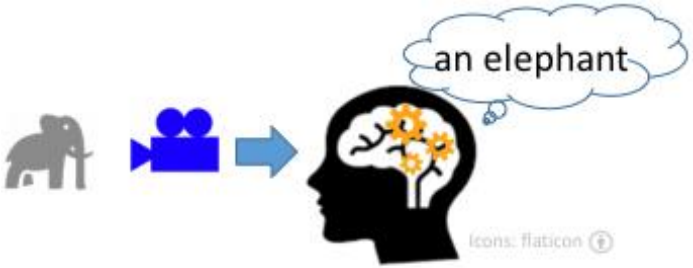
AI in computer vision

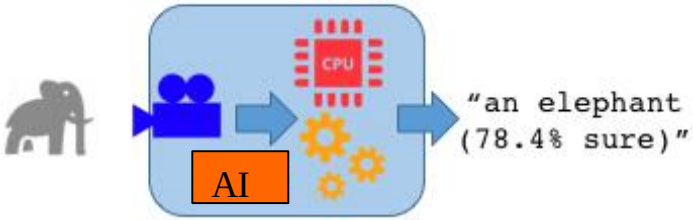
Basic idea:

use computer/AI processing to go beyond image capture, towards image interpretation.



Standard Camera:

- Captures video
- Sends video to human user
- Human brain interprets what the video contains



AI Camera:

- Captures video
- Processes video to interpret contents
- Sends interpretation to human, robot, or another computer

A good historical overview and perspective

- Slides by Justin Johnson at UMich

- https://github.com/Andrew-Ng-s-number-one-fan/EECS498-Deep-Learning-for-Computer-Vision

Computer Vision is everywhere!



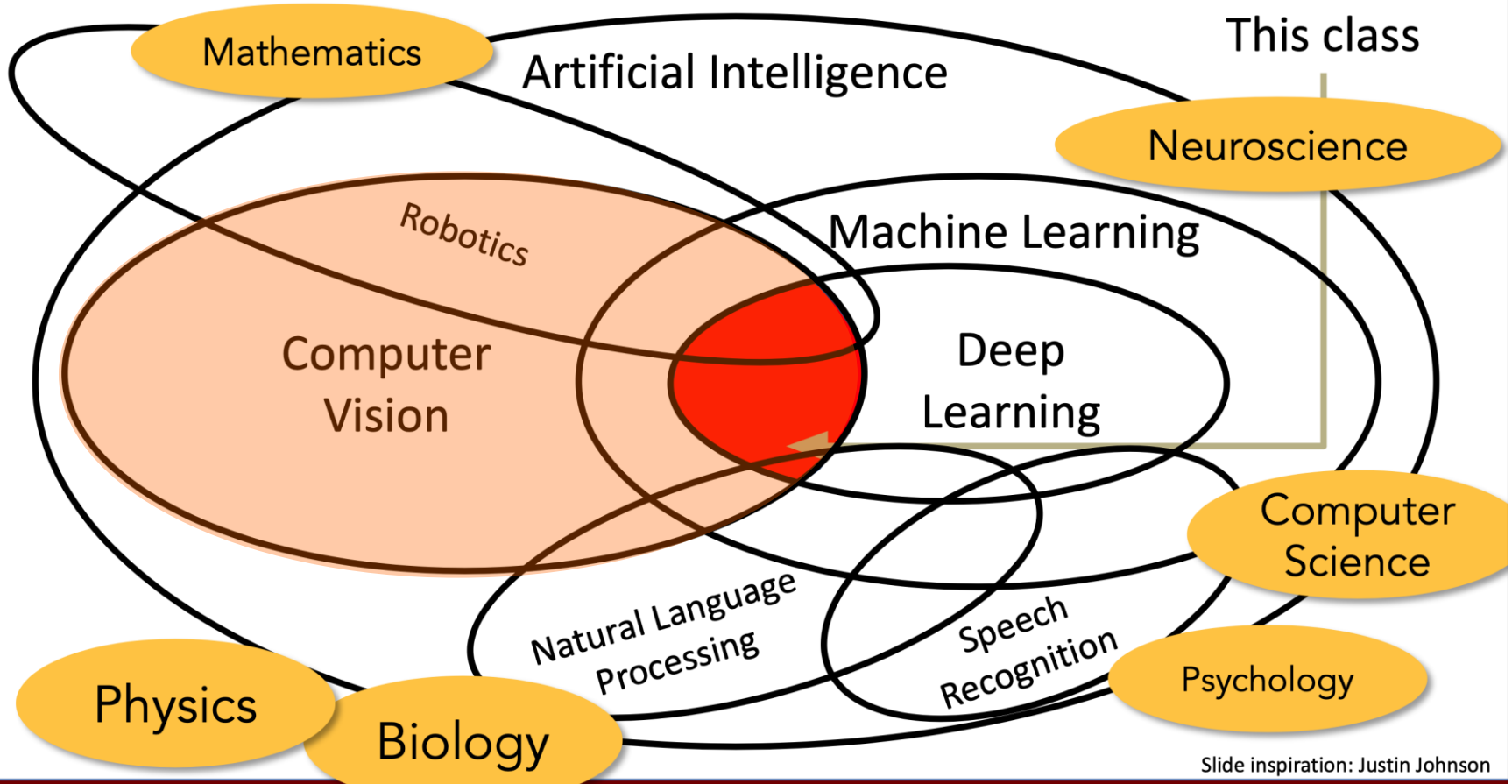

Left to right:
Image by Roger H Goun is licensed under CC BY 2.0
Image is CC0 1.0 public domain
Image is CC0 1.0 public domain
Image is CC0 1.0 public domain




Left to right:
Image is free to use
Image is CC0 1.0 public domain
Image by NASA is licensed under CC BY 2.0
Image is CC0 1.0 public domain




Bottom row, left to right
Image is CC0 1.0 public domain
Image by Derek Keats is licensed under CC BY 2.0; changes made
Image is public domain
Image is licensed under CC-BY 2.0; changes made



Slide inspiration: Justin Johnson

Starting point: optics and image formation

From 3D world to 2D image: loss of information

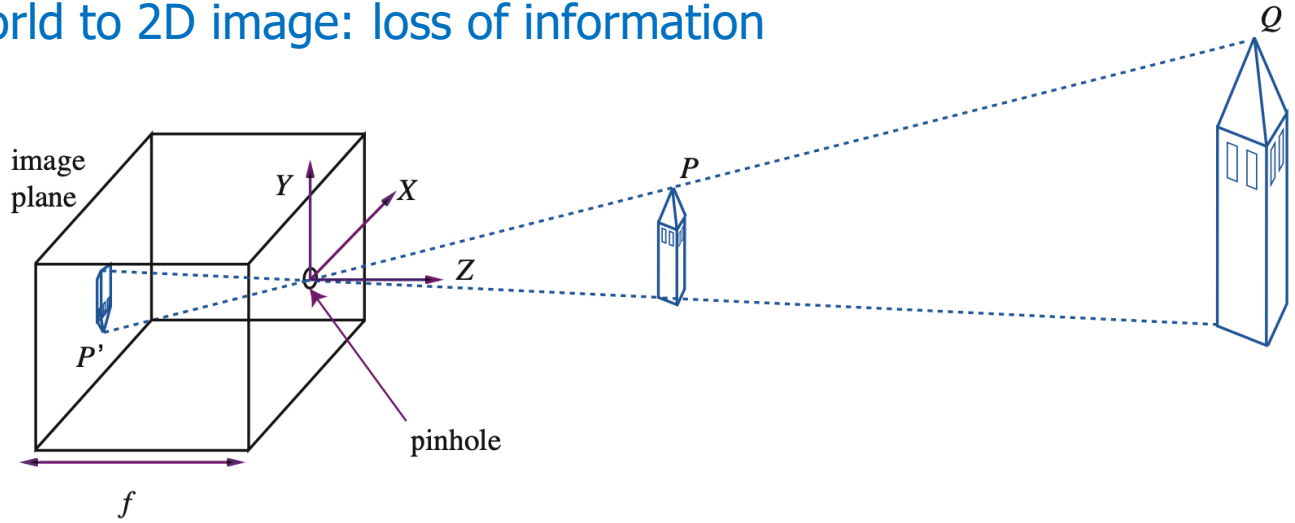
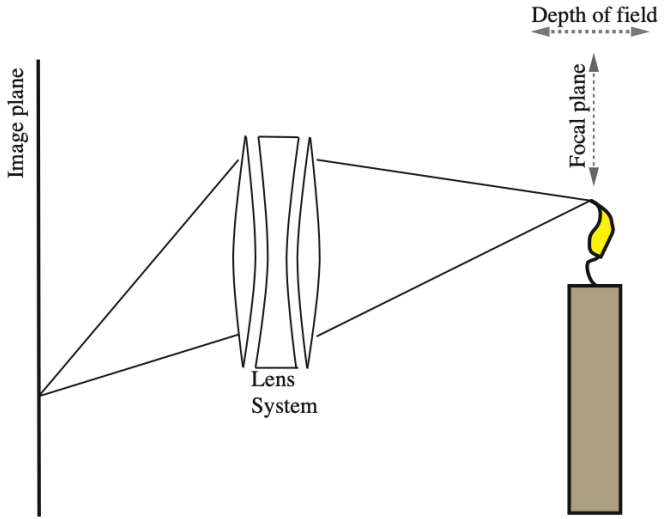
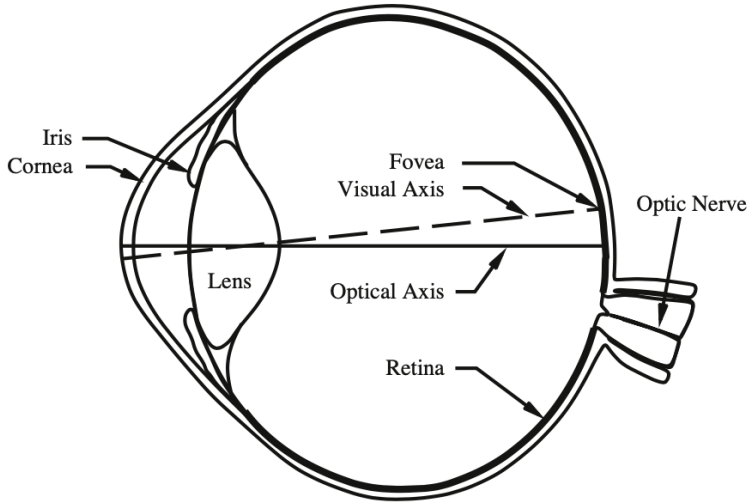


Figure 25.2 Each light sensitive element at the back of a pinhole camera receives light that passes through the pinhole from a small range of directions. If the pinhole is small enough, the result is a focused image behind the pinhole. The process of projection means that large, distant objects look the same as smaller, nearby objects—the point P' in the image plane could have come from a nearby toy tower at point P or from a distant real tower at point Q.

Starting point: optics and image formation

Using lenses yields greater flexibility (field of view, zoom factor, depth of field, etc) but also complicates the geometry

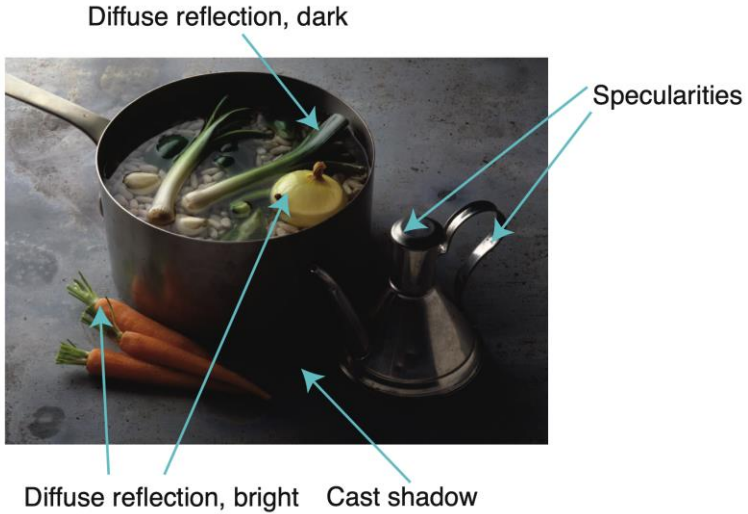


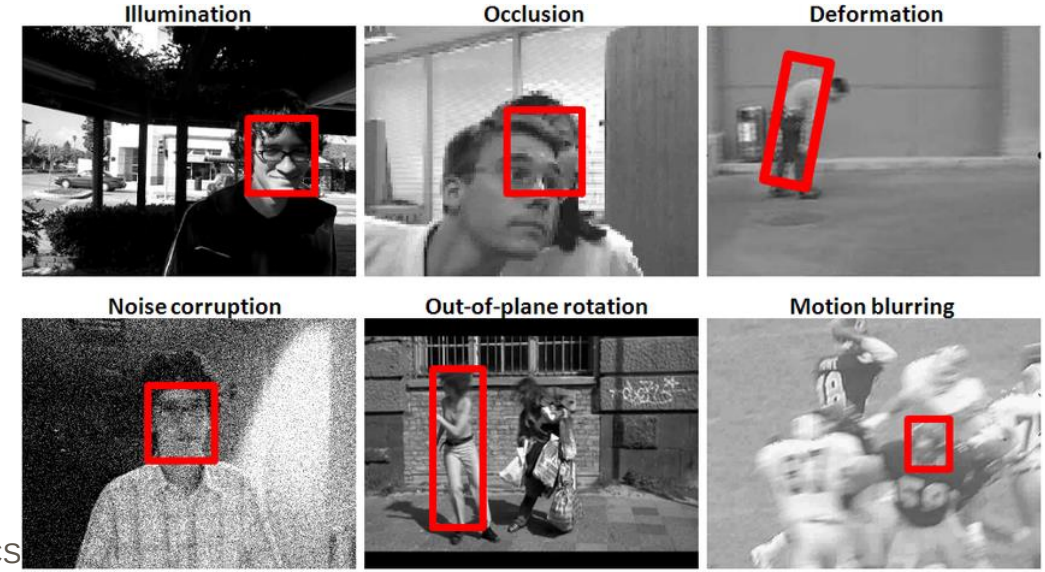


Complications

Because of projection and complications, inferring what was in the real world that gave rise to an image is an ill-posed problem.

Need to overcome "nuisance factors" such as noise, occlusions, changes in pose or illumination, deformations, etc.





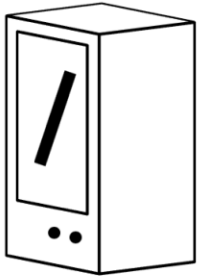
Traditional and new CV approaches

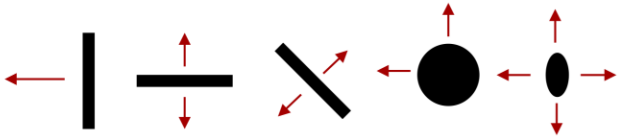
- Traditionally, CV has been rooted in signal processing.
- Algorithms detect ”features” in images, often with optimality guarantees.
- Downstream tasks (recognition, detection, tracking, etc) solved in feature space.

- In recent years: these approaches have been largely replaced by deep learning:
  - Discover features from large datasets.
  - Learn the downstream tasks too.

Hubel and Wiesel, 1959

Measure brain activity



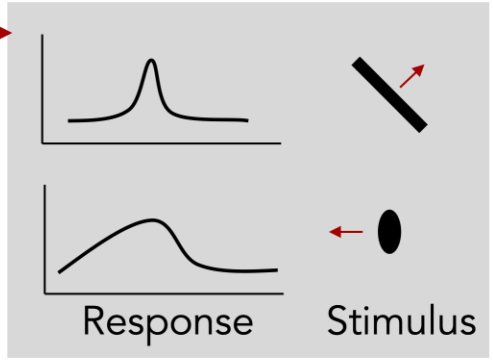


Simple cells:
Response to light orientation

Complex cells:
Response to light orientation and movement

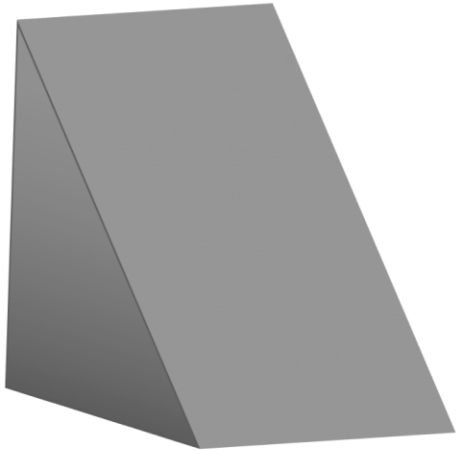
Hypercomplex cells:
response to movement with an end point

No response

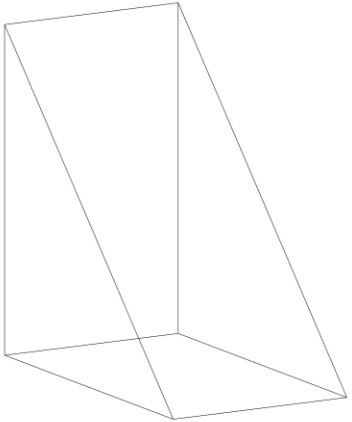


1959
Hubel & Wiesel

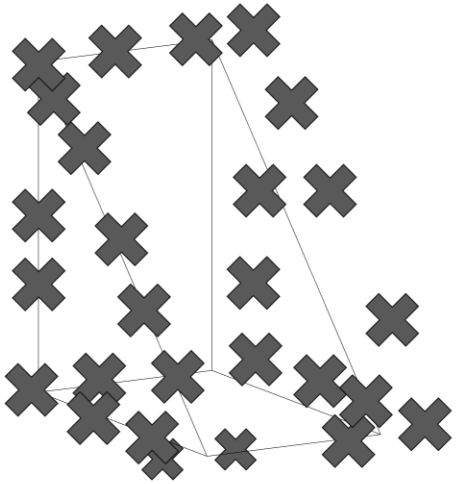
Larry Roberts, 1963



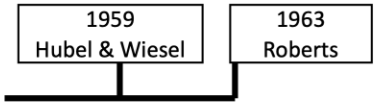
(a) Original picture



(b) Differentiated picture



(c) Feature points selected



MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

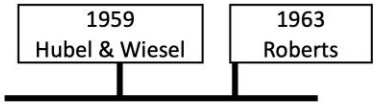
Artificial Intelligence Group
Vision Memo. No. 100.

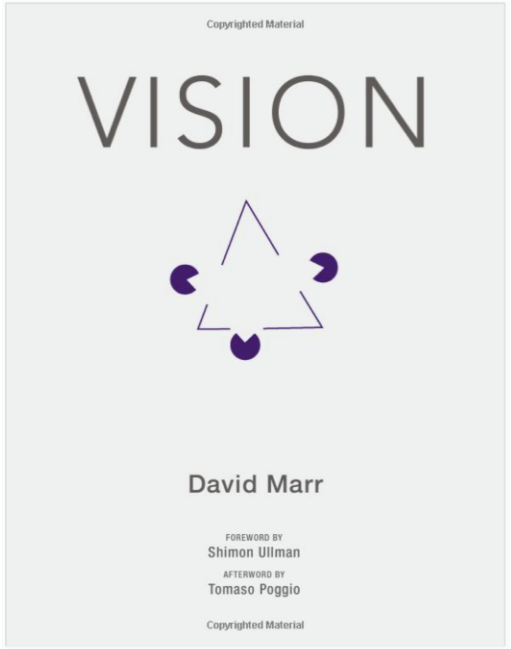
July 7, 1966

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".





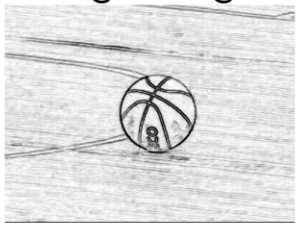
1959 Hubel & Wiesel

1963 Roberts

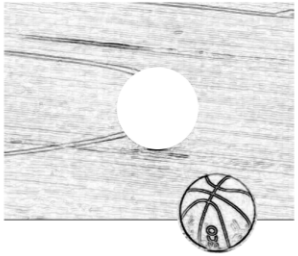
1970s David Marr

Input image

Edge image



2 ½-D sketch

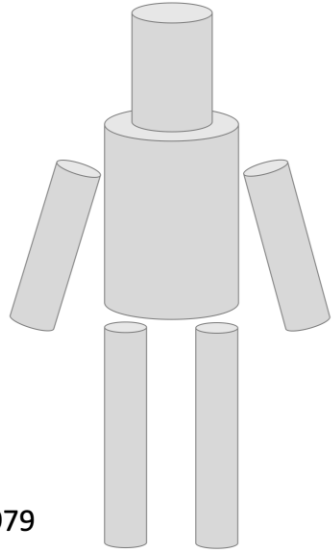


3-D model

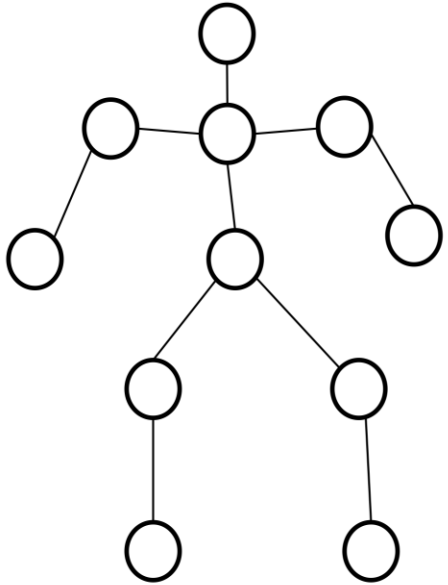
Input Image	Primal Sketch	2 ½-D Sketch	3-D Model Representation
Perceived intensities	Zero crossings, blobs, edges, bars, ends, virtual lines, groups, curves boundaries	Local surface orientation and discontinuities in depth and in surface orientation	3-D models hierarchically organized in terms of surface and volumetric primitives

Stages of Visual Representation, David Marr, 1970s

Recognition via Parts (1970s)

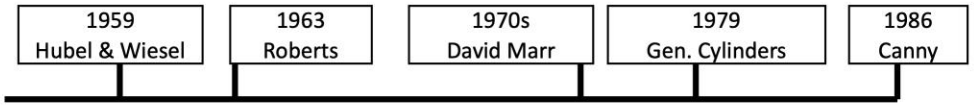


Generalized Cylinders,
Brooks and Binford, 1979



Pictorial Structures,
Fischler and Elshlager, 1973

Recognition via Edge Detection (1980s)



John Canny, 1986
David Lowe, 1987

Recognition via Grouping (1990s)

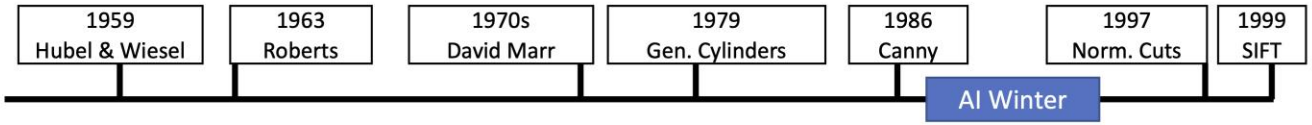
1959 Hubel & Wiesel
1963 Roberts
1970s David Marr
1979 Gen. Cylinders
1986 Canny
AI Winter
1997 Norm. Cuts

Normalized Cuts, Shi and Malik, 1997

Recognition via Matching (2000s)

Image is public domain

Image is public domain

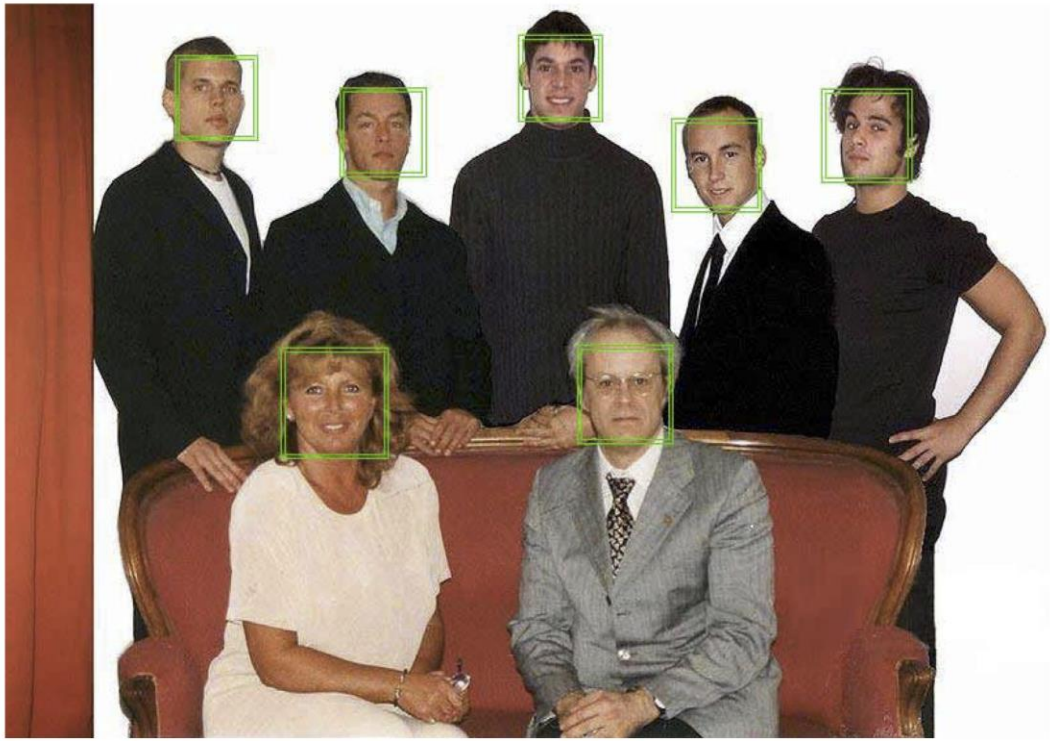


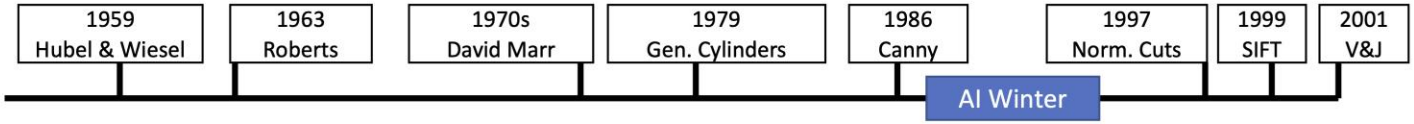
SIFT, David Lowe, 1999

Face Detection

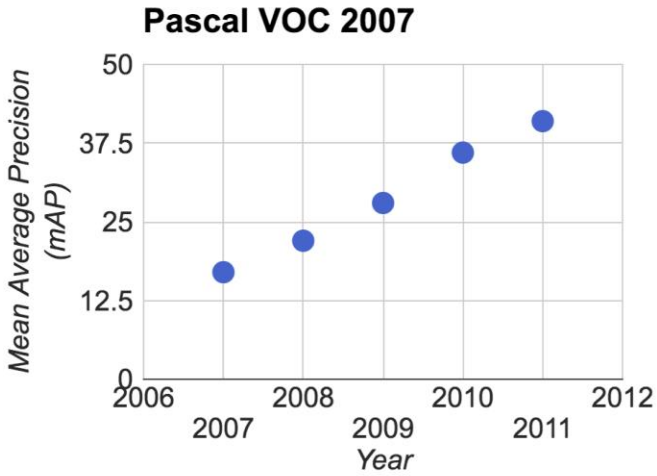
Viola and Jones, 2001

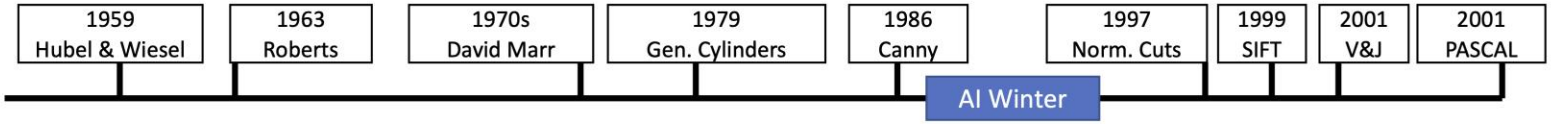
One of the first successful applications of machine learning to vision





PASCAL Visual Object Challenge





IMAGENET Large Scale Visual Recognition Challenge

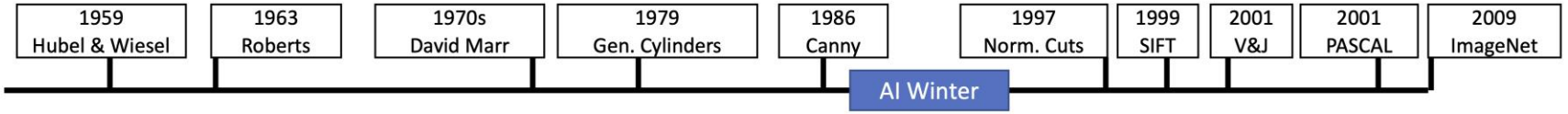
The Image Classification Challenge:

1,000 object classes

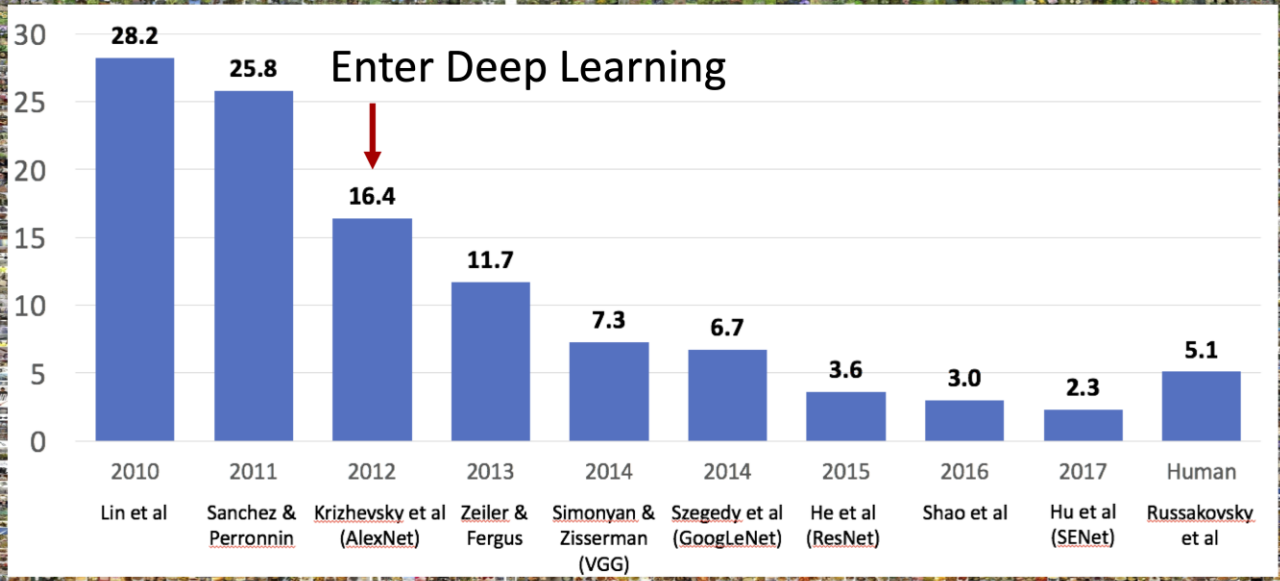
1,431,167 images

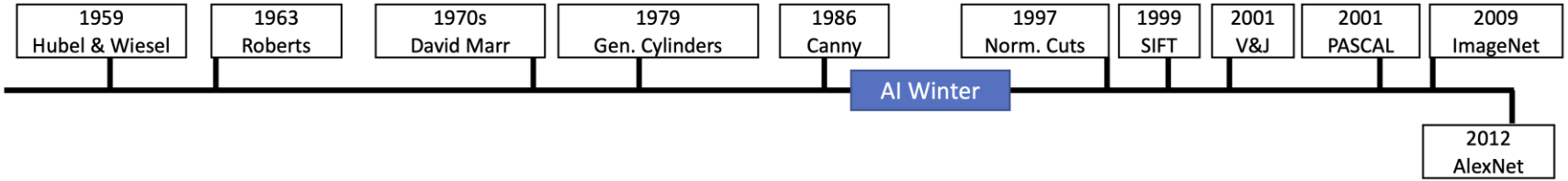
Output:
Scale
T-shirt
Steel drum
Drumstick
Mud turtle

Deng et al, 2009
Russakovsky et al. IJCV 2015

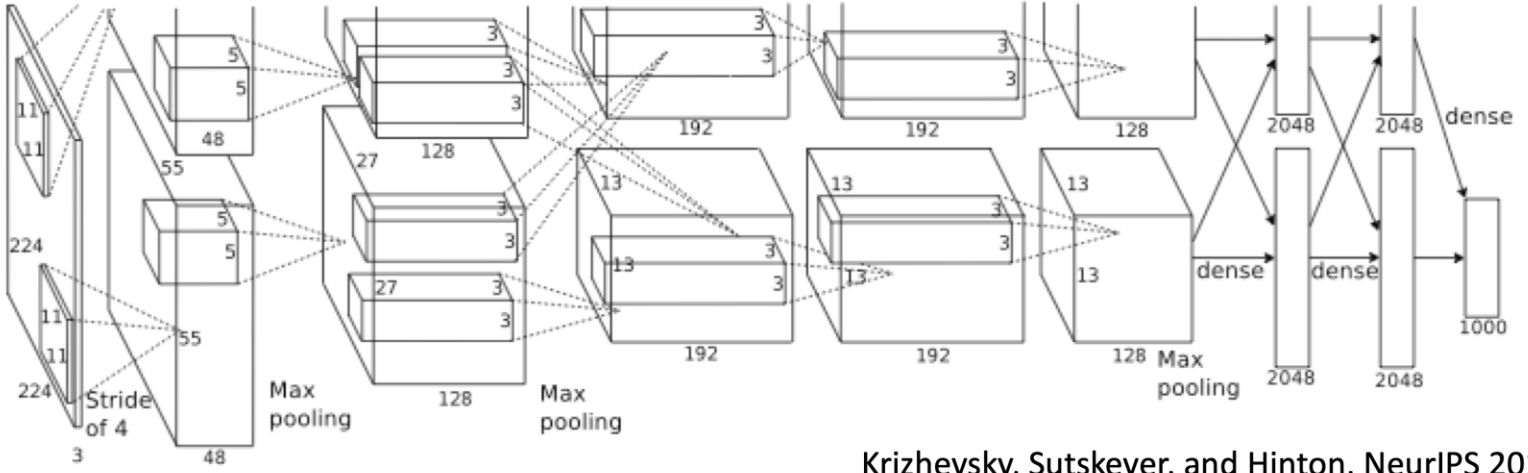


IMAGENET Large Scale Visual Recognition Challenge

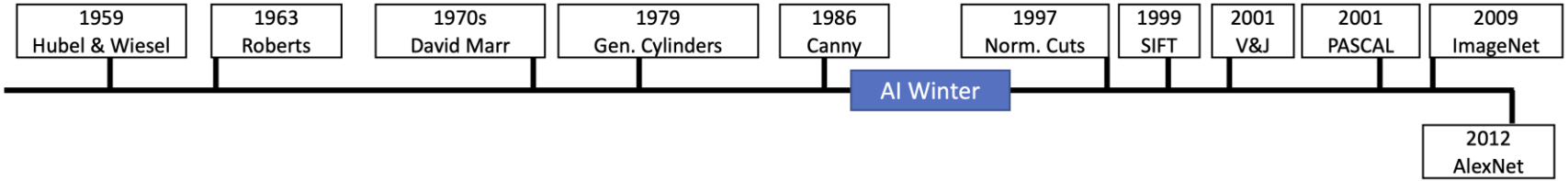




AlexNet: Deep Learning Goes Mainstream



Krizhevsky, Sutskever, and Hinton, NeurIPS 2012



Perceptron

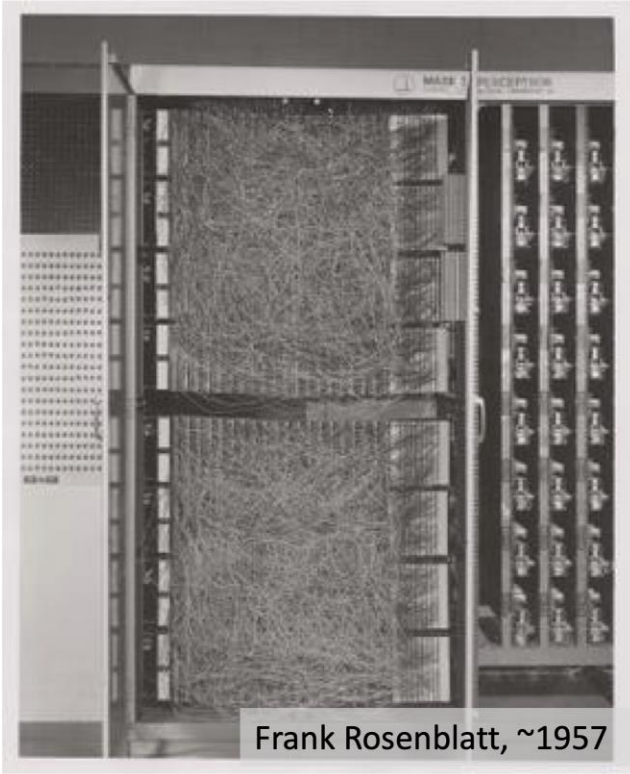
One of the earliest algorithms that could learn from data

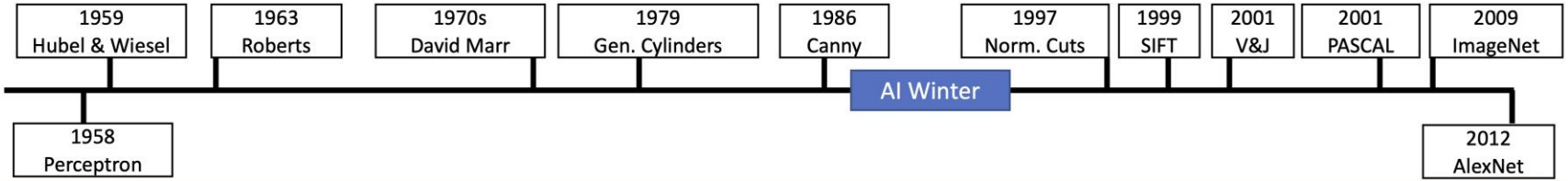
Implemented in hardware! Weights stored in potentiometers, updated with electric motors during learning

Connected to a camera that used 20x20 cadmium sulfide photocells to make a 400-pixel image

Could learn to recognize letters of the alphabet

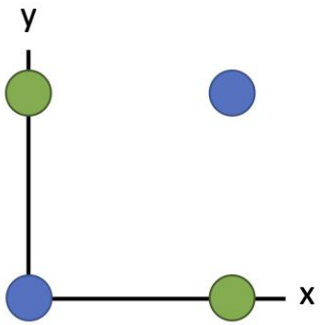
Today we would recognize it as a **linear classifier**

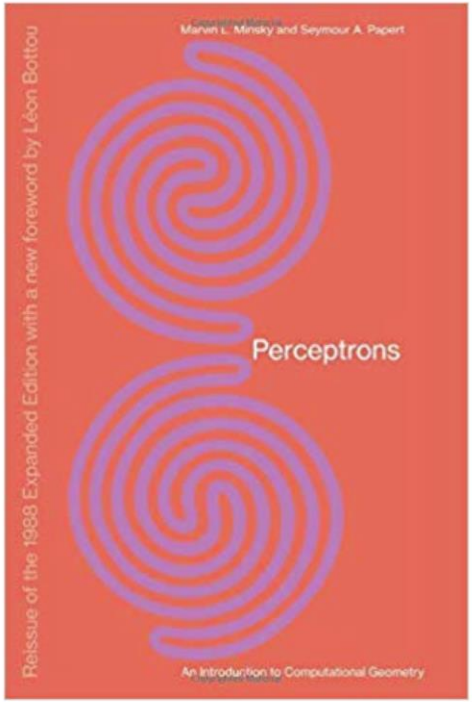




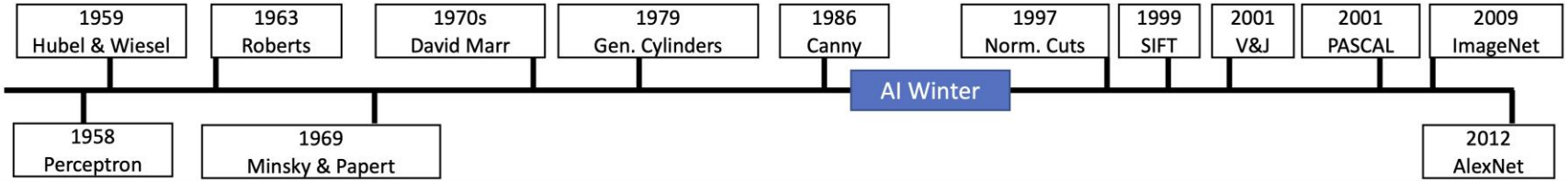
Minsky and Papert, 1969

X	Y	F(x,y)
0	0	0
0	1	1
1	0	1
1	1	0





Showed that Perceptrons could not learn the XOR function
Caused a lot of disillusionment in the field

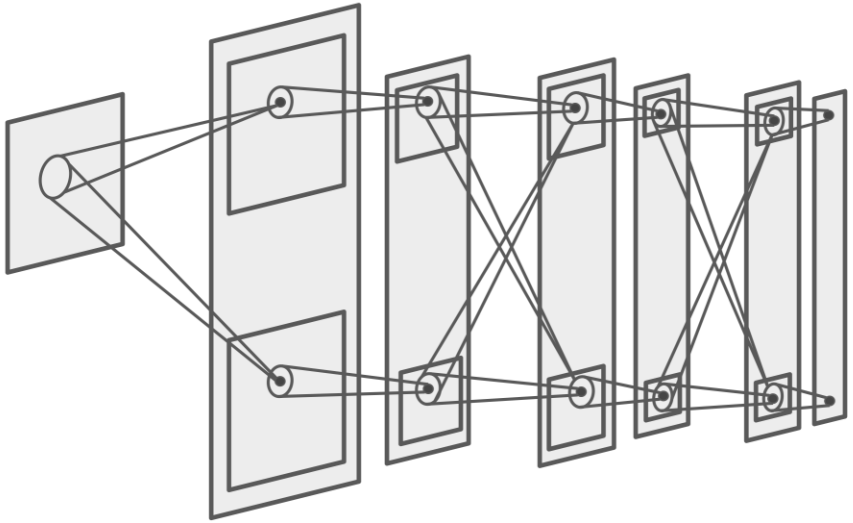


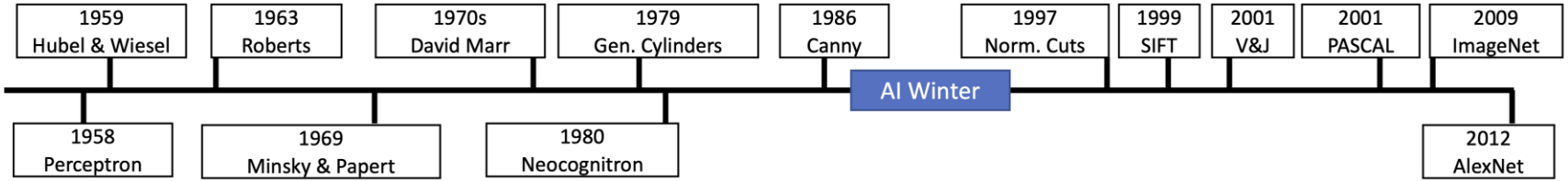
Neocognitron: Fukushima, 1980

Computational model the visual system, directly inspired by Hubel and Wiesel's hierarchy of complex and simple cells

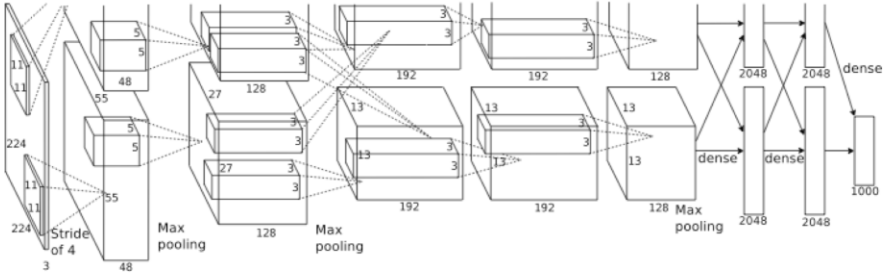
Interleaved simple cells (convolution) and complex cells (pooling)

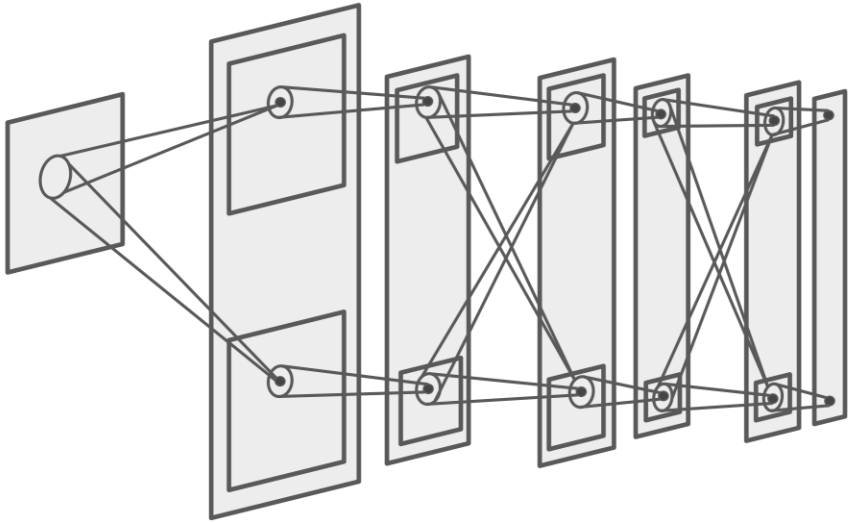
No practical training algorithm



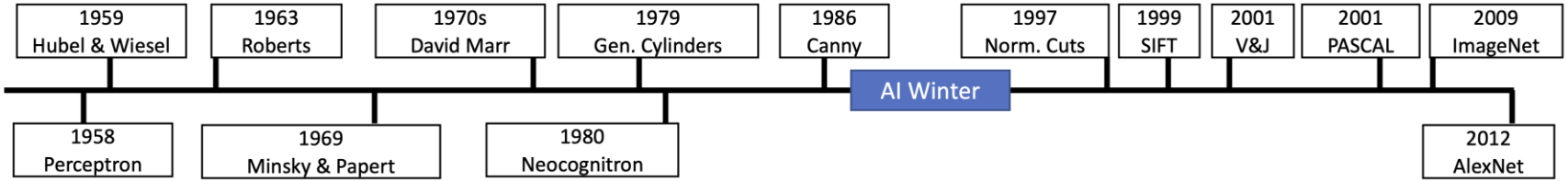


Neocognitron: Fukushima, 1980





Looks a lot like AlexNet
more than 32 years later!



Backprop: Rumelhart, Hinton, and Williams, 1986

Introduced backpropagation for computing gradients in neural networks

Successfully trained perceptrons with multiple layers

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial w_{ji}}$$

recognizable math

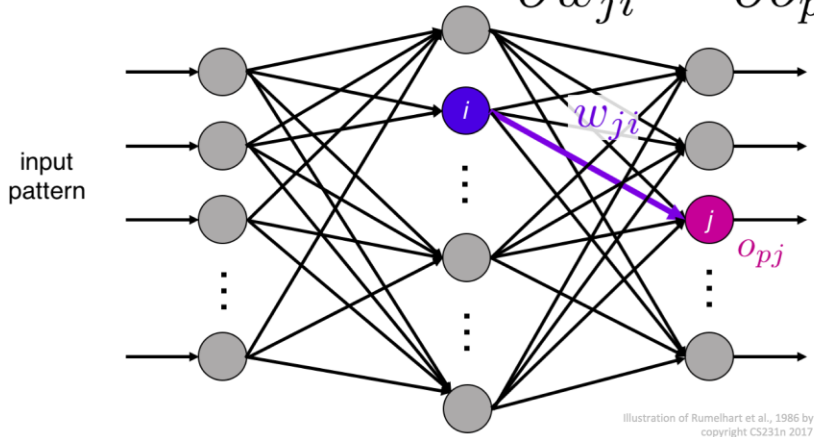
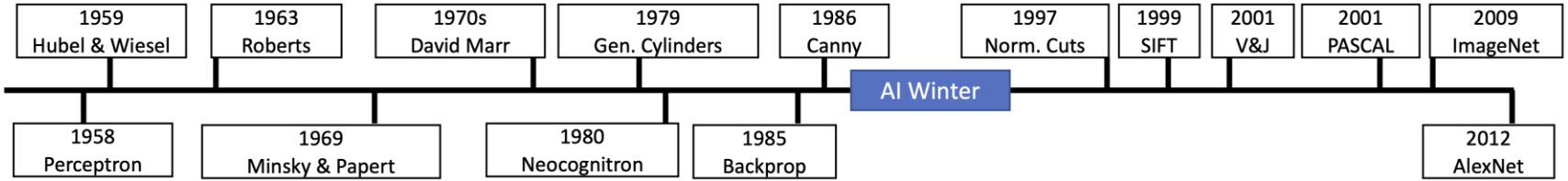
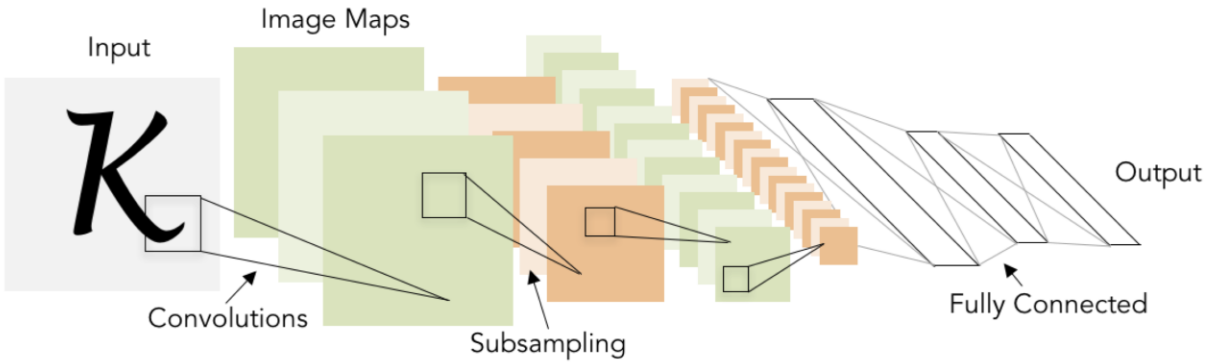


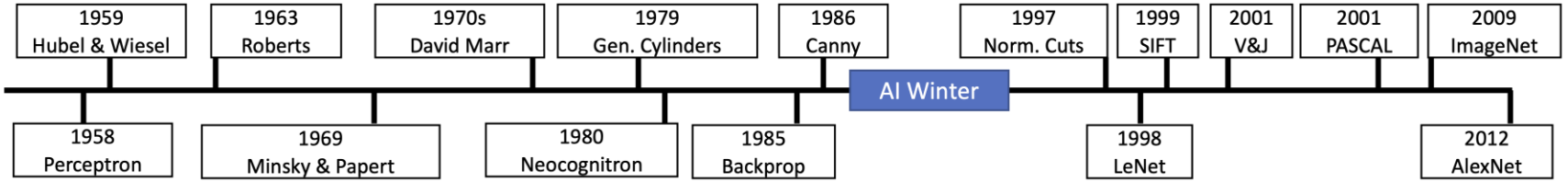
Illustration of Rumelhart et al., 1986 by Lane McIntosh, copyright CS231n 2017



Convolutional Networks: LeCun et al, 1998



Applied backprop algorithm to a Neocognitron-like architecture
Learned to recognize handwritten digits
Was deployed in a commercial system by NEC, processed handwritten checks
Very similar to our modern convolutional networks!

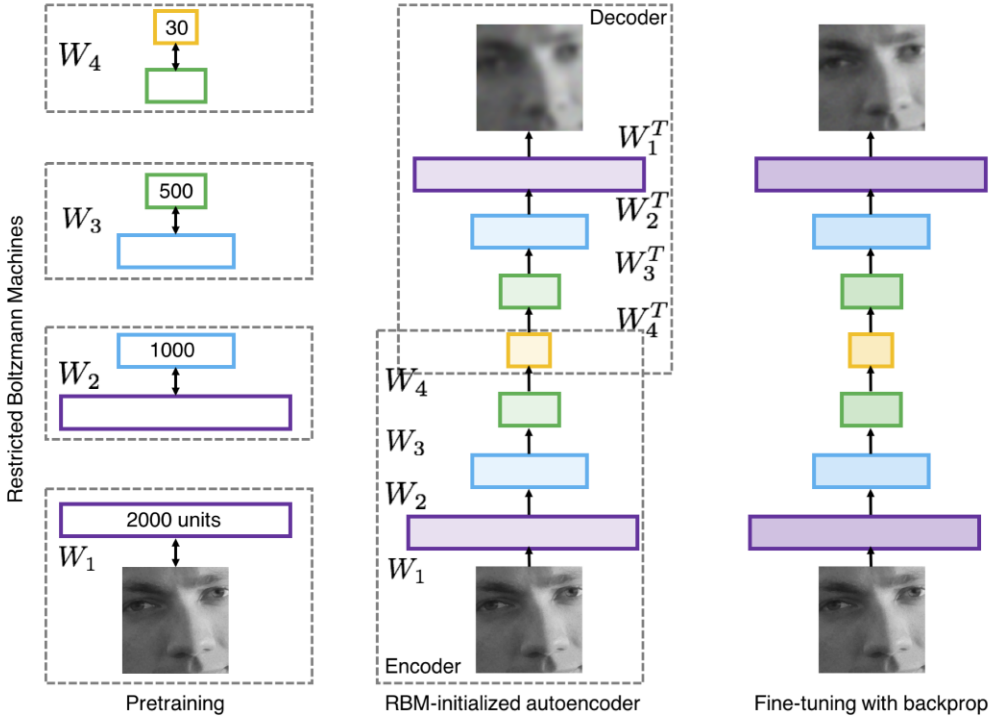


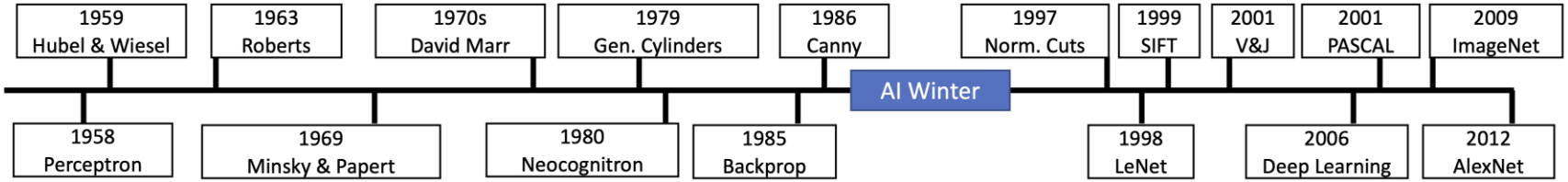
2000s: “Deep Learning”

People tried to train neural networks that were deeper and deeper

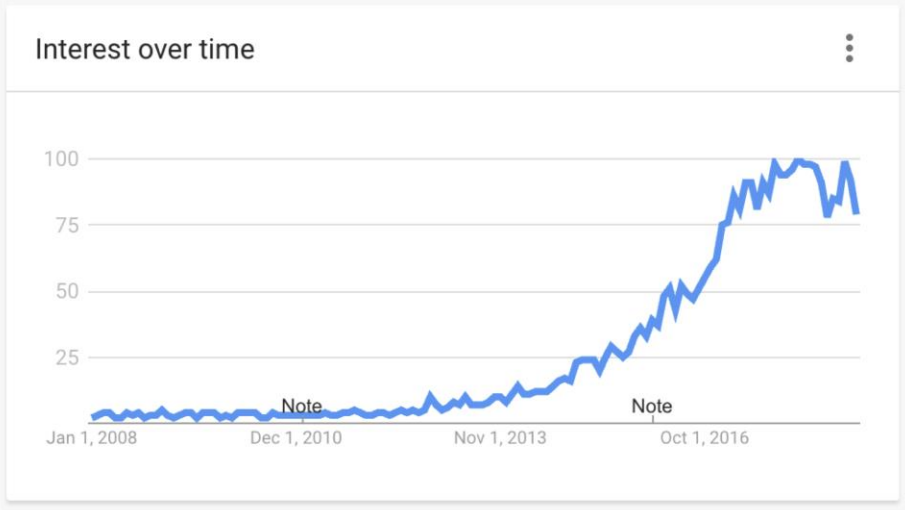
Not a mainstream research topic at this time

Hinton and Salakhutdinov, 2006
Bengio et al, 2007
Lee et al, 2009
Glorot and Bengio, 2010

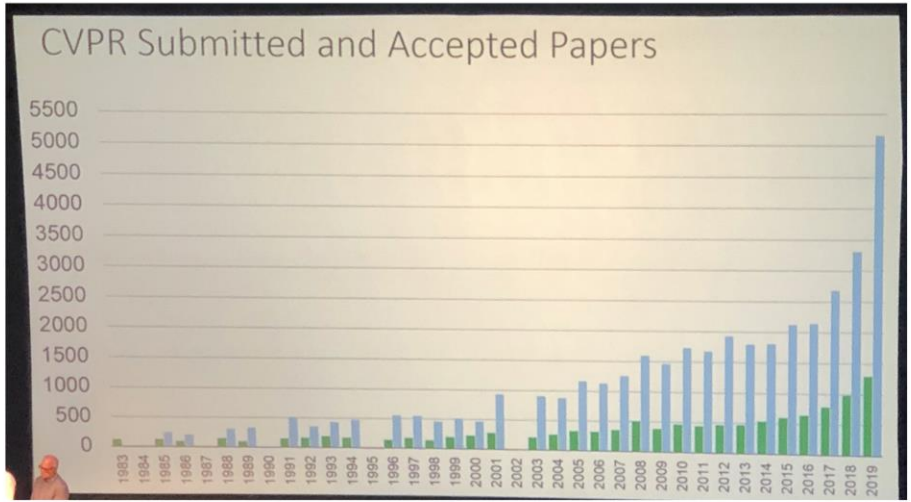




2012 to Present: Deep Learning Explosion



Google Trends: "Deep Learning"



Publications at top Computer Vision conference

1959 Hubel & Wiesel
1963 Roberts
1970s David Marr
1979 Gen. Cylinders
1986 Canny
1997 Norm. Cuts
1999 SIFT
2001 V&J
2001 PASCAL
2009 ImageNet

AI Winter

1958 Perceptron
1969 Minsky & Papert
1980 Neocognitron
1985 Backprop
1998 LeNet
2006 Deep Learning
2012 AlexNet

2012 to Present: ConvNets are everywhere

Image Classification

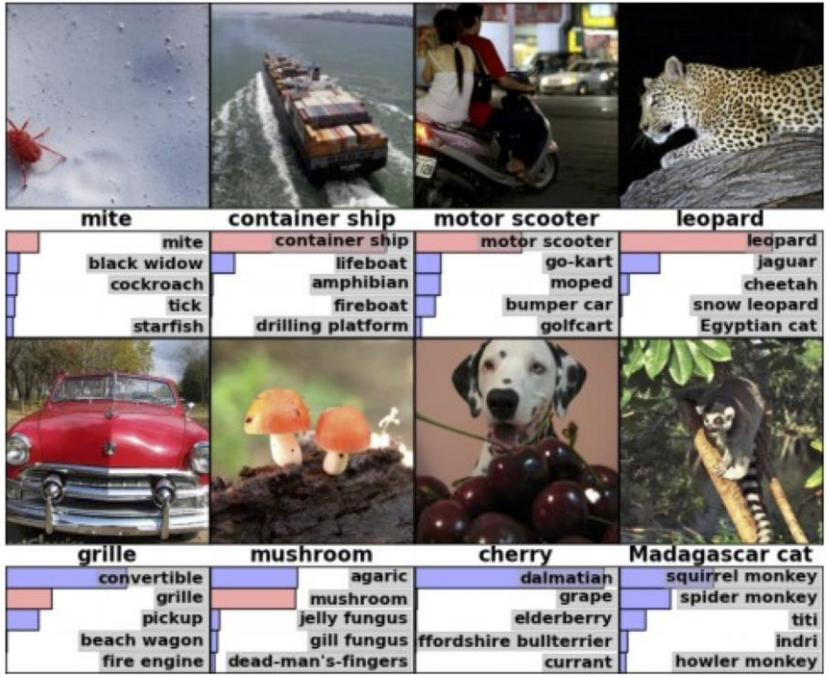


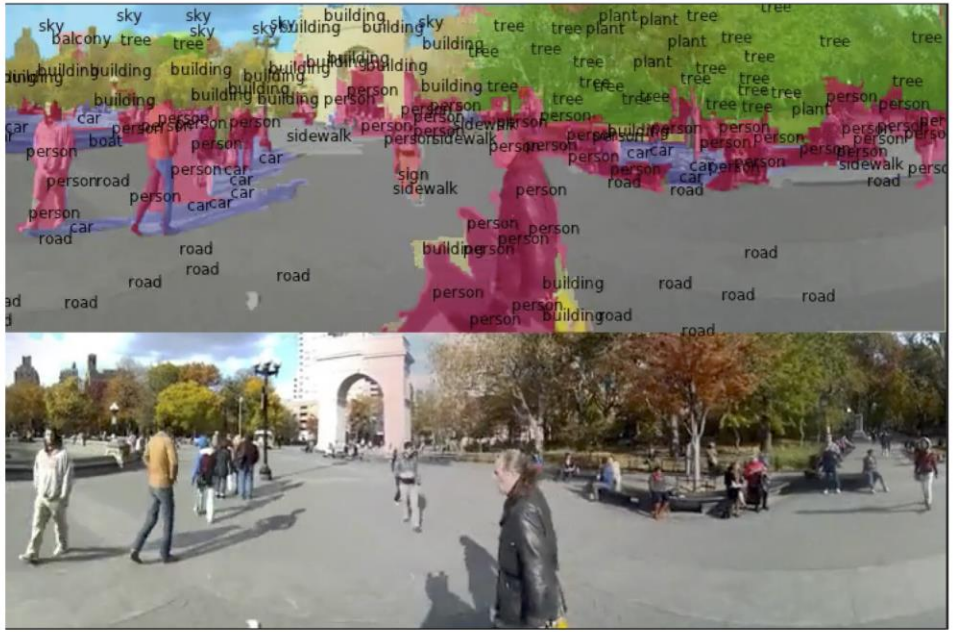
Image Retrieval

2012 to Present: ConvNets are everywhere

Object Detection

Ren, He, Girshick, and Sun, 2015

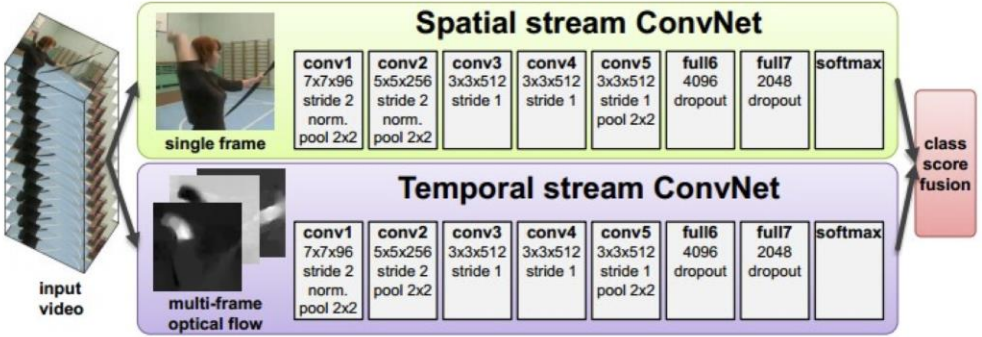
Image Segmentation



Fabaret et al, 2012

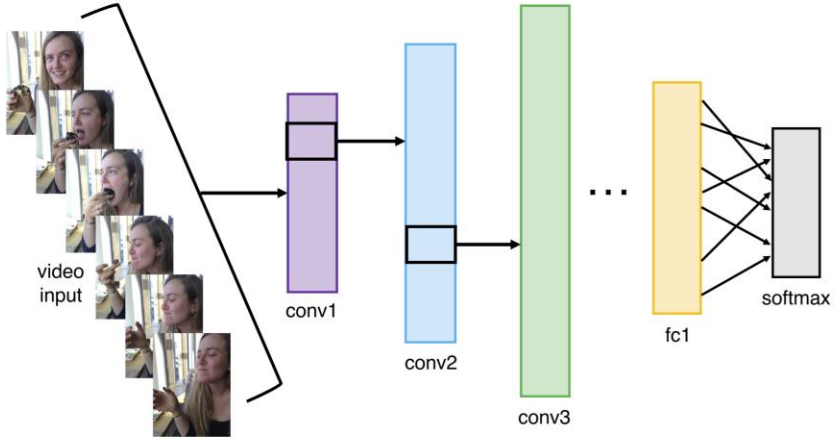
2012 to Present: ConvNets are everywhere

Video Classification



Simonyan et al, 2014

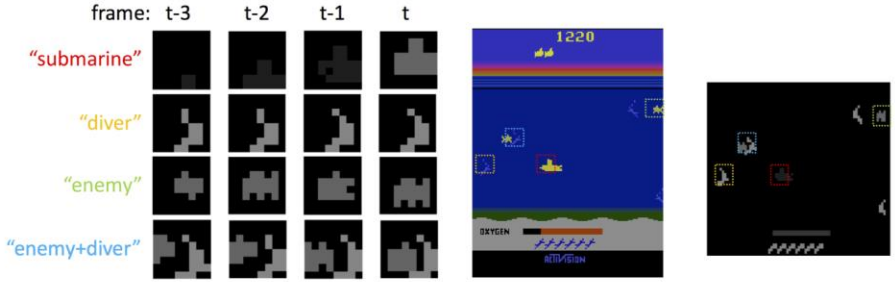
Activity Recognition

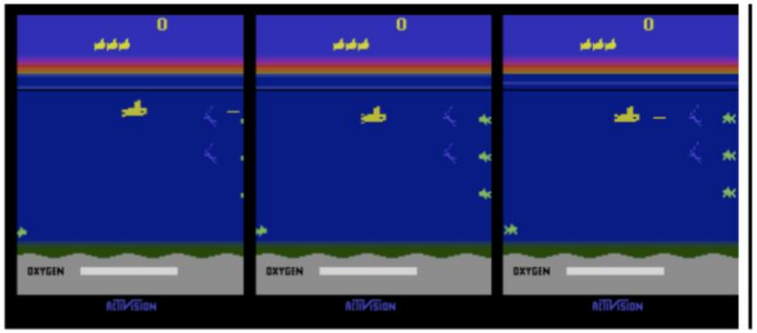


2012 to Present: ConvNets are everywhere

Pose Recognition (Toshev and Szegedy, 2014)

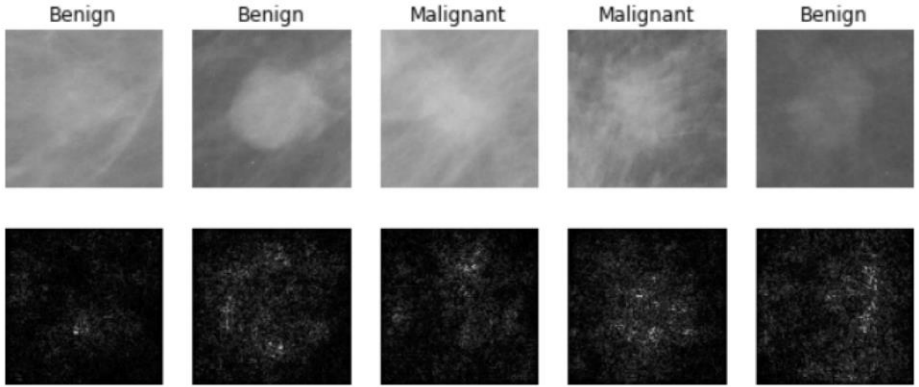
Playing Atari games (Guo et al, 2014)





2012 to Present: ConvNets are everywhere

Medical Imaging



Levy et al, 2016 Figure reproduced with permission

Galaxy Classification

Dieleman et al, 2014

From left to right: public domain by NASA, usage permitted by ESA/Hubble, public domain by NASA, and public domain.

Whale recognition

Kaggle Challenge

This image by Christin Khan is in the public domain and originally came from the U.S. NOAA.

2012 to Present: ConvNets are everywhere

Image Captioning

Vinyals et al, 2015
Karpathy and Fei-Fei, 2015

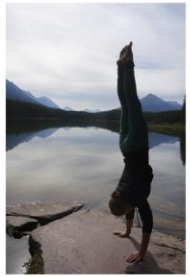
A white teddy bear sitting in the grass

A man in a baseball uniform throwing a ball

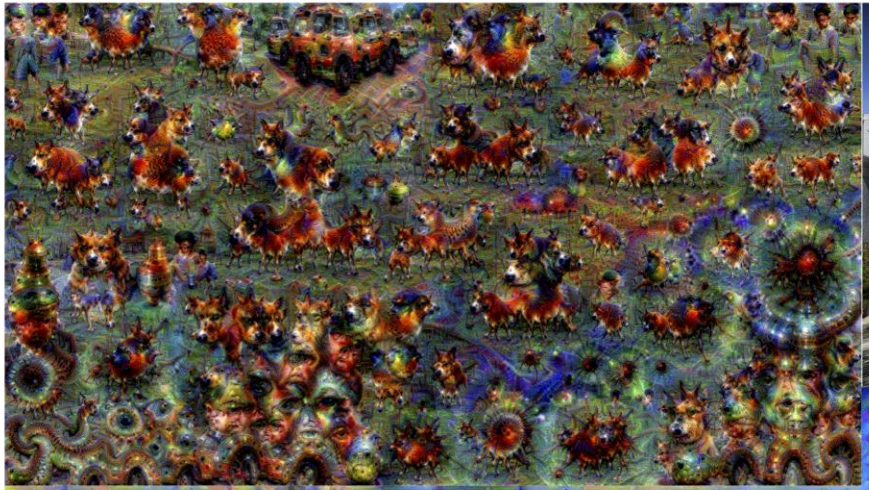
A woman is holding a cat in her hand

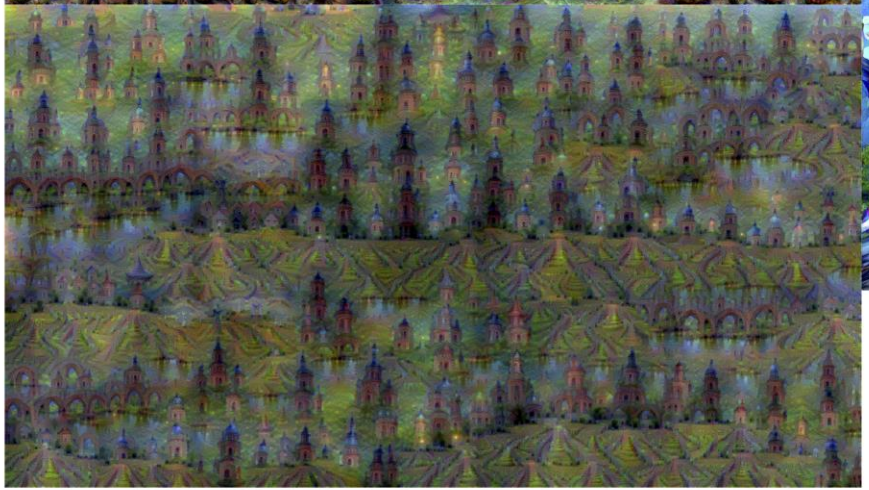
A man riding a wave on top of a surfboard

A cat sitting on a suitcase on the floor



A woman standing on a beach holding a surfboard





Figures copyright Justin Johnson, 2015. Reproduced with permission. Generated using the Inceptionism approach from a blog post by Google Research.

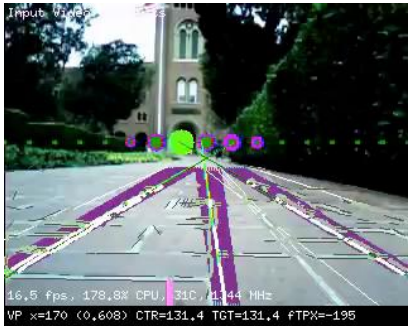
Original image is CC0 public domain
Starry Night and Tree Roots by Van Gogh are in the public domain
Bokeh image is in the public domain
Stylized images copyright Justin Johnson, 2017; reproduced with permission

Mordvinsev et al, 2015
Gatys et al, 2016

Non-AI computer vision

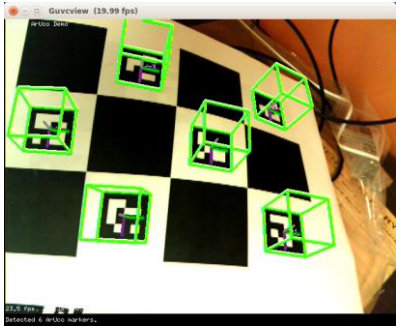
Many algorithms that work really well!

Barcode/QRcode



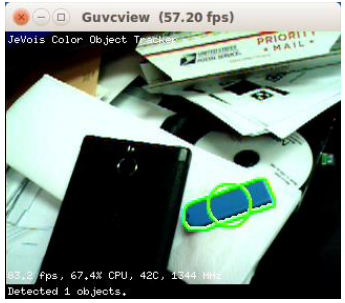
Road following

Attention/saliency



VR markers

Object matching



Tracking

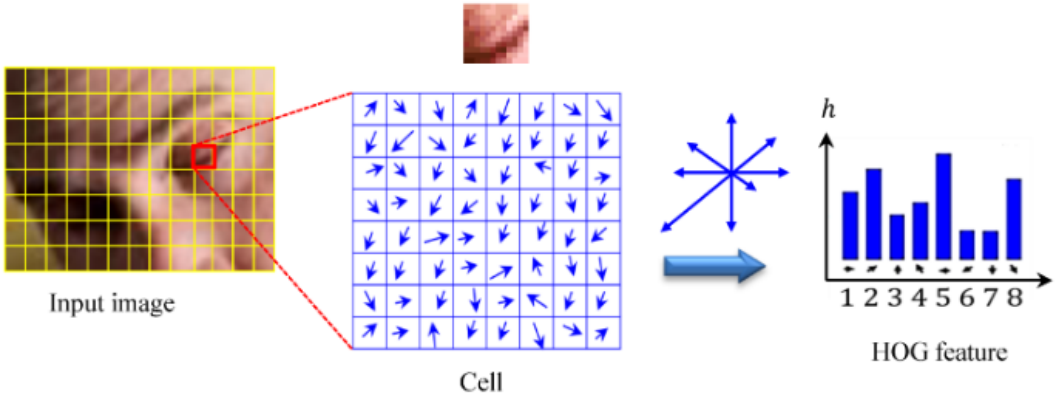
Counting

etc etc…

Non-AI computer vision

Not as good for detecting and recognizing objects/animals/things in the wild…

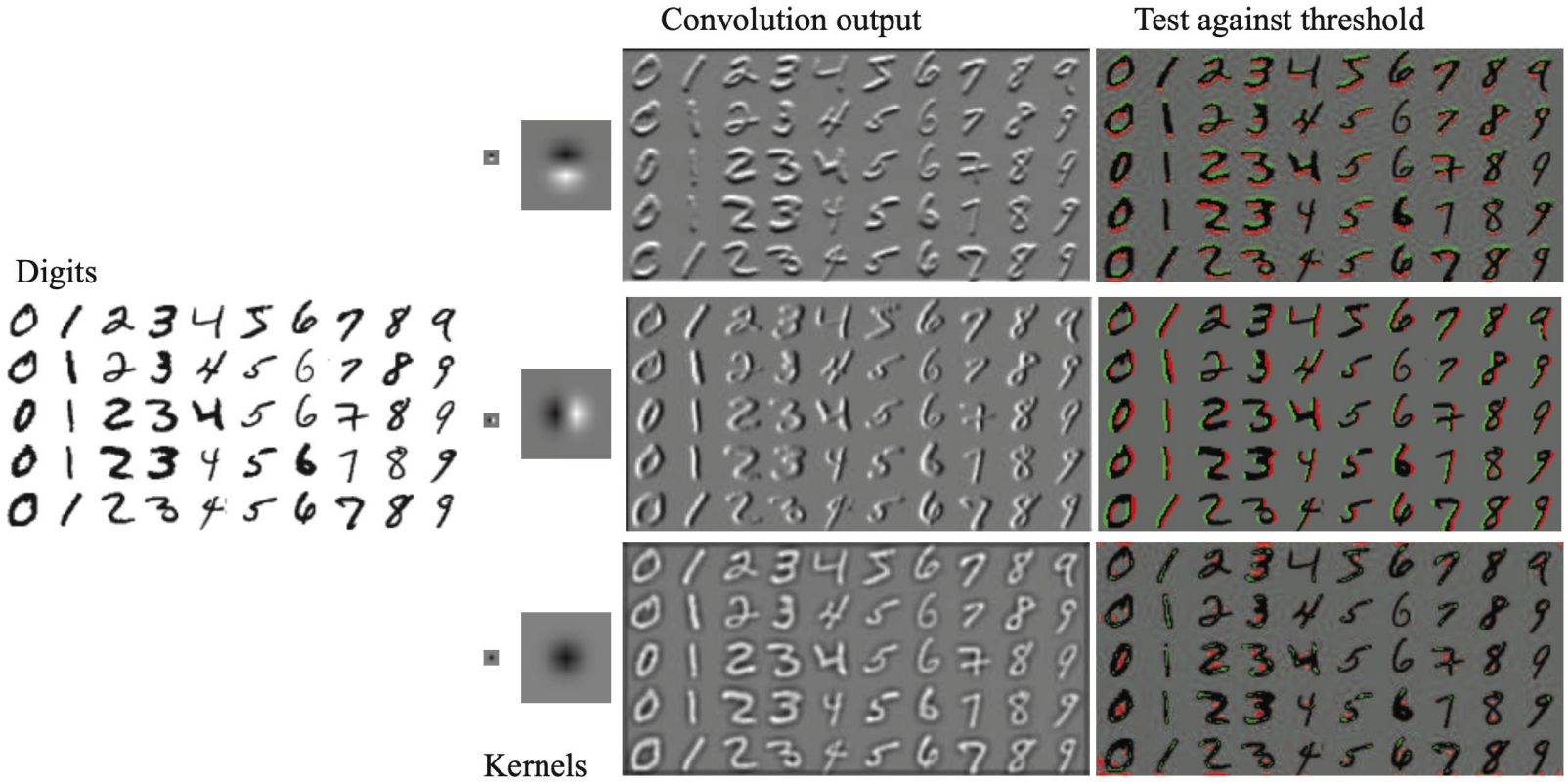
State of the art before AlexNet: Histograms of Oriented Gradients (HoG) features + SVM classifier



https://medium.com/@danyang95luck/comparison-of-hog-histogram-of-oriented-gradients-and-sift-scale-invariant-feature-transform-e2b17f61c9a3

AI / deep learning for computer vision

Green for > threshold
Red for < threshold



AI / deep learning for computer vision

Green for > threshold
Red for < threshold

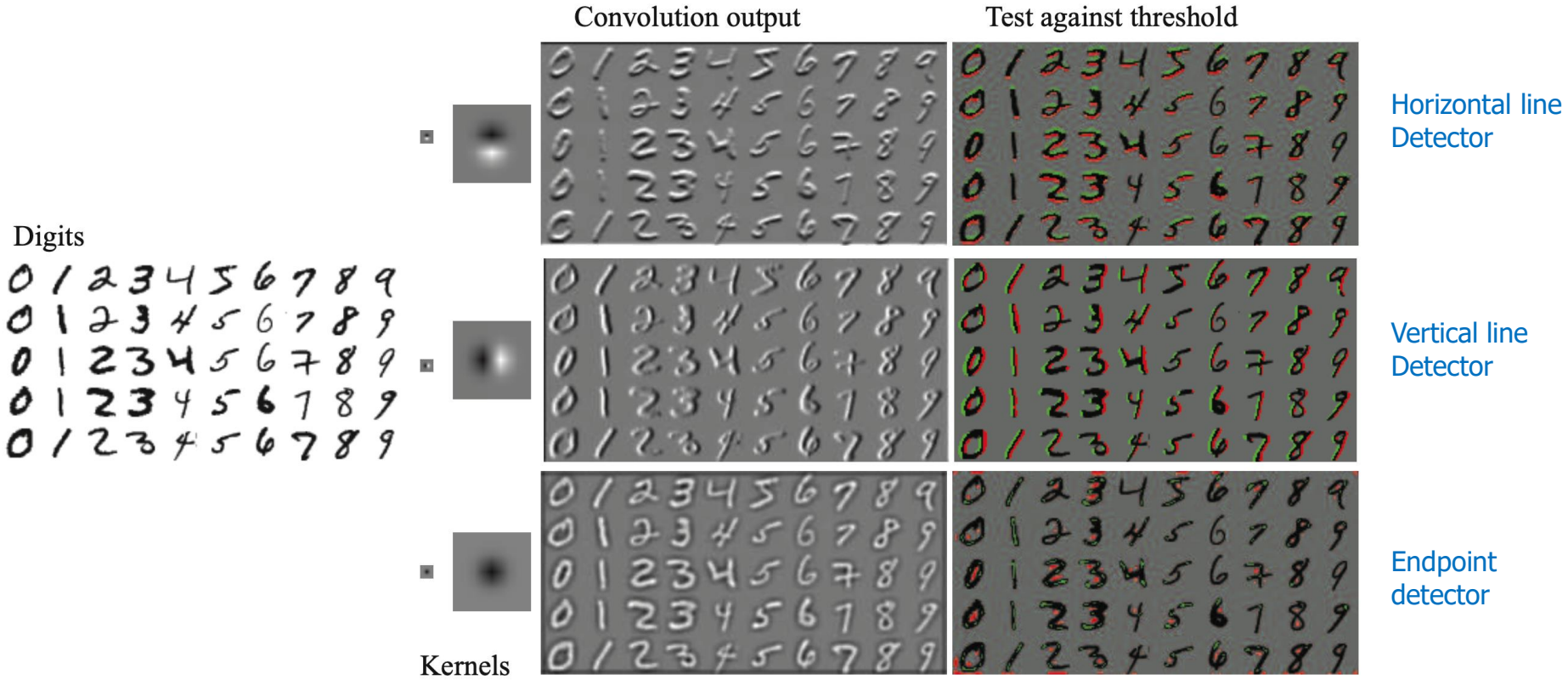
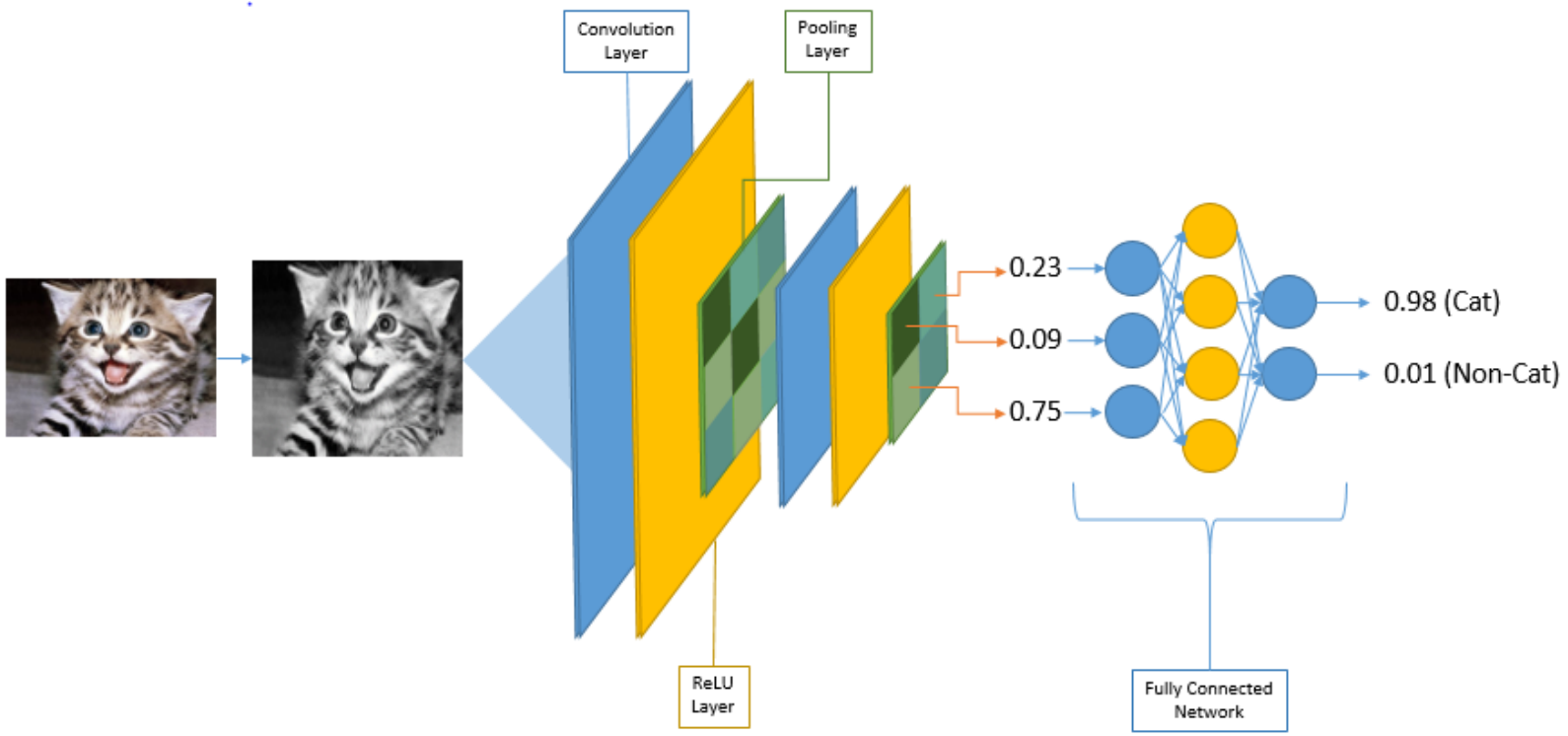


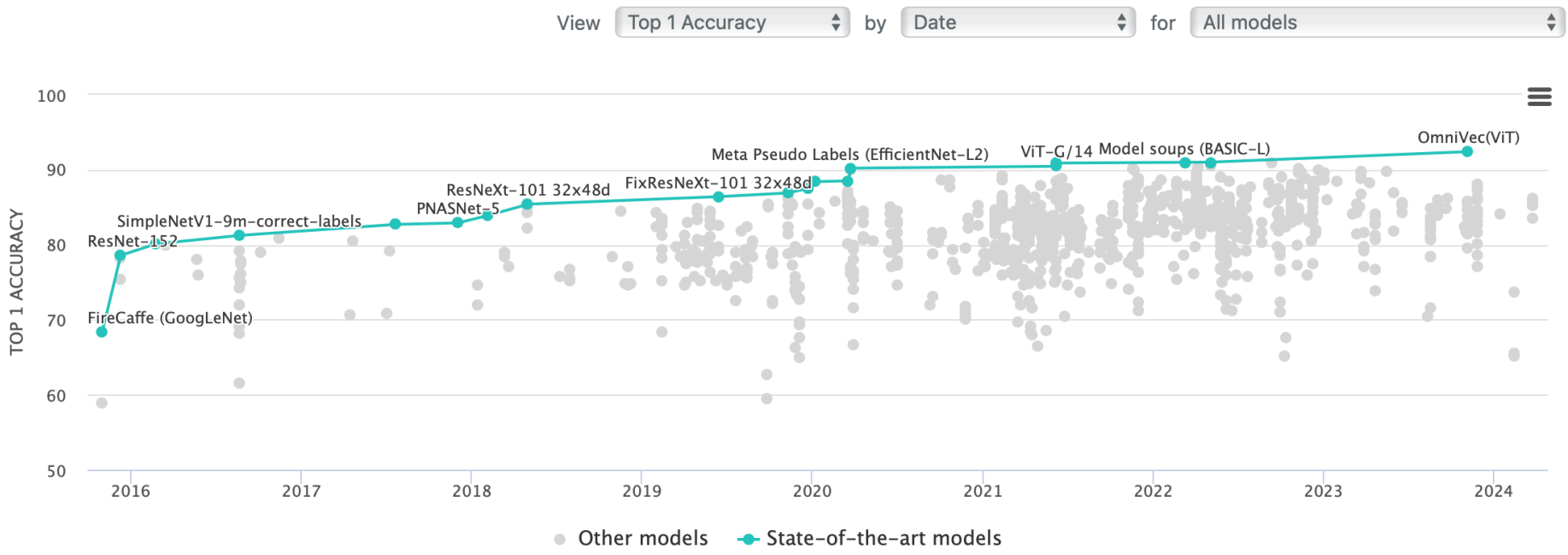
Image classification



https://www.codeproject.com/Articles/5160467/Image-Classification-Using-Neural-Networks-in-NET

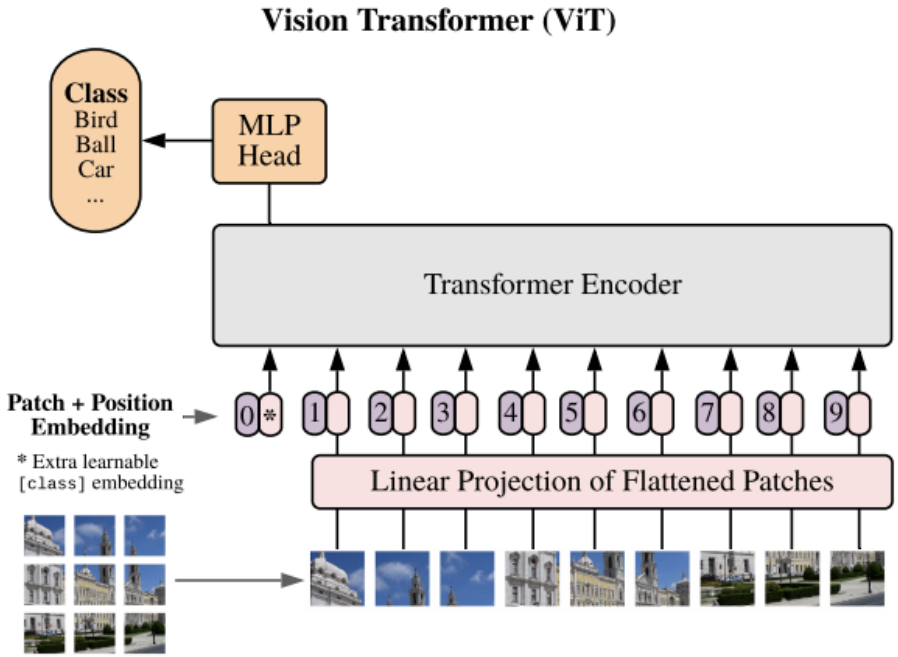
Image Classification on ImageNet

Leaderboard Dataset



https://paperswithcode.com/sota/image-classification-on-imagenet

Vision transformer



https://arxiv.org/abs/2010.11929v2

Image classification: OmniVec

Key ideas:
- Visual transformer
- Leverage knowledge across multiple modalities to achieve better embedding & generalization
- Learning multiple tasks on multiple modalities yields a stronger network

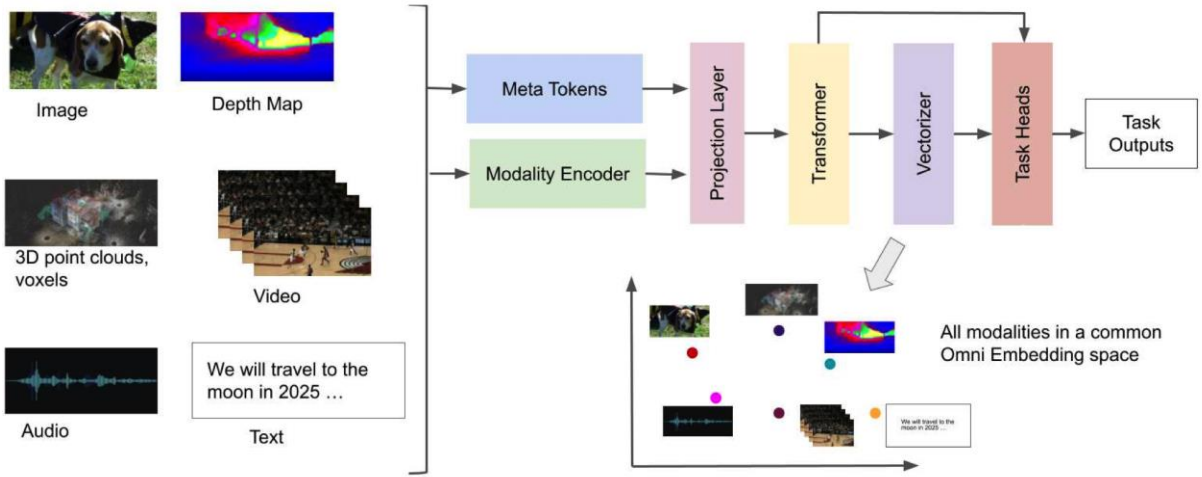


Figure 1. **OmniVec:** The proposed method takes data from one of the modalities and pass it through the modality encoder and combine it with the meta token and then pass through the projection layer to embedd the feature onto a common embedding space. Then it is passes through the common backbone of Transformer layers which is then vectorized by the vectorizer. Finally, the task heads are used for task specific outputs.

https://arxiv.org/pdf/2311.05709v1.pdf

Object detection + recognition

Early approaches:
Cascade 2 networks:

- First detect candidate boxes that may contain objects
- Then apply an image classification network to each box

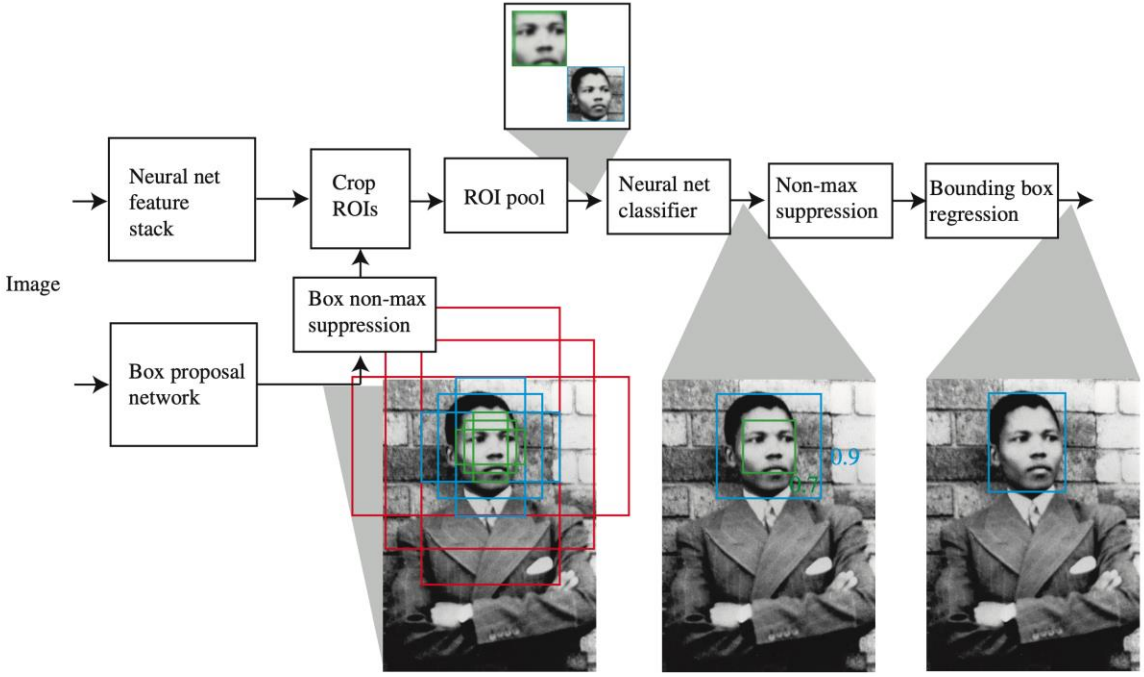


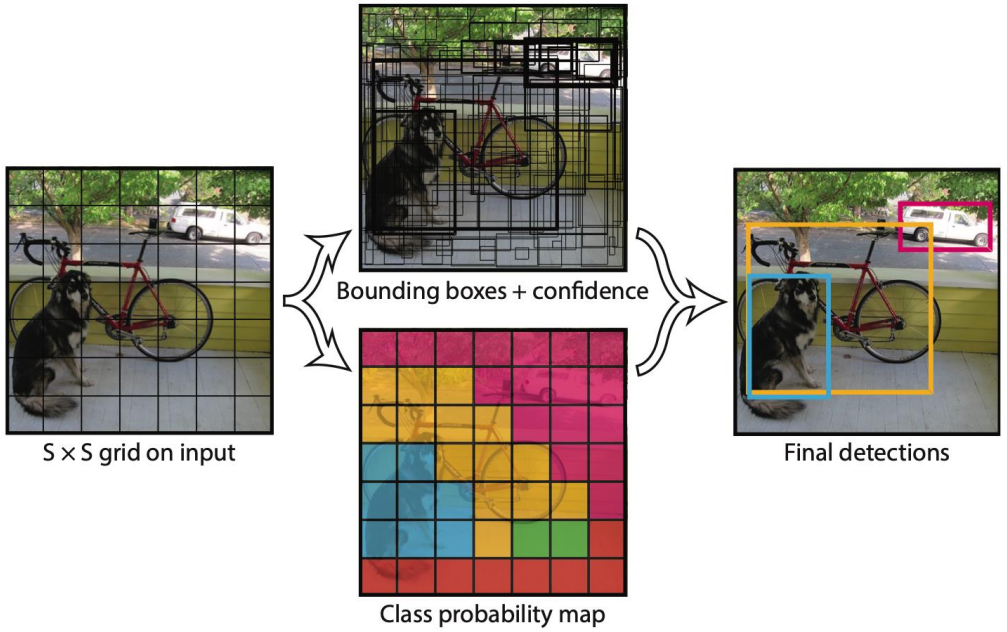
Figure 25.13 Faster RCNN uses two networks. A picture of a young Nelson Mandela is fed into the object detector. One network computes "objectness" scores of candidate image boxes, called "anchor boxes," centered at a grid point. There are nine anchor boxes (three scales, three aspect ratios) at each grid point. For the example image, an inner green box and an outer blue box have passed the objectness test. The second network is a feature stack that computes a representation of the image suitable for classification. The boxes with highest objectness score are cut from the feature map, standardized in size with ROI pooling, and passed to a classifier. The blue box has a higher score than the green box and overlaps it, so the green box is rejected by non-maximum suppression. Finally, bounding box regression the blue box so that it fits the face. This means that the relatively coarse sampling of locations, scales, and aspect ratios does not weaken accuracy. Photo by Sipa/Shutterstock.

Object detection + recognition: YOLO (you only look once)

Basic idea (https://pjreddie.com/darknet/yolo/):

- Prior detection systems repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections.

- We use a totally different approach. We apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.



https://arxiv.org/pdf/1506.02640.pdf

Object detection + recognition: YOLO (you only look once)

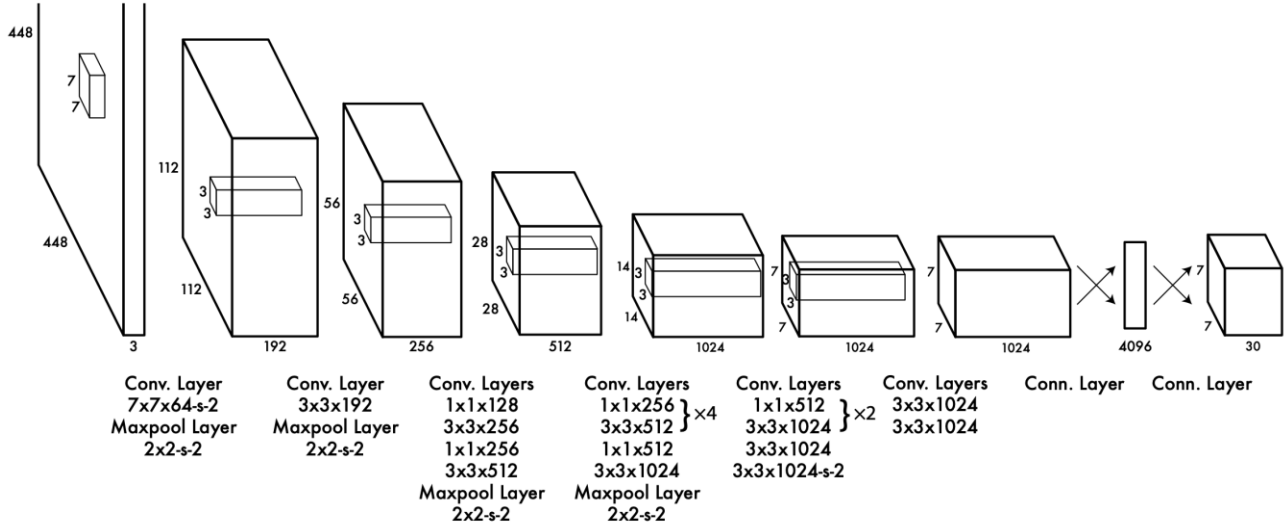


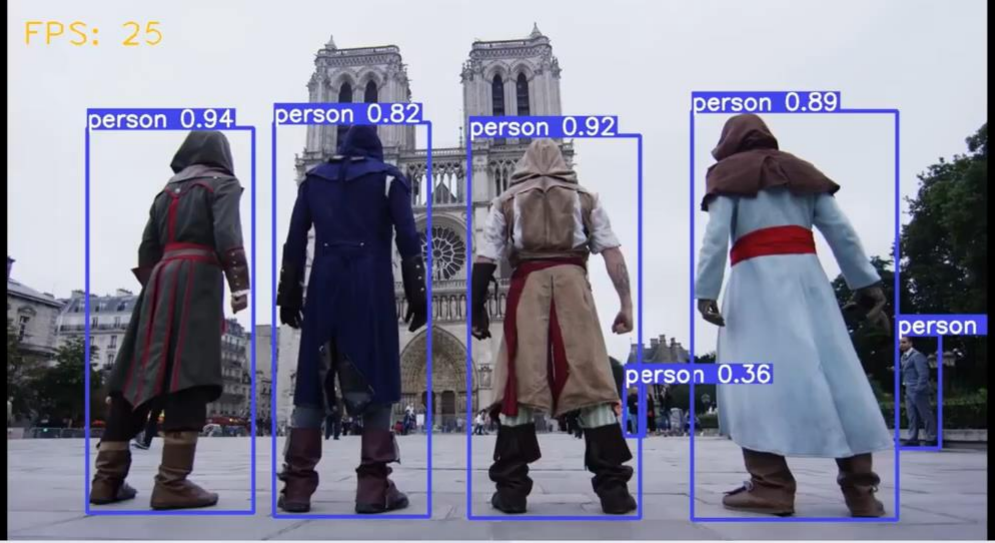
Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

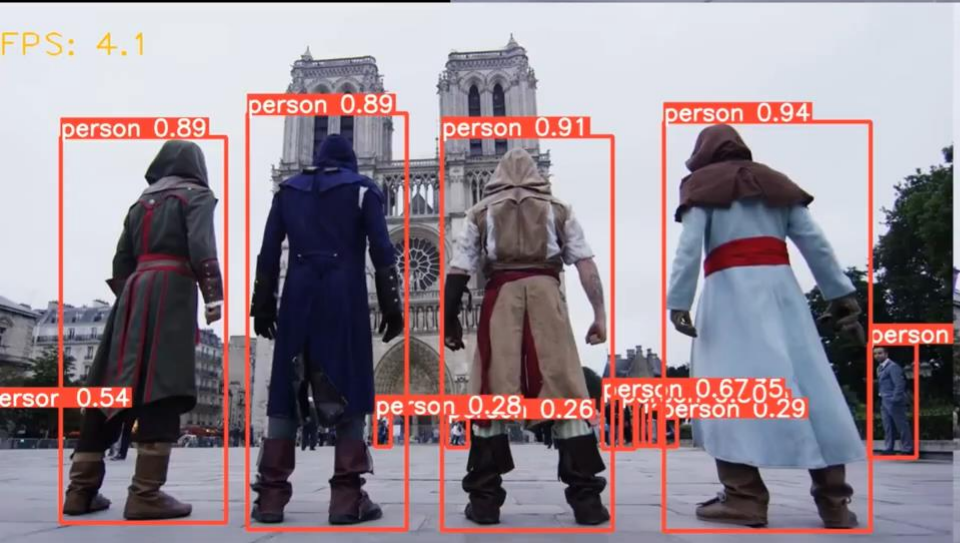
https://arxiv.org/pdf/1506.02640.pdf

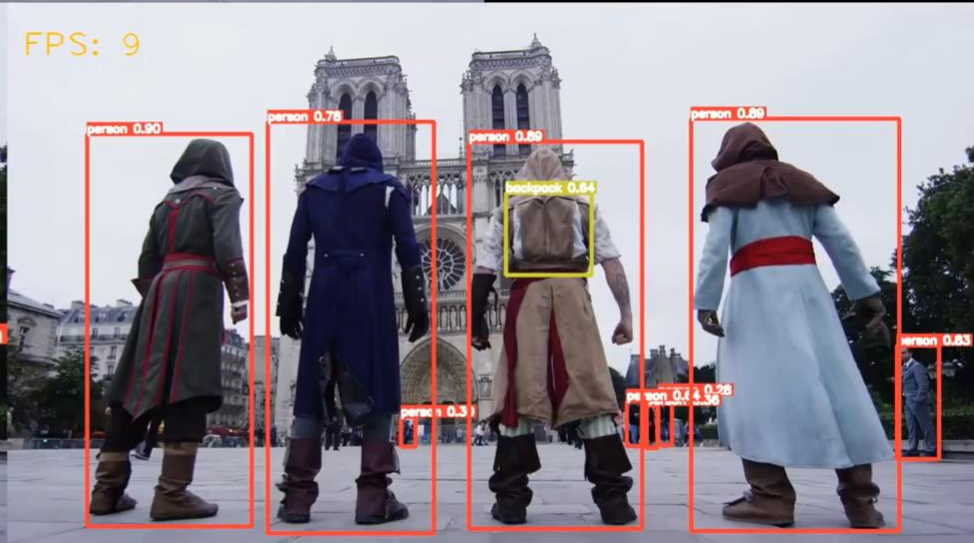

p0:OpenCV:Detect:YoloV3-Tiny-Coco
PreProc: 4.6ms (215.2fps)
Network: 358.7ms (2.8fps)
PstProc: 7.4ms (135.8fps)
OVERALL: 337.54ms/inference
54.58
p1:NPU:Detect:YoloV2
PreProc: 1.9ms (524.1fps)
Network: 37.3ms (26.8fps)
PstProc: 1.4ms (725.3fps)
OVERALL: 42.81ms/inference
30.38
p2:TPU:Detect:MobileDetSSD-Coco
PreProc: 1.5ms (657.3fps)
Network: 8.4ms (119.3fps)
PstProc: 28.8us (34733.1fps)
OVERALL: 13.36ms/inference
54.69
p3:VPU:Detect:pedestrian-and-vehicle-detector-ad 0001
PreProc: 6.2ms (161.1fps)
Network: 134.7ms (7.4fps)
PstProc: 38.7us (25862.5fps)
OVERALL: 141.60ms/inference
vehicle 85
81.93
30.5 fps, 387.2% CPU, 86C, 2208 MHz, Camera: YUYV:1920x1080 + RGB3:1024x576, Display: RGBA:1920x1080


29.7 fps, 279.3% CPU, 89C, 2208 MHz, Camera: YUYV:1920x1080 + RGB3:1024x576, Display: RGBA:1920x1080

AUGMENTED STARTUPS
Computer Vision | AI | Drones

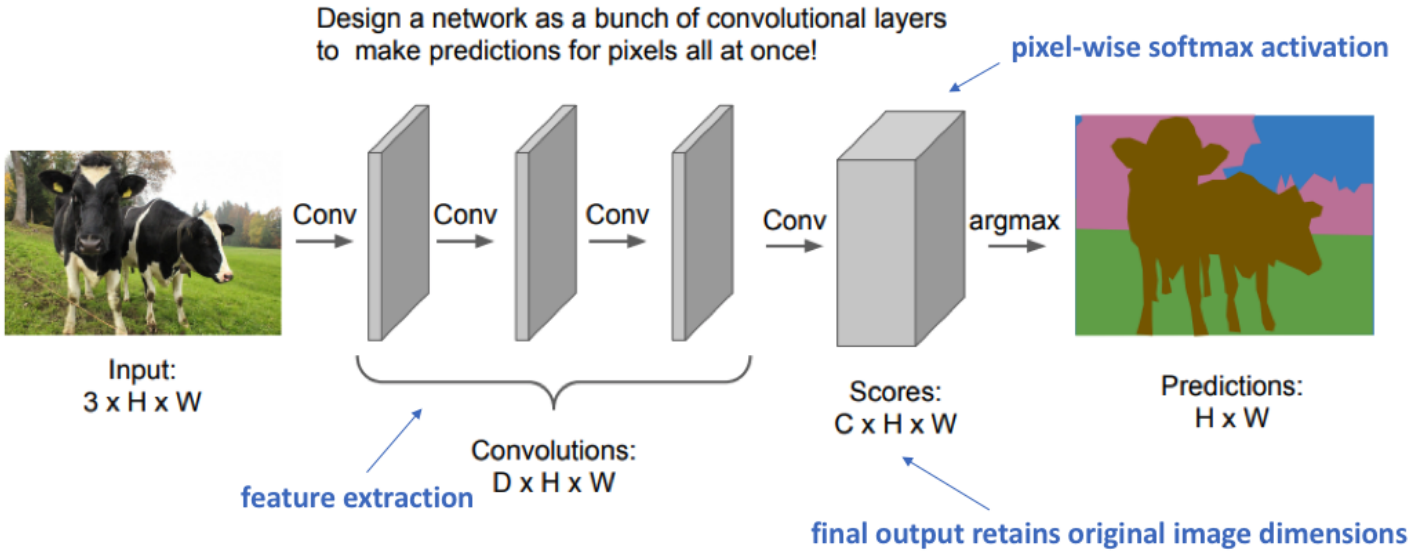






Semantic segmentation

Goal: assign a class label to every pixel in an image



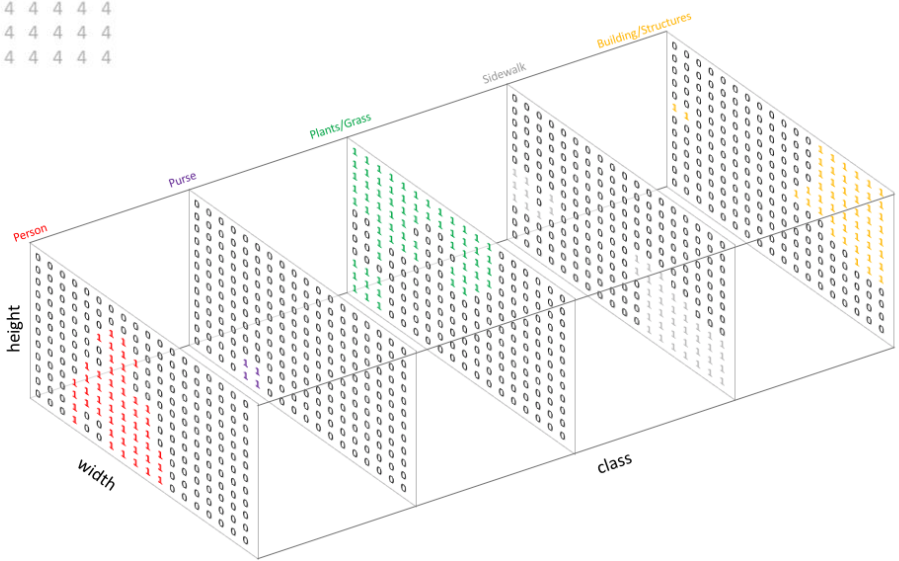
Downside: Preserving image dimensions throughout entire network will be computationally expensive.

https://www.jeremyjordan.me/semantic-segmentation/

Semantic segmentation

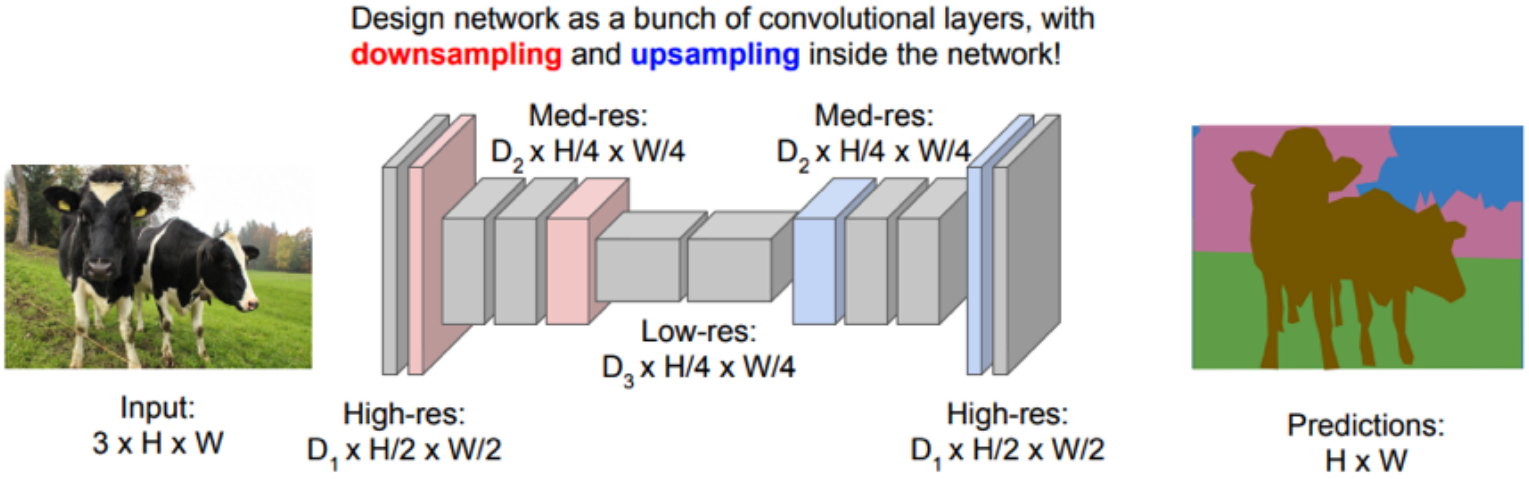
Input

Semantic Labels



https://www.jeremyjordan.me/semantic-segmentation/

Semantic segmentation



Solution: Make network deep and *work at a lower spatial resolution* for many of the layers.

e.g., UNet


Pipeline: TPU Segment DeepLabV3-dm0.5
PreProc: 6.5ms (153.9fps)
Network: 77.8ms (12.9fps)
PostProc: 1.3ms (796.6fps)
OVERALL: 87.85ms/inference
29.9 fps, 89.0% CPU, 49C, 2208 MHz, Camera: YUYV:1920x1080 + RGB3:1024x576, Display: RGBA:1920x1080

Visual question answering (VQA)

Q. What is the cat wearing?
A. Hat

Q. What is the weather like?
A. Rainy

Q. What surface is this?
A. Clay

Q. What toppings are on the pizza?
A. Mushrooms

Q. How many holes are in the pizza?
A. 8

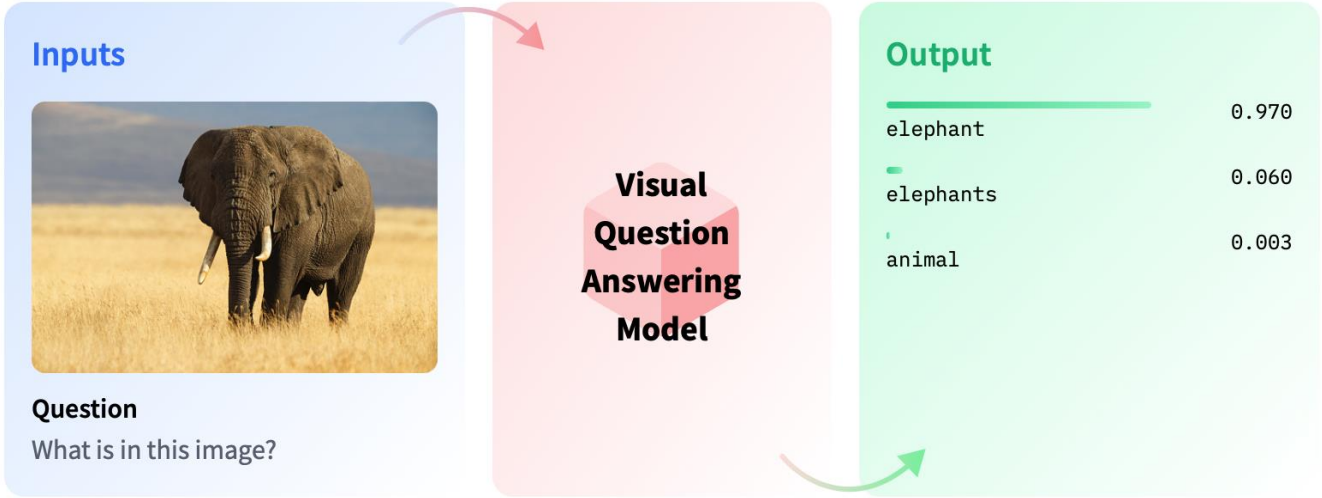
Q. What letter is on the racket?
A. w

Q. What color is the right front leg?
A. Brown

Q. Why is the sign bent?
A. It's not

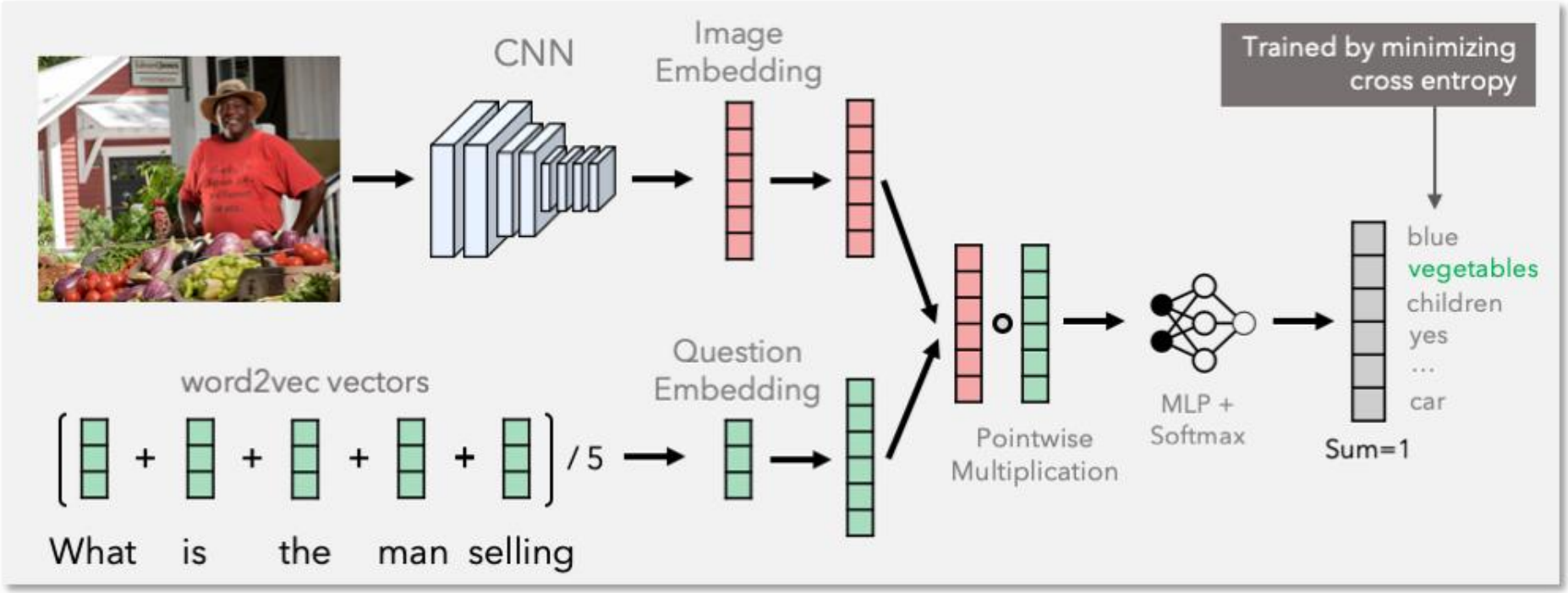
Visual question answering (VQA)

Visual Question Answering is the task of answering open-ended questions based on an image.
They output natural language responses to natural language questions.



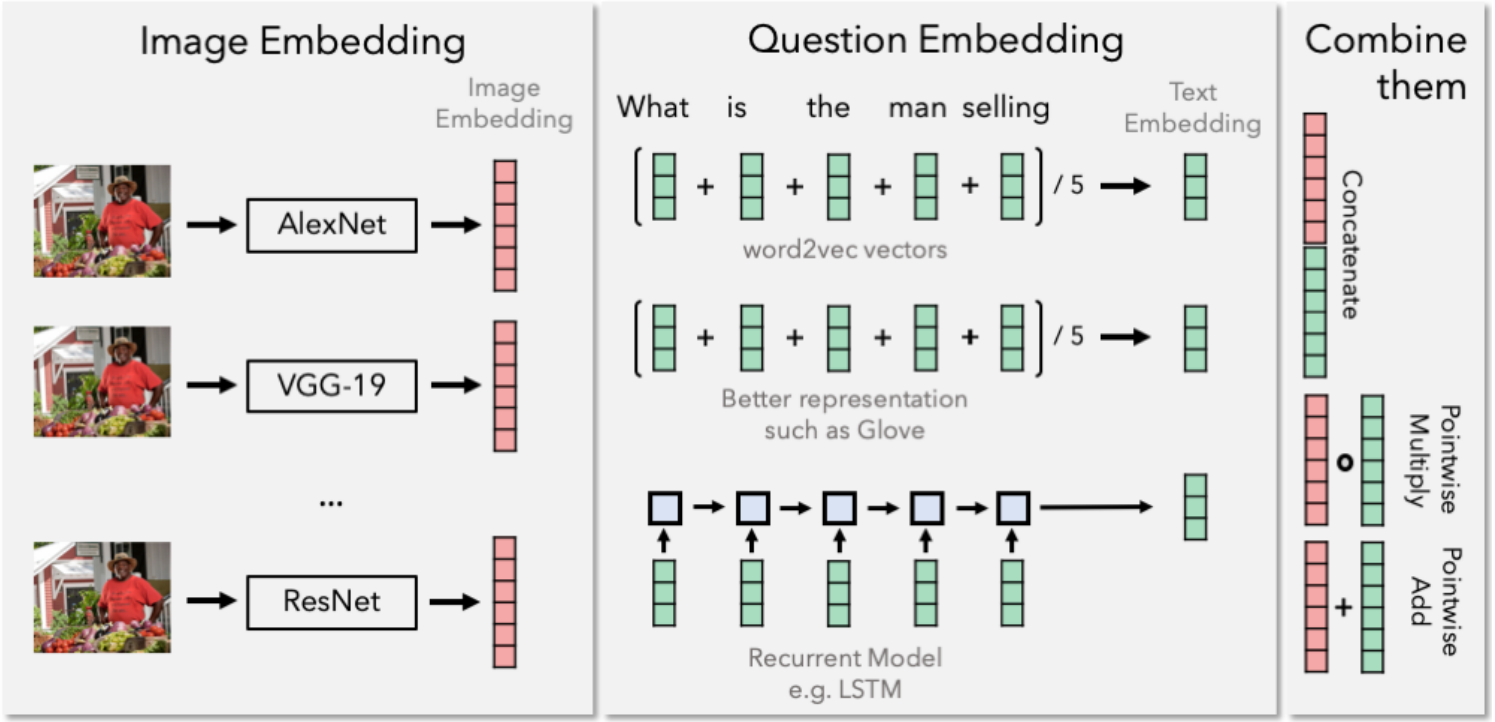
https://huggingface.co/tasks/visual-question-answering

Visual question answering (VQA)



https://blog.allenai.org/vanilla-vqa-adcaaaa94336

Visual question answering (VQA): variants



https://blog.allenai.org/vanilla-vqa-adcaaaa94336

Vision-Language Models (VLM)

["Lion and cheetah",
"Cat and dog",
"Rabbit and bird"]

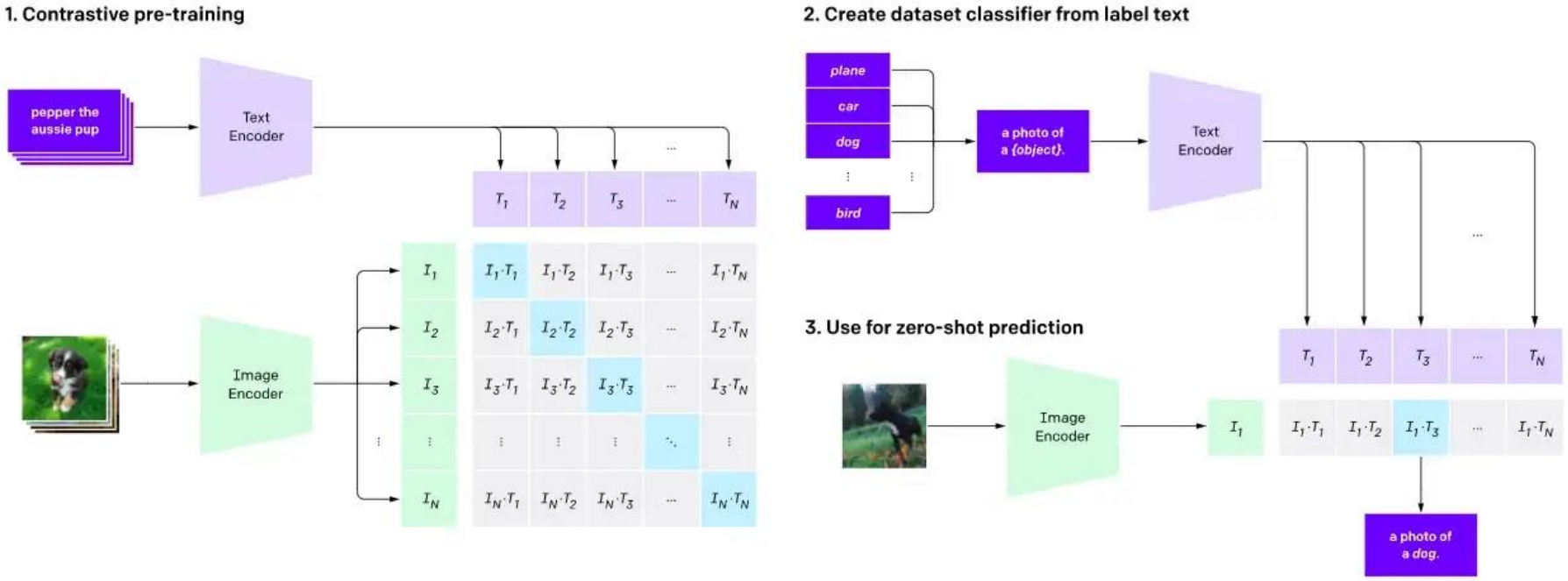
Vision-Language Model

[{'score': 0.9950, 'label': 'cat and dog'},
{'score': 0.0048, 'label': 'rabbit and lion'},
{'score': 8.81e-05, 'label': 'lion and cheetah'}]

https://huggingface.co/blog/vision_language_pretraining

Vision-Language Models (VLM): contrastive learning

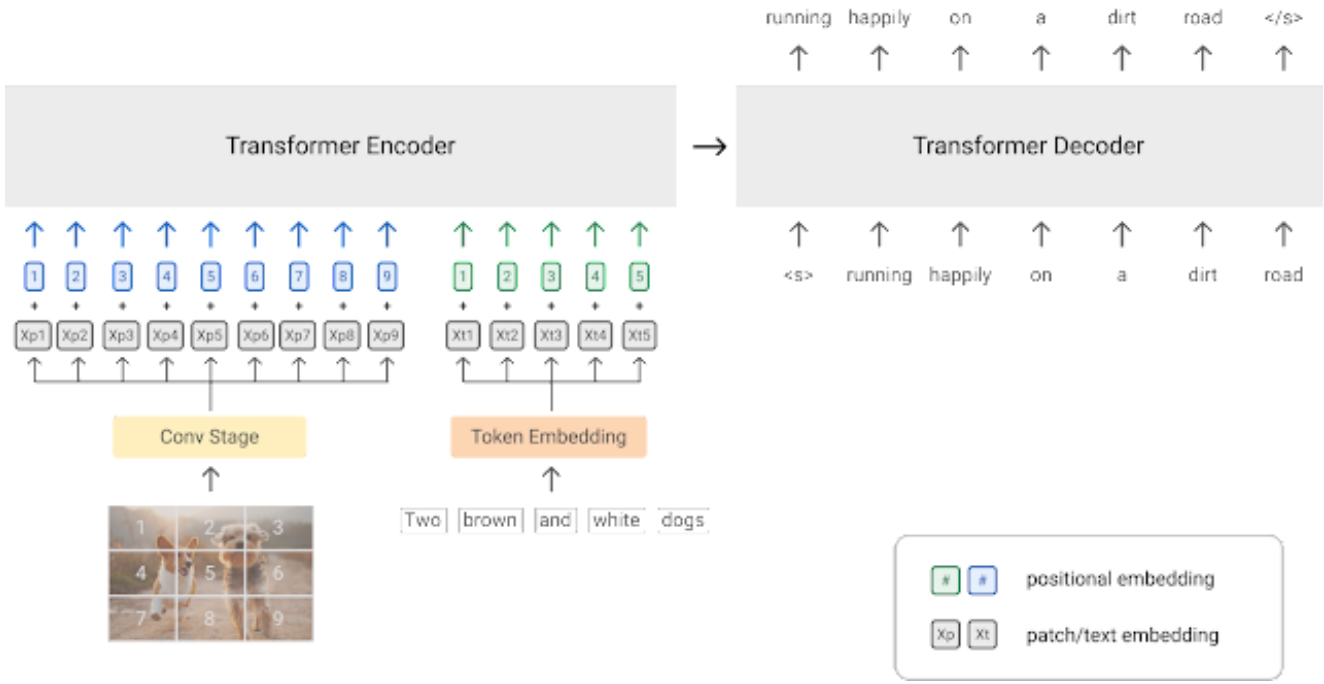
Goal: embed images and their captions into the same latent space; e.g., CLIP model.



https://huggingface.co/blog/vision_language_pretraining

Vision-Language Models (VLM): prefixing

Use a transformer to process both image and text; e.g., PrefixLM



https://huggingface.co/blog/vision_language_pretraining

Vision-Language Models (VLM)

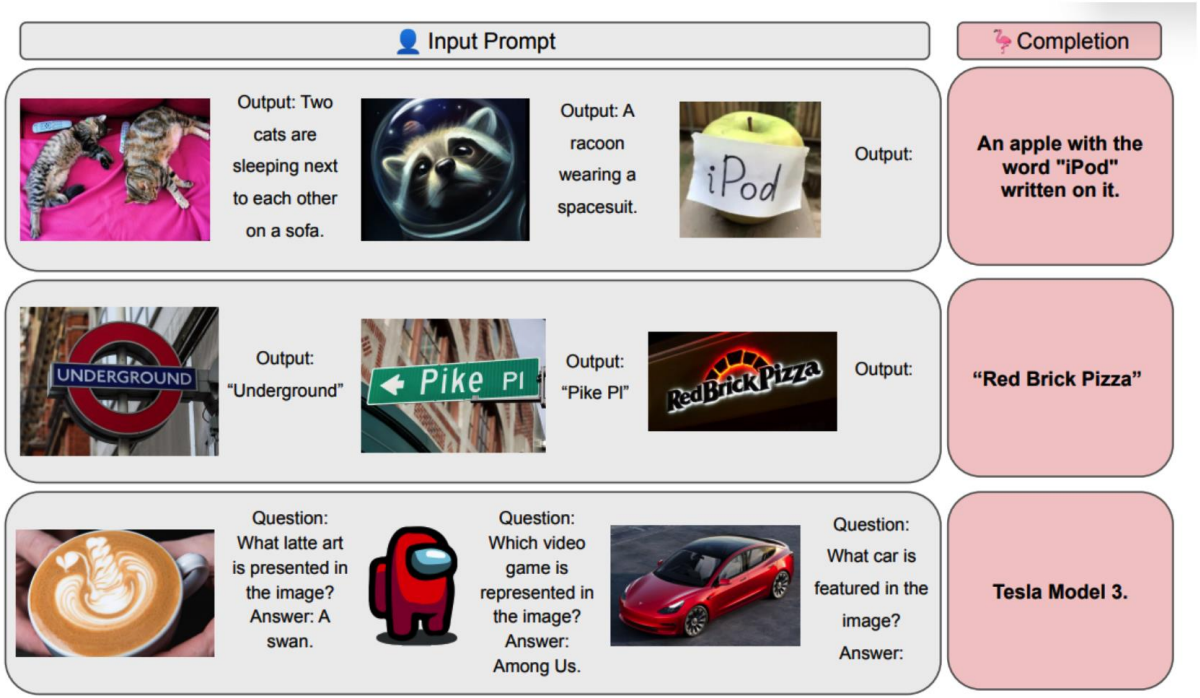


Figure 2: OpenFlamingo-9B (pictured) can process interleaved image-and-text sequences. This interface allows OpenFlamingo to learn many vision-language tasks through in-context demonstrations.

https://syncedreview.com/2023/08/10/open-source-large-autoregressive-vision-language-models-institutions-join-forces-to-replicate-deepminds-flamingo-models/

Outlook

- Very active research area
- Transformers & LLMs are taking over, or are they?
- Many new opportunities to delve deeper.