

Examples of successful applications



Examples of successful applications

- **Image Processing**
 - Image Classification and Detection
 - Image Segmentation, Scene understanding
 - Style transfer
- **Natural Language Processing**
 - Typical problems
 - Relevant architectures
 - GPT3 playground, ChatGPT
- **Generative modeling**
 - The generative problem
 - Conditional generation
- **Deep Reinforcement Learning**
 - Robot navigation and autonomous driving
 - Planning and simulation

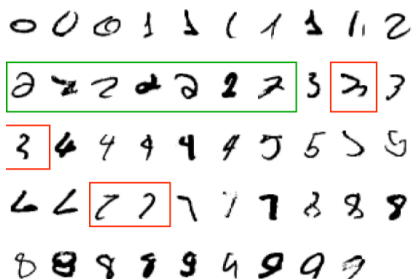


Image Processing



Modified National Institute of Standards and Technology database

- ▶ grayscale images of handwritten digits, 28×28 pixels each
- ▶ 60,000 training images and 10,000 testing images



A comparison of different techniques



Classifier	Error rate
Linear classifier	7.6
K-Nearest Neighbors	0.52
SVM	0.56
Shallow neural network	1.6
Deep neural network	0.35
Convolutional neural network	0.21

See LeCun's page [the mnist database](#) for more data.

ImageNet

ImageNet (@Stanford Vision Lab)

- ▶ high resolution color images covering 22K object classes
- ▶ over 15 million labeled images from the web

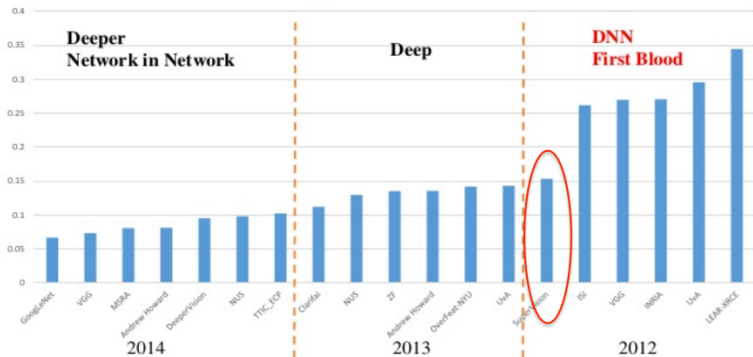


ImageNet samples



ImageNet results

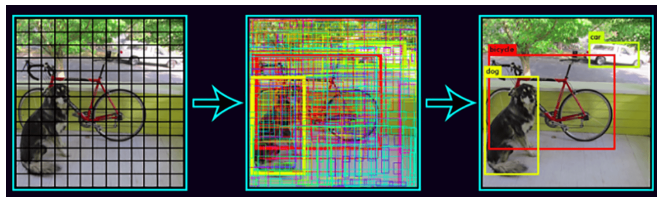
ImageNet Classification error throughout years and groups



Li Fei-Fei: ImageNet Large Scale Visual Recognition Challenge, 2014 <http://image-net.org/>



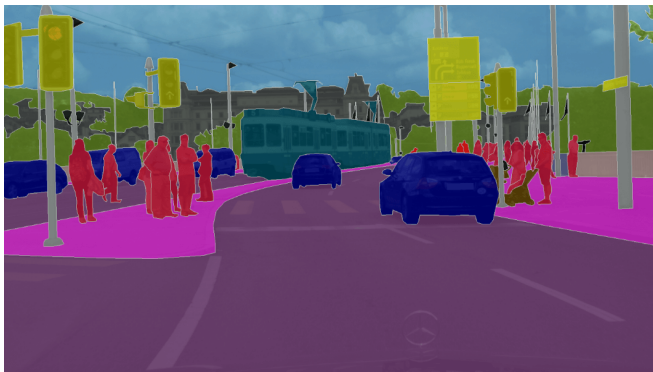
YOLO: Real-Time Object Detection



You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Pascal Titan X it processes images at 30 FPS and has a mAP of 57.9



Video-to-Video Synthesis



Mimicking style

A neural algorithm of artistic style

L.A. Gatys, A.S. Ecker, M. Bethge



Change the style of an image, preserving the content.



Natural Language Processing

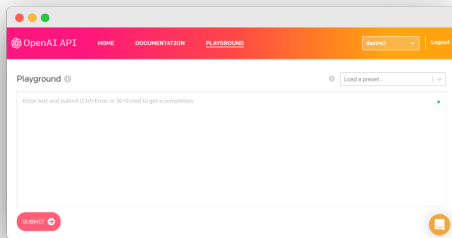


Typical problems in NLP

- ▶ Classification, e.g. sentiment analysis: classify a document according to its “polarity” .
- ▶ Language Modeling (predict the next word/token in a sentence)
- ▶ Text summarization/completion
- ▶ Machine Translation
- ▶ Text Generation: a truly generative task
- ▶ Chatbox: a dialog system
- ▶ Speech recognition/generation

Examples: GPT3 playground, ChatGPT

Generative Pre-trained Transformer 3 (GPT-3) is a recent language model created by OpenAI showing amazing performance. It is one of the largest neural network ever created, with 175 billion parameters, trained over almost all documents available on internet.



Key Technologies

tokenization splitting the input sentence into relevant lexical components (characters/words/**subwords**), and coding them into numbers. Byte-Pair Encoding, WordPiece, SentencePiece, etc.
see this [tokenizer summary](#)

transformers a feed-forward deep learning model adopting **self-attention** for weighting the mutual significance of tokens in the sentence

word embeddings A semantic embedding of words, mostly used for text similarity, text retrieval, code search, etc.
Examples are Word2Vec, Glove.

Transformers do not use them: they learn their own embeddings.
see [this blog](#) for a comparison of state-of-the-art text embeddings.

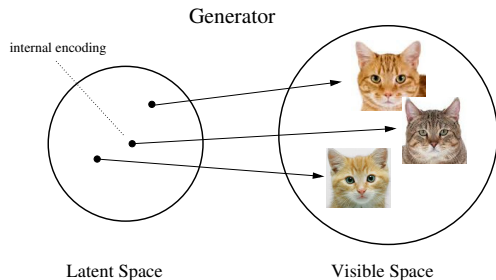


Generative Modeling



Generative modeling

Train a **generator** able to sample original data similar to those in the training set, implicitly learning the **distribution of data**.

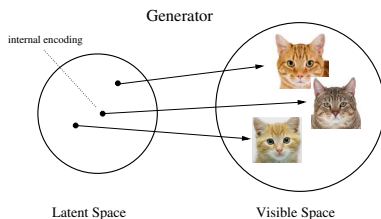


the randomness of the generator is provided by a **random seed** (noise) received as input

The latent space

Generation is thus a two stage process (ancestral sampling):

- sample z according to a **known prior distribution**
- pass z as input to the generator, and process it



- ▶ the seed contains **all information** needed to generate a sample, hence it can be seen as an **encoding** (latent representation) of the given sample.
- ▶ seeds must be disseminated with a **known, regular distribution** in their space (the so called prior distribution)
- ▶ generation is a continuous process: small modifications of the seed produce small modifications of the output

Generative Modeling

Example: Face generation video by Nvidia



Conditional generation



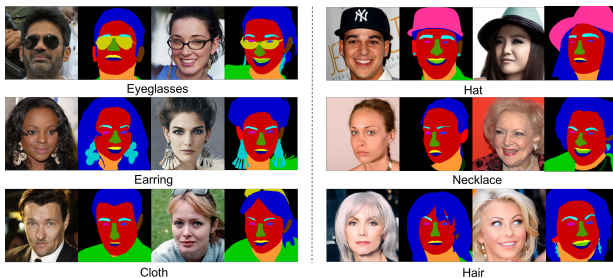
Deep Single Image Portrait Relighting



Interpreting the Latent Space of GANs for Semantic Face Editing



Conditional generation



MaskGAN: Towards Diverse and Interactive Facial Image Manipulation



Caption-conditioning

Dall·E, Imagen and others are new AI systems able to create realistic images and art from a description in natural language.



Embedding

Embedding is the inverse process of generation, namely from visible space to latent space.

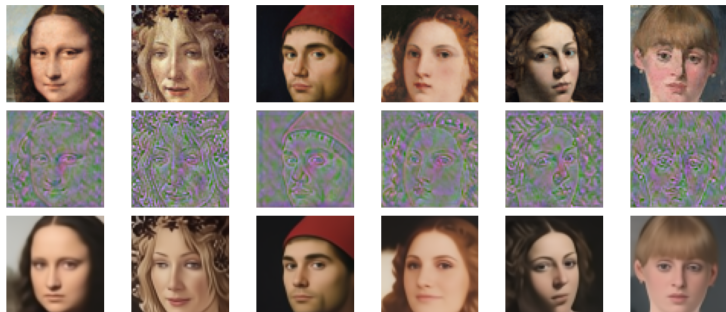


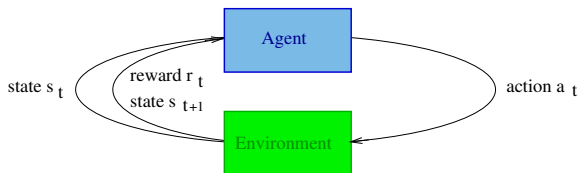
Image Embedding for Denoising Generative Models

Reinforcement Learning



Reinforcement Learning

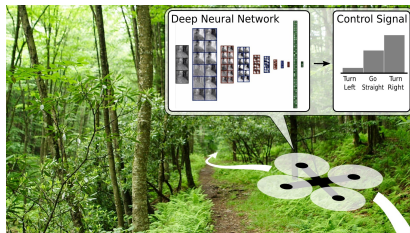
Problems involving an **agent** interacting with an **environment**, which provides numeric **rewards**



Goal: learn how to take actions in order to maximize the future **cumulative** reward.



Quadcopter Navigation in the Forest using Deep Neural Networks



Robotics and Perception Group, University of Zurich, Switzerland &
Institute for Artificial Intelligence (IDSIA), Lugano Switzerland

Based on **Imitation Learning**

Autonomous driving

Develop intelligent, fully automatic driving functions for vehicles.



Merging of signals collected by different sensors (camera, lidar, sonar, dots). Needs to accurately evaluate distances and speeds.

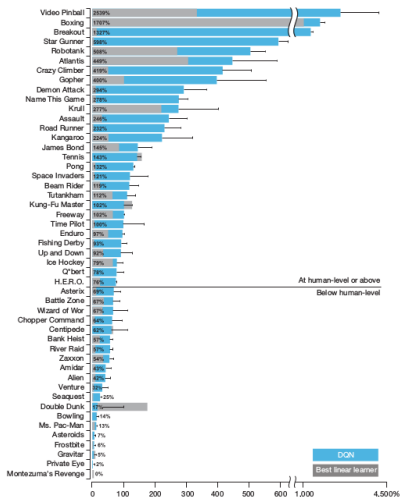


Turn observations into actions.

Several competitions around. We took part to the 2018 Audi Autonomous Driving Cup



Game Simulation



Google DeepMind's system playing Atari games (2013)

The same network architecture was applied to all games

End-to-end training starting from screen frames

Works well for reactive games; problems with planning...

but see [An investigation of Model-Free planning \(ICML 2019\)](#)



Open AI-gym/(gymnasium)



OpenAI Gym (now gymnasium) is a Gym is a toolkit for developing and comparing reinforcement learning algorithms (DQN, A3C, A2C, Acer, PPO, ...).

It offers many learning scenarios, from walking to playing games like Pong or Pinball, as well as other classical physical “equilibrium” problems.

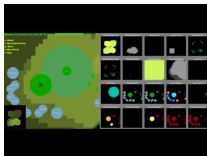


Multi agent DRL

It requires interaction and cooperation of multiple agents.

Examples:

StarCraft II: a RL environment based on the game StarCraft II. The environment consists of three sub-components: a Linux StarCraft II binary, the StarCraft II API providing programmatic control over the game, and a python wrapper over the API called PyC2.



Flatland: a train **rescheduling** problem on a complex grid world environment.

Flatland is organized every year by **Alcrowd** in collaboration with the **Swiss Federal Railways, SBB**