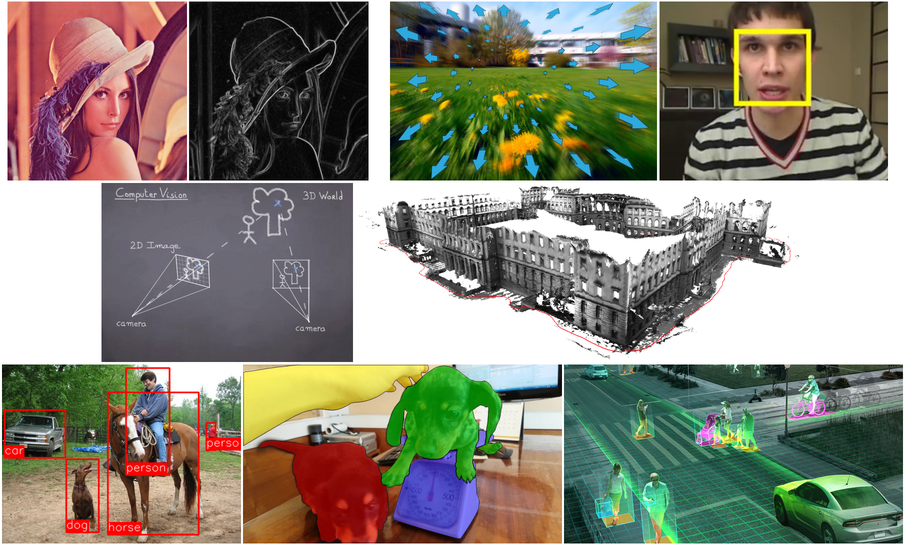
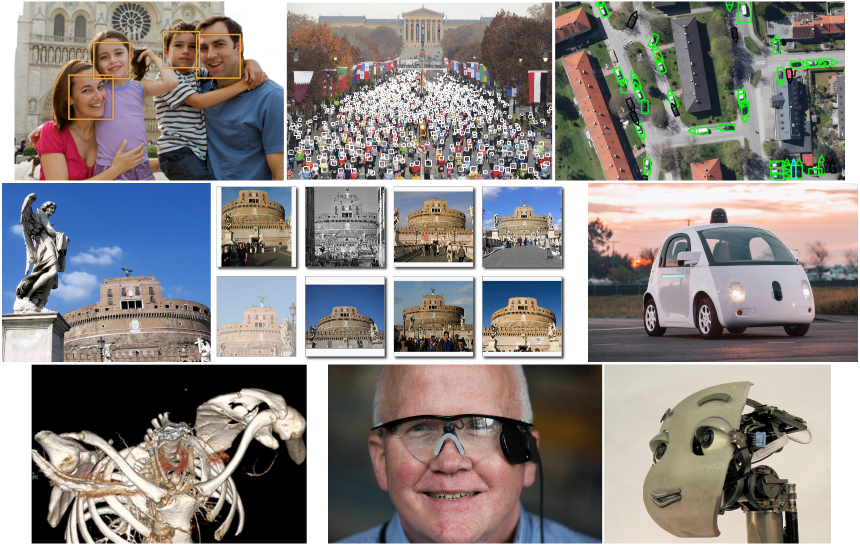
# Computer vision and image processing introduction

Ronan Sicre
Credits to Yannis Avrithis https://sif-dlv.github.io/

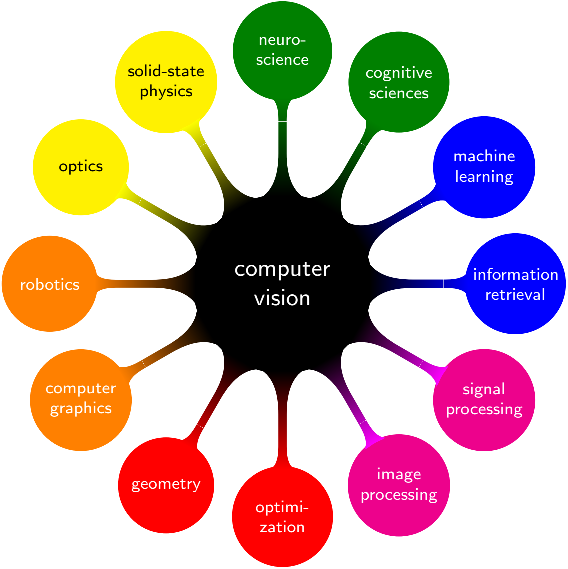
# computer vision in images



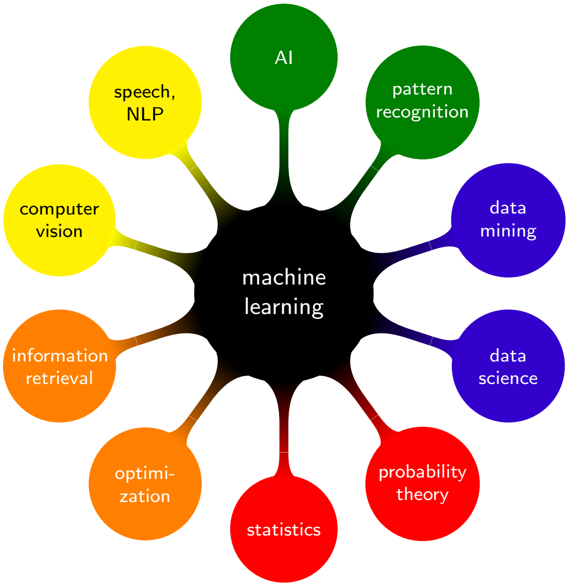
# computer vision in images



# computer vision—related fields



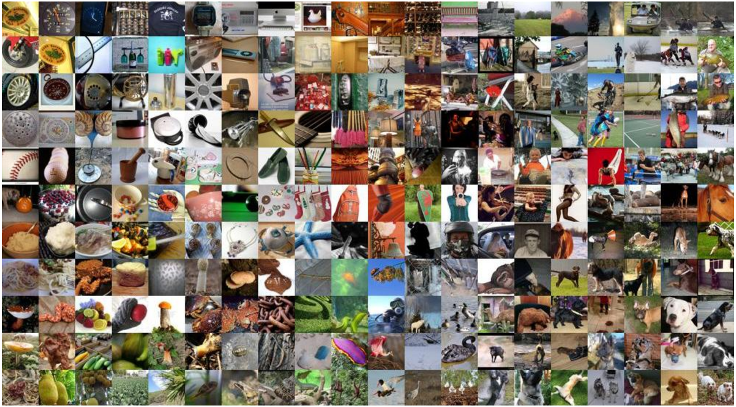
# machine learning—related fields



# modern deep learning

# ImageNet

[Russakovsky et al. 2014]

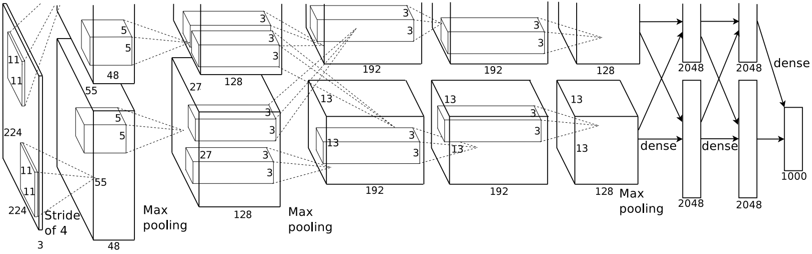


- 22k classes, 15M samples
- ImageNet Large-Scale Visual Recognition Challenge (ILSVRC): 1000 classes, 1.2M training images, 50k validation images, 150k test images

Russakovsky, Deng, Su, Krause, *et al.* 2014. Imagenet Large Scale Visual Recognition Challenge.

# AlexNet

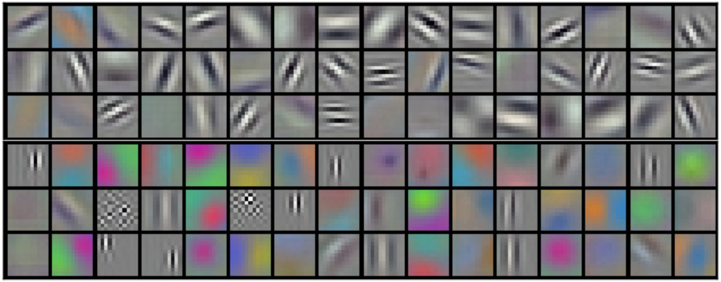
[Krizhevsky et al. 2012]




- implementation on two GPUs; connectivity between the two subnetworks is limited
- ReLU, data augmentation, local response normalization, dropout
- outperformed all previous models on ILSVRC by 10%

Krizhevsky, Sutskever, Hinton. NIPS 2012. Imagenet Classification with Deep Convolutional Neural Networks.

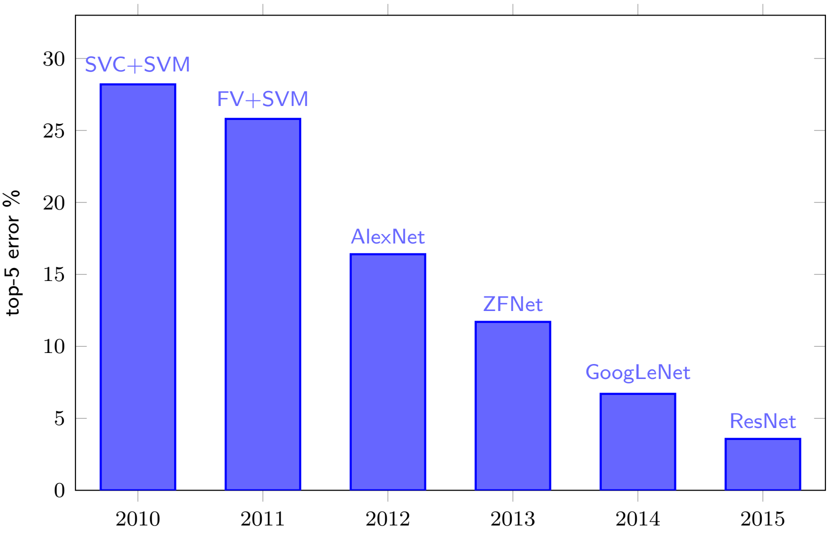
# learned layer 1 kernels



- 96 kernels of size $11 \times 11 \times 3$
- top: 48 GPU 1 kernels; bottom: 48 GPU 2 kernels

Krizhevsky, Sutskever, Hinton. NIPS 2012. Imagenet Classification with Deep Convolutional Neural Networks.

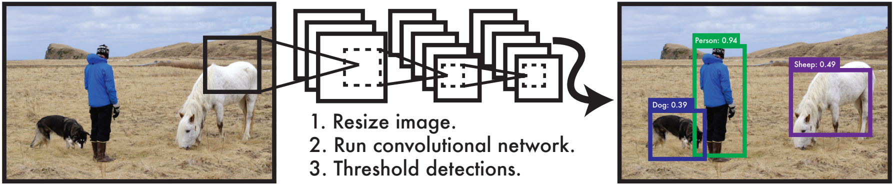
# ImageNet classification performance




Russakovsky, Deng, Su, Krause, *et al.* 2014. Imagenet Large Scale Visual Recognition Challenge.

# object detection

[Redmon et al. 2016]



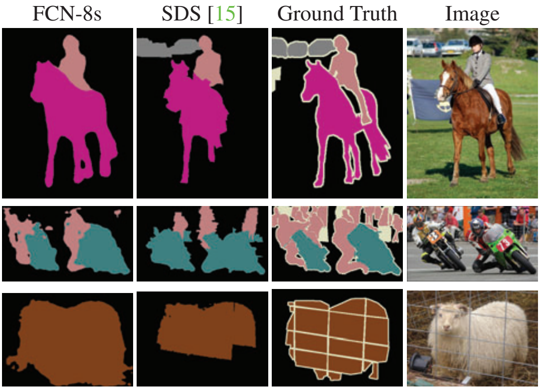

1. Resize image.
2. Run convolutional network.
3. Threshold detections.

- learn to detect objects as a single classification and regression task, without scanning the image or detecting candidate regions
- first object detector to operate at 45fps

Redmon, Divvala, Girshick, Farhadi. CVPR 2016. You Only Look Once: Unified, Real-Time Object Detection.

# semantic segmentation

[Long et al. 2015]




- learn to upsample
- apply to pixel-dense prediction tasks

Long, Shelhamer, Darrell. CVPR 2015. Fully Convolutional Networks for Semantic Segmentation.

# instance segmentation and pose estimation

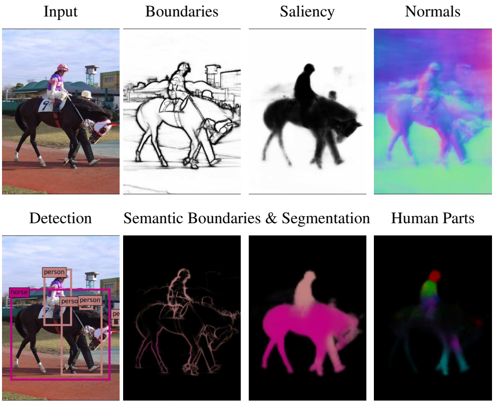
[He et al. 2017]

- semantic segmentation per detected region
- pose estimation as regression

He, Gkioxari, Dollar, Girshick. ICCV 2017. Mask R-CNN.

# multi-task learning

[Kokkinos 2017]

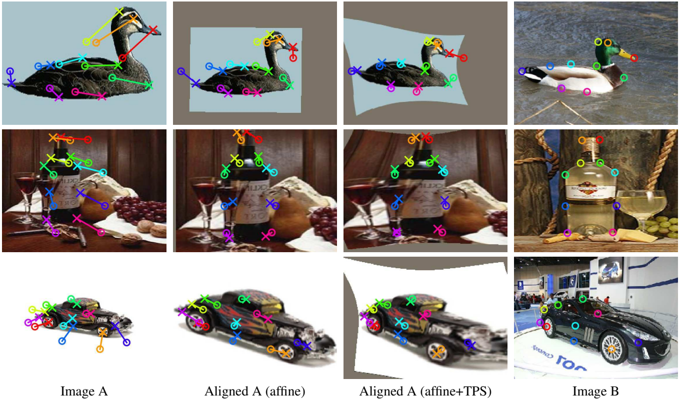



- learn several vision tasks with a joint network architecture including task-specific skip layers

Kokkinos. CVPR 2017. Ubernet: Training a Universal Convolutional Neural Network for Low-, Mid-, and High-Level Vision Using Diverse Datasets and Limited Memory.

# geometric matching

[Rocco et al. 2017]

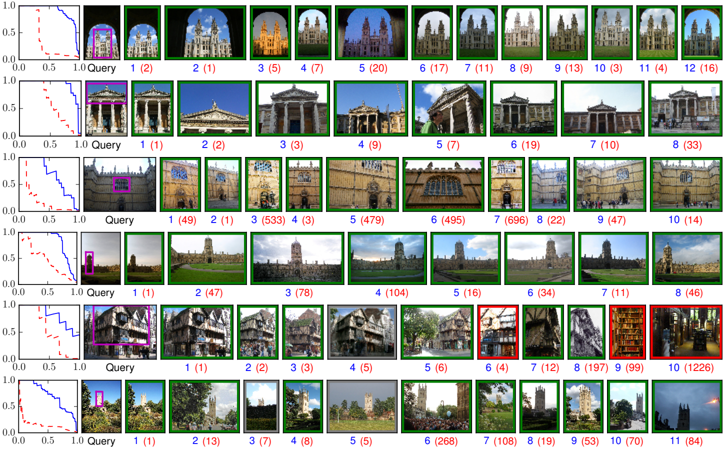



- mimic the standard steps of feature extraction, matching and simultaneous inlier detection and model parameter estimation
- still trainable end-to-end

Rocco, Arandjelovic, Sivic. CVPR 2017. Convolutional Neural Network Architecture for Geometric Matching.

# image retrieval

[Gordo et al. 2016]

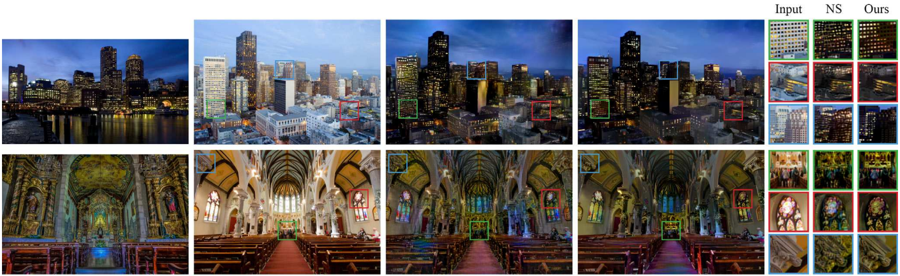



- learn to match
- apply as generic feature extractor

Gordo, Almazan, Revaud, Larlus. ECCV 2016. Deep Image Retrieval: Learning Global Representations for Image Search.

# photorealistic style transfer

[Luan et al. 2017]




(a) Reference style image (b) Input image (c) Neural Style (distortions) (d) Our result (e) Insets

- generate same scene as input image
- transfer style from reference image
- photorealism regularization

Luan, Paris, Shechtman, Bala. CVPR 2017. Deep Photo Style Transfer.

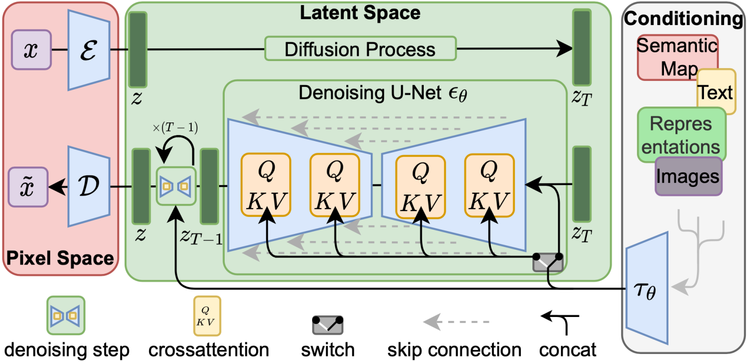
# image captioning

[Vinyals et al. 2017]


- image description by deep CNN
- language generation by RNN

Vinyals, Toshev, Bengio and Erhan. PAMI 2017. Show and Tell: Lessons Learned From the 2015 MSCOCO Image Captioning Challenge.

GAN, Diffusion, VAE, MAE, DAE.



# Self-supervised models

RotNet, Deep Cluster, BYOL, DINO, iBOT

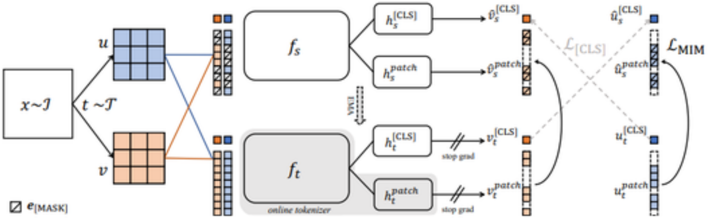



Figure 3: **Overview of iBOT framework, performing masked image modeling with an *online tokenizer*.** Given two views $\boldsymbol{u}$ and $\boldsymbol{v}$ of an image $\boldsymbol{x}$, each view is passed through a teacher network $h_t \circ f_t$ and a student network $h_s \circ f_s$. iBOT minimizes two losses. The first loss $\mathcal{L}_{[CLS]}$ is self-distillation between cross-view [CLS] tokens. The second loss $\mathcal{L}_{MIM}$ is self-distillation between in-view patch tokens, with some tokens masked and replaced by $\boldsymbol{e}_{[MASK]}$ for the student network. The objective is to reconstruct the masked tokens with the teacher networks' outputs as supervision.