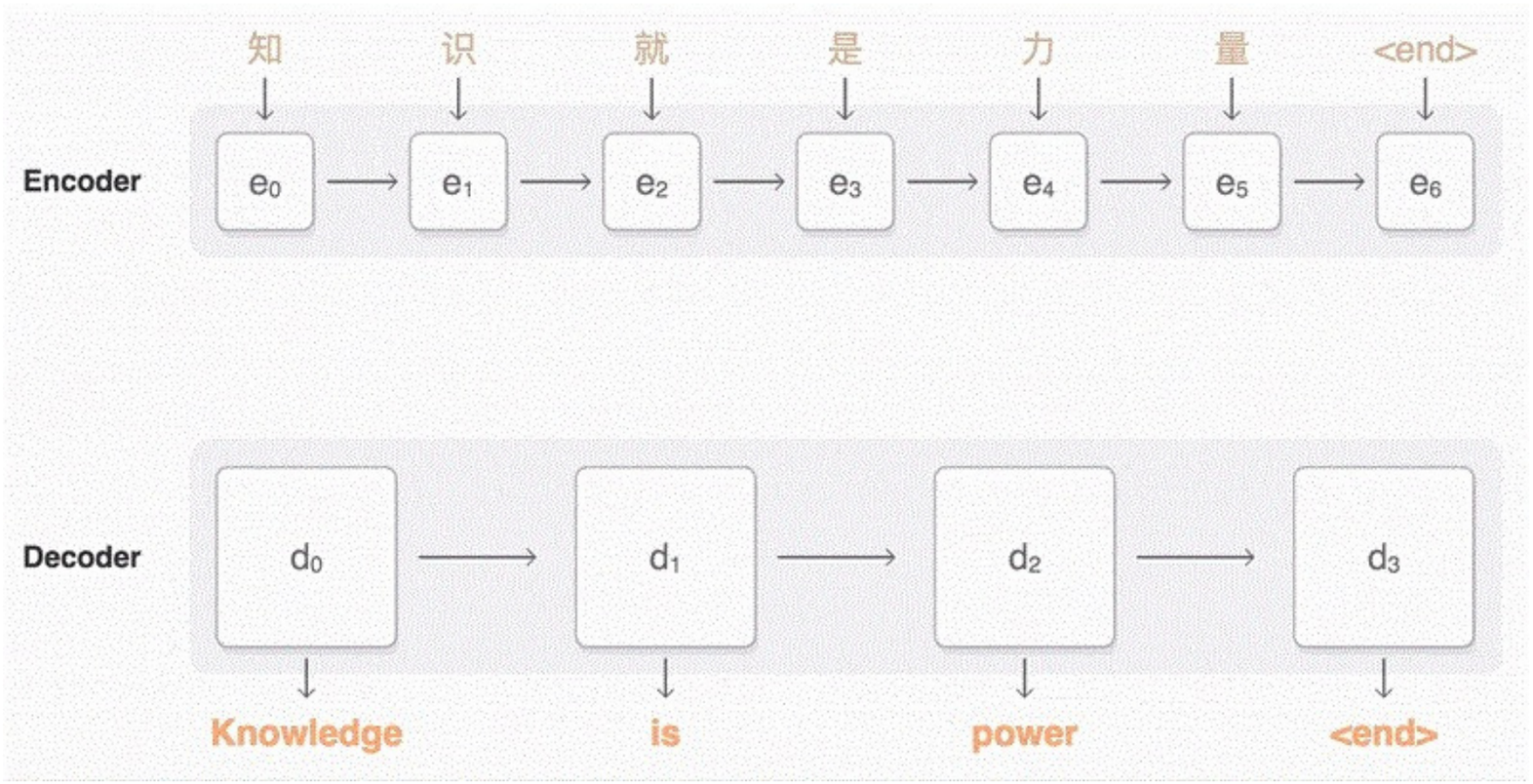

Figure 12: Language translation

Code Generation

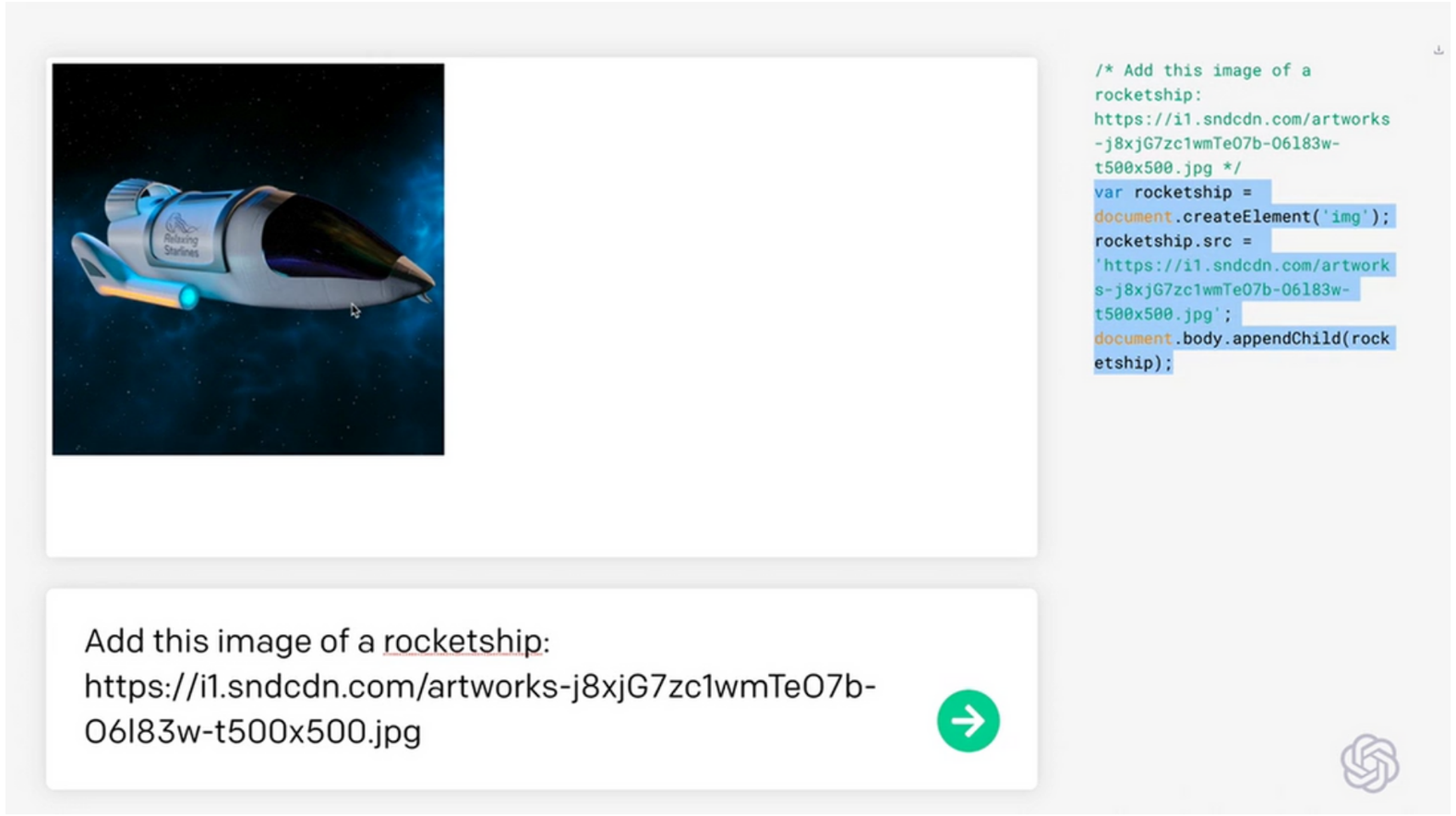

Figure 13: Code Generation. https://openai.com/blog/openai-codex/

Auto enco ders

- Family of neural networks for which the input is the same as the output. They work by compressing the input into a latent-space representation, and then reconstructing the output from this representation.
- The idea is project the input into a latent space and then reconstruct the input from that latent space representation
- Consist of two parts: Encoder and decode.
  - Encoder projects the input to a latent space Z.
  - Decoder takes the encoded embedding vector and reconstructs the input from it.
  - We also use altered versions of input as output which can be even more interesting.

Autoencoders (cont.)

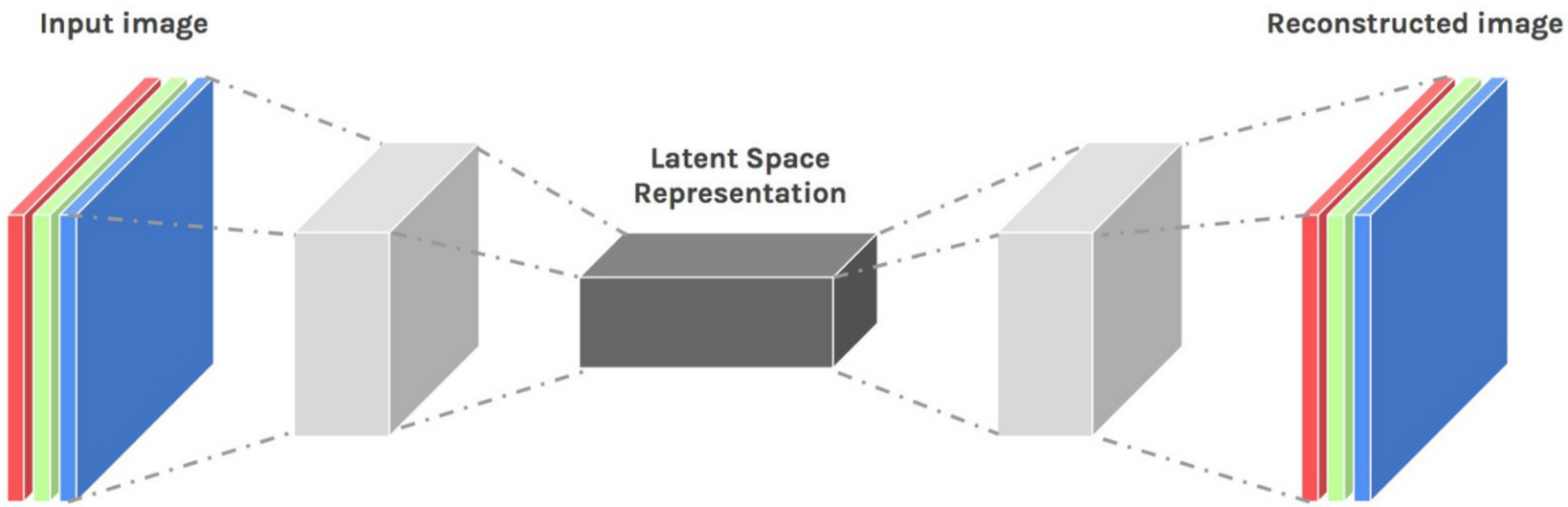

Figure 29: Autoencoder architecture

Autoencoders (cont.)

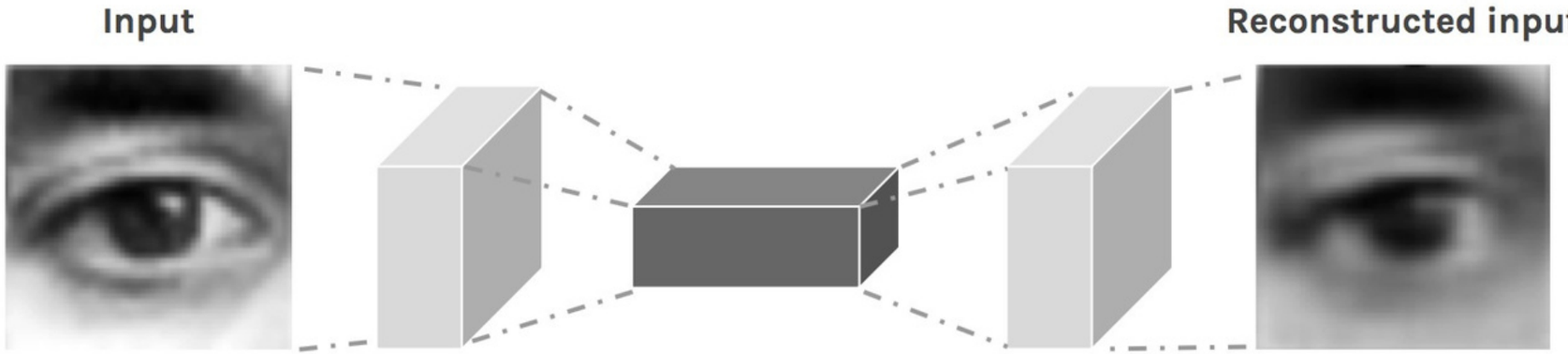

Figure 30: Sample Autoencoder

Autoencoders - Interactive Demo

https://douglasduhaime.com/p osts/visualizing-latent-spaces.html

Autoencoders as generative models ?

- Autoencoders project data into a latent space Z.
- What if we sample a new embedding vector from Z and then have the decoder reconstruct the image from it?
- Does not work. Autoencoders just learn a function that maps input to output. The learned latent space is too discontinous to work as a generative model.

Autoencoders as generative models (cont.)

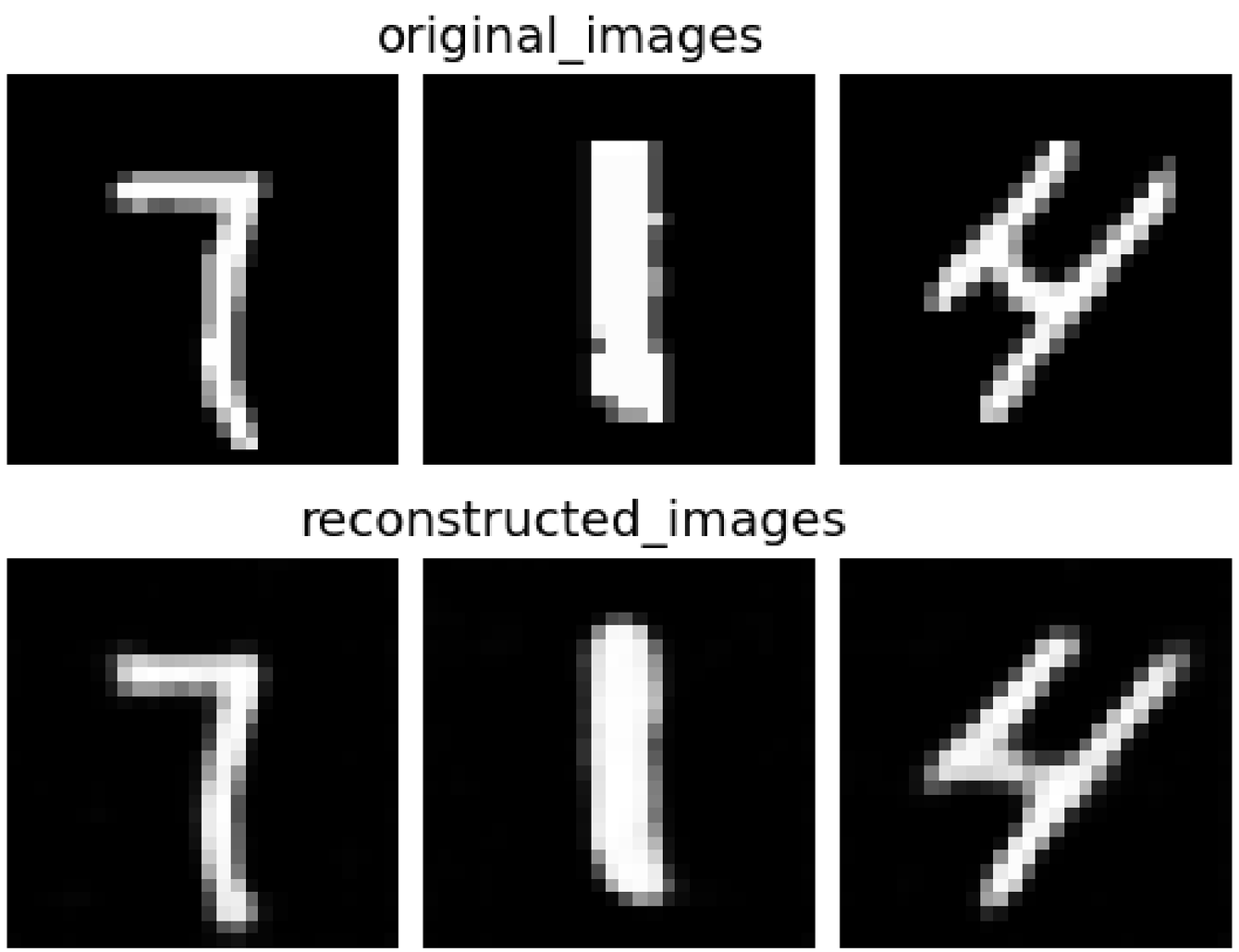

Figure 31: Image reconstruction with autoencoder trained on MNIST digits

Autoencoders as generative models (cont.)

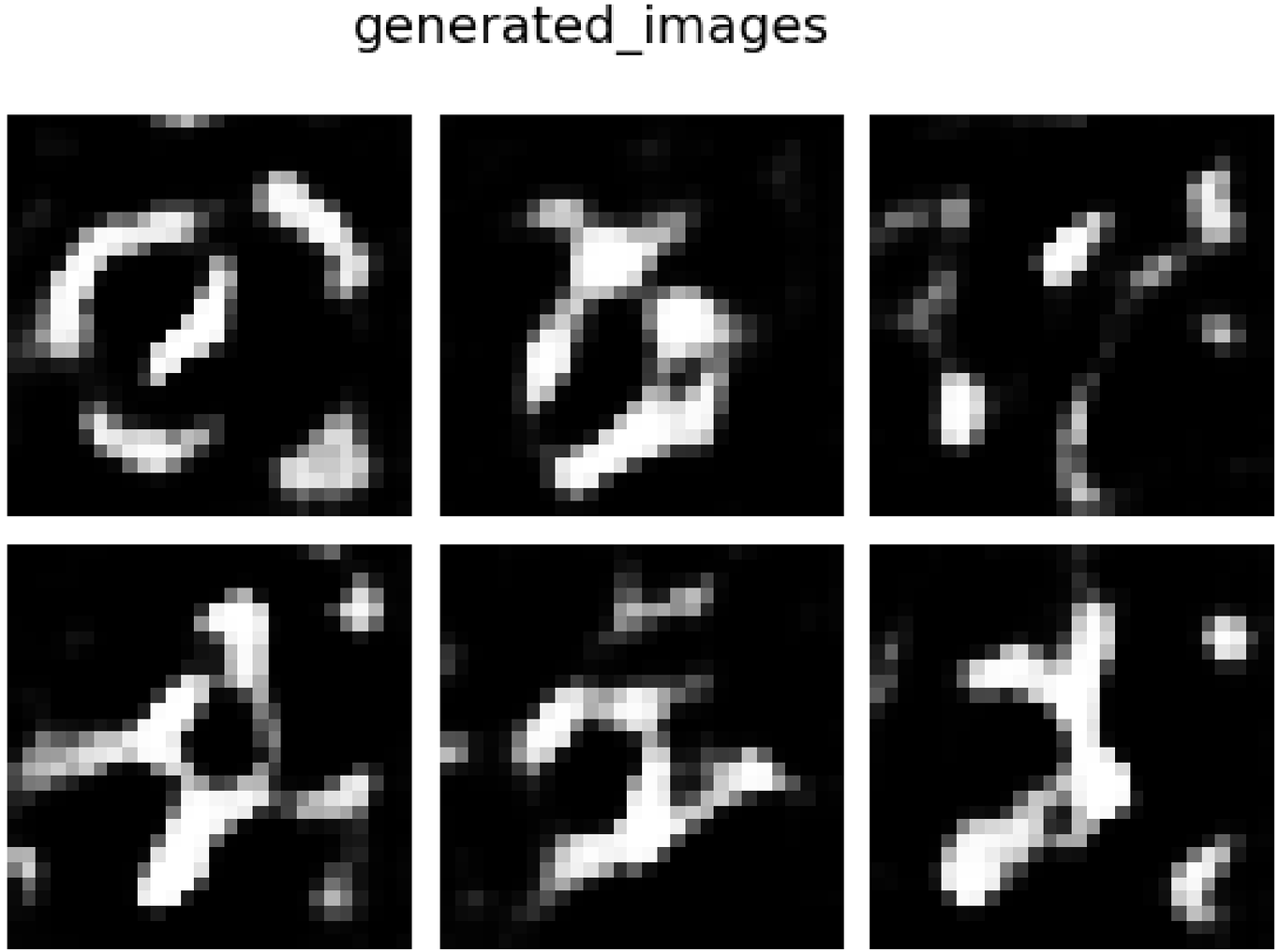

Figure 32: Image generation with autoencoder trained on MNIST digits. Encoding vector sampled from latent space Z and the passed to decoder.

Autoencoders - Applications

- While autoencoders themselves have very low generative power, we will soon talk about a type of autoencoders called Variational Autoencoders which are specifically designed for generative mo deling.
- Other use cases of Autoencoders include:
  - Data encoding and dimensionality
  - reduction Image denoising and
  - super-resolution
  - Image completion

  Image colorization

Autoencoders - Applications (cont.)

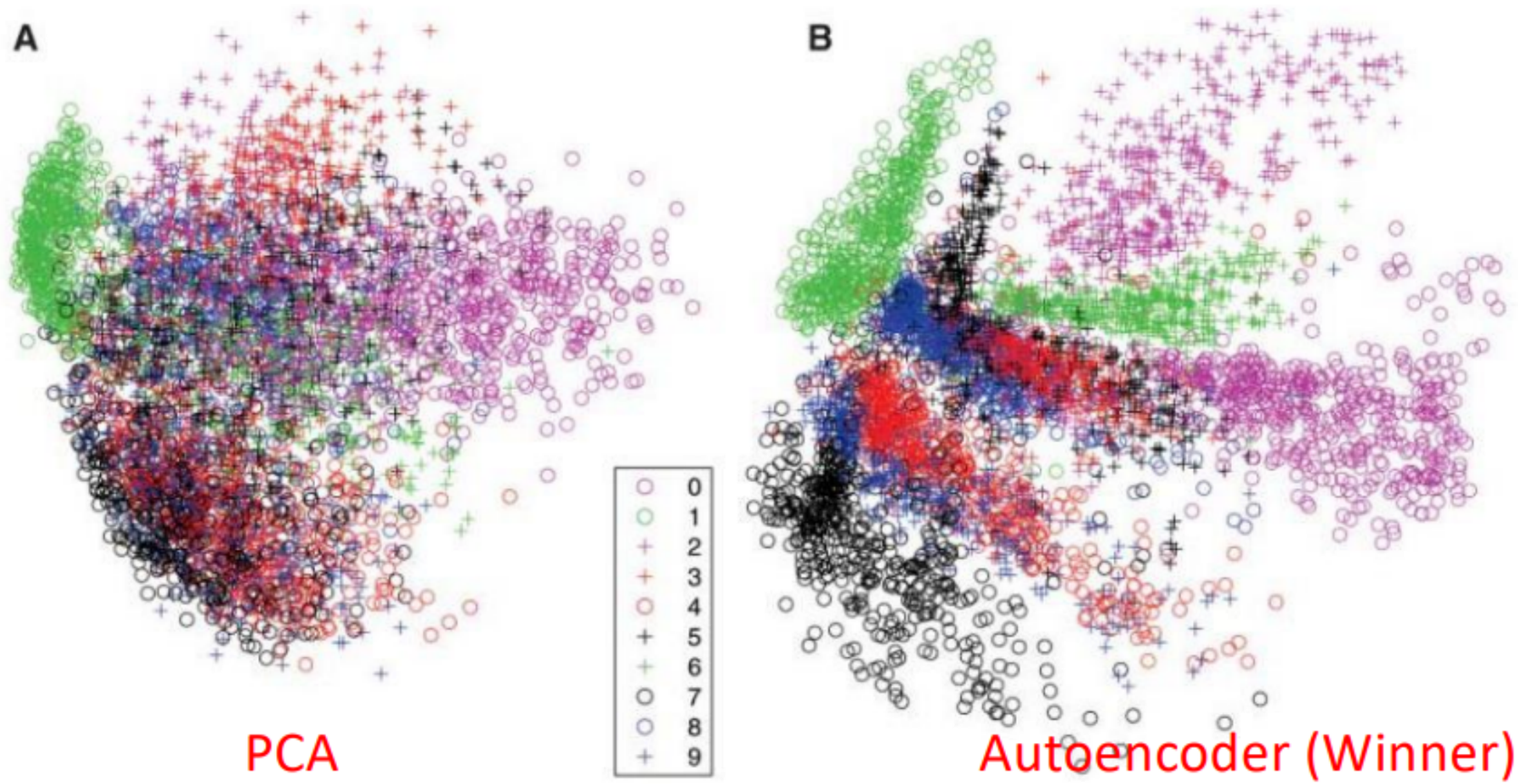

Figure 33: t-SNE visualization on MNIST digits dataset. PCA vs. Autoencoders. The image vector is projected into R2.

Autoencoders - Applications (cont.)

Figure 34: Image super-resolution using Autoencoders

Autoencoders - Applications (cont.)

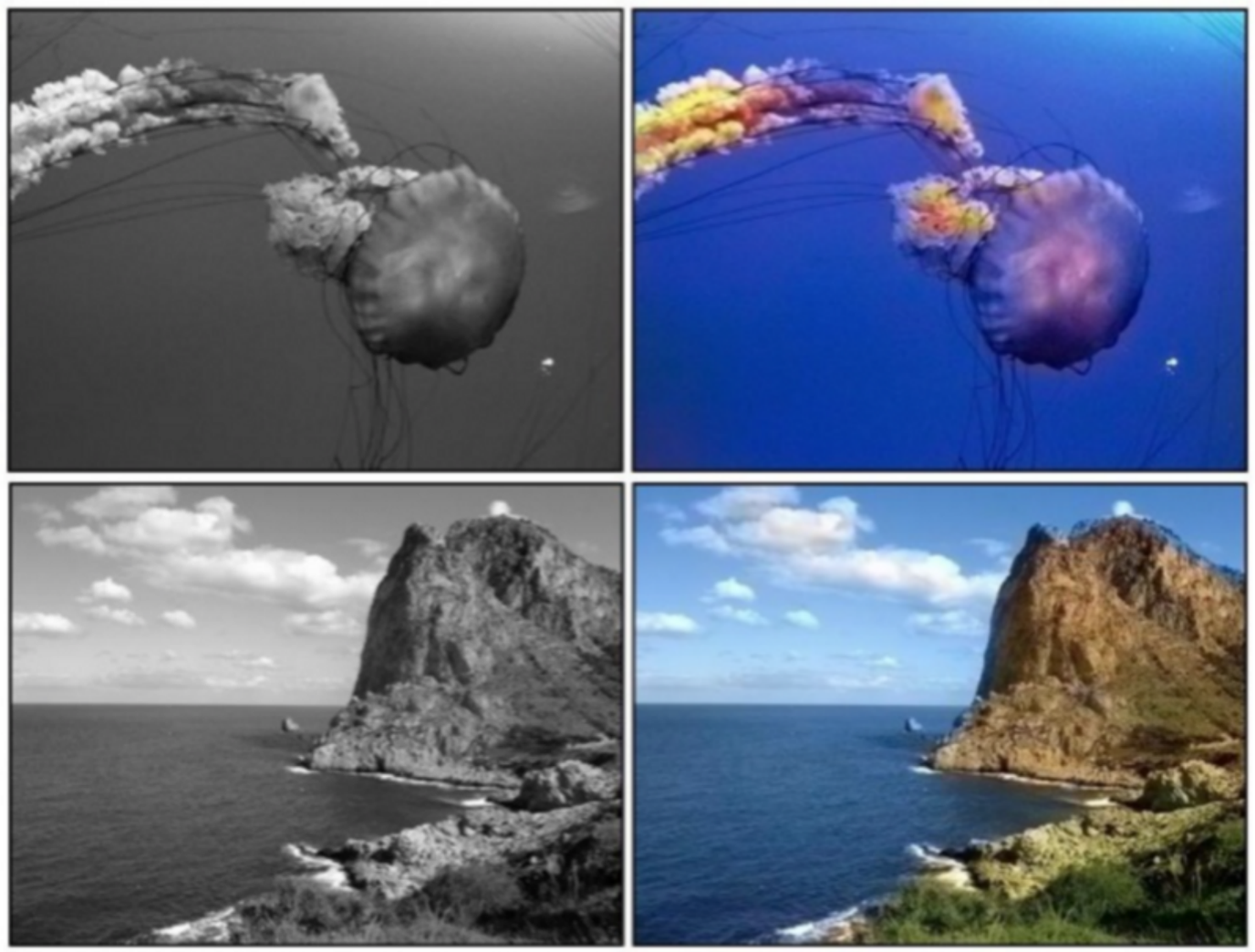

Figure 35: Image colorization using Autoencoders

References

Reference Slides

- Fei-Fei Li ”Generative Deep Learning” CS231
- Hao Dong ”Deep Generative Models”