BIRDS-OF-A-FEATHER

AI at UFIT Research Computing

Fundamentals of Image Processing and Machine Learning on HiPerGator-AI

Yunchao Yang

UF Research Computing

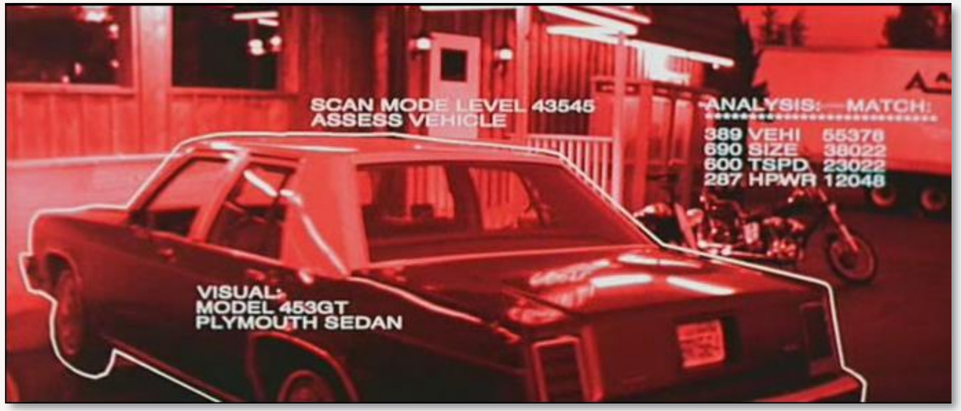
AI Research Facilitator

yunchyang@ufl.edu

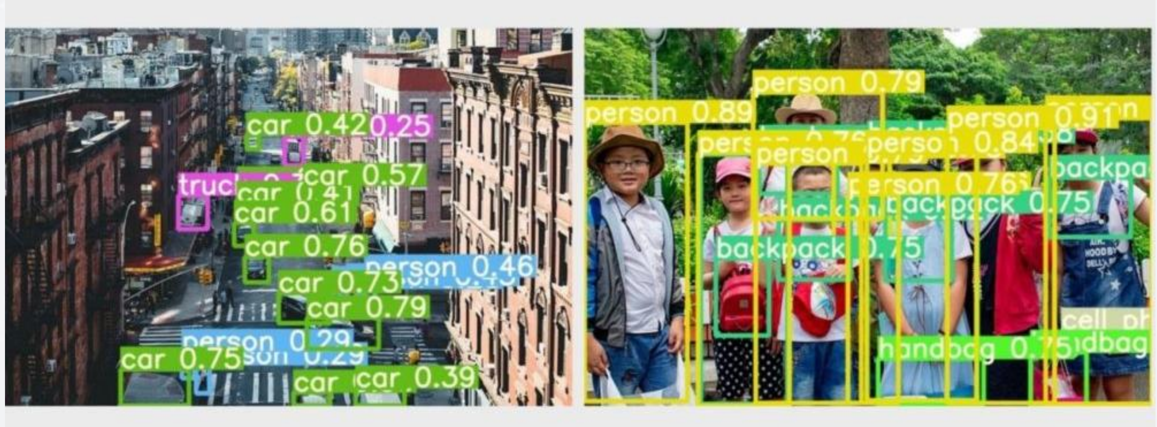
UF | Information Technology
UNIVERSITY *of* FLORIDA

AI in computer vision

- Progress of AI in computer vision



Terminator 2, 1991



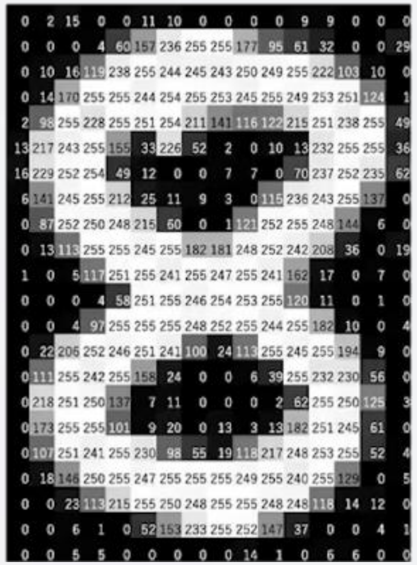
Object detection YOLOv5 algorithm, 2022

Outline

- What you will learn in this workshop?
  - Intro to Image Processing/Computer Vision
  - Hands-on tutorials of basic Imaging processing on HiPerGator using OpenCV and TorchVision
  - Intro to advanced Deep Learning models in Computer Vision
  - Demo of applying PyTorch on a Transfer Learning example
  - Demo of applying PyTorch on a real-time Object detection example
  - Intro to Medical Imaging Processing/Analysis
  - Demo of applying MONAI on a Medical Imaging processing
- What you can learn offline?
  - Neural network models used in the tutorial
  - Math and statistics of deep learning model

How are images represented in Computer?

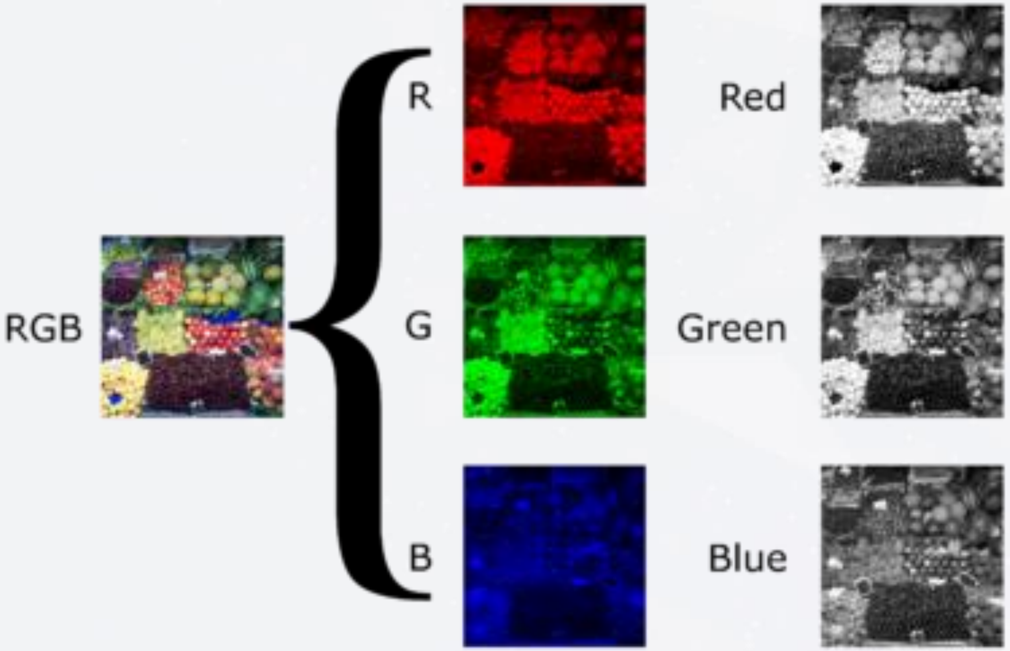
Grayscale



0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

1Byte 8-bit grayscale
numerical representations: [0, 255]

Color images



Color images are often built of several stacked color channels:
RGB: red, green, blue primary color
CMYK: cyan, magenta, yellow, black ink plate.

Basic image operations and manipulation

- Reading and Writing Image/Video I/O
- Pre-processing
  - Scaling, Interpolations, And Re-Sizing
  - Cropping
  - Padding
  - Perspective Transformation
  - Changing Color spaces: Gray-BGR, BGR-HSV
  - Thresholding, Adaptive Thresholding
  - Sharpening
  - Dilation, Opening, Closing, And Erosion
  - Blurring
  - Contours
  - Counting Circles And Ellipses
  - Image Pyramids
  - Mask
  - …

Feature extraction

Finding Lines, edges and ridges

Localized interest points such as corners, blobs or points.

Image/Video Processing Libraries

- C++
  - OpenCV
  - ITK (Insight Segmentation and Registration Toolkit)
- Python
  - OpenCV-Python
  - SimpleITK
  - Pillow/PIL
  - Sk-Image
  - Scipy
  - Rendering
    - Matplotlib
    - Seaborn
- Matlab

Data augmentation

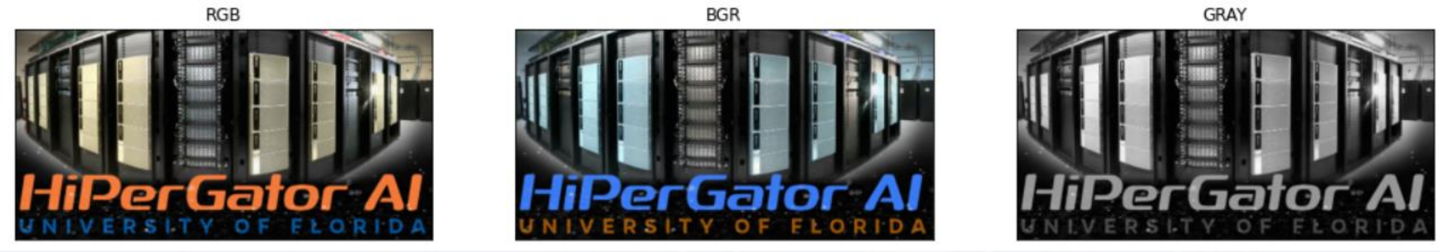
TorchVision

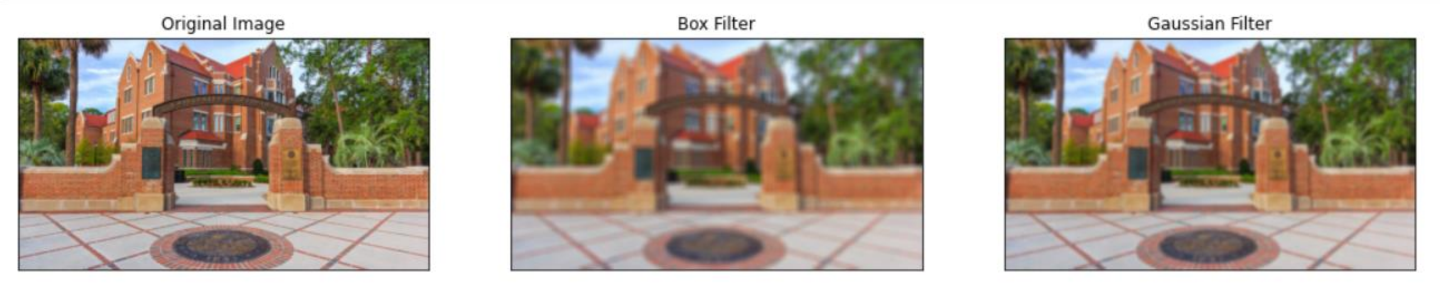
Albumentations

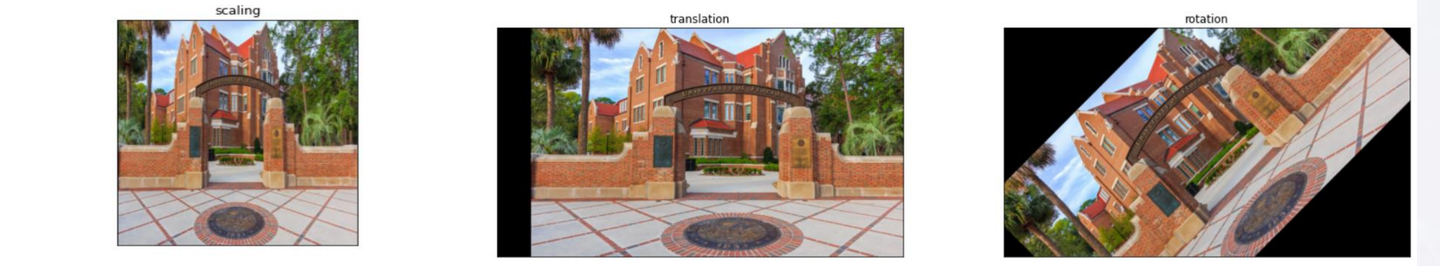
Fast and flexible image augmentations

Image Processing in OpenCV

- Modules
  - Image Filtering
  - Geometric Image Transformations
  - Miscellaneous Image Transformations
  - Drawing Functions
  - Color Space Conversions
  - Histograms
  - Feature Detection
  - Object Detection
  - Image Segmentation
  - …



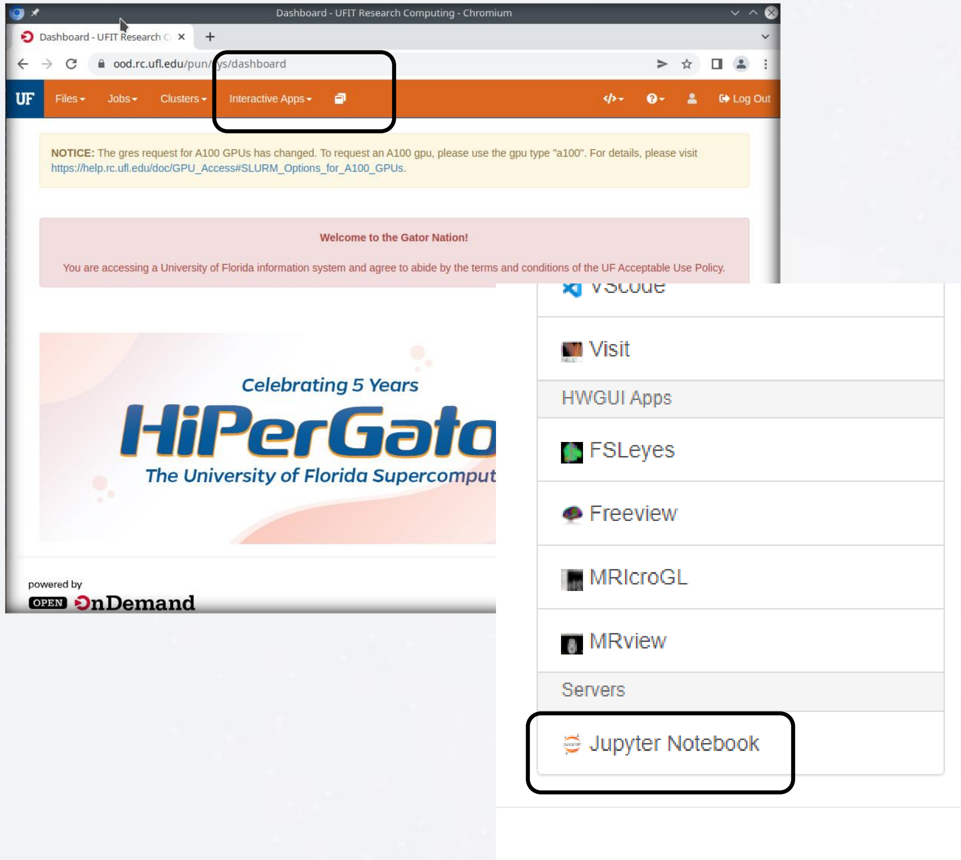




OpenCV HiPerGator Demo

Access jupyter notebook on HiPerGator

- To access the jupyterlab notebook on HPG with A100,
- 1 using Open On Demand application
  - Connect your web browser to: https://ood.rc.ufl.edu
  - Please note, if you are off campus or on the wifi, you will need to connect to the campus VPN service first: https://vpn.ufl.edu
  - Input resources to be allocated to this job.
- 2 using jupyter hub,
  - Connect your web browser to: https://jhub.rc.ufl.edu,
  - select the resources configuration with A100 GPU.
- More details can be found at UFRC help page: JupyterHub - UFRC (ufl.edu)


Dashboard - UFIT Research Computing - Chromium
Dashboard - UFIT Research C
ood.rc.ufl.edu/pun/sys/dashboard
UF
Files
Jobs
Clusters
Interactive Apps
Log Out

NOTICE: The gres request for A100 GPUs has changed. To request an A100 gpu, please use the gpu type "a100". For details, please visit https://help.rc.ufl.edu/doc/GPU_Access#SLURM_Options_for_A100_GPUs.

Welcome to the Gator Nation!

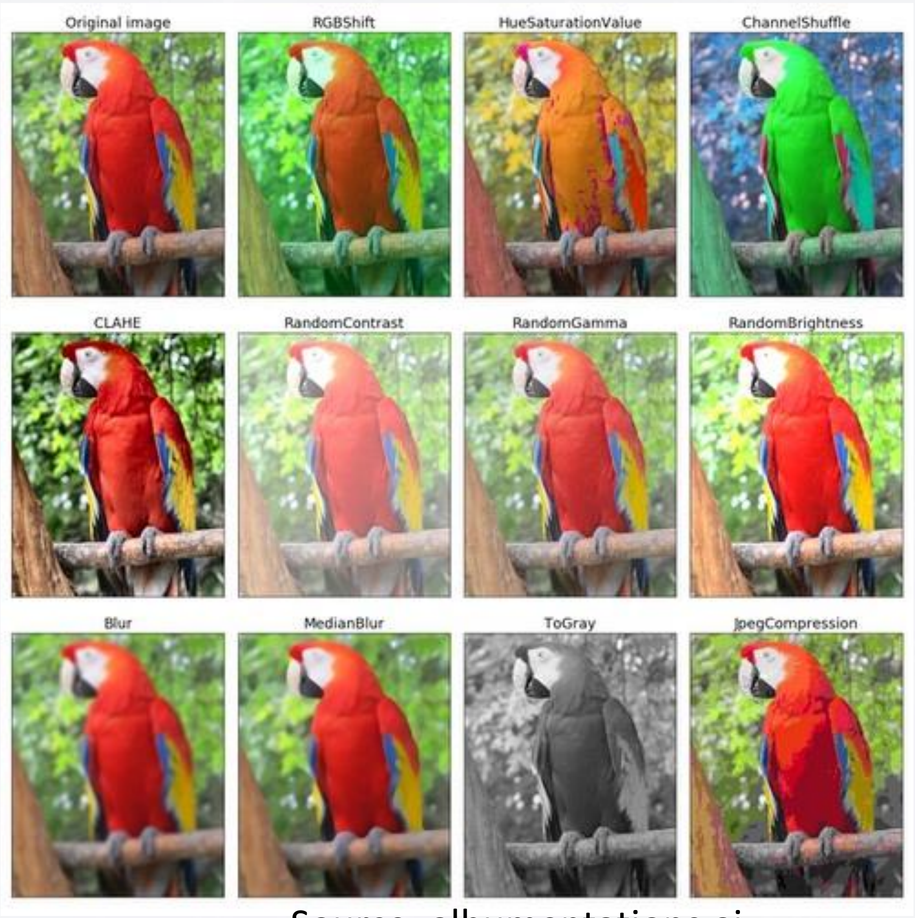
You are accessing a University of Florida information system and agree to abide by the terms and conditions of the UF Acceptable Use Policy.

torchvision.io

TorchVision

- Transforming and augmenting images
  - torchvision.transform on PIL image or tensor
    - Compose()
  - Functional Transforms
- Models and pre-trained weights
  - torchvision.models
  - torch.hub
- Datasets
  - subclasses of torch.utils.data.Dataset
  - Built-in dataset and Custom dataset
- Utils
  - draw_bounding_box,
  - draw_segmentation_mask,
  - draw_keypoints,
  - make_grid
  - save_image

Data augmentation

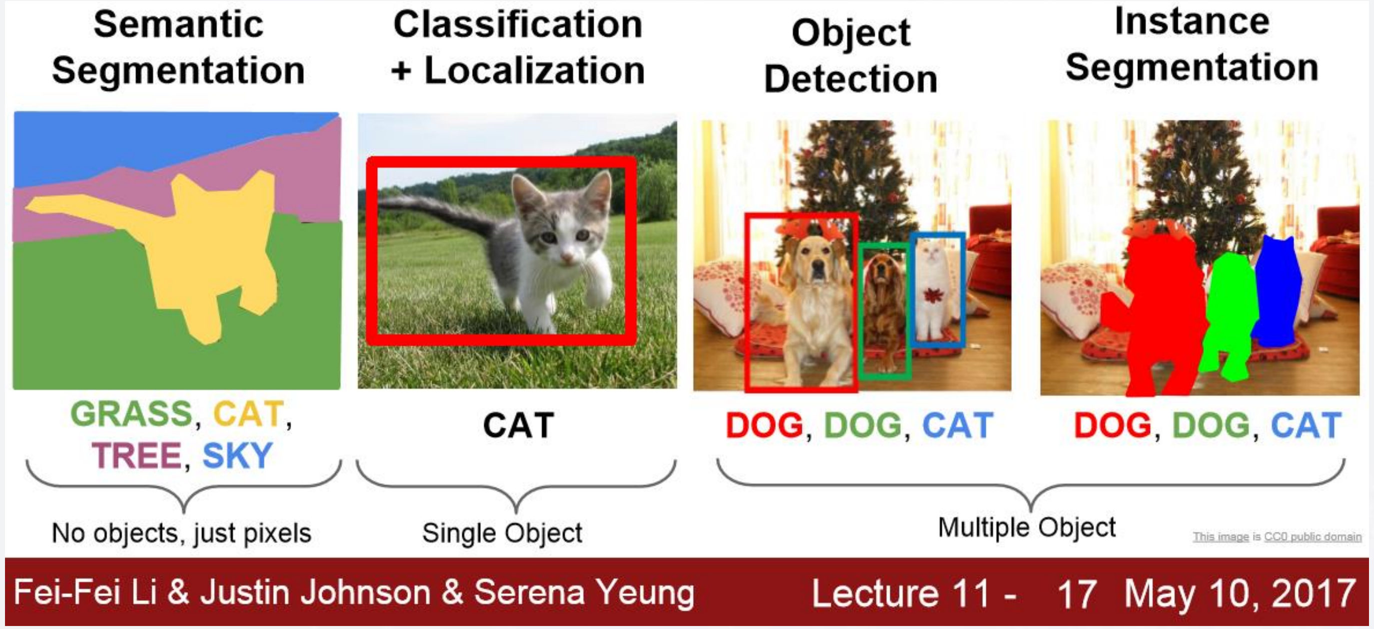


Source: albumentations.ai

TorchVision HiPerGator Demo

Computer Vision tasks

- Image classification
  - Backbone/feature extraction
  - Classifier
- Semantic Segmentation
  - Pixel-level segmentation
- Image Localization
  - Coordinates of bounding box
- Object Detection
  - Image classification + Localization
- Instance Segmentation
  - Semantic + Multiple objects segmentation



Semantic Segmentation | Classification + Localization | Object Detection | Instance Segmentation

GRASS, CAT, TREE, SKY | CAT | DOG, DOG, CAT | DOG, DOG, CAT

No objects, just pixels | Single Object | Multiple Object

This image is CC0 public domain

Fei-Fei Li & Justin Johnson & Serena Yeung Lecture 11 - 17 May 10, 2017

Image Classification

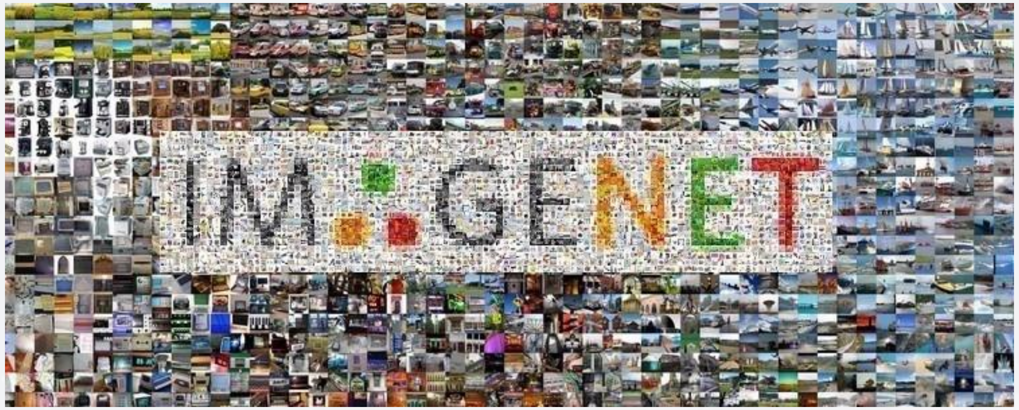
- fundamental task
- Models (keep evolving):
  - AlexNet, 2012
  - VGG
  - Inception
  - ResNet
  - …

Convolution models
Supervised learning

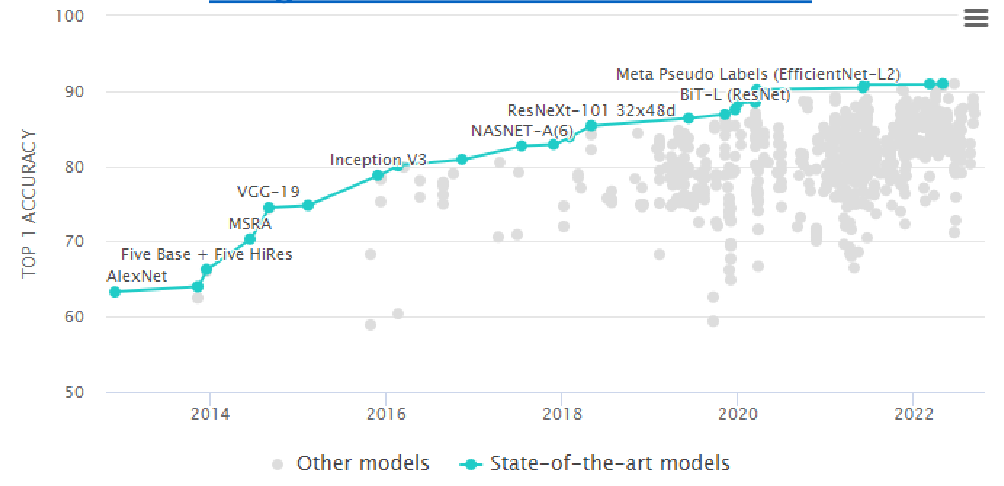
Transformer

  - ViT, 2020
  - ConvNext, 2022
  - CoAtNet, 2022
  - CoCa, 2022

Self-supervised learning

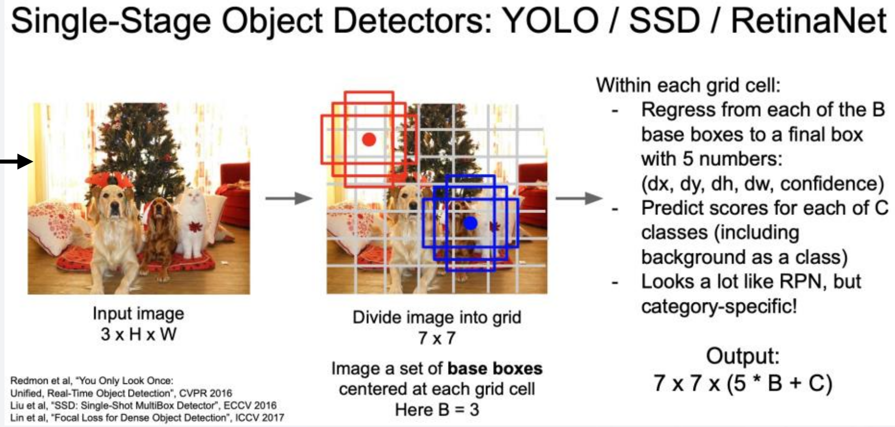


ImageNet benchmark Leaderboard



Detection/Segmentation

- One-stage Object Detection
  - SSD, 2015
  - YOLO, 2016
  - retinaNet, 2017
- Two-stages Region Proposals
  - R-CNN, 2013
  - Fast-RCNN, 2015
  - Faster-RCNN, 2015
  - Mask-RCNN, 2017
- Transformer based
  - DETR,2020
  - DINO, 2022
  - SwinV2, 2022



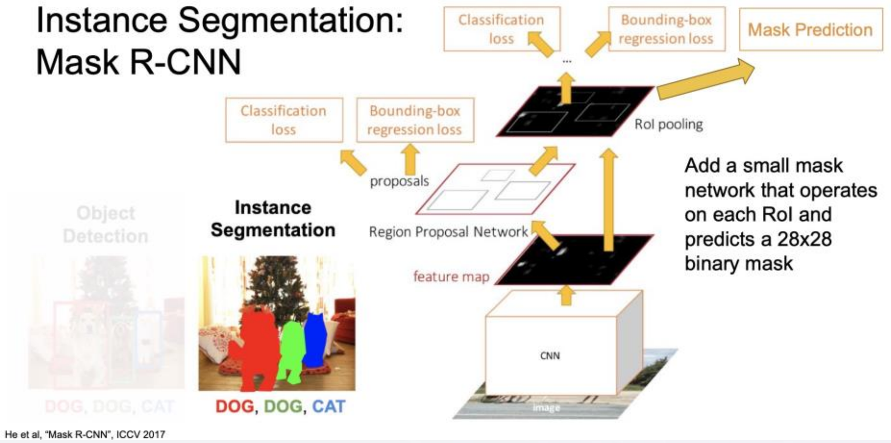
Single-Stage Object Detectors: YOLO / SSD / RetinaNet

Image a set of **base boxes** centered at each grid cell
Here B = 3

Within each grid cell:
- Regress from each of the B base boxes to a final box with 5 numbers:
  (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)
- Looks a lot like RPN, but category-specific!

Output:
7 x 7 x (5 * B + C)

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016
Lin et al, "Focal Loss for Dense Object Detection", ICCV 2017



Instance Segmentation:
Mask R-CNN

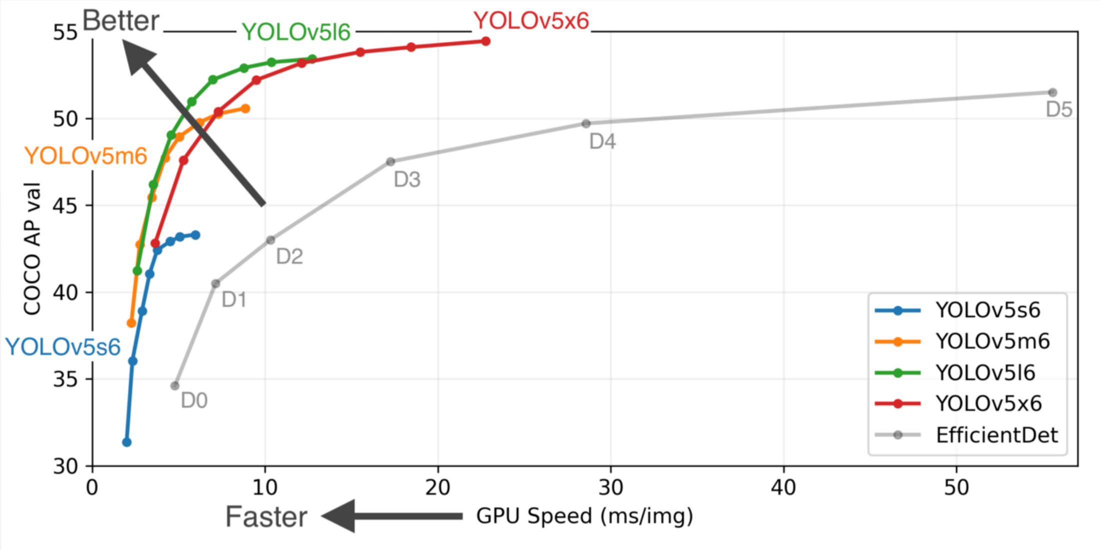
Add a small mask network that operates on each RoI and predicts a 28x28 binary mask

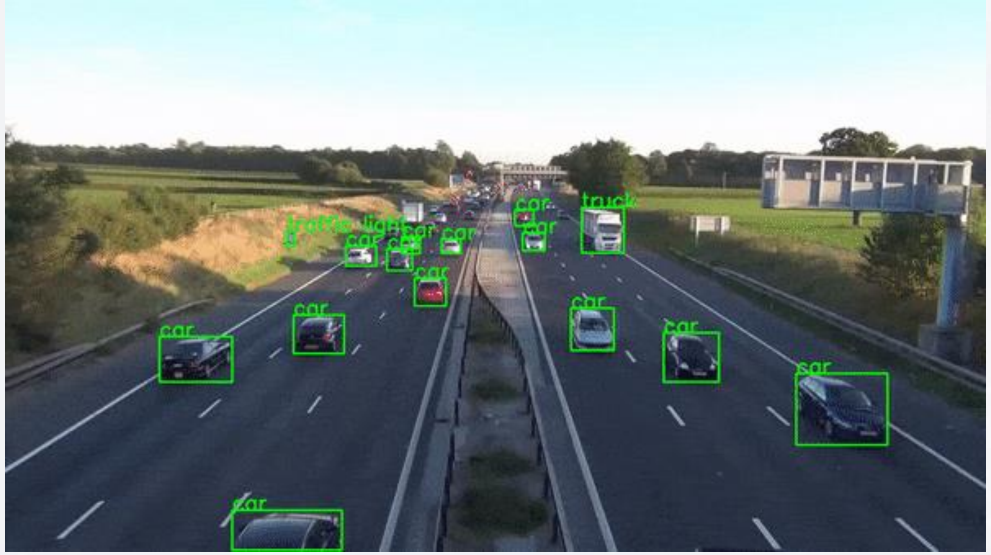
He et al, "Mask R-CNN", ICCV 2017

Real time Object Detection using PyTorch and OpenCV Demo

- Object detection
  - Using pre-trained YOLOv5 model on torch.hub
  - Reading The Video Stream or a webcam
  -

[Real time Object Detection Demo]()

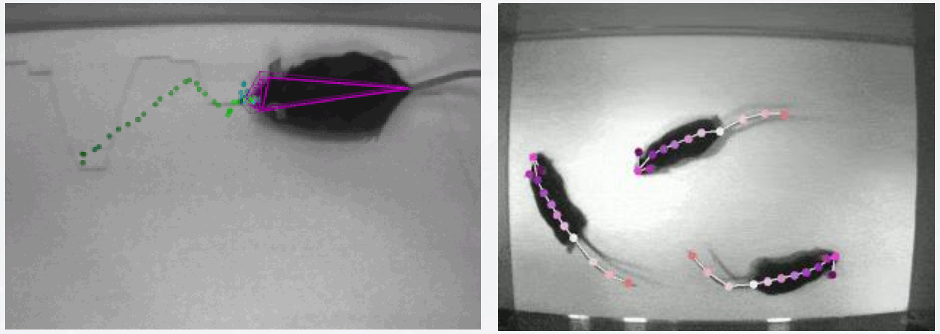




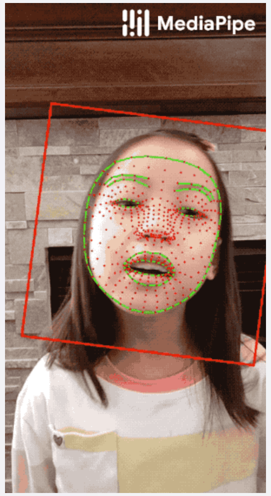
Other Computer Vision Applications

- **Pose Estimation**
- **Image generation**
- **Domain Adaptation**
- **Face Mesh/3D Face Animation**
- **Image registration**

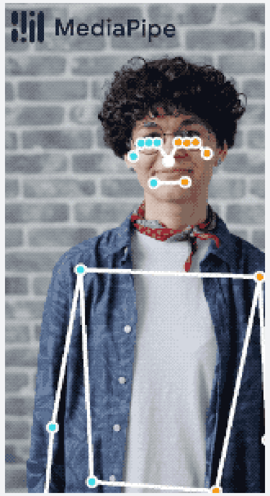
animal pose estimation Source: DeepLabCut



Face Detection



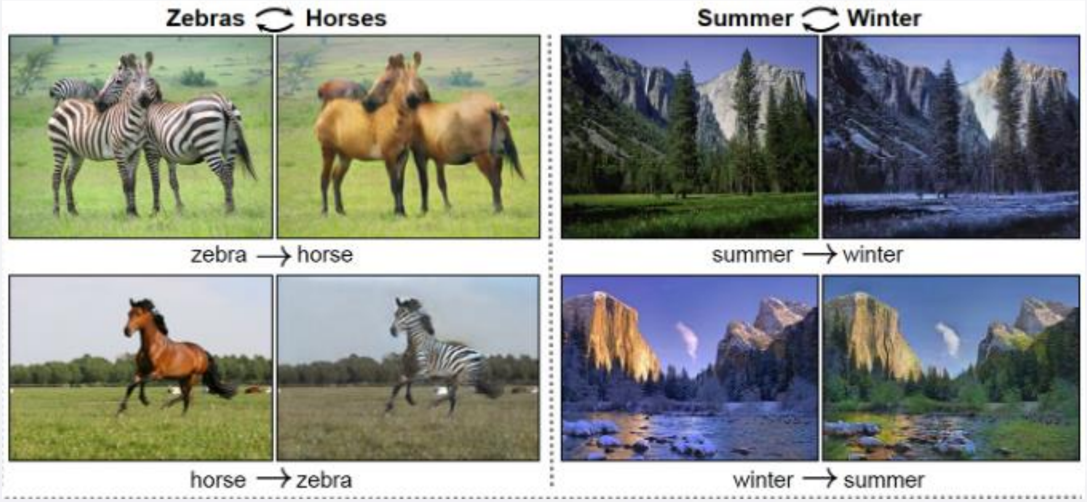
Face Mesh



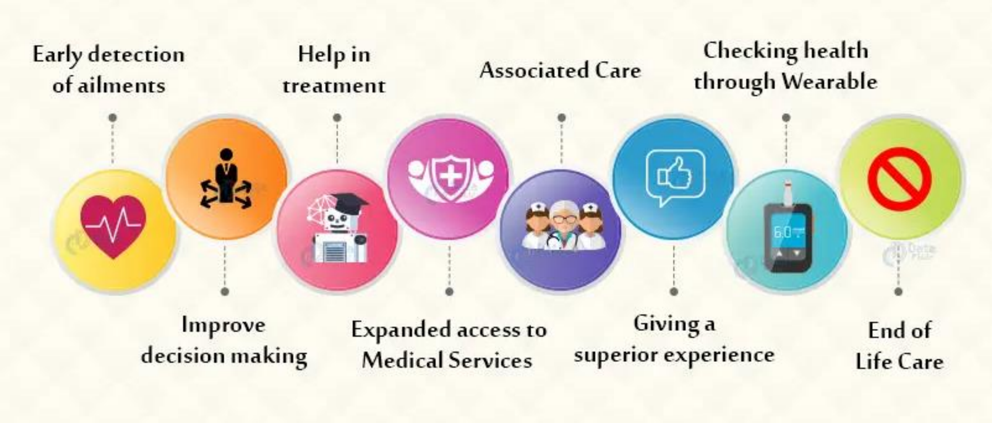
Pose estimation

Source: Mediapipe

Image-to-image translation

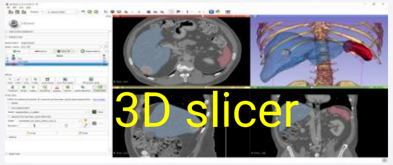


CV in Healthcare and Medicine


Early detection of ailments
Help in treatment
Associated Care
Checking health through Wearable
Improve decision making
Expanded access to Medical Services
Giving a superior experience
End of Life Care

- **Medical image processing**
  - Registration/labeling
  - Detection of Tutor
- **Drug discovery**
- **Machine-assisted Diagnosis and Treatment**
- **AI in genetics**

- **Tools/Libraries:**
  - Deep Learning framework
    - MONAI
    - MONAI label
  - SimpleITK
  - Visualization
    - 3D slicer
    - Itk-snap
    - …

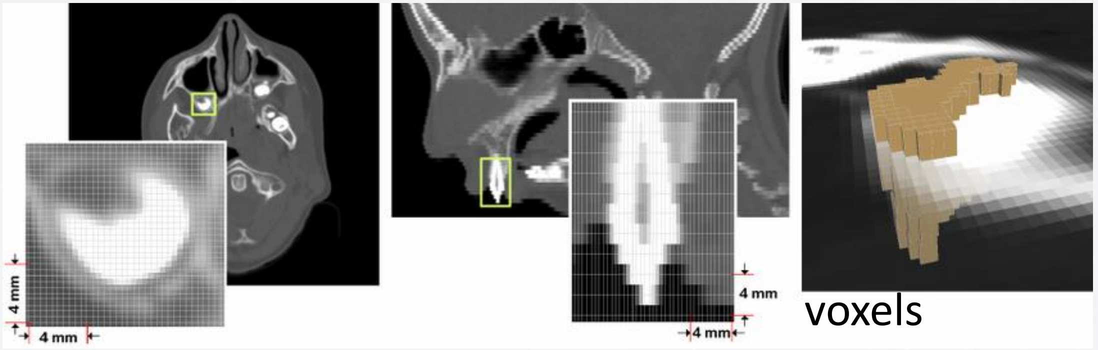


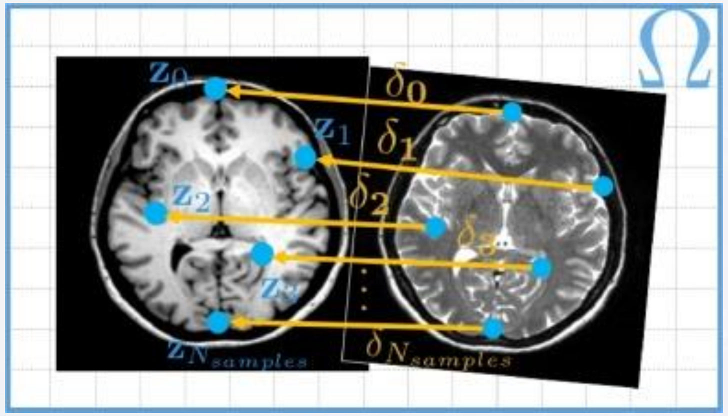
- **Modality**

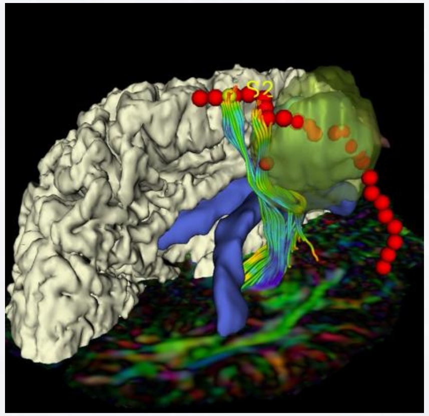
- **Format**
  - DICOM
  - Nifti
  - …
  - …

Intro to Medical Imaging Processing/Analysis

- 3-D voxels volume elements
- Modalities:
- Enhancement
  - Increase contrast
  - Remove noise
  - Emphasize edges: Edge boost, Unsharp masking.
  - Modify shapes
- Segmentation
- Registration
- Visualization
- Communication



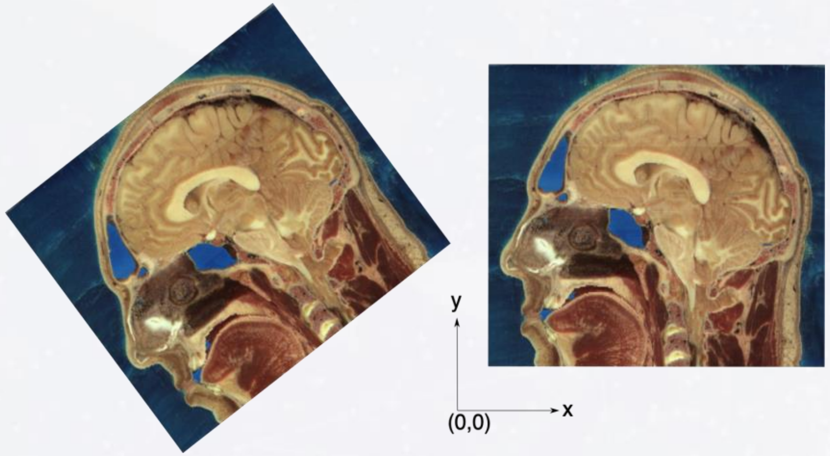




Process medical images in Python

- Medical images might have different extensions, "DICOM" and "NifTi"
- SimpleITK
  - a simplified programming interface to the algorithms and data structures of the Insight Toolkit (ITK) for segmentation, registration and advanced image analysis.
  - Fundamental concepts
    - Images
    - Transforms
    - Resampling
    - Registration
  - https://simpleitk.org/TUTORIAL/
- Nibabel
  - Support common neuroimaging file formats
- PyDICOM
  - DICOM medical image datasets, storage and transfer

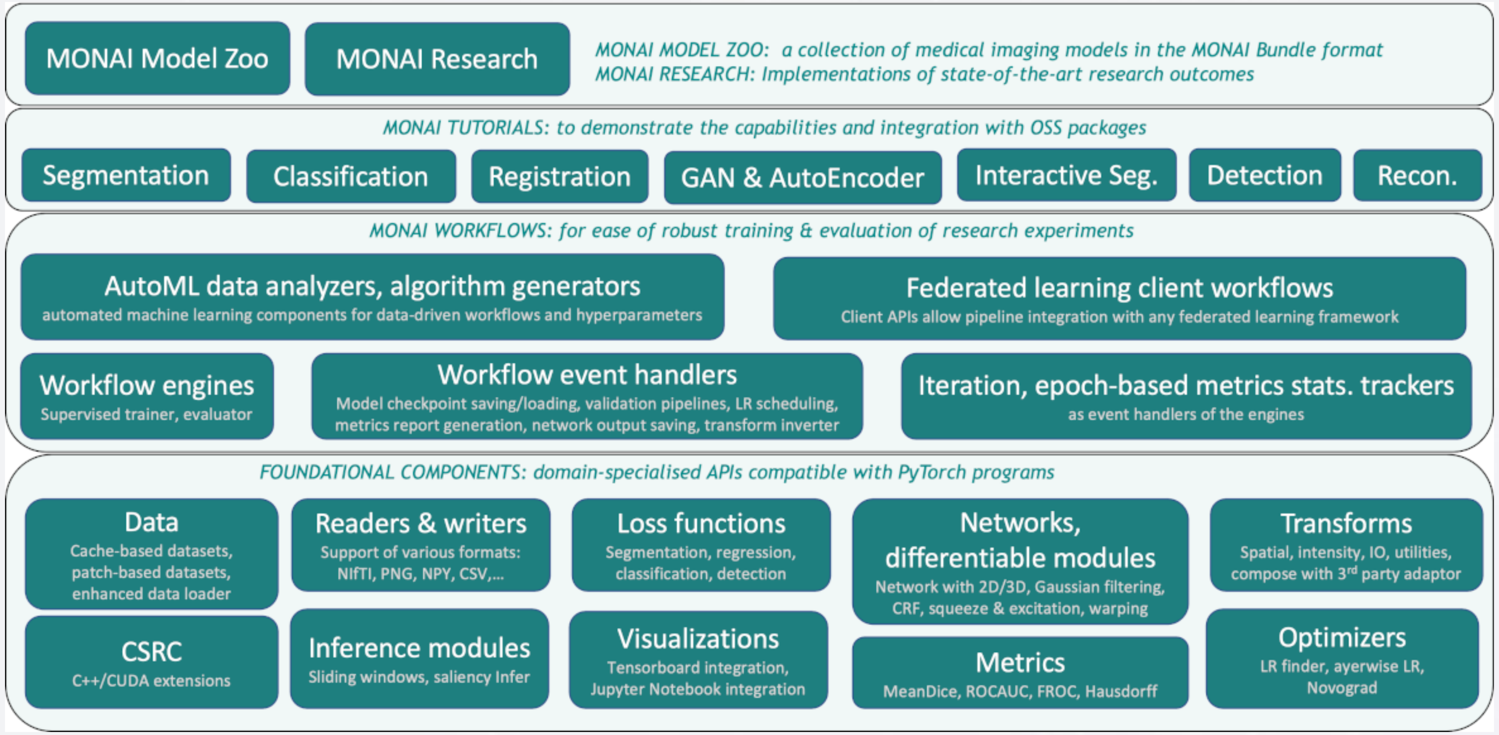
Exactly the same intensities, but in a different physical location.



MONAI (Medical Open Network for A. I.)

Pytorch-based open-source framework

- I/O, processing and augmentation
- Datasets and Data Loading
- Differentiable components, networks, losses and optimizers
- Evaluation
- Visualization
- Workflows
- Bundle
- Federated Learning
- Auto3dseg
- Multi-GPU parallel support



MONAI Model Zoo

MONAI Research

MONAI MODEL ZOO: a collection of medical imaging models in the MONAI Bundle format
MONAI RESEARCH: Implementations of state-of-the-art research outcomes

MONAI TUTORIALS: to demonstrate the capabilities and integration with OSS packages

Segmentation | Classification | Registration | GAN & AutoEncoder | Interactive Seg. | Detection | Recon.

MONAI WORKFLOWS: for ease of robust training & evaluation of research experiments

AutoML data analyzers, algorithm generators
automated machine learning components for data-driven workflows and hyperparameters

Federated learning client workflows
Client APIs allow pipeline integration with any federated learning framework

Workflow engines
Supervised trainer, evaluator

Workflow event handlers
Model checkpoint saving/loading, validation pipelines, LR scheduling, metrics report generation, network output saving, transform inverter

Iteration, epoch-based metrics stats. trackers
as event handlers of the engines

FOUNDATIONAL COMPONENTS: domain-specialised APIs compatible with PyTorch programs

Data
Cache-based datasets, patch-based datasets, enhanced data loader

Readers & writers
Support of various formats: NIfTI, PNG, NPY, CSV,...

Loss functions
Segmentation, regression, classification, detection

Networks, differentiable modules
Network with 2D/3D, Gaussian filtering, CRF, squeeze & excitation, warping

Transforms
Spatial, intensity, IO, utilities, compose with 3rd party adaptor

CSRC
C++/CUDA extensions

Inference modules
Sliding windows, saliency Infer

Visualizations
Tensorboard integration, Jupyter Notebook integration

Metrics
MeanDice, ROCAUC, FROC, Hausdorff

Optimizers
LR finder, ayerwise LR, Novograd

MONAI HiPerGator Demo

Questions?

Open datasets and CV models

Dataset hosted

- ImageNet
- CIFAR-10/CIFAR-100
- CelebA
- COCO
- Google Landmarks Dataset v2
- Open Images Dataset
- VisualGenome
- ADE20K
- Kinetics-400
- …

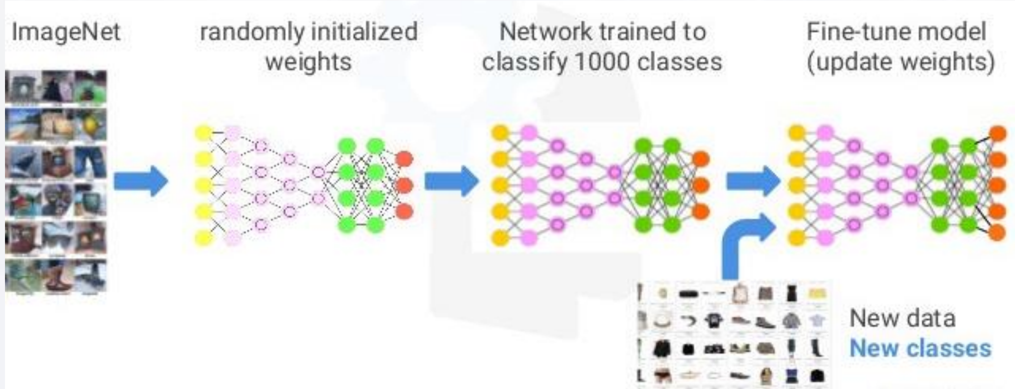
https://help.rc.ufl.edu/doc/AI_Reference_Datasets

COCO
Common Objects in Context

Transfer learning

- using Pretrained Model weights
- The backbone neural architecture can be download from the official release of Tensorflow or TorchVision.
- For example:

```
model_ft = models.resnet18(pretrained=use_pretrained)
```


ImageNet
randomly initialized weights
Network trained to classify 1000 classes
Fine-tune model (update weights)
New data
New classes

https://pytorch.org/vision/stable/models.html
https://www.tensorflow.org/tutorials/images/transfer_learning