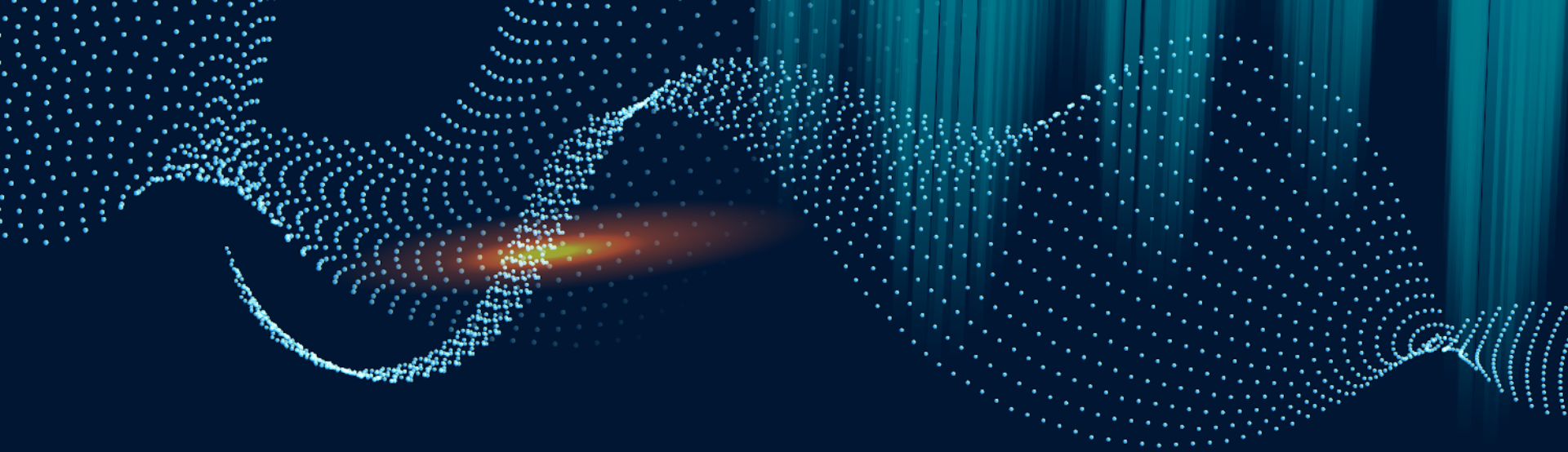
Computer Vision

Katie Huang
March 12th, 2021
Metis Investigation

Outline

1. What is vision?
2. History of computer vision
3. Beginner friendly tools
4. Future of computer vision



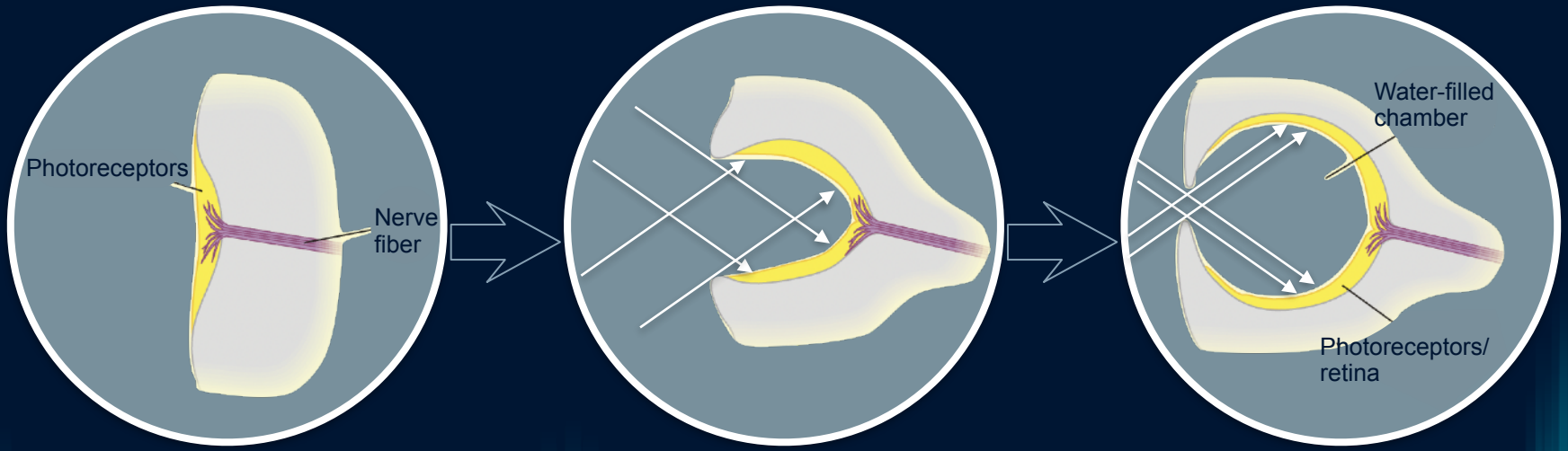
01 Vision

Its origin and what we have today

How it began

- Photoreceptor proteins
- Where is the light?

As time goes by...



Eyespot

“Pit” eye

“Pinhole” eye

Our eyes today

Direction

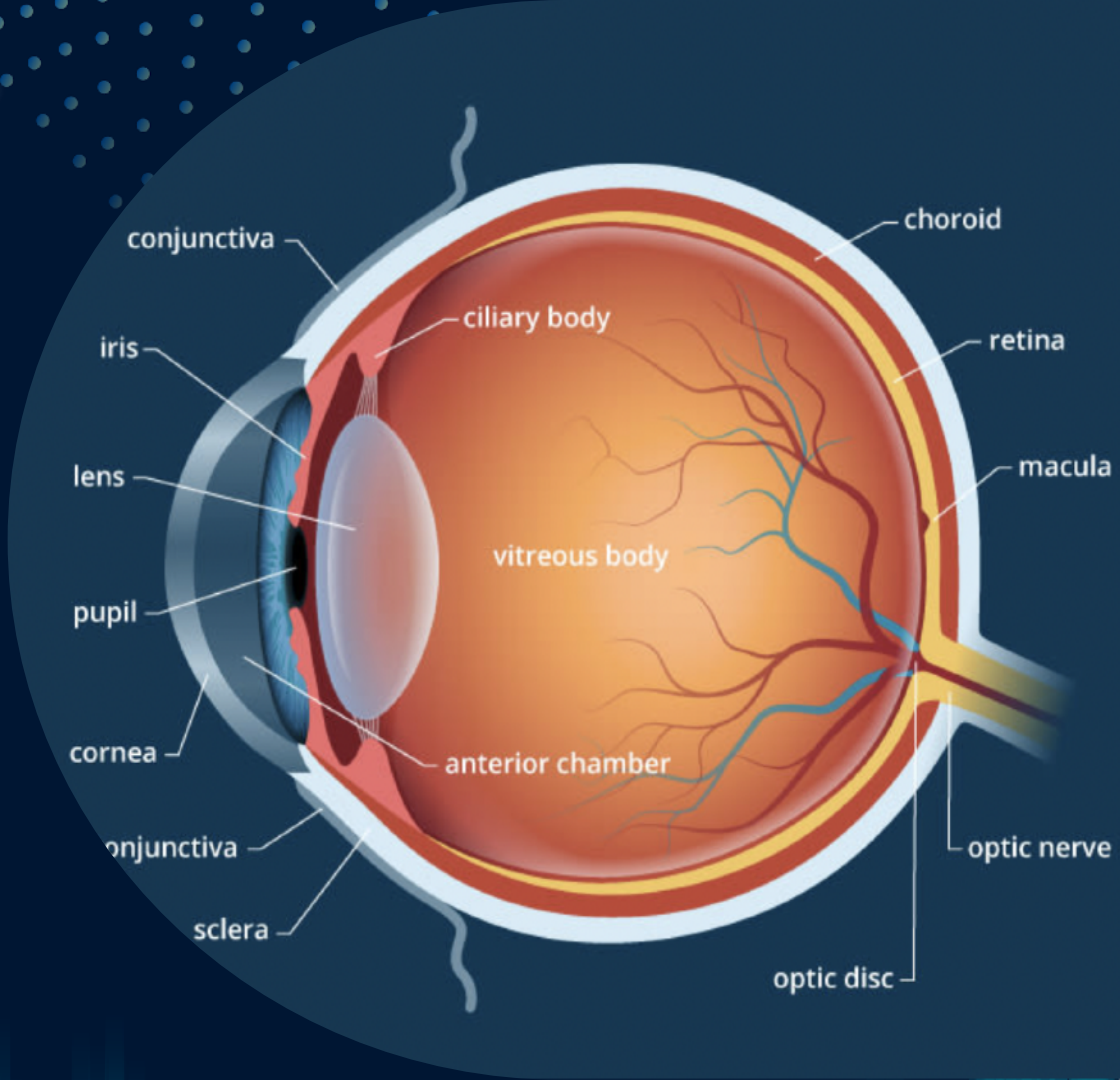
Focus

Colors

...

Collecting signals

And then?

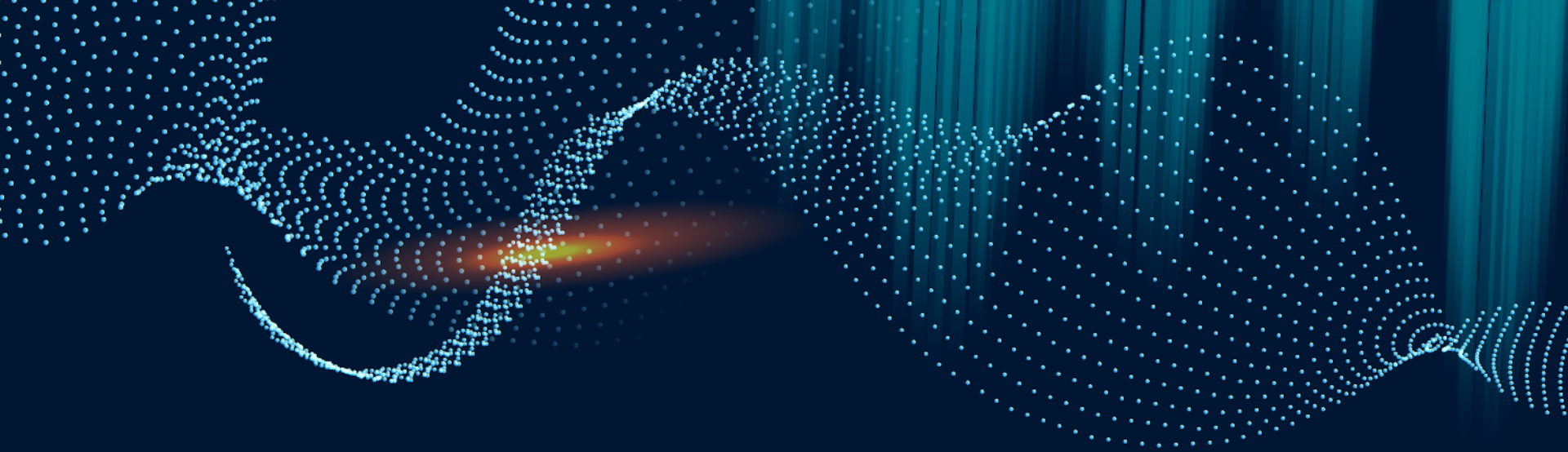


Visual processing

Sensory input

Process the signals

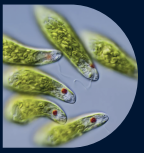
Vision



02 History

The beginning and foundation of computer vision

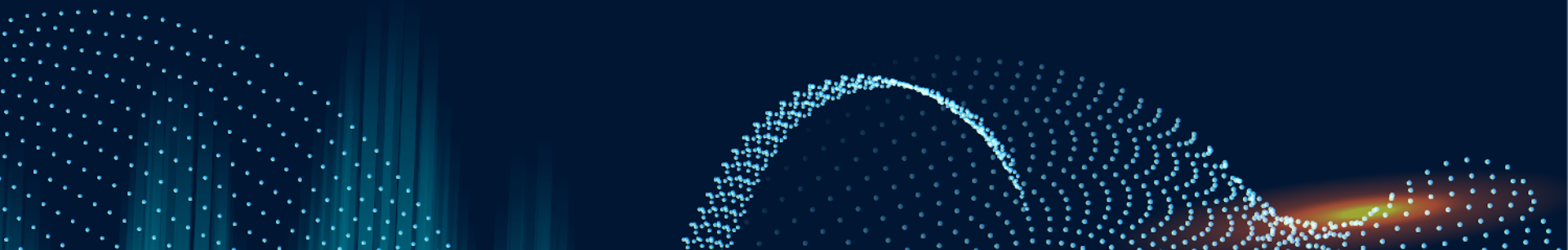
A brief history



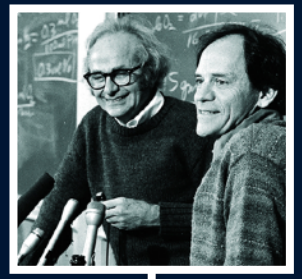
A brief history

1956

“Artificial Intelligence”



A brief history



Neurophysiology: Discover vision is hierarchical

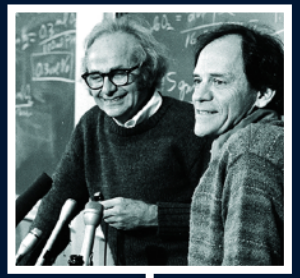
(Hubel & Wiesel)

1956

1959

“Artificial Intelligence”

A brief history



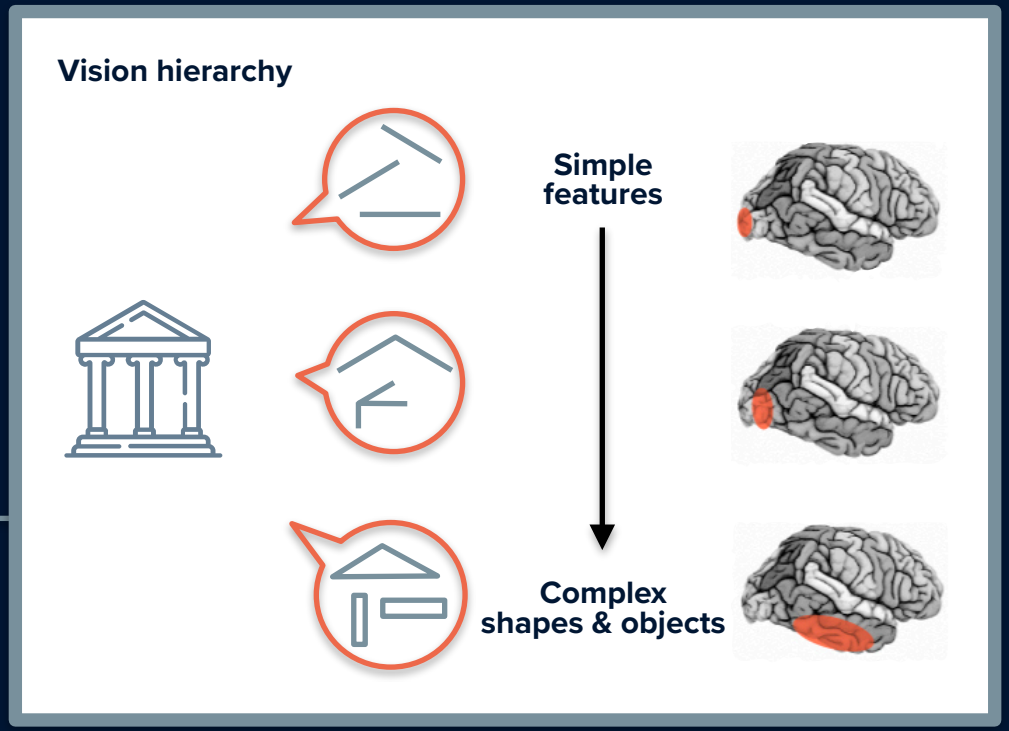
Neurophysiology: Discover vision is hierarchical

(Hubel & Wiesel)

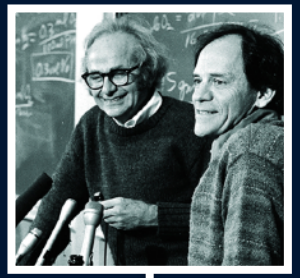
1956

1959

“Artificial Intelligence”



A brief history



Neurophysiology: Discover vision is hierarchical

(Hubel & Wiesel)

Foundation of computer vision:

Hierarchical approach to perceiving and analyzing visual stimuli

1956

"Artificial Intelligence"

1959

From human vision to computer vision

Sensory input

Process the signals

Vision

Image

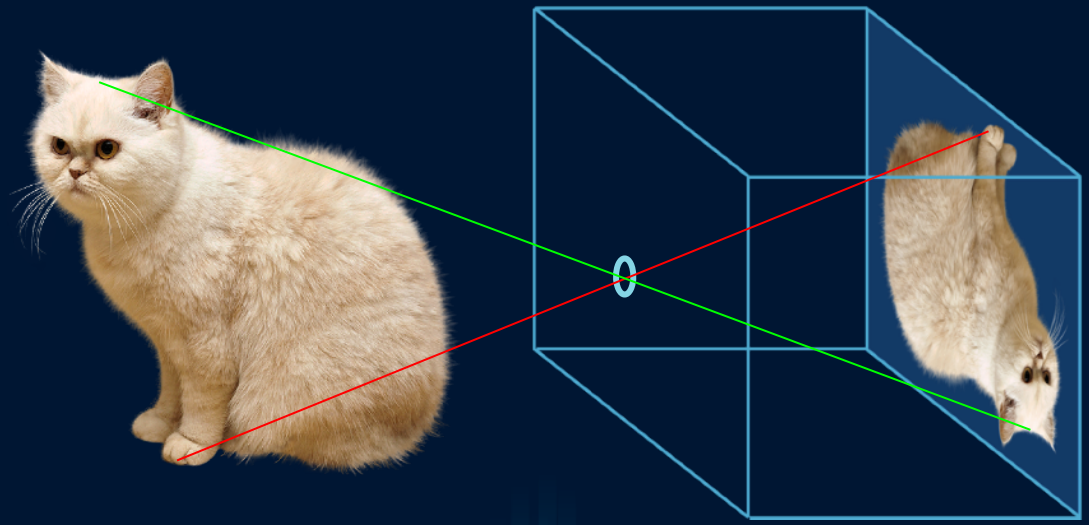
Low-level
image processing

Mid-level
interactions

High-level
semantics

Computer vision

Image

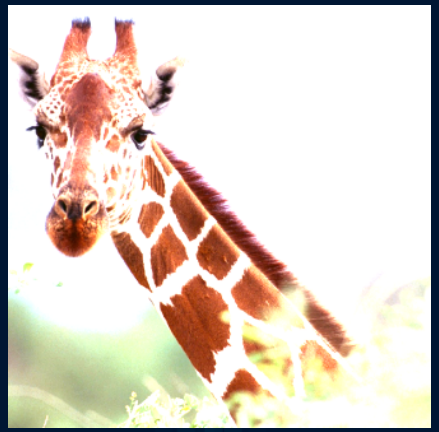


3D

2D plane

Computer vision

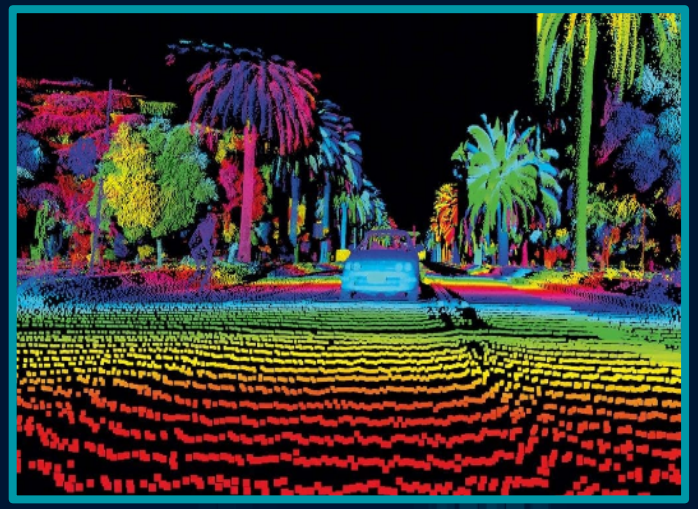
Low-level
image processing

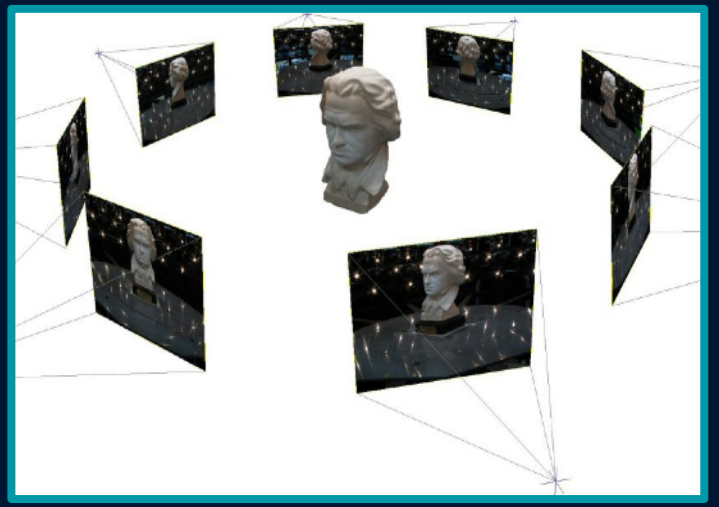
resize, exposure, hue etc

Computer vision

Mid-level
interaction with world



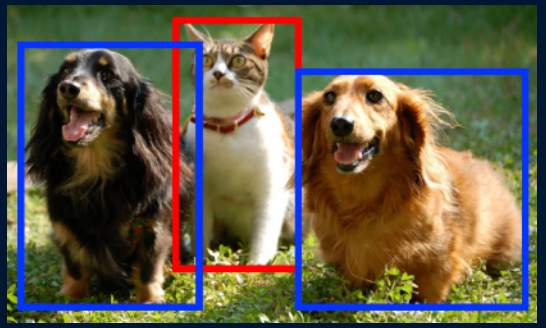
LiDAR



Stereo reconstruction

Computer vision

Classification



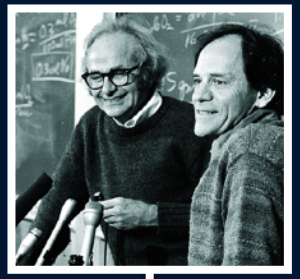
Detection, Segmentation

High-level

image → ← semantics

GAN, Deep Fake

A brief history



Neurophysiology: Discover vision is hierarchical

(Hubel & Wiesel)

Foundation of computer vision:

Hierarchical approach to perceiving and analyzing visual stimuli

1956

"Artificial Intelligence"

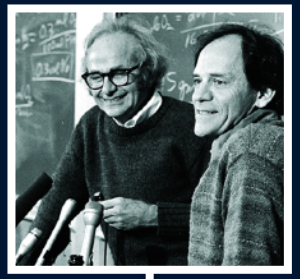
1959

1960s

First computer vision project

(MIT summer project)

A brief history



Neurophysiology: Discover vision is hierarchical

(Hubel & Wiesel)

1956

"Artificial Intelligence"

1959

1960s

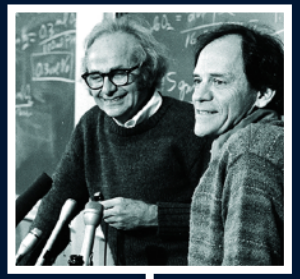
First computer vision project

(MIT summer project)

1970s

"AI winter"

A brief history



Neurophysiology: Discover vision is hierarchical

(Hubel & Wiesel)

Rudimentary neural networks

1956 1959 1960s 1970s 1990s

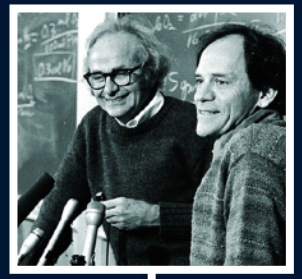
“Artificial Intelligence”

First computer vision project

(MIT summer project)

“AI winter”

A brief history



Neurophysiology: Discover vision is hierarchical

(Hubel & Wiesel)

Rudimentary neural networks

1956 1959 1960s 1970s 1990s 2001

“Artificial Intelligence”

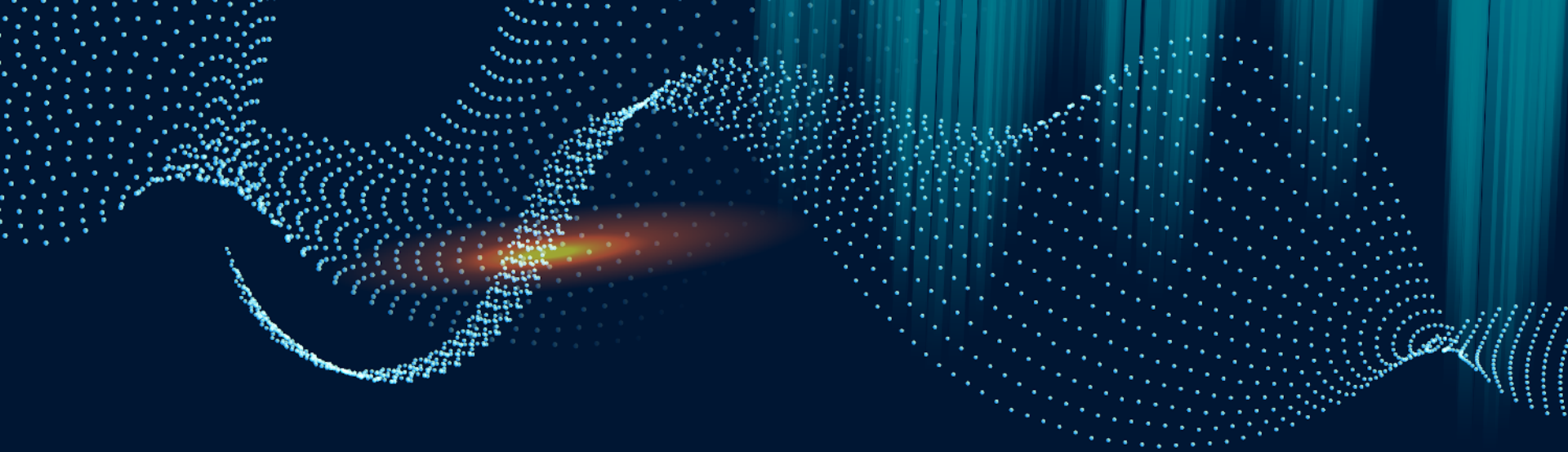
First computer vision project

(MIT summer project)

“AI winter”

Face detection

(Viola & Jones)

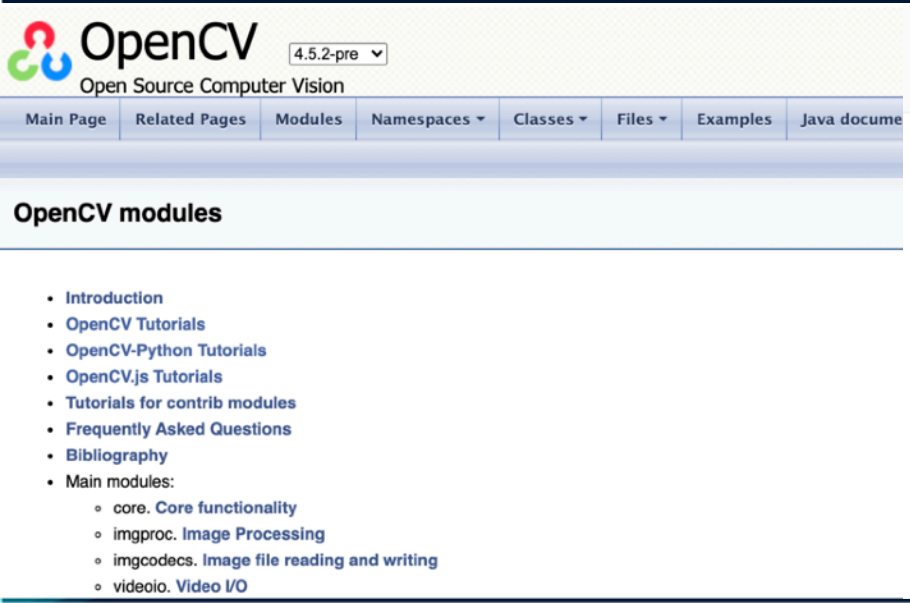


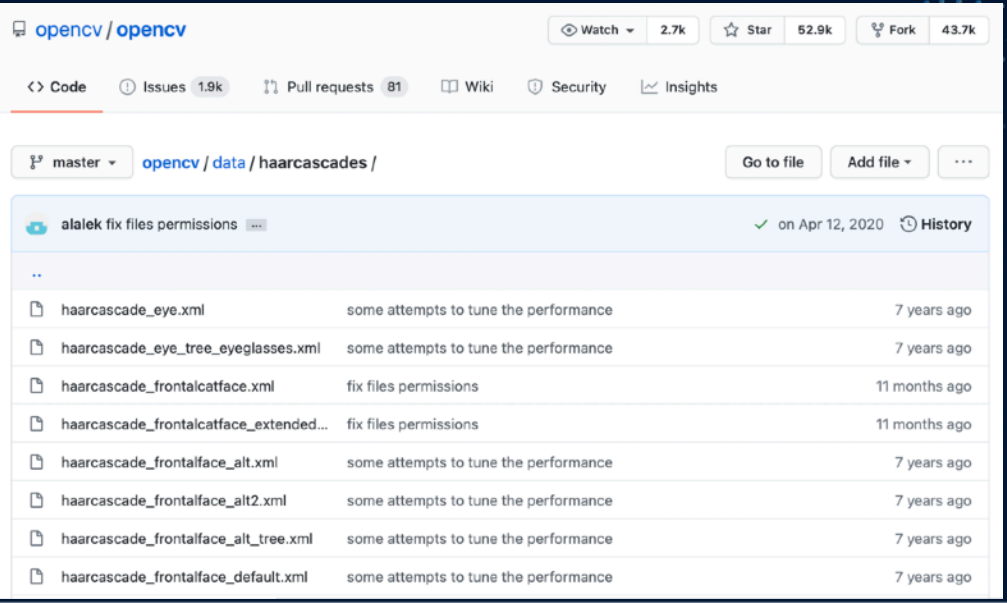
03 Tools

Beginner friendly: OpenCV and YOLO

OpenCV

- A library aimed at real-time computer vision
- Supports deep learning frameworks like Tensorflow, Py-Torch, YOLO etc



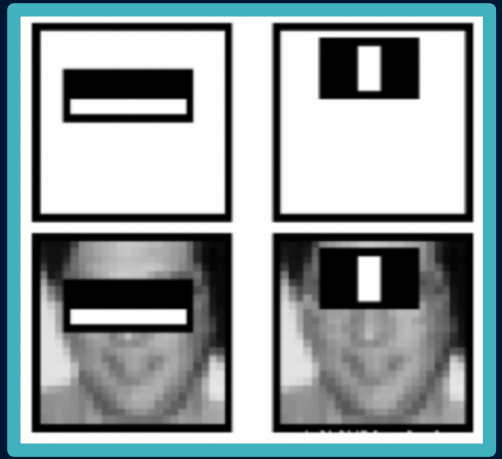


OpenCV - Face detection

- Haar features
- Cascade of classifiers

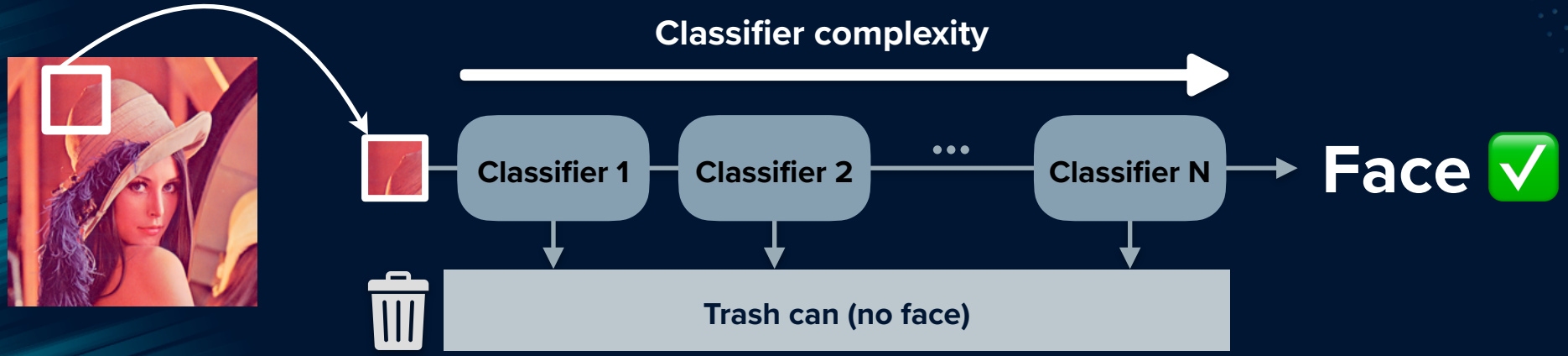
OpenCV - Face detection

- Haar features
- Cascade of classifiers



OpenCV - Face detection

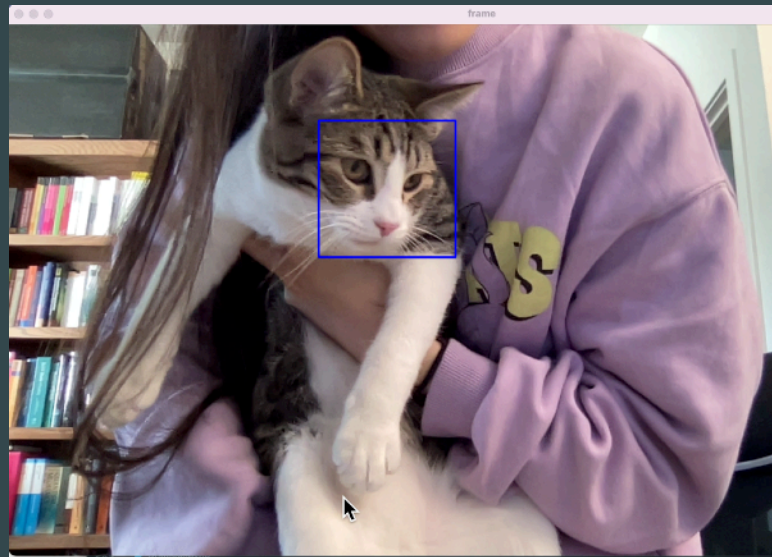
- Haar features
- Cascade of classifiers



OpenCV - Demo

demo.py > ...

```python
import cv2 as cv
cat_face=cv.CascadeClassifier(cv.data.haarcascades+'haarcascade_frontalcatface_extended.xml')

cap = cv.VideoCapture(0)
while True:
    ret, frame = cap.read()
    detections = cat_face.detectMultiScale(frame,1.3,5)

    if (len(detections)>0):
        (x,y,w,h) = detections[0]
        frame = cv.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)

    cv.imshow('frame',frame)
    if cv.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv.destroyAllWindows()
```

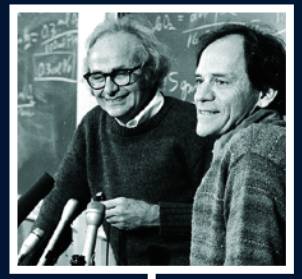


OpenCV - Demo

demo.py > ...

```python
import cv2 as cv
cat_face=cv.CascadeClassifier(cv.data.haarcascades+'haarcascade_frontalcatface_extended.xml')

cap = cv.VideoCapture(0)
while True:
    ret, frame = cap.read()
    detections = cat_face.detectMultiScale(frame,1.3,5)

    if (len(detections)>0):
        (x,y,w,h) = detections[0]
        frame = cv.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)

    cv.imshow('frame',frame)
    if cv.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv.destroyAllWindows()
```

A brief history



Neurophysiology: Discover vision is hierarchical

(Hubel & Wiesel)

Rudimentary neural networks

1956 1959 1960s 1970s 1990s 2001

“Artificial Intelligence”

First computer vision project

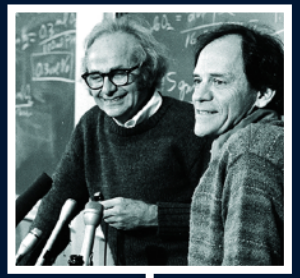
(MIT summer project)

“AI winter”

Face detection

(Viola & Jones)

A brief history



Neurophysiology: Discover vision is hierarchical

(Hubel & Wiesel)

AlexNet (Krizhevsky et al.)

Rudimentary neural networks

1956 1959 1960s 1970s 1990s 2001 2012

"Artificial Intelligence"

First computer vision project

(MIT summer project)

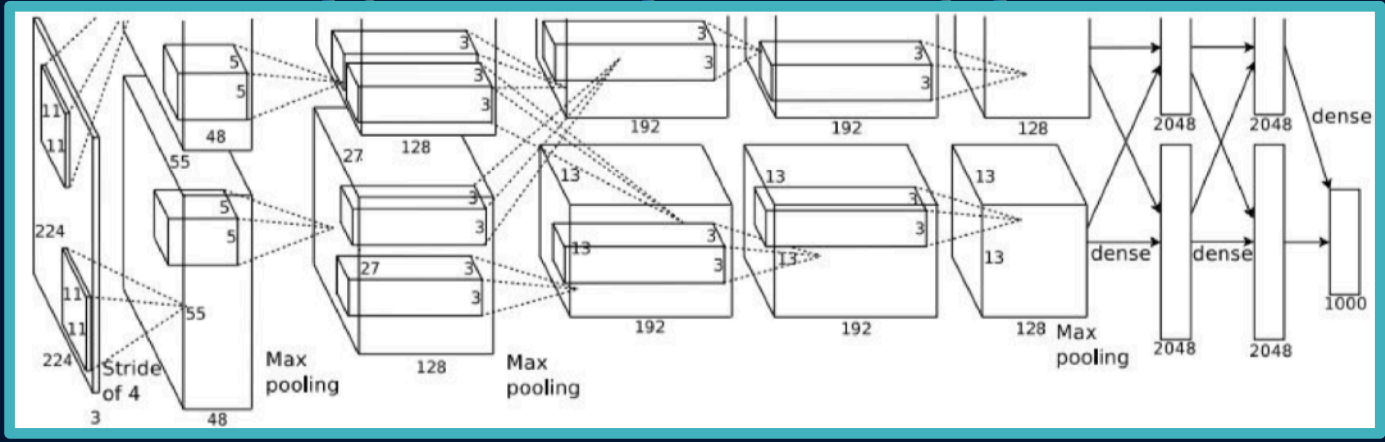
"AI winter"

Face detection

(Viola & Jones)

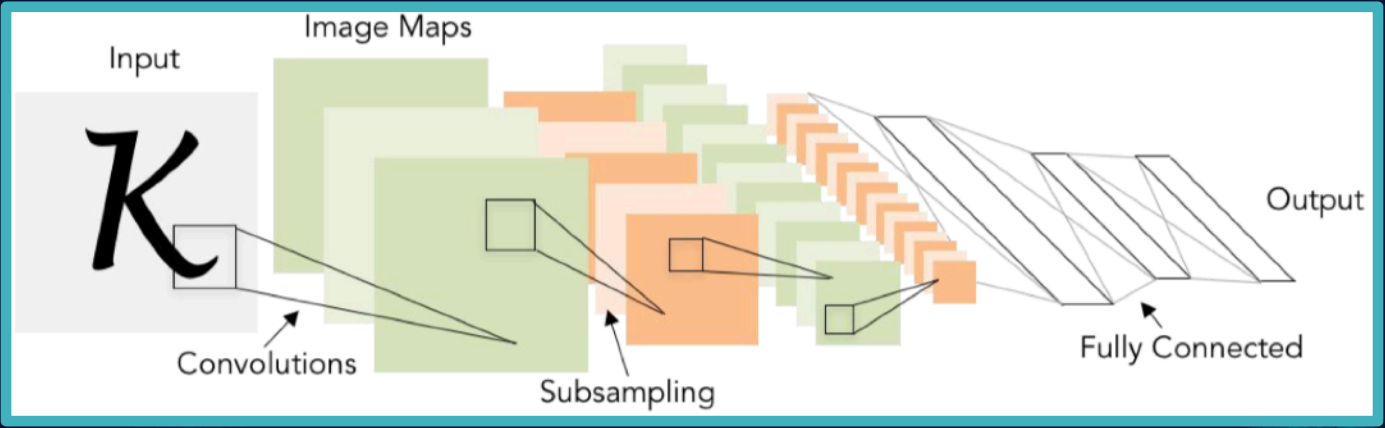
AlexNet, 2012

(Krizhevsky et al.)



LeNet, 1998

(LeCun et al.)

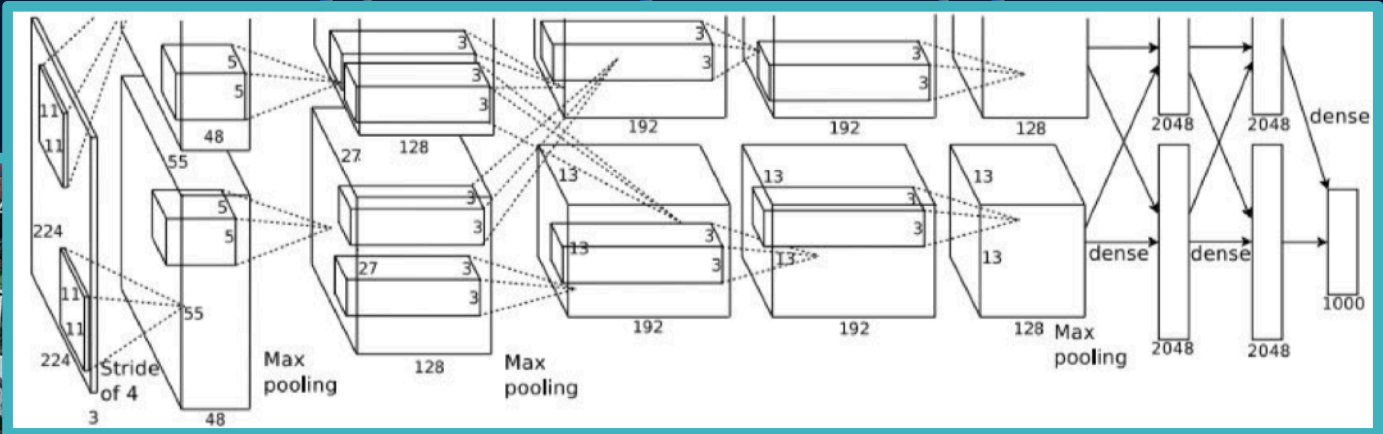


Standing on the shoulders of giants...

Algorithms

Data

Computation



A brief history

AlexNet (Krizhevsky et al.)

2001

2012

VGG, GoogLeNet, ResNet, SENet...

Face detection

(Viola & Jones)

A brief history

AlexNet (Krizhevsky et al.)

2001

Face detection

(Viola & Jones)

2012

darknet

- ImageNet Classification
- RNNs in Darknet
- DarkGo
- Tiny Darknet
- Nightmare
- ...

A brief history

AlexNet (Krizhevsky et al.)

2001

Face detection
(Viola & Jones)

2012

darknet

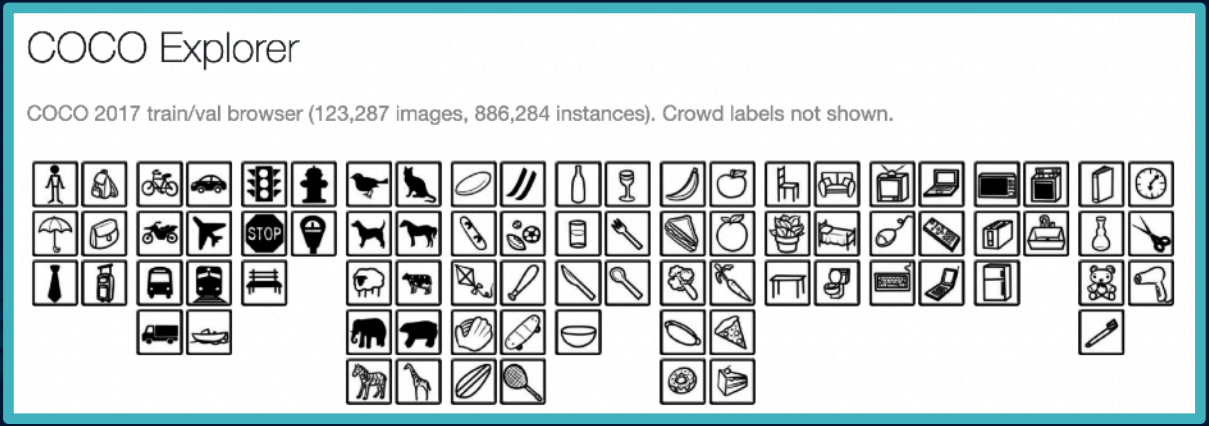
- **YOLO**

(You Only Look Once)
Real-Time Object Detection

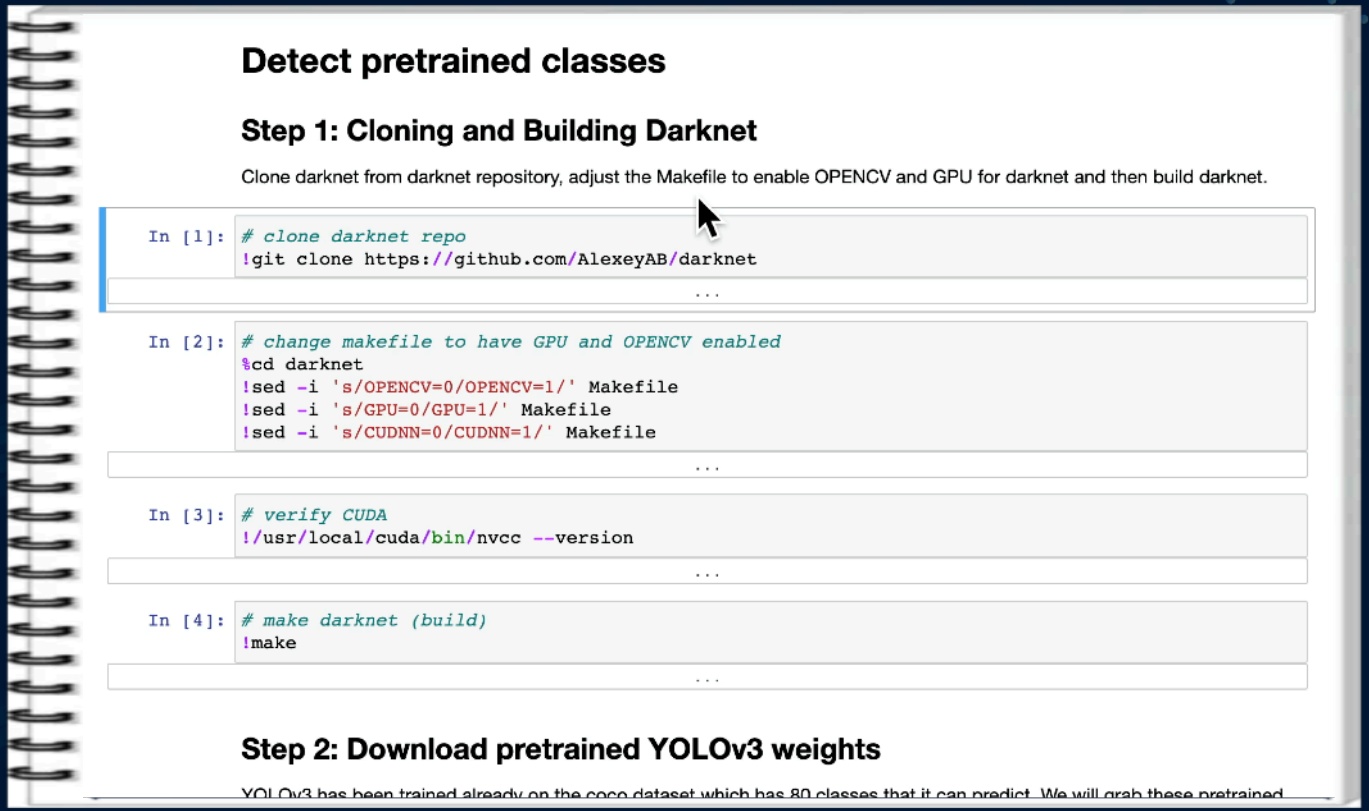
Extremely fast & accurate!

Darknet YOLO - Pretrained

- 80 classes from COCO dataset



Darknet YOLO - Pretrained



Detect pretrained classes

Step 1: Cloning and Building Darknet

Clone darknet from darknet repository, adjust the Makefile to enable OPENCV and GPU for darknet and then build darknet.

In [1]:
```
# clone darknet repo
!git clone https://github.com/AlexeyAB/darknet
```

...

In [2]:
```
# change makefile to have GPU and OPENCV enabled
%cd darknet
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
```

...

In [3]:
```
# verify CUDA
!/usr/local/cuda/bin/nvcc --version
```

...

In [4]:
```
# make darknet (build)
!make
```

...

Step 2: Download pretrained YOLOv3 weights

YOLOv3 has been trained already on the coco dataset which has 80 classes that it can predict. We will grab these pretrained

Darknet YOLO - Custom

1. Prepare custom dataset
2. Prepare 4 files: .cfg, obj.names, obj.data, train.txt

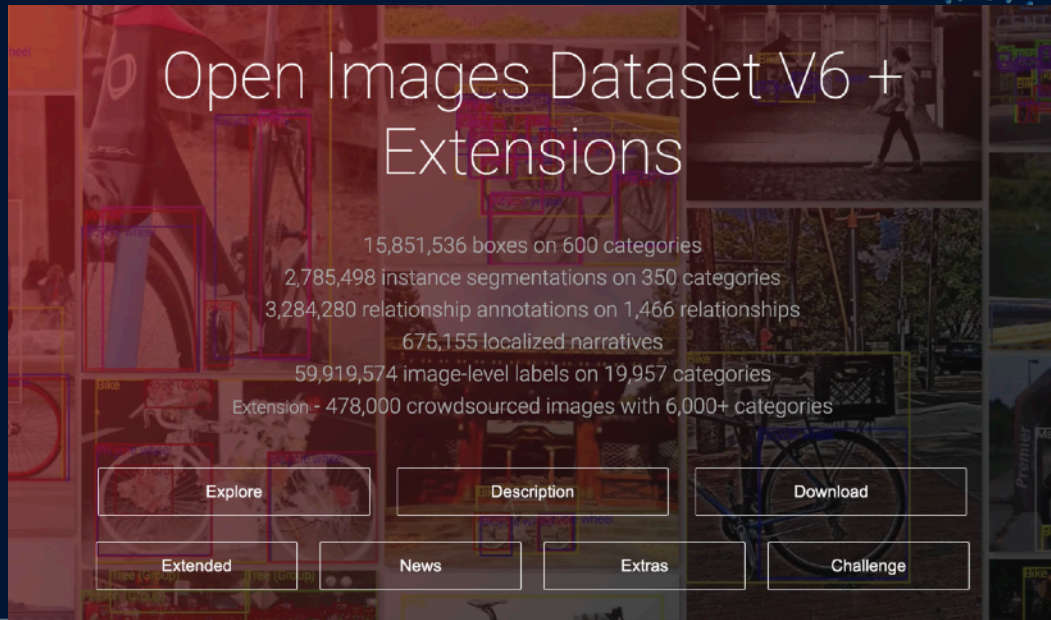
Darknet YOLO - Custom

1. Prepare custom dataset (images and labels!)
2. Prepare 4 files: .cfg, obj.names, obj.data, train.txt

Darknet YOLO - Custom

1. Prepare custom dataset (images and labels!)
2. Prepare 4 files: .cfg, obj.names, obj.data, train.txt

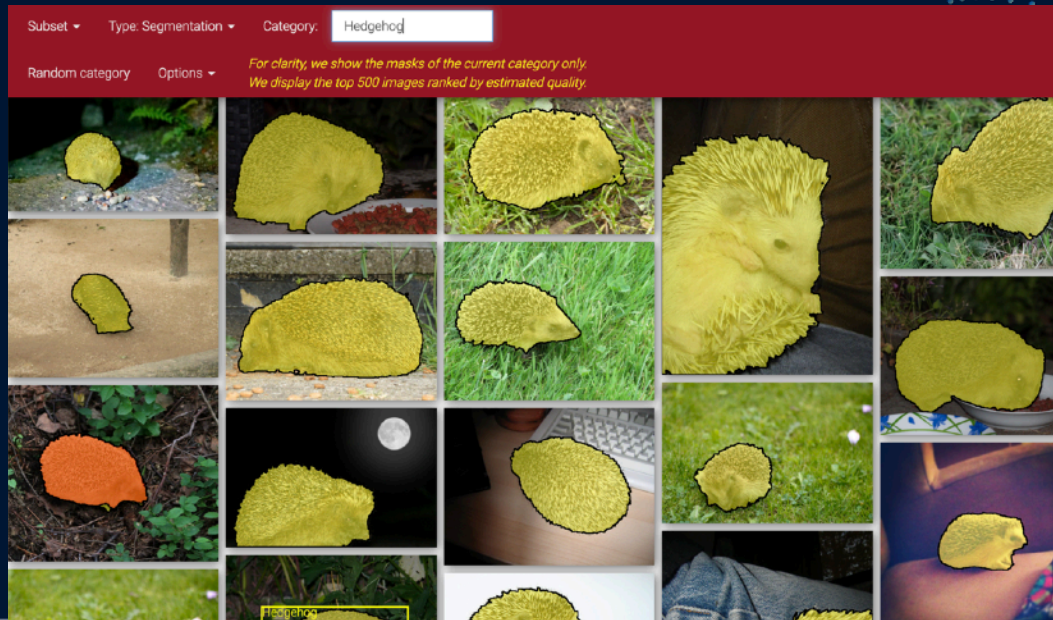
Method 1: Google's Open Images Dataset



Darknet YOLO - Custom

1. Prepare custom dataset (images and labels!)
2. Prepare 4 files: .cfg, obj.names, obj.data, train.txt

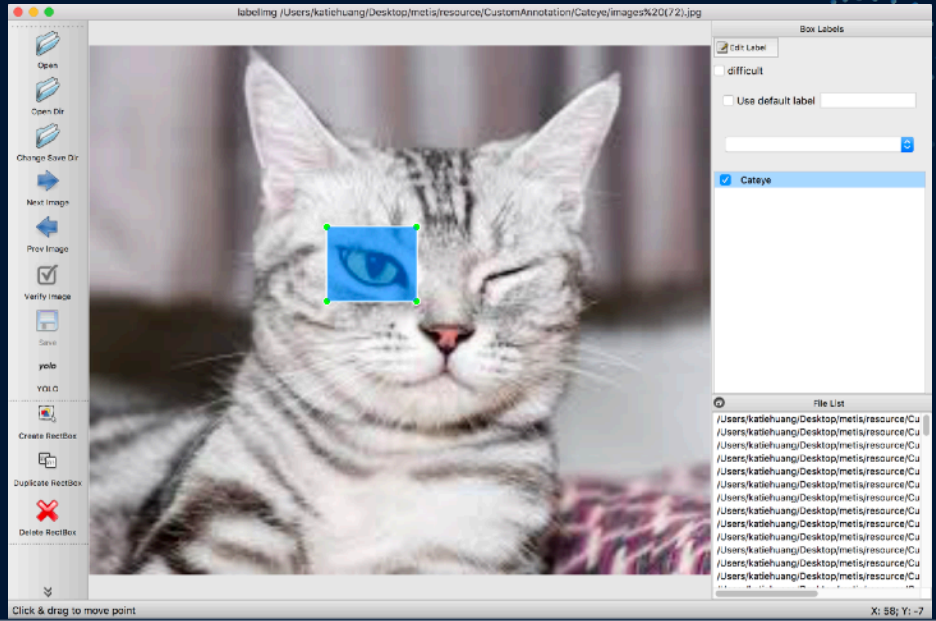
Method 1: Google's Open Images Dataset



Darknet YOLO - Custom

1. Prepare custom dataset (images and labels!)
2. Prepare 4 files: .cfg, obj.names, obj.data, train.txt

Method 2: Manually label all the images 😵



Darknet YOLO - Custom

Training a Custom YOLOv3 Object Detector in the Cloud!

Step 1: Gather (Label) a Custom Dataset. Move it to Your Cloud VM.

Method 1: Using Google's Open Images Dataset

This method is the method I recommend as you can gather thousands of images and auto-generate their labels within minutes! Gathering a dataset from Google's Open Images Dataset and using OIDv4 toolkit to generate labels is easy and time efficient. The dataset contains labeled images for over 600 classes! Explore the Dataset Here!

Method 2: Manually Labeling Images with Annotation Tool

In [15]:
```
# this is where my zip is stored (I created a yolov3 folder where I will get my required files
!ls /mydrive/yolov3_cat_eye
```

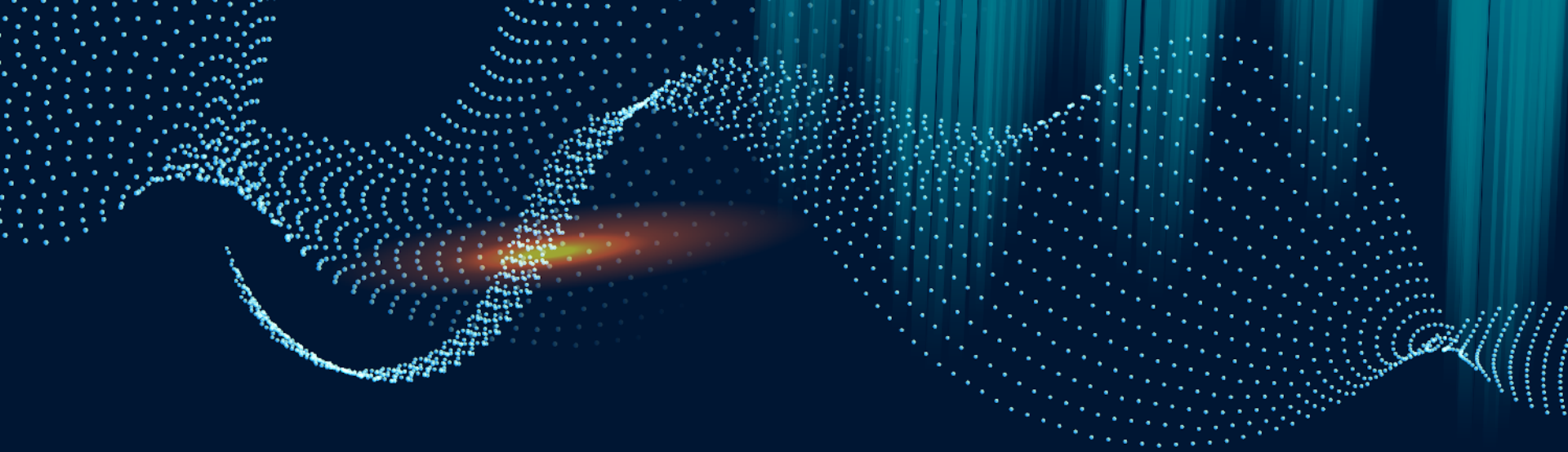
```
backup  generate_train.py  obj.data  obj.names  obj.zip  yolov3_custom.cfg
```

In [16]:
```
# copy the .zip file into the root directory of cloud VM
!cp /mydrive/yolov3_cat_eye/obj.zip ../
```

In [17]:
```
# unzip the zip file and its contents should now be in /darknet/data/obj
!unzip ../obj.zip -d data/
```

...

Step 2: Configuring Files for Training



04 Future

What’s next?

Customized training is great,

but…


labeling is not

""Practically speaking, it's impossible to label everything in the world. There are also some tasks for which there's simply not enough labeled data, such as training translation systems for low-resource languages.""

—Yann LeCun

Self-Supervised Learning

Learn directly from the data without labeling

SElf-supERvised Learning

SEER

84.2% top-1 accuracy
on ImageNet

BBC | Home | News | Sport | Reel | Worklife | Travel | Future | More | Search

NEWS

Home | Coronavirus | Video | World | US & Canada | UK | Business | Tech | Science | Stories | More

Entertainment & Arts

Tech

Instagram photos help Facebook AI 'teach itself'

Mar 8, 2021

More details...

Blog

FACEBOOK AI Research Publications

RESEARCH | COMPUTER VISION

SEER: The start of a more powerful, flexible, and accessible era for computer vision

March 4, 2021

arXiv paper

Self-supervised Pretraining of Visual Features in the Wild

Priya Goyal[1] Mathilde Caron[1,2] Benjamin Lefaudeux[1] Min Xu[1] Pengchao Wang[1] Vivek Pai[1]
Mannat Singh[1] Vitaliy Liptchinsky[1] Ishan Misra[1] Armand Joulin[1] Piotr Bojanowski[1]

[1] Facebook AI Research [2] Inria*

Code: https://github.com/facebookresearch/vissl

Abstract

Recently, self-supervised learning methods like MoCo [22], SimCLR [8], BYOL [20] and SwAV [7] have reduced the gap with supervised methods. These results have been achieved in a control environment, that is the highly curated ImageNet dataset. However, the premise of self-supervised learning is that it can learn from any random image and from any unbounded dataset. In this work, we explore if self-supervision lives to its expectation by training large models on random, uncurated images with no supervision. Our final SElf-supERvised (SEER) model, a RegNetY with 1.3B parameters trained on 1B random images with 512 GPUs achieves 84.2% top-1 accuracy, surpassing the best self-supervised pretrained model by 1% and confirming that self-supervised learning works in a real world setting. Interestingly, we also observe that self-supervised models are good few-shot learners achieving 77.9% top-1 with access to only 10% of ImageNet.

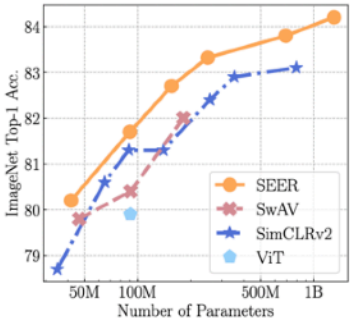
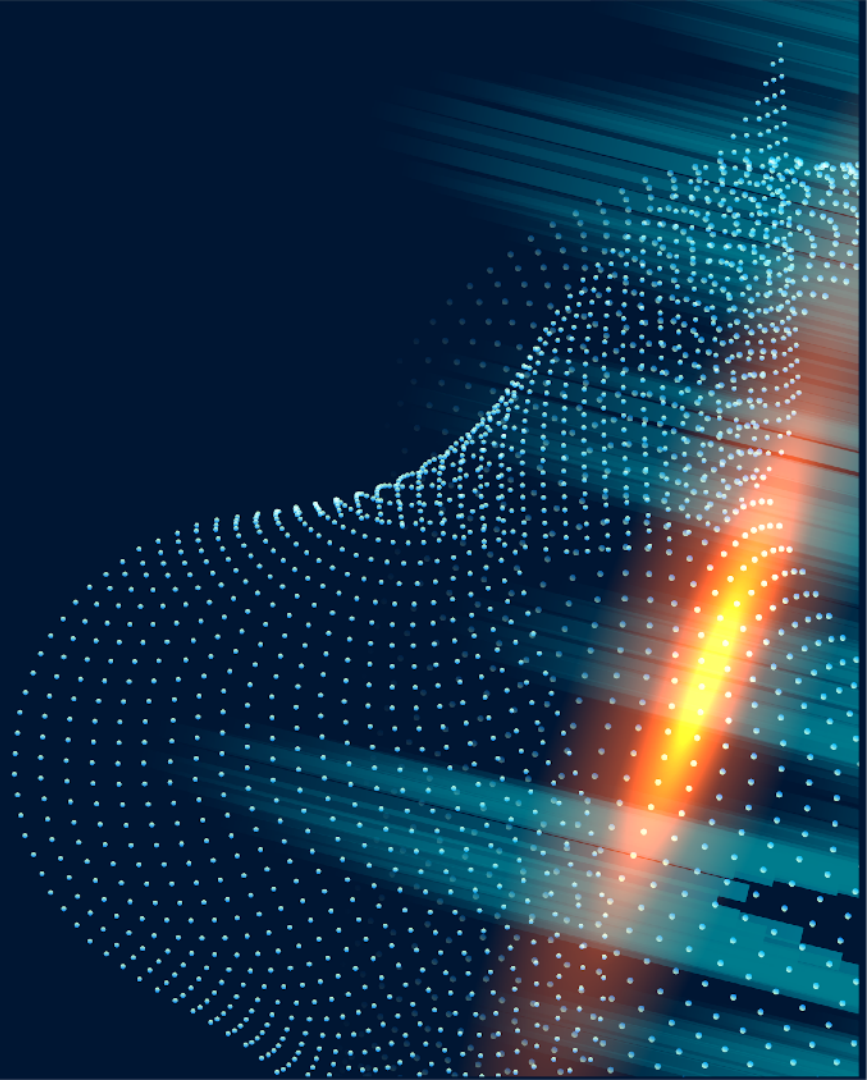
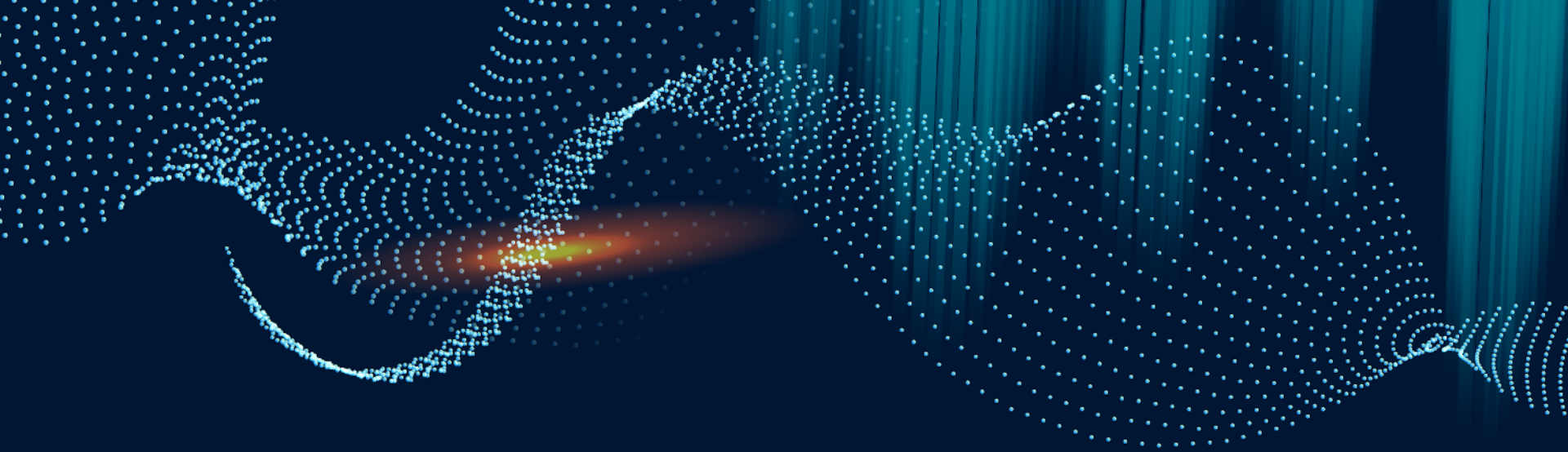


Figure 1: Performance of large pretrained models on ImageNet. We pretrain our SEER models n an uncurated and random images. They

THANKS!

Do you have any questions?





05 Appendix

ILSVRC erro rates

A brief history

AlexNet (Krizhevsky et al.)

2001

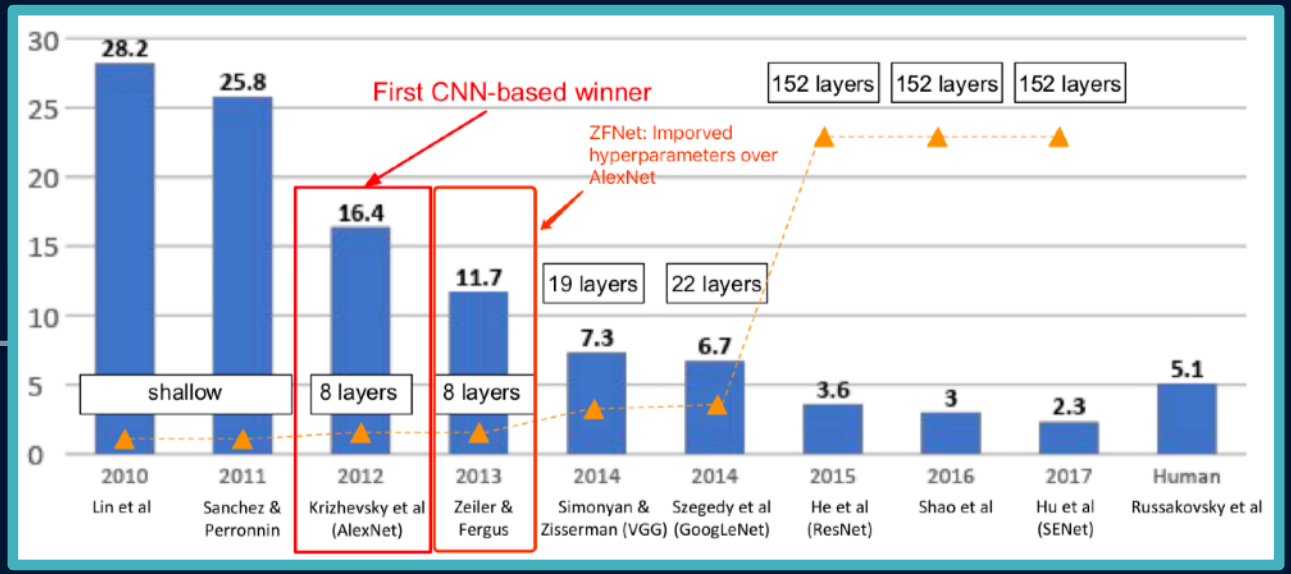
2012

Face detection

(Viola & Jones)

ILSVRC Winners

(ImageNet Large Scale Visual Recognition Challenge)



Slide from Fei-Fei Li *et al*.

A brief history

AlexNet (Krizhevsky et al.)

2001

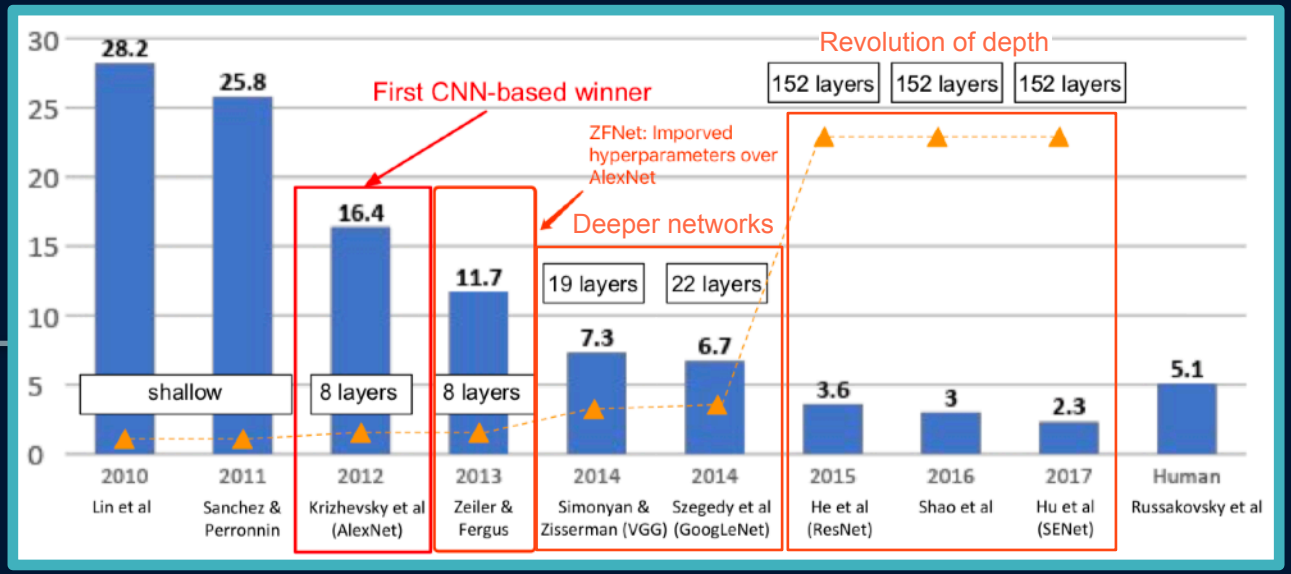
2012

Face detection

(Viola & Jones)

ILSVRC Winners

(ImageNet Large Scale Visual Recognition Challenge)



Slide from Fei-Fei Li *et al*.


slidesgo