In the name of God

Computer Vision

Alireza Saharkhiz

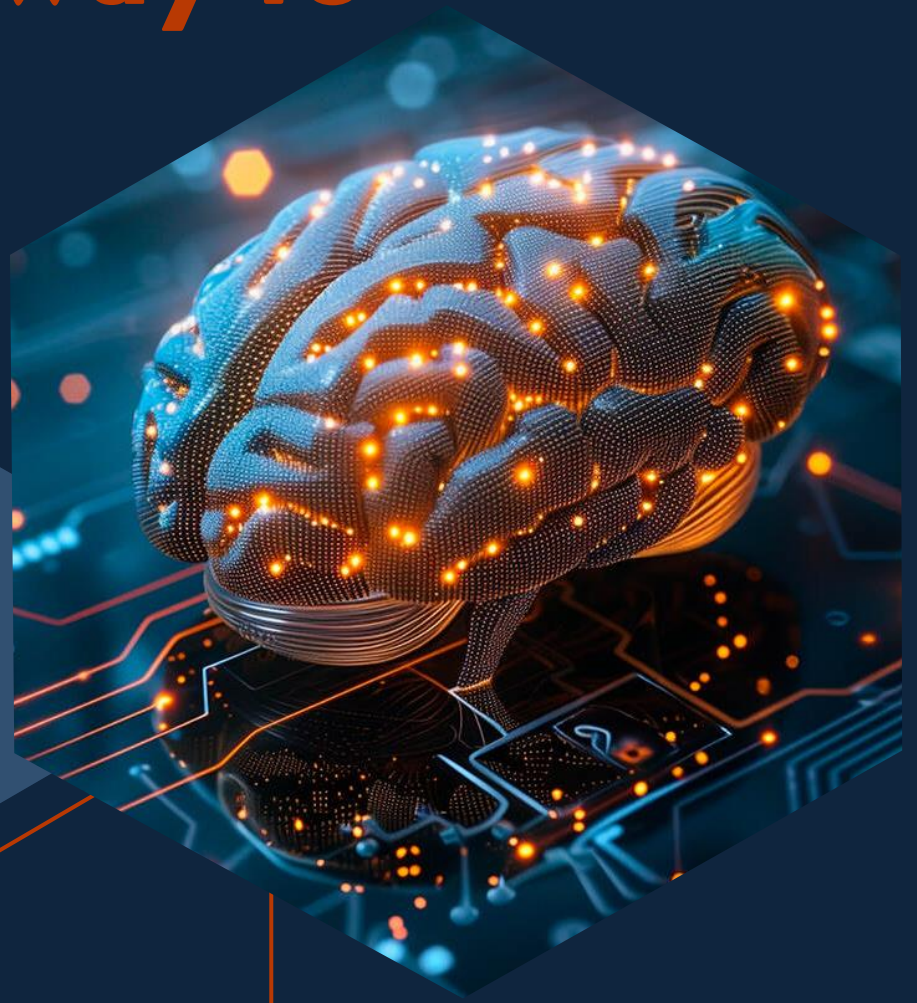
Supervisor:

Dr. Mohammad Arashi

Contents:

1. Introduction: A Gateway to the World of Artificial Intelligence

From self-driving cars navigating complex city streets to medical diagnoses aided by image analysis, Artificial Intelligence is already reshaping our lives. But how do these machines 'see' and interpret the visual data that surrounds us? This booklet opens the door to the captivating world of Computer Vision, the key technology that empowers AI to understand images and videos, unlocking a realm of possibilities.



Artificial Intelligence: A Foundation for Transforming the World

A Journey Through the History of AI

Artificial Intelligence (AI) may seem new, but the idea of creating intelligent machines has existed for centuries. Scientists and philosophers have long wondered how to build machines that think like humans.

The term "Artificial Intelligence" was first used at the 1956 Dartmouth conference, marking the official start of the field. However, AI development has had its ups and downs—periods of rapid progress, known as "AI summers," were followed by slowdowns or pauses, called "AI winters."

In the early stages, AI researchers focused on teaching computers to solve problems and reason, like playing chess or solving puzzles, by setting predefined rules. Later, "machine learning" emerged as a new approach, where computers learn from data instead of being explicitly programmed.

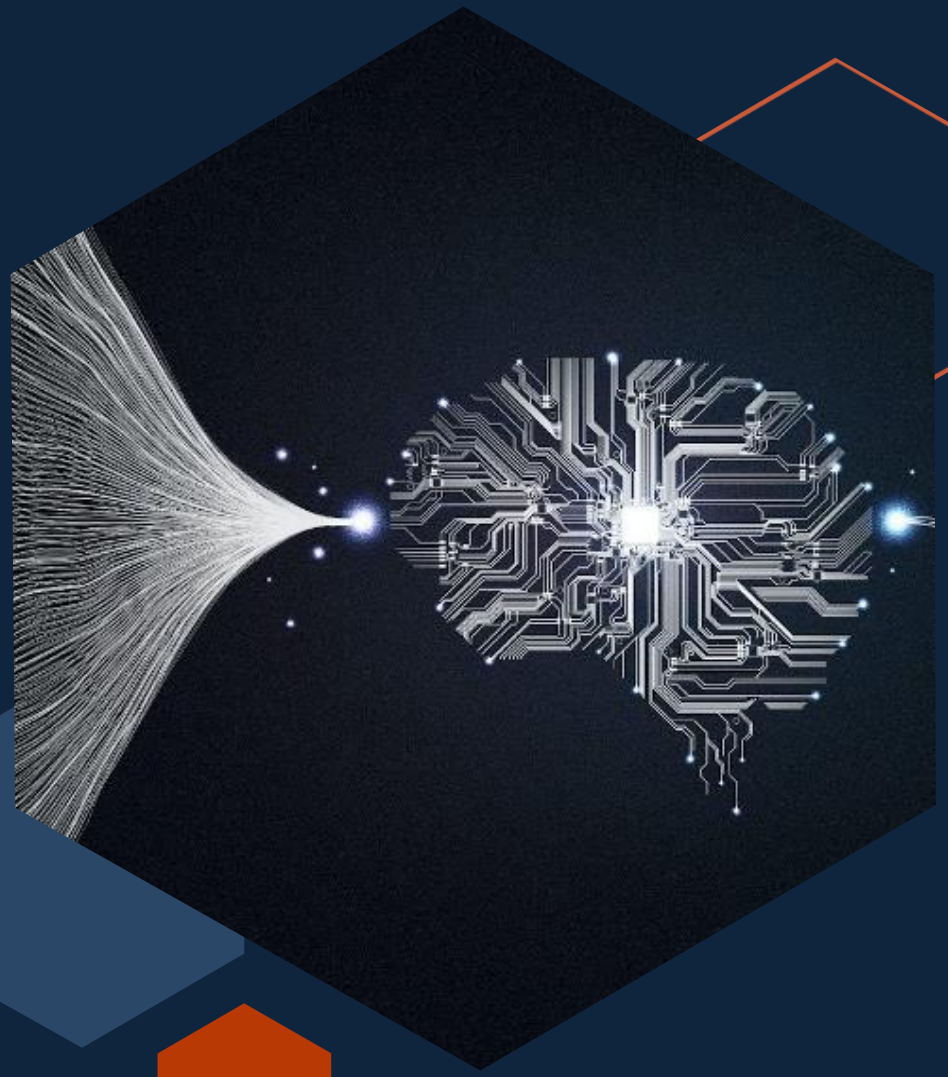
A key part of machine learning is "neural networks," which mimic the human brain (in a simpler way). Thanks to neural networks, computers can now perform complex tasks like recognizing faces and translating languages. From Alan Turing's early ideas to today's widespread AI, the ultimate goal has always been to create machines that think and learn like humans.

2. Machine Learning: The Beating Heart of AI

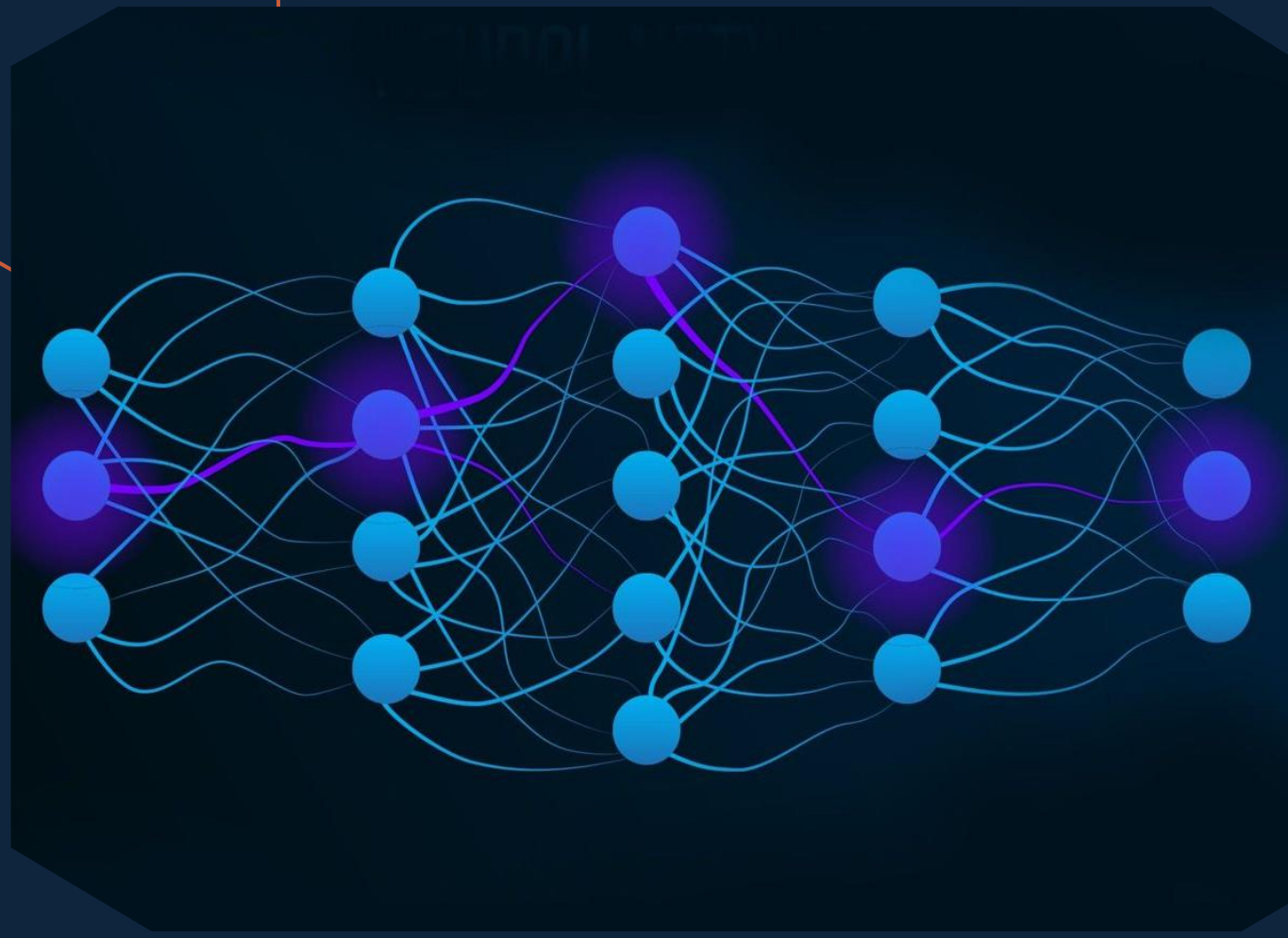
How do computers actually learn? It's not magic, it's Machine Learning. Prepare to explore the Beating Heart of AI and discover the secrets of artificial intelligence.

Machine Learning: How Computers Learn

Machine Learning revolutionized AI by introducing a paradigm shift: learning from data instead of explicit programming. This fundamental change unlocked AI's ability to solve complex problems, adapt to new conditions, and achieve breakthroughs across numerous domains, marking a true AI revolution.

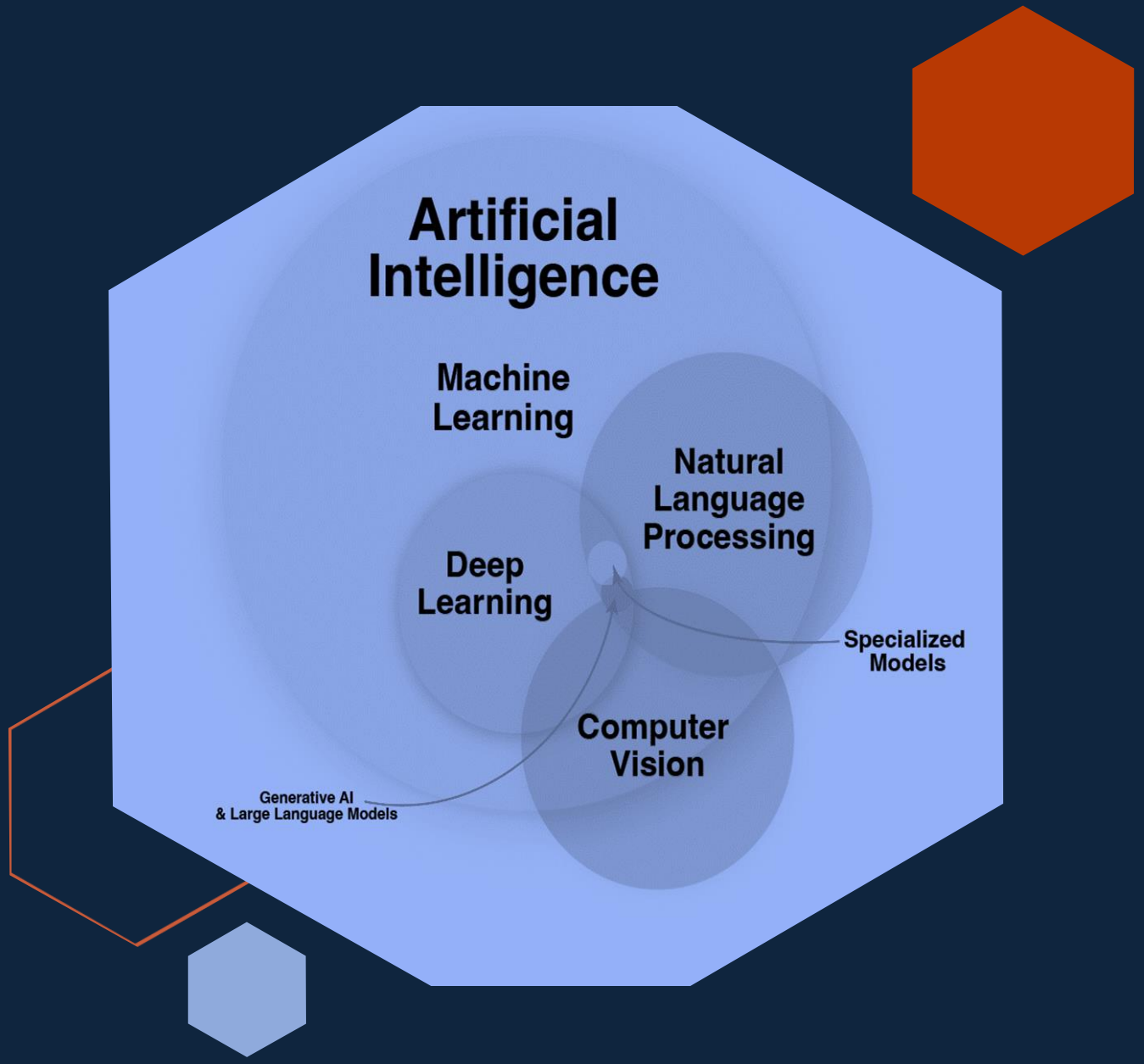


Artificial Neural Networks: Inspired by the Human Brain



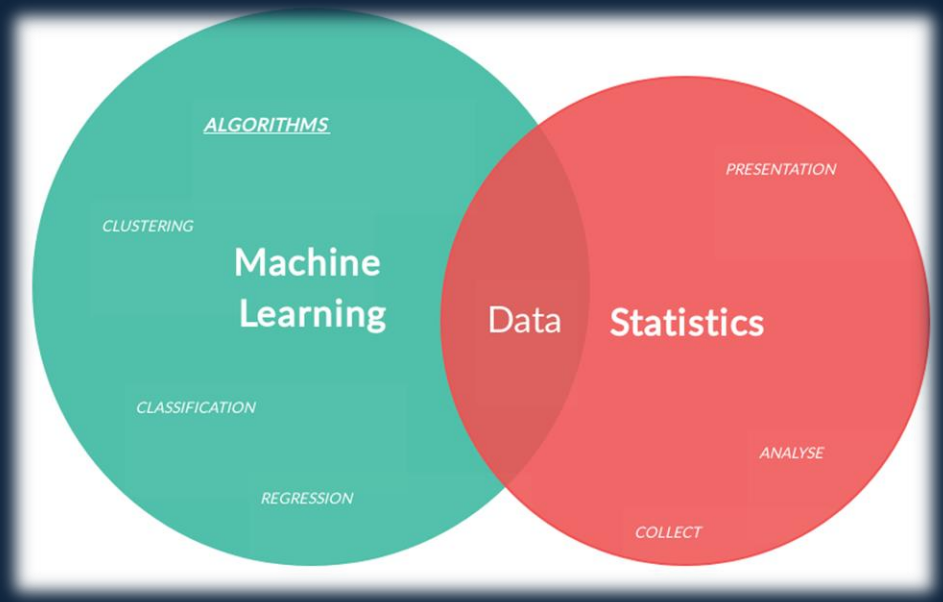
Machine Learning Vs Deep Learning

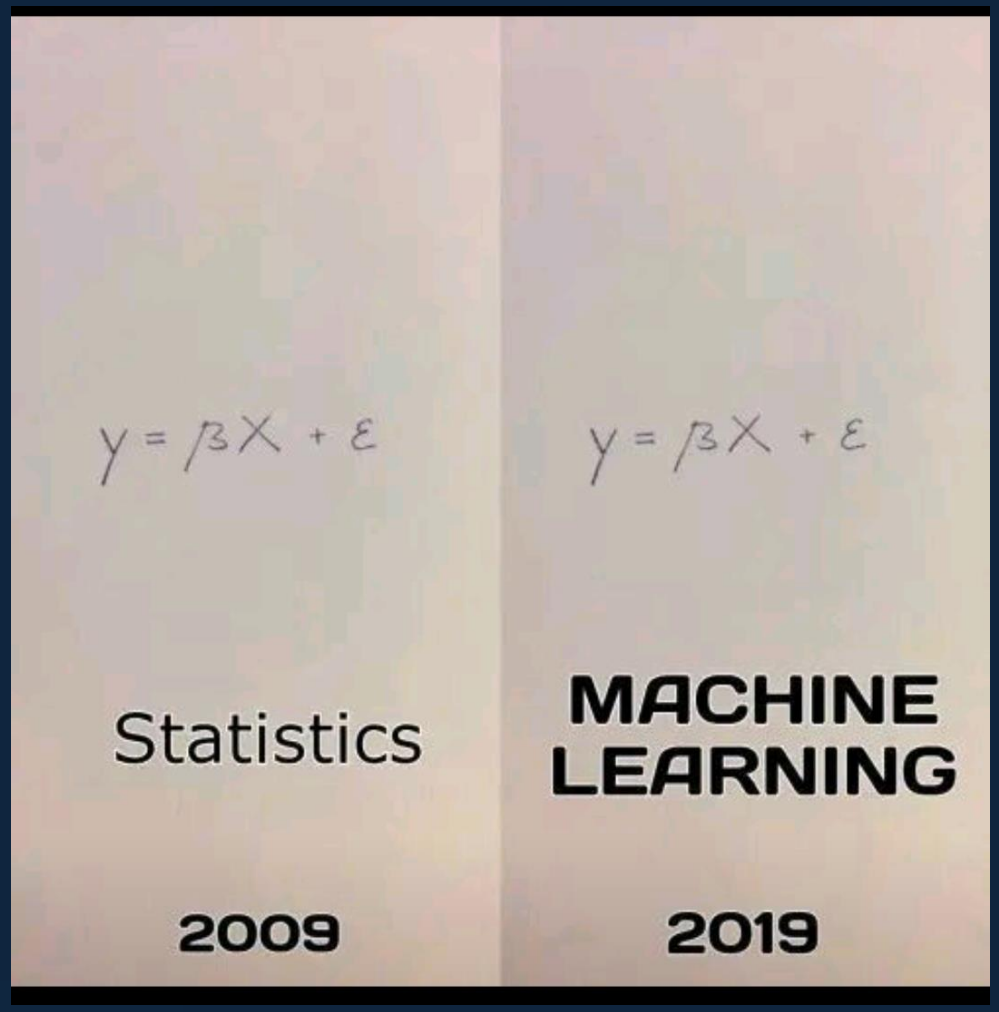
Aspect	Deep Learning	Machine Learning
Learning Approach	Mimics human brain, complex neural networks	Relies on statistical models and algorithms
Feature Extraction	Automated feature extraction, less human intervention	Manual feature engineering required
Data Requirement	Large datasets required for accurate predictions	Smaller datasets can be sufficient
Computational Power	High computational power needed, longer training time	Less computational power required, faster training
Application	Image recognition, natural language processing, speech synthesis	Predictive analytics, recommendation systems, fraud detection



Subsets of Artifical Intelligence

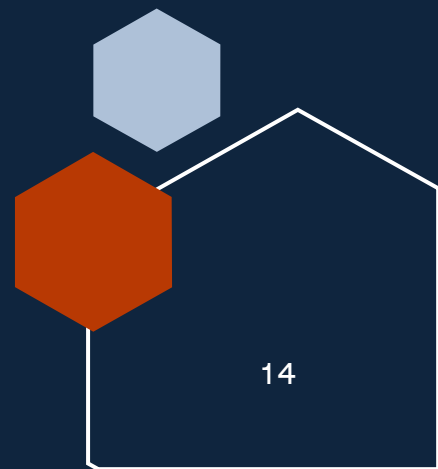
3. Statistics and Machine Learning: An Inseparable Bond




$y = \beta X + \varepsilon$
Statistics
2009
$y = \beta X + \varepsilon$
MACHINE LEARNING
2019

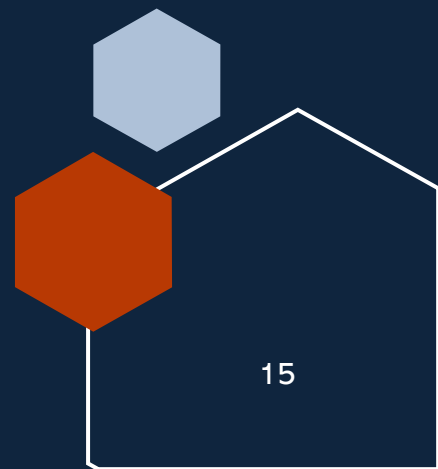
The Role of Statistics in Machine Learning

- Understanding and describing data
- Data preprocessing
- Feature Selection
- Dimensionality Reduction
- Machine learning algorithms
- Model Evaluation



Statistics at the heart of deep learning models

- Softmax Function
- Loss Functions
- Weight Initialization
- Batch Normalization
- Overfitting
- Evaluation



4. Computer Vision: Seeing the World Through Machine Eyes

The Concept of Computer Vision

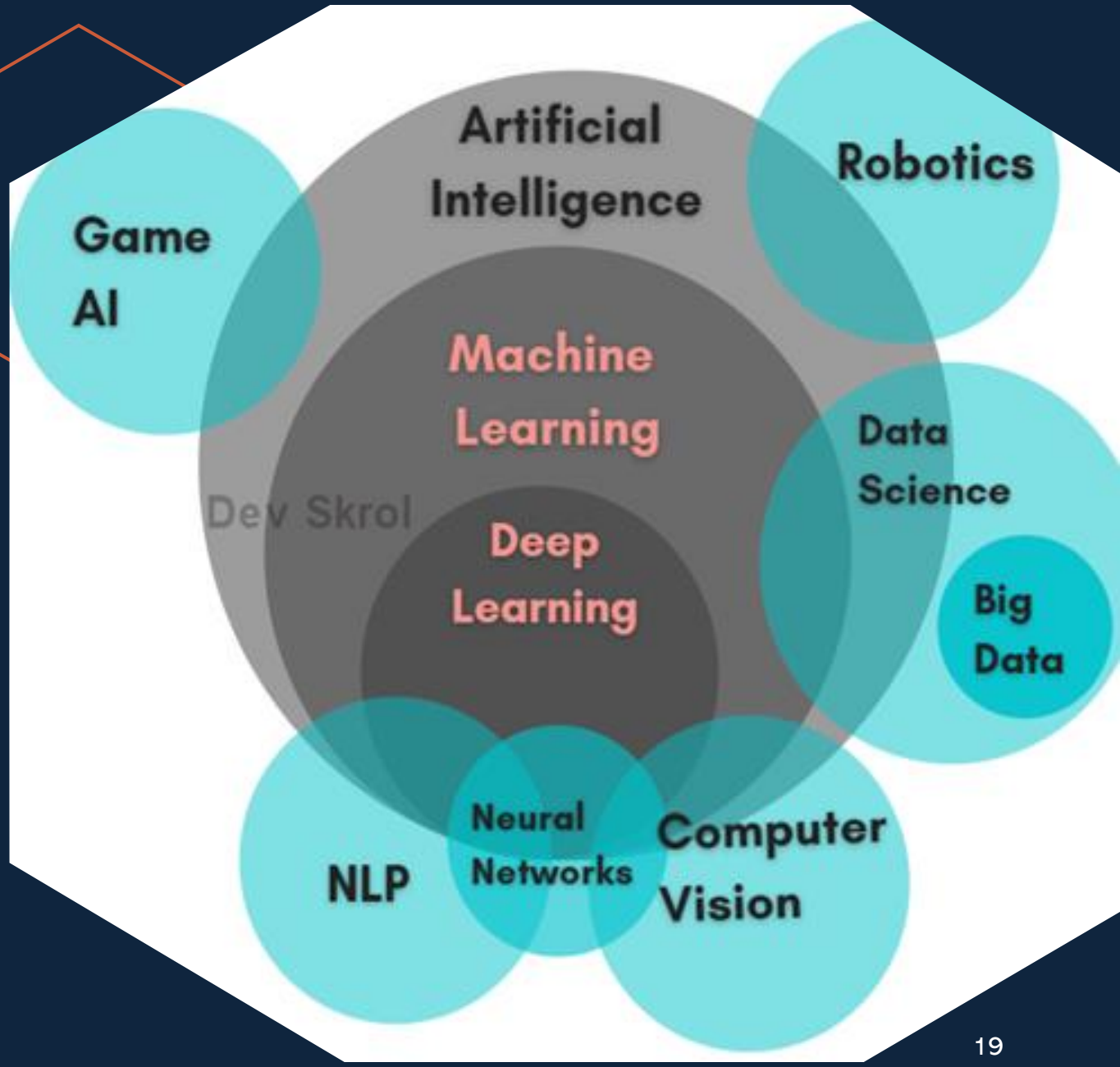
Computer Vision (CV) is a branch of Artificial Intelligence (AI) that enables computers to understand the visual world. Although it may seem simple at first glance, "seeing" for computers is much more complex than a digital camera that merely captures light. The main goal of computer vision is to extract meaningful and useful information from images and videos, and then use this information to perform specific tasks.

The Relationship Between Computer Vision and Machine Learning

Machine Learning (ML) plays a central and fundamental role in Computer Vision. In fact, it can be said that modern computer vision is virtually inconceivable without machine learning. Machine learning acts as the driving engine behind the recent significant advancements in the field of computer vision.

Transitioning from Classical Computer Vision to the Age of Deep Learning

Computer vision, like many other fields of Artificial Intelligence, has experienced a fundamental paradigm shift from "classic" approaches to "deep learning" (Deep Learning - DL) methods. This transition marks a revolution in the field, leading to significant advancements in the performance and capabilities of computer vision systems.



Classical CV vs Modern CV

Features	Classical CV	Modern CV
Feature Extraction	Hand-crafted: Feature engineering by humans using domain knowledge and traditional image processing algorithms	Automatic: Features are automatically learned by deep neural networks from raw data (pixels). No manual feature engineering is required.
Algorithms	Traditional machine learning algorithms (SVM, decision trees, KNN) after hand-crafted feature extraction. Image processing methods such as filters, edge detection, and rule-based methods.	Deep Neural Networks (CNNs, RNNs, Transformers) as the dominant architecture. Optimization algorithms (such as gradient descent) to train networks. Generative models (GANs, VAEs).
Training Data	Usually required less training data.	Requires a vast amount of training data to effectively train deep neural networks.
Performance and Accuracy	Limited accuracy and generalization ability, especially when dealing with the diversity and complexity of real-world data.	Much higher accuracy and much better generalization ability thanks to learning powerful and flexible features from large datasets.
Interpretability	Generally higher interpretability due to the use of hand-crafted features and simpler algorithms.	Lower interpretability (often referred to as "black boxes"). Ongoing research to improve the interpretability of deep learning models.
Computational Requirements	Requires less computational power.	Requires significantly more computation, powerful GPUs, and distributed computing platforms for training deep neural networks.

5. Tools and Challenges: The Toolkit for Computer Vision

Python: The Language of Machine Learning and Computer Vision

Python Libraries and Frameworks: The Computer Vision Toolbox

- OpenCV: The heart of Python's computer vision toolbox. OpenCV is a vast collection of algorithms and functions for image and video processing.
- NumPy: The computational foundation of Python's computer vision. Images in computers are represented as multi-dimensional arrays of numbers (pixels). NumPy provides efficient tools for manipulating and performing mathematical operations on these arrays.
- TensorFlow and PyTorch: Deep learning frameworks that have revolutionized computer vision. These frameworks provide the necessary tools to build and train deep neural networks.

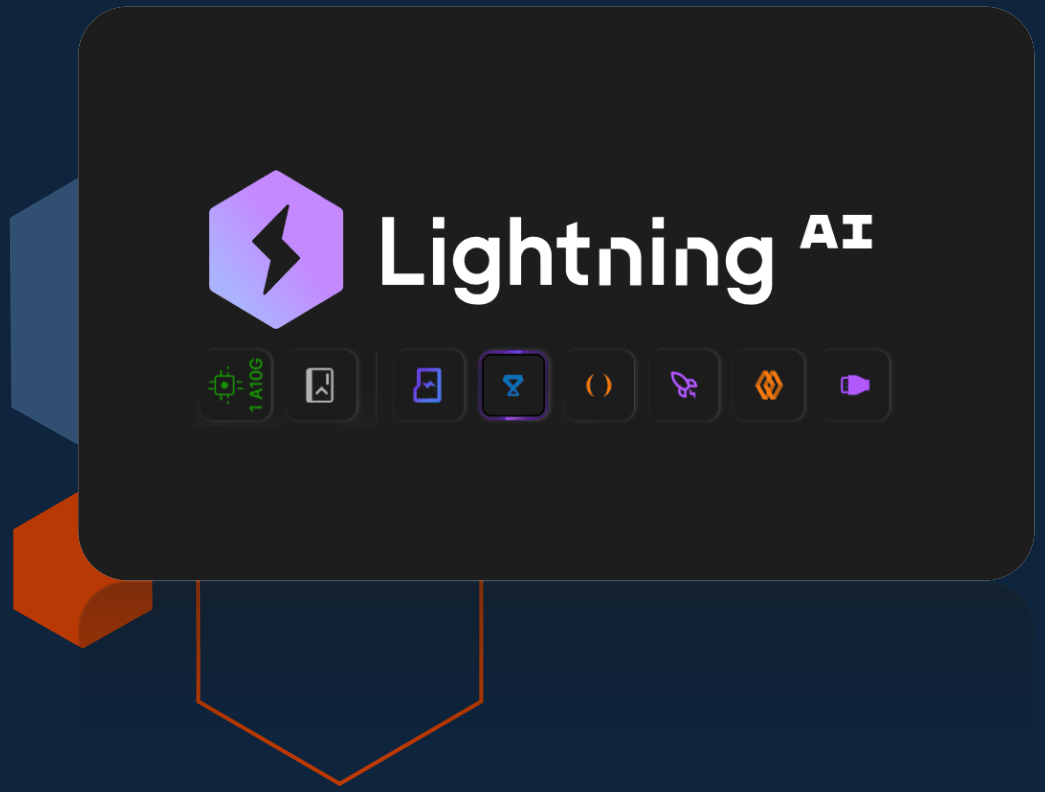
Pytorch VS Tensorflow

TensorFlow

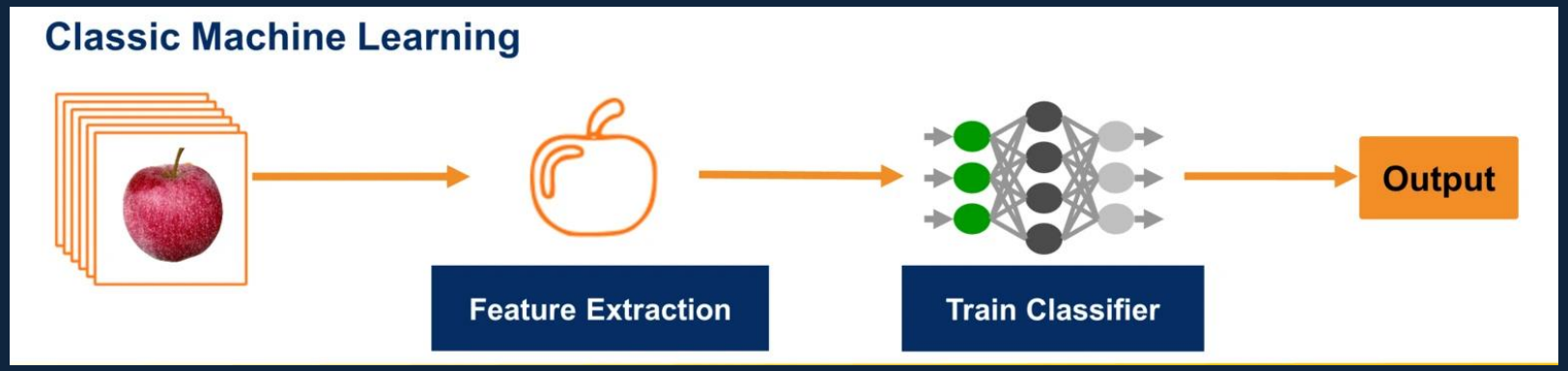
Modeling Requires Hardware and GPUs

Lightning Ai Studio

Nvidia L40S
Price: $9,000



6. Classical Computer Vision



Task:

- Image Processing
- Video Processing
- Object Detection
- Face Detection
- Text Detection (OCR)

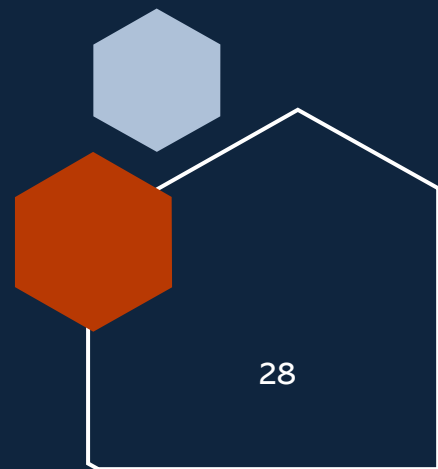


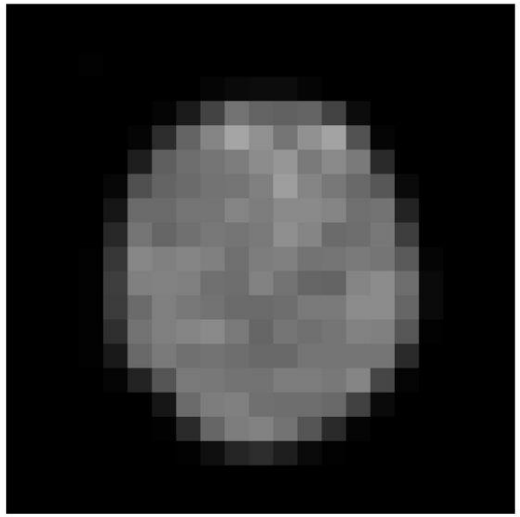
Image Processing

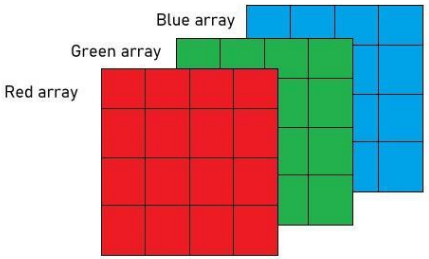
- Import openCV and working with image
- Drawing on image
- Transformation on image
- Operation on image
- Filter

Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Classical%20Computer%20Vision/Image%20Processing/ImageProcessing.ipynb

Working with Image

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	2	1	1	2	1	1	1	2	1	2	1	2	1	1	1	1	1	1	1	1	1
2	2	1	2	3	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1
3	2	1	1	2	2	2	2	2	4	6	7	7	4	2	2	1	1	1	1	1	1
4	2	2	2	2	2	2	4	13	29	48	42	41	42	29	7	2	1	1	1	1	1
5	2	2	2	2	2	4	20	42	55	68	60	49	60	67	35	4	1	1	1	1	1
6	1	2	2	2	2	14	40	46	49	52	58	62	51	57	51	19	2	1	1	1	1
7	1	1	2	2	5	34	42	45	48	49	53	65	55	50	44	37	6	1	1	1	1
8	2	2	2	2	11	45	45	49	49	55	52	57	57	55	47	48	16	2	1	1	1
9	1	1	2	2	18	49	43	47	50	49	51	59	55	45	46	50	27	2	1	1	1
10	1	1	2	3	24	56	53	55	53	47	52	53	48	49	51	49	38	4	1	1	1
11	1	2	2	3	27	53	54	53	47	46	54	48	43	42	55	58	46	6	1	1	1
12	1	2	2	3	26	45	53	50	48	45	45	46	51	51	58	58	43	6	2	1	1
13	1	1	2	3	21	48	53	55	57	51	43	49	47	49	54	53	37	3	1	1	1
14	1	1	2	3	12	44	51	47	48	46	44	44	49	49	49	49	19	2	2	1	1
15	2	1	1	2	4	16	36	51	50	48	48	52	52	50	55	30	4	2	2	2	1
16	1	1	1	2	2	2	11	45	52	53	47	46	46	44	28	6	2	2	2	1	1
17	1	1	1	2	2	1	3	19	43	52	54	48	36	19	4	2	2	1	1	1	1
18	1	1	1	1	2	1	2	3	8	17	20	14	6	3	2	1	1	1	2	1	1
19	1	1	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1	2	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

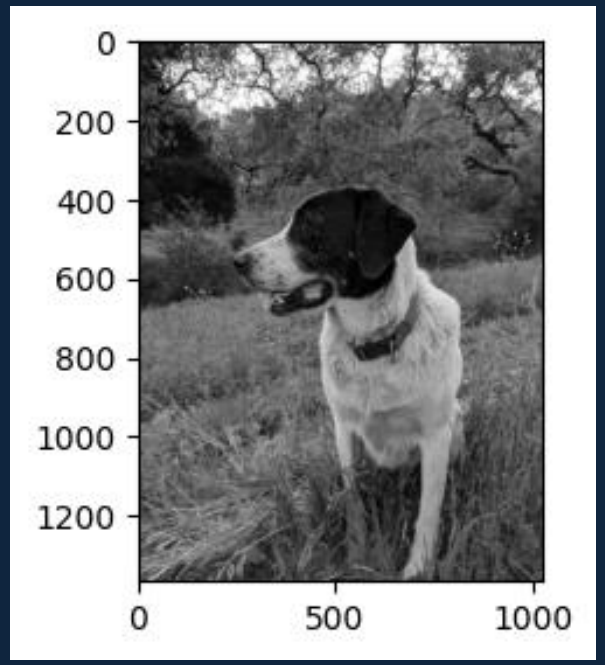


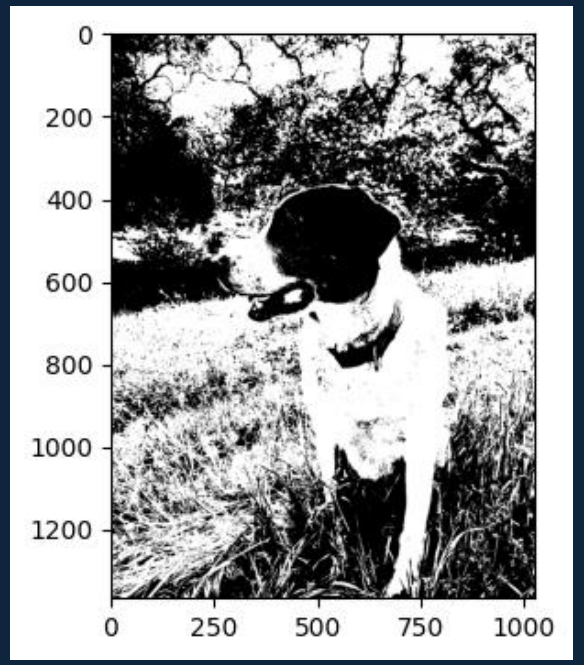


Arrrays stacked over each other
to form a Digital Image.

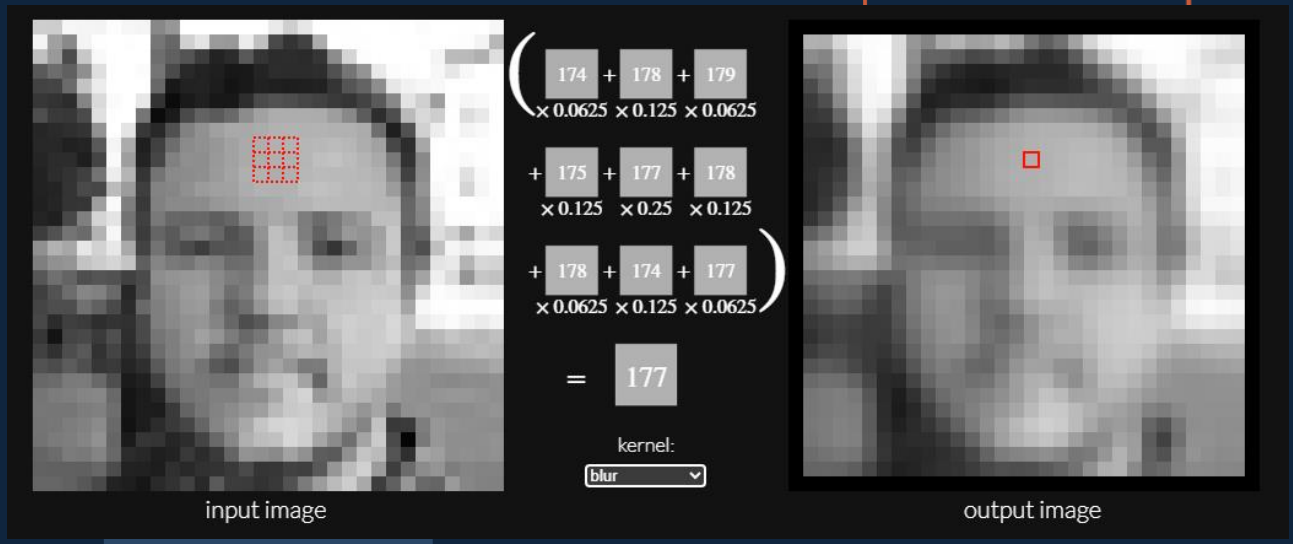
operation on image

thresholding:





Filter



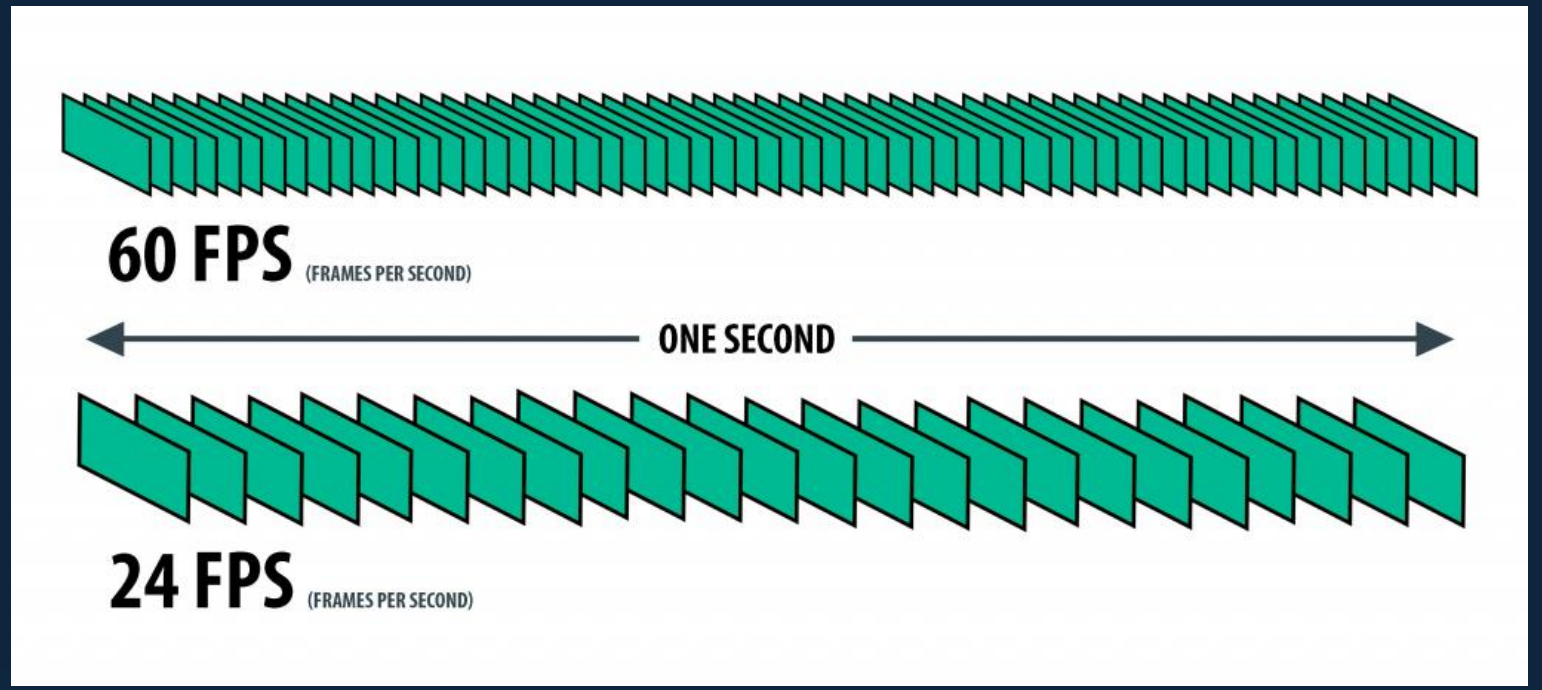
Link site: https://setosa.io/ev/image-kernels/

Video Processing

- Webcam
- live sketch with webcam
- video file
- save video

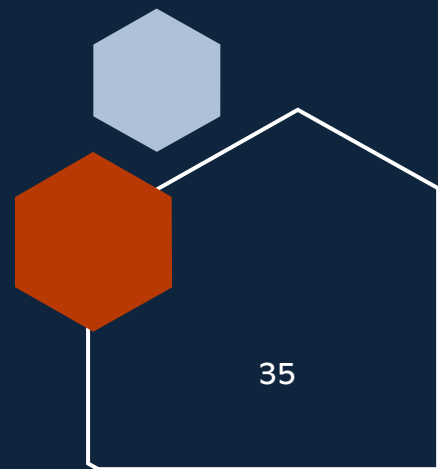
Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Classical%20Computer%20Vision/Video%20Processing/WorkingWithVideo.ipynb

Frame



Object Detection

- dilation, erosion
- edge detection
- contour detection
- corner detection
- template matching
- feature matching
- watershed algorithm
- pedestrian detection with Haar Cascade Classifier
- car detection with Haar Cascade Classifier

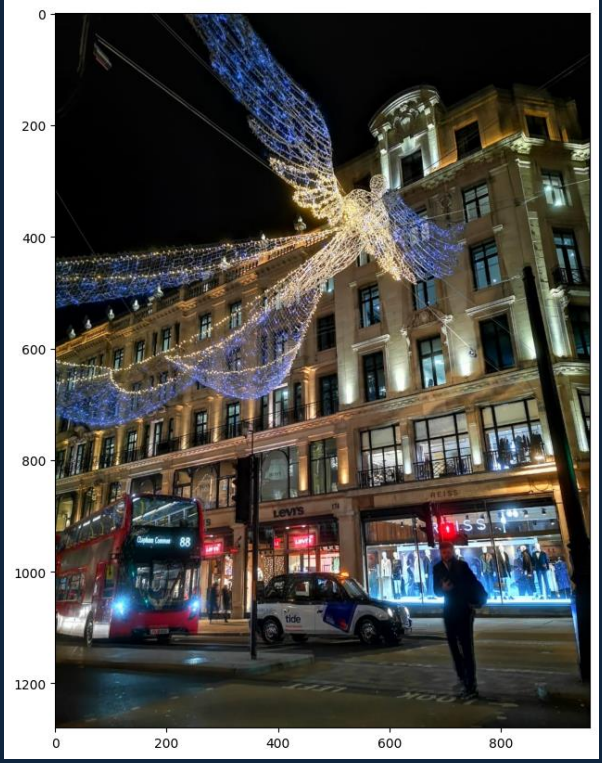


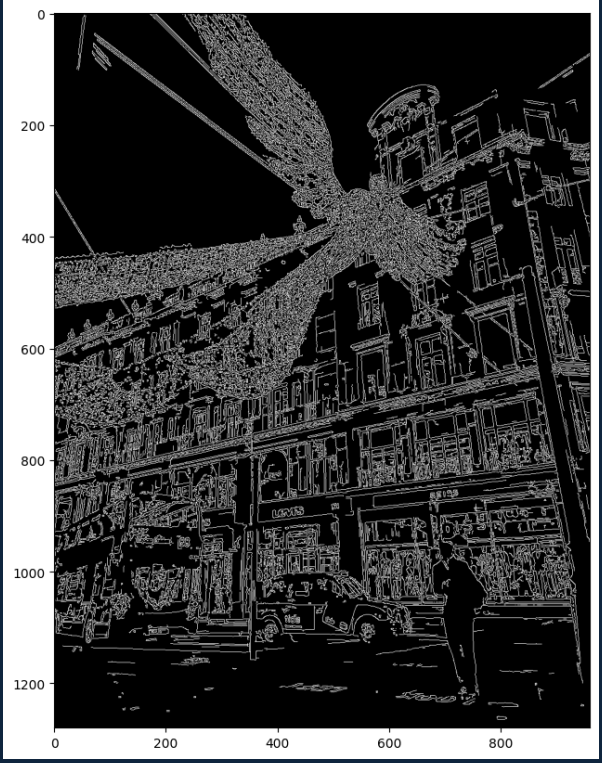
Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Classical%20Computer%20Vision/Object%20Detection/ObjectDetection.ipynb

Edge Detection

Canny Edge Detection Algorithm:

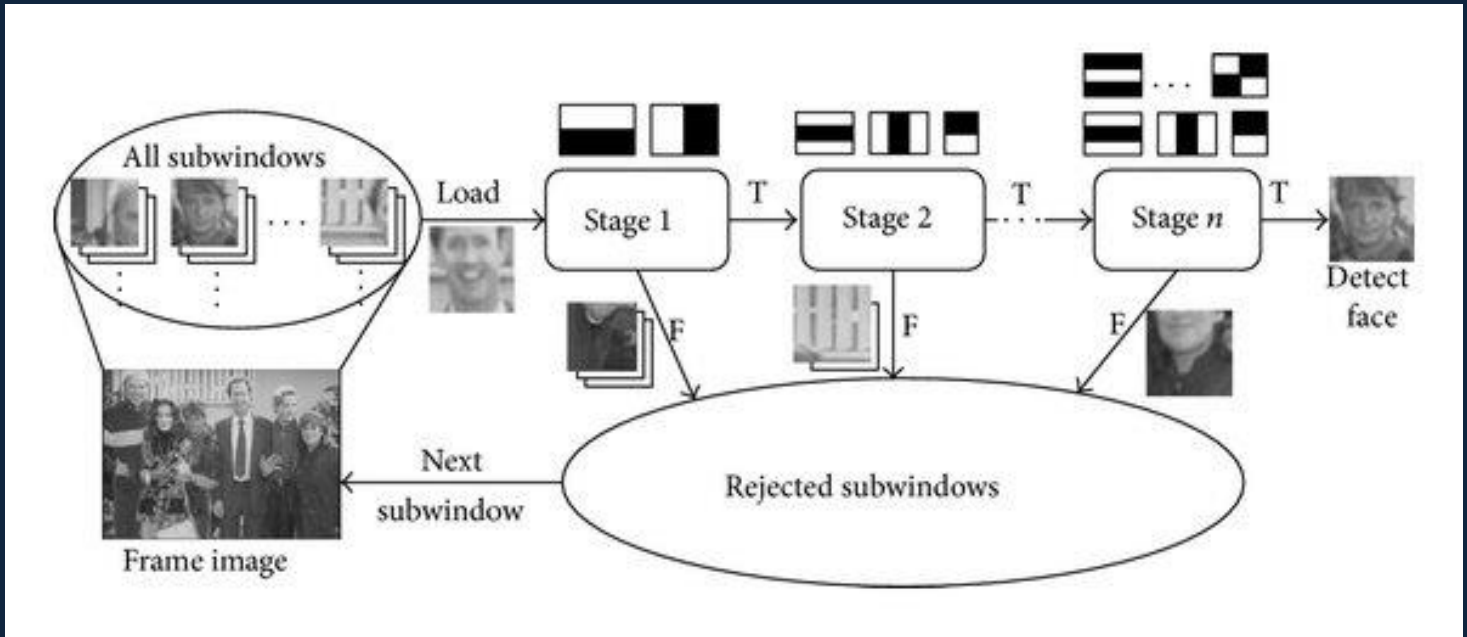
1. Noise Reduction
2. Gradient Calculation
3. Non-Maximum Suppression
4. Thresholding





Haar Cascades

- Haar cascades was proposed by P. Viola and M. Jones in 2001
- It is a machine learning method
- Trained on both positive and negative images



pedestrian detection with Haar Cascade Classifier

Face Detection

- Face detection in the image
- Face detection in the webcam
- Eye detection

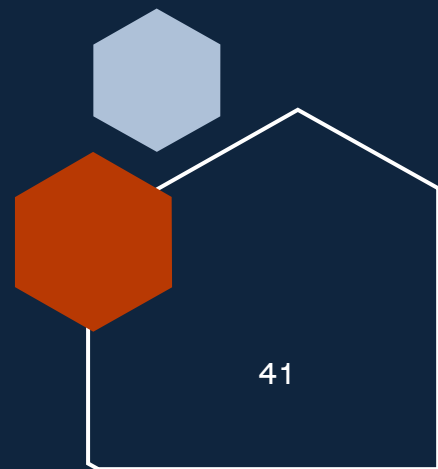
Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Classical%20Computer%20Vision/Face%20Detection/FaceDetection.ipynb

Face Detection Using Haar Cascades

Text Detection (OCR)

Optical Character Recognition

- car license plate detection
- blurring a car license plate
- OCR with pyTesseract
- OCR with easyocr
- convert car license plate to text



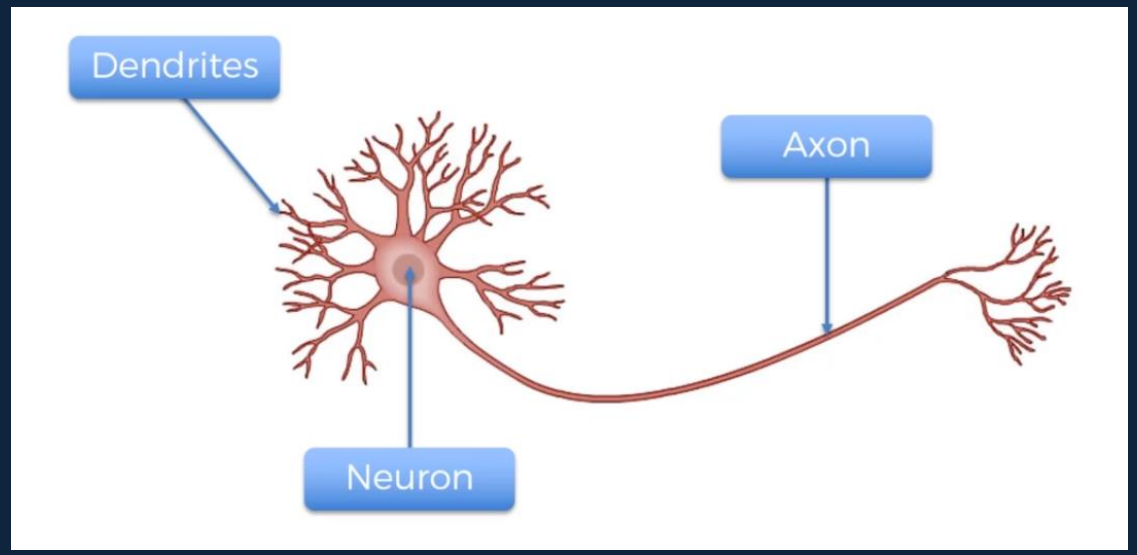
Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Classical%20Computer%20Vision/OCR/TextDetection.ipynb

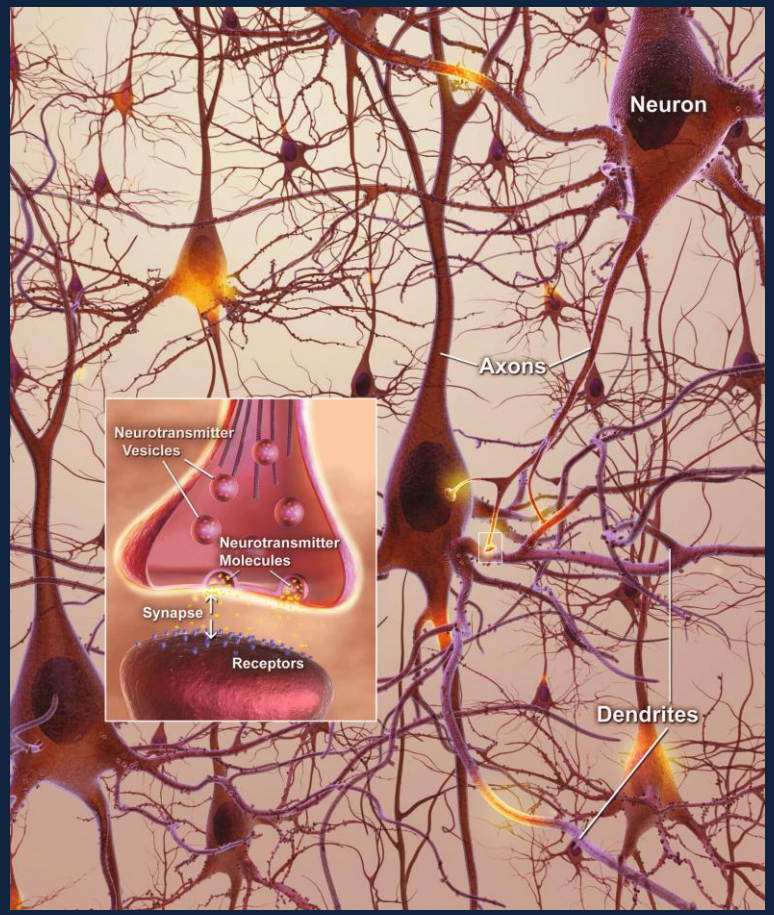
7. Modern Computer Vision: The Age of Deep Learning

Artificial Neural Network Architectures (ANN)

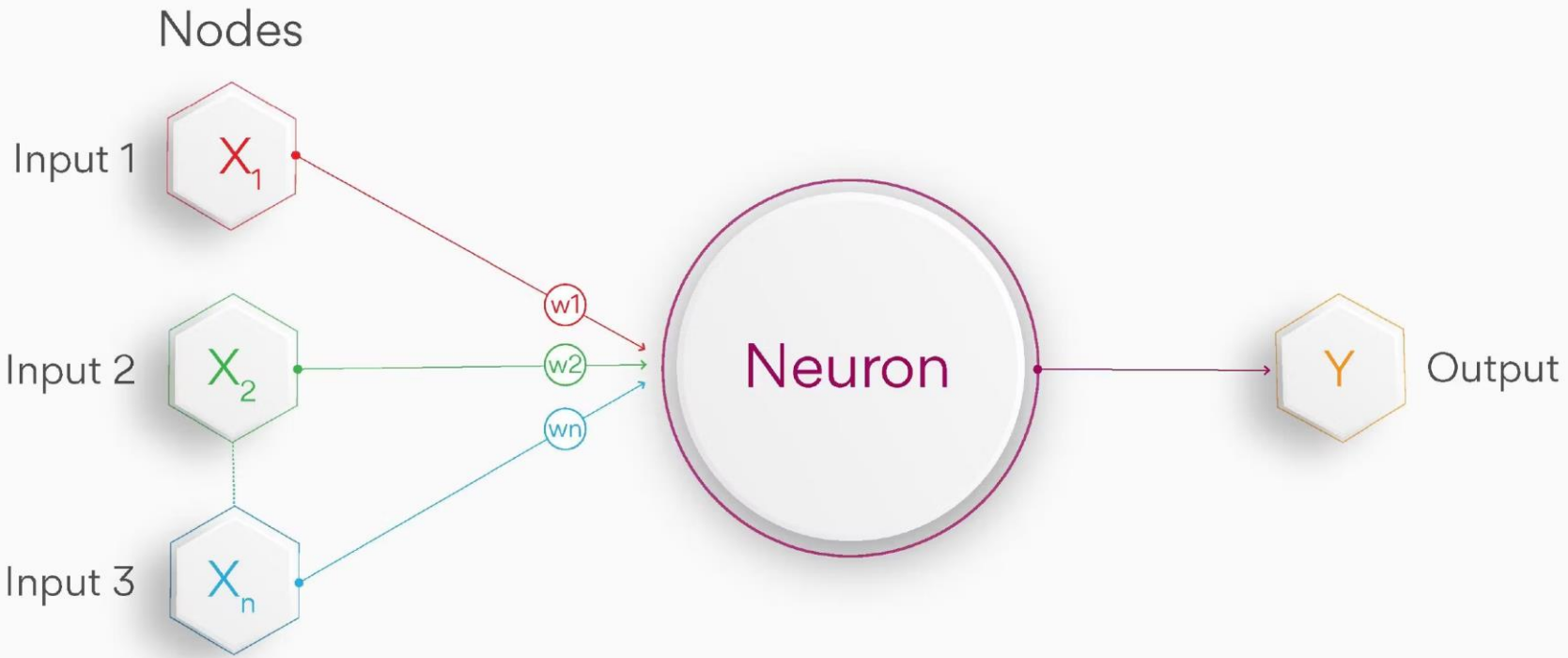
Geoffrey Hinton

neuron

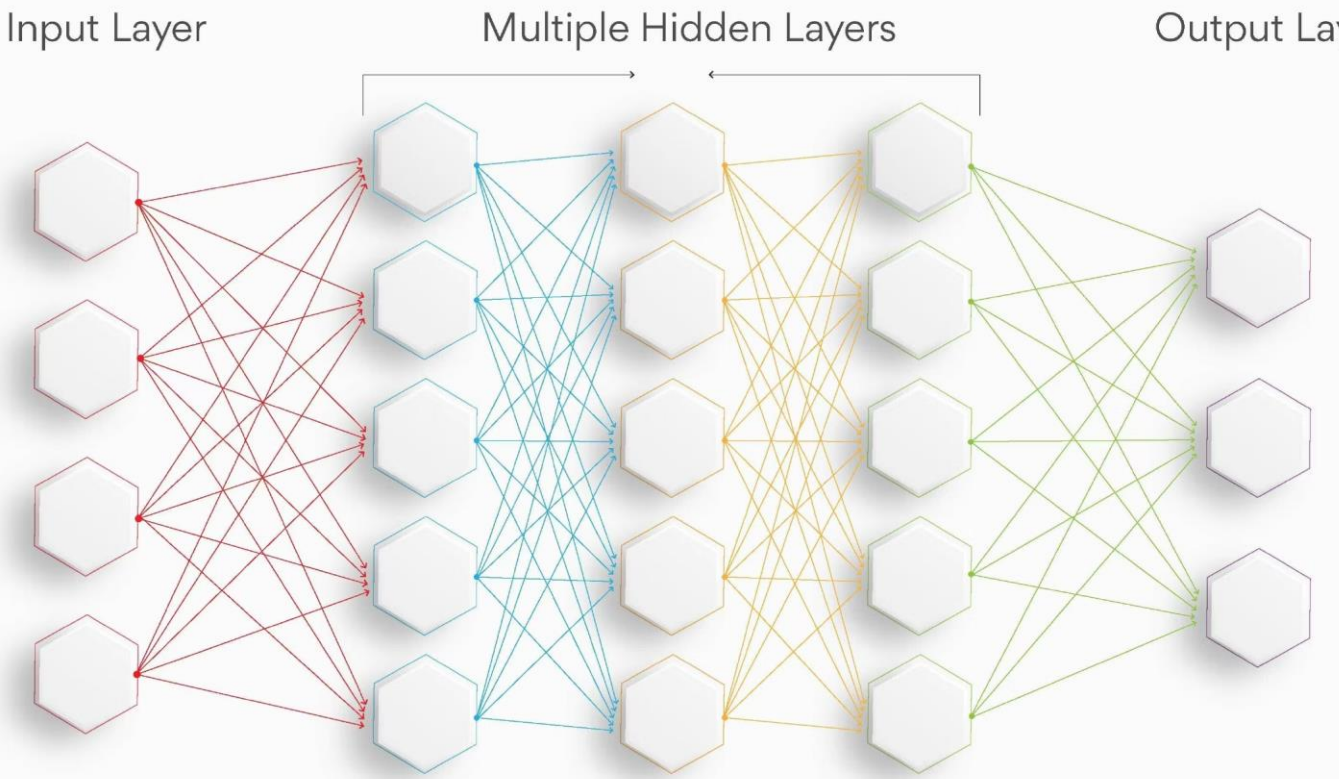

Dendrites
Axon
Neuron


Neuron
Axons
Neurotransmitter Vesicles
Neurotransmitter Molecules
Synapse
Receptors
Dendrites

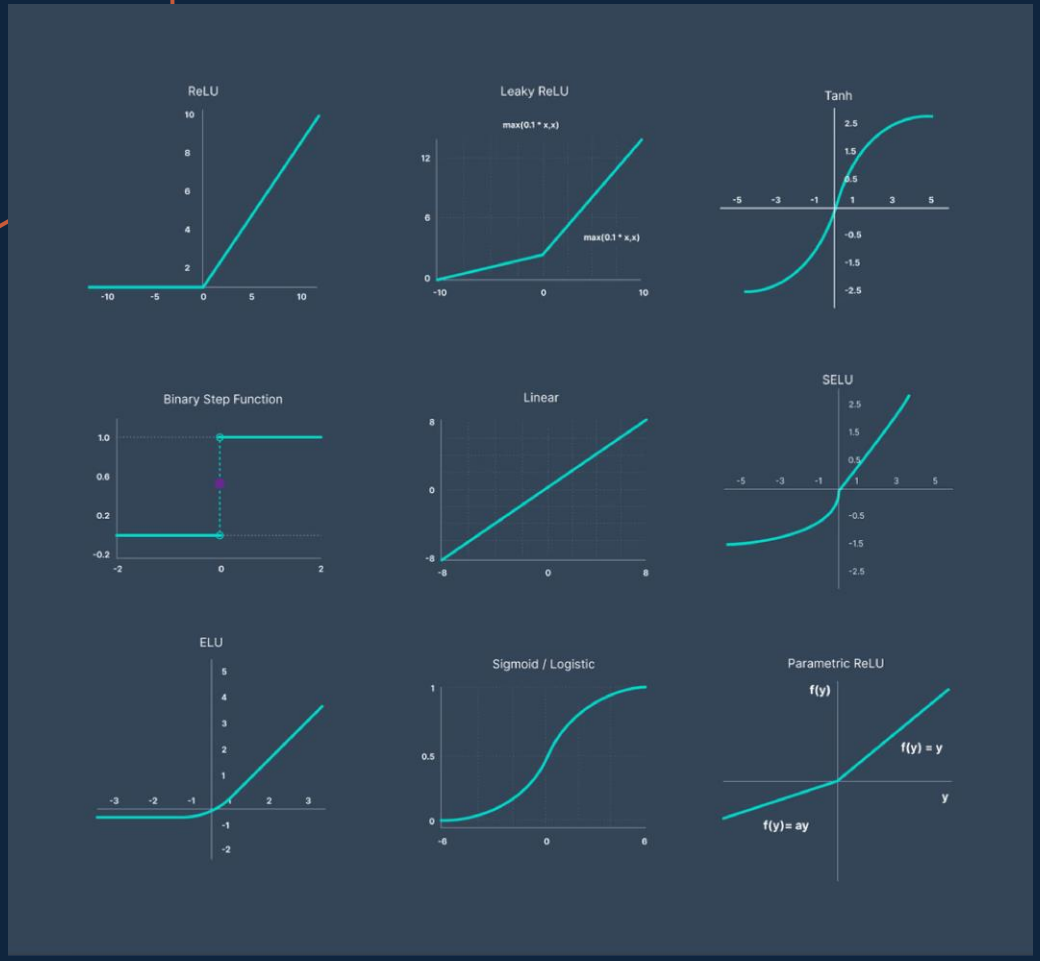
Artificial Neural Network



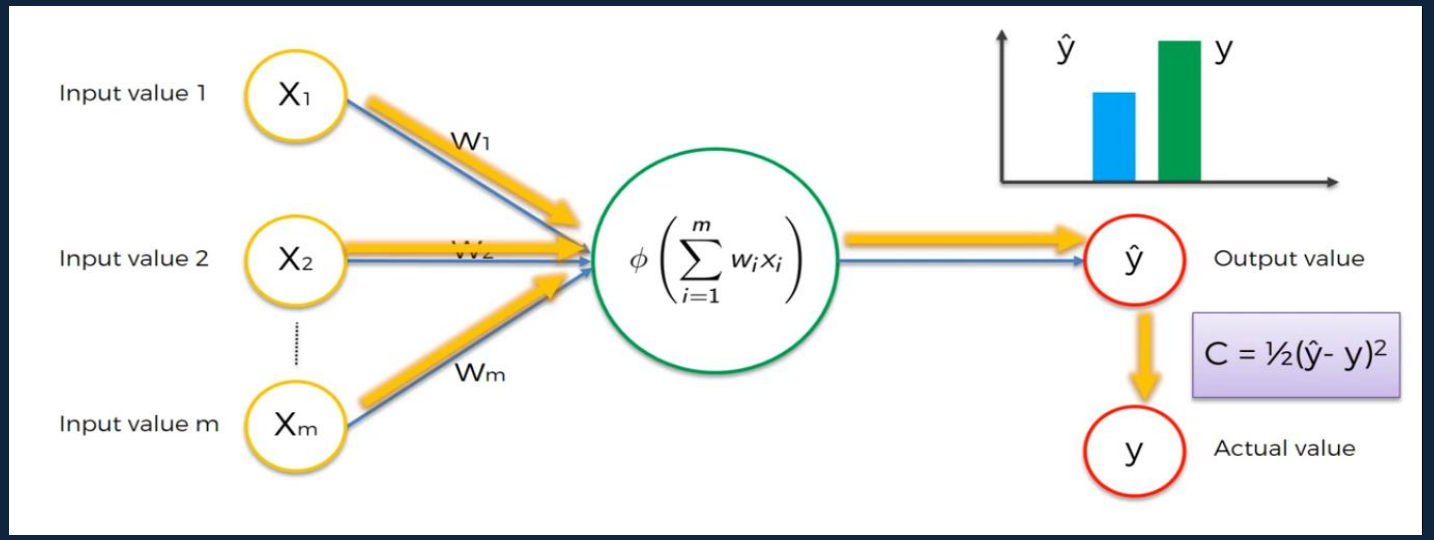
Deep Neural Network

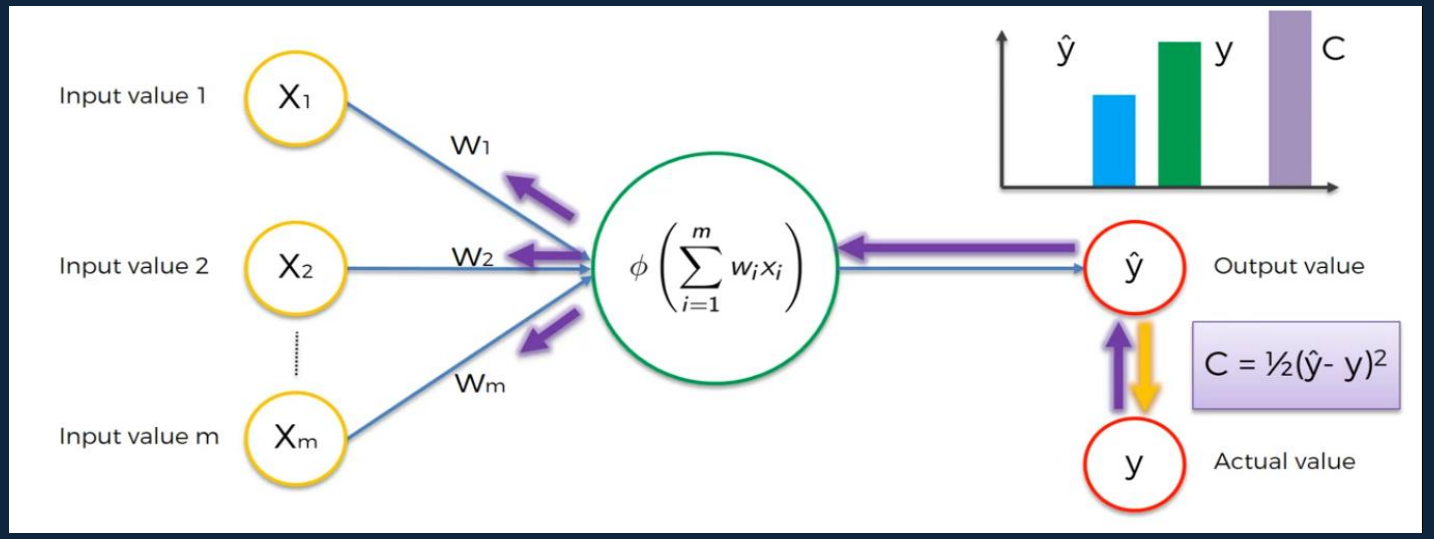


Activation Function

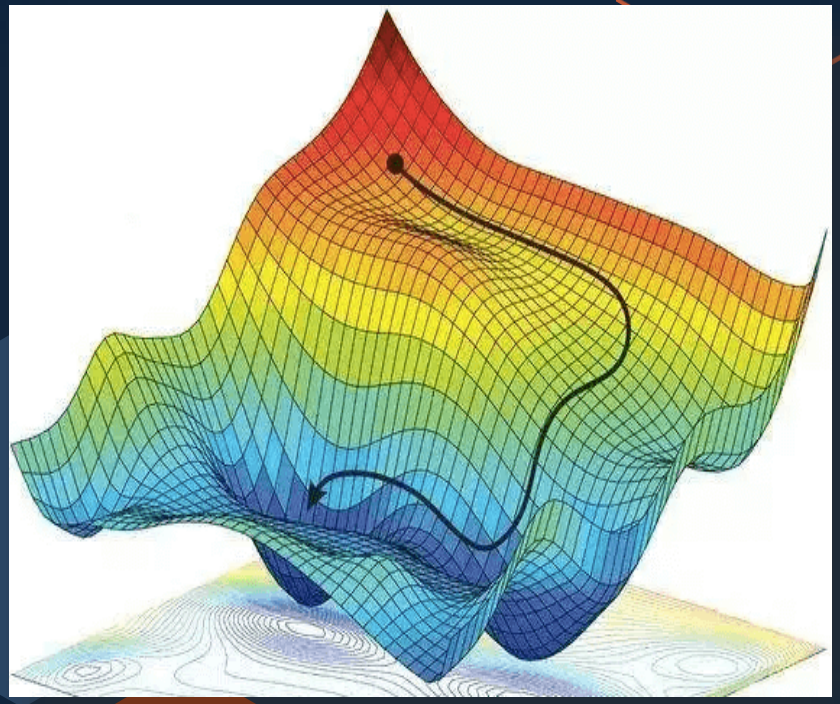


How do Neural Networks learn?

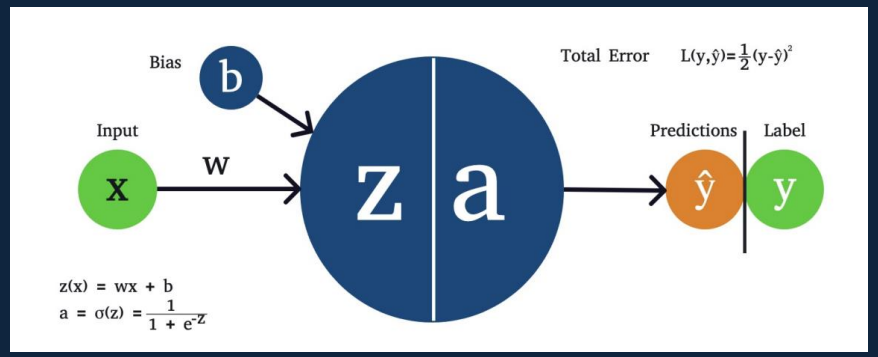


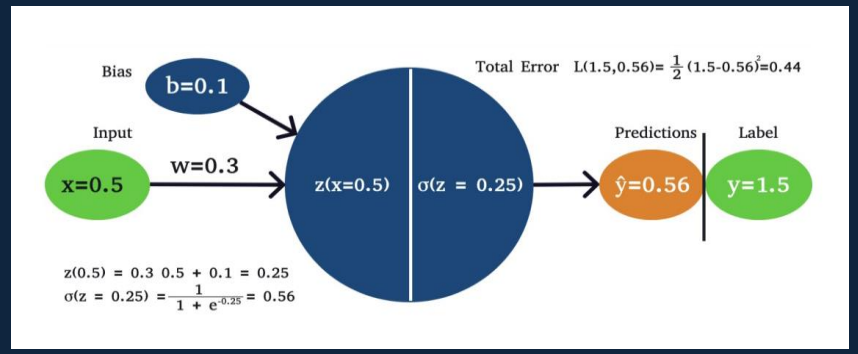


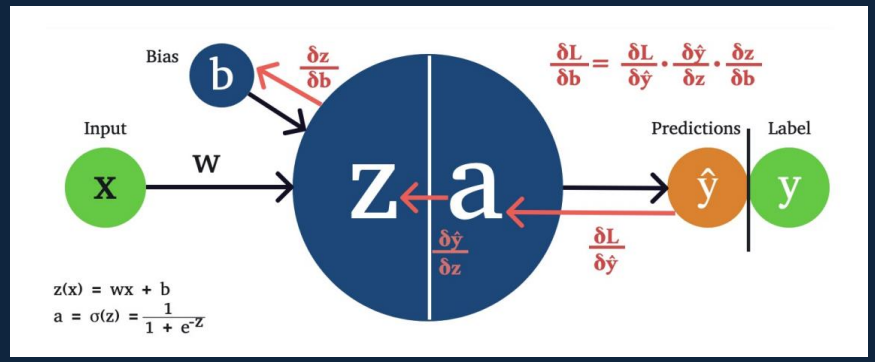
Gradient Descent

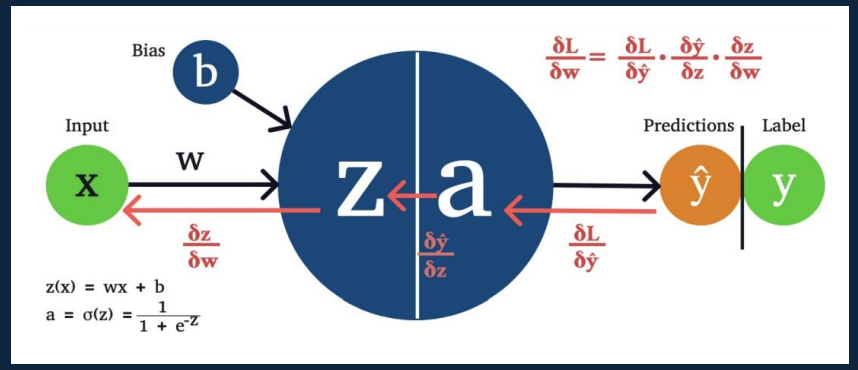


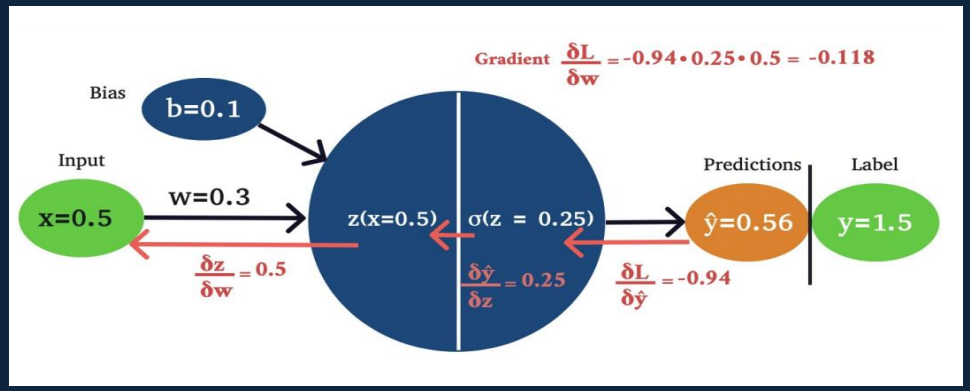
backpropagation





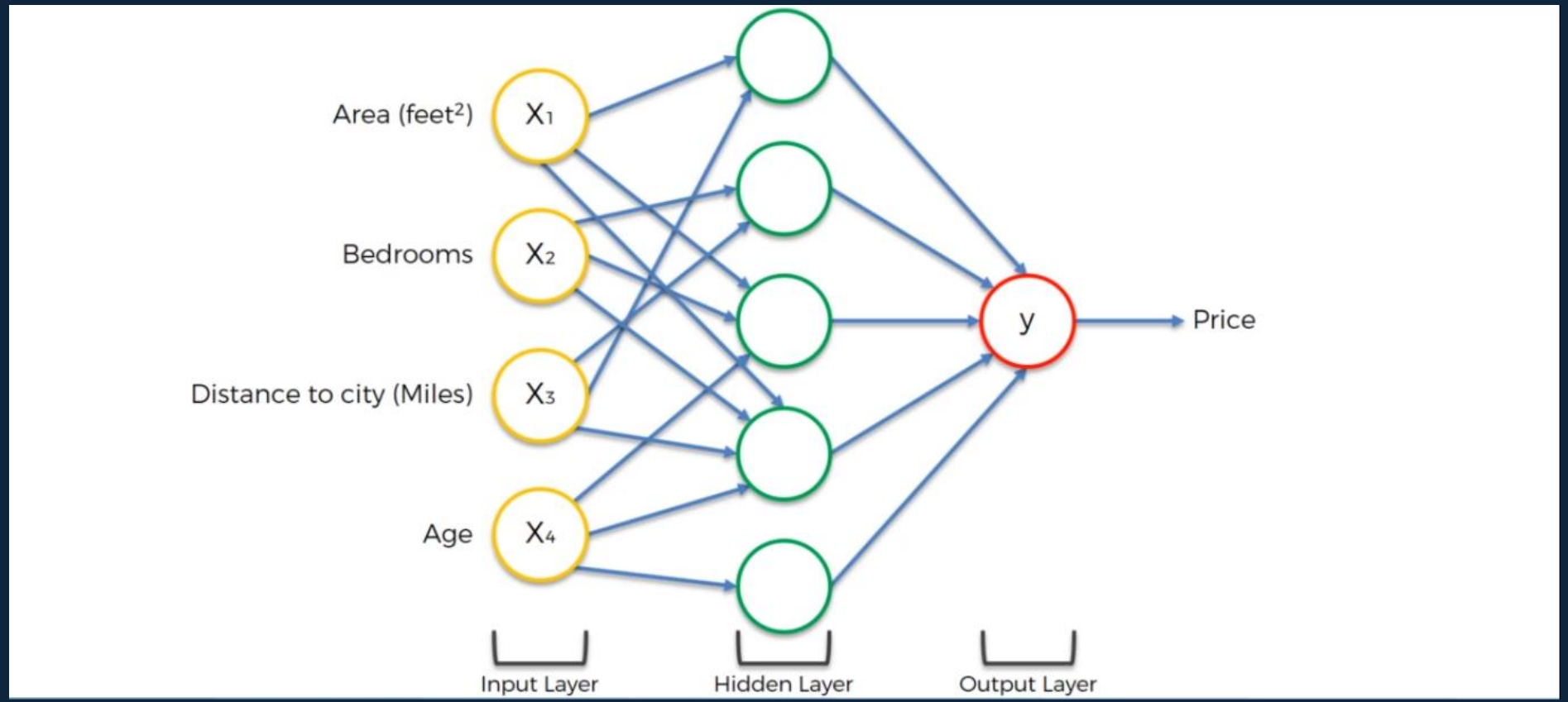






Explainability

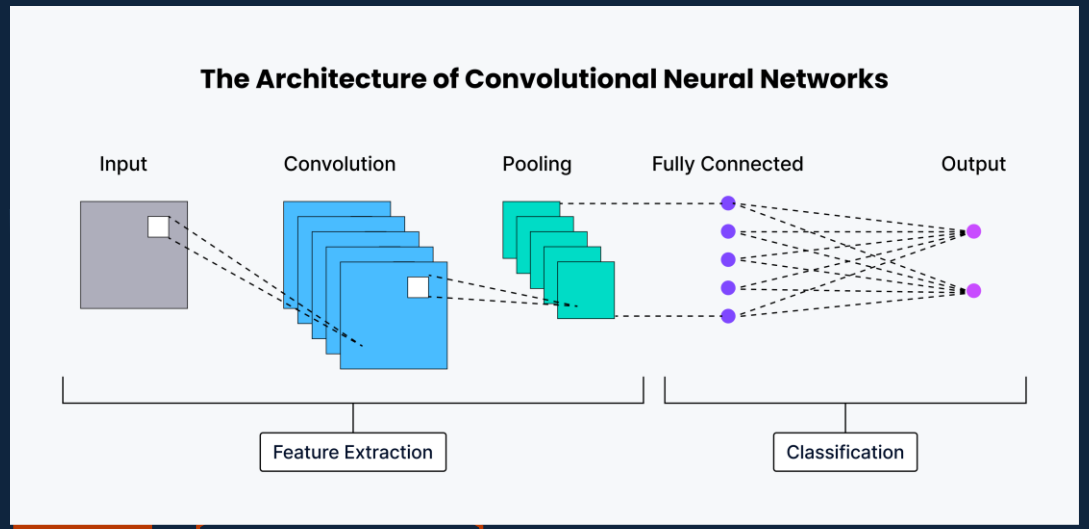
Black Box



Types of Neural Networks

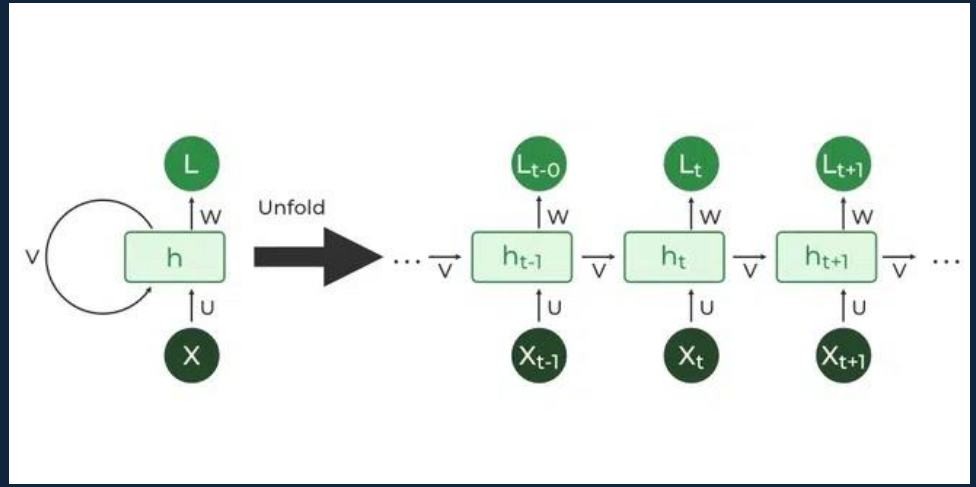
Convolutional Neural Networks (CNNs)

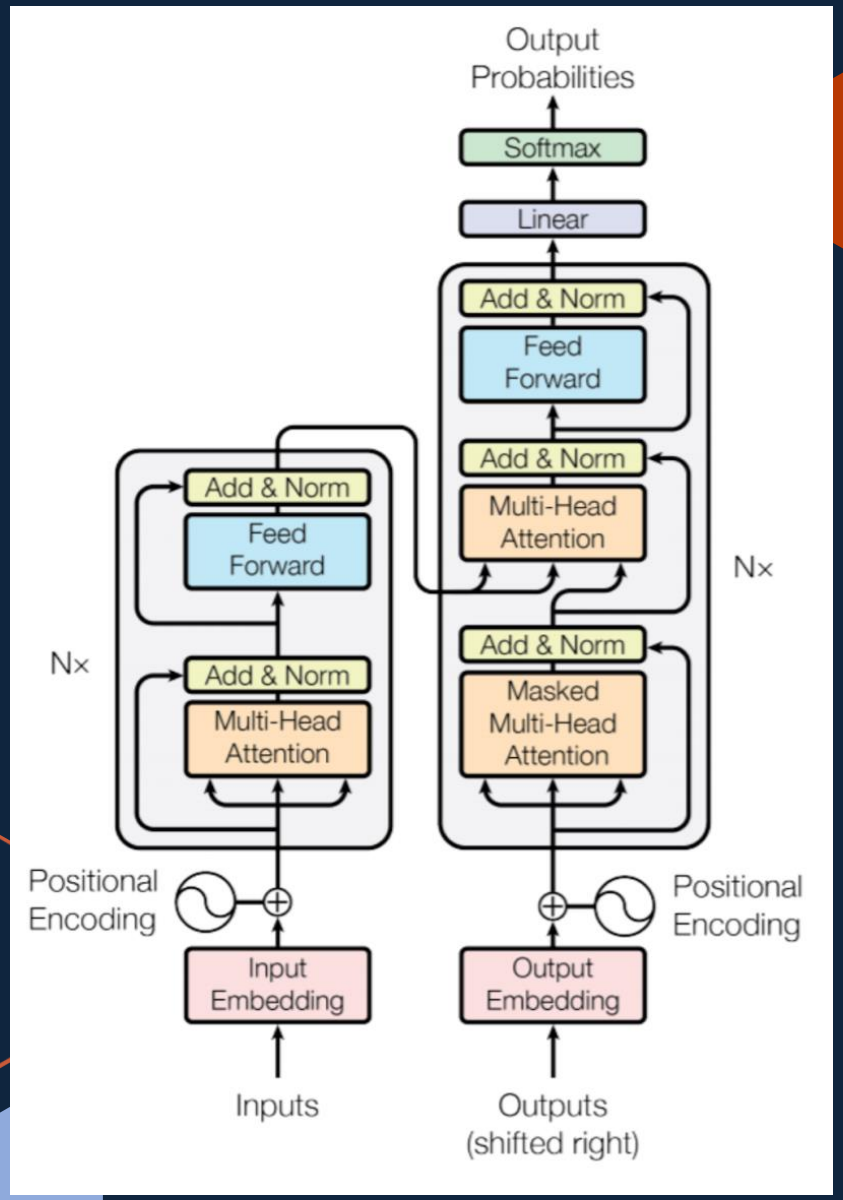
convolution



Recurrent Neural Networks (RNN)

memory





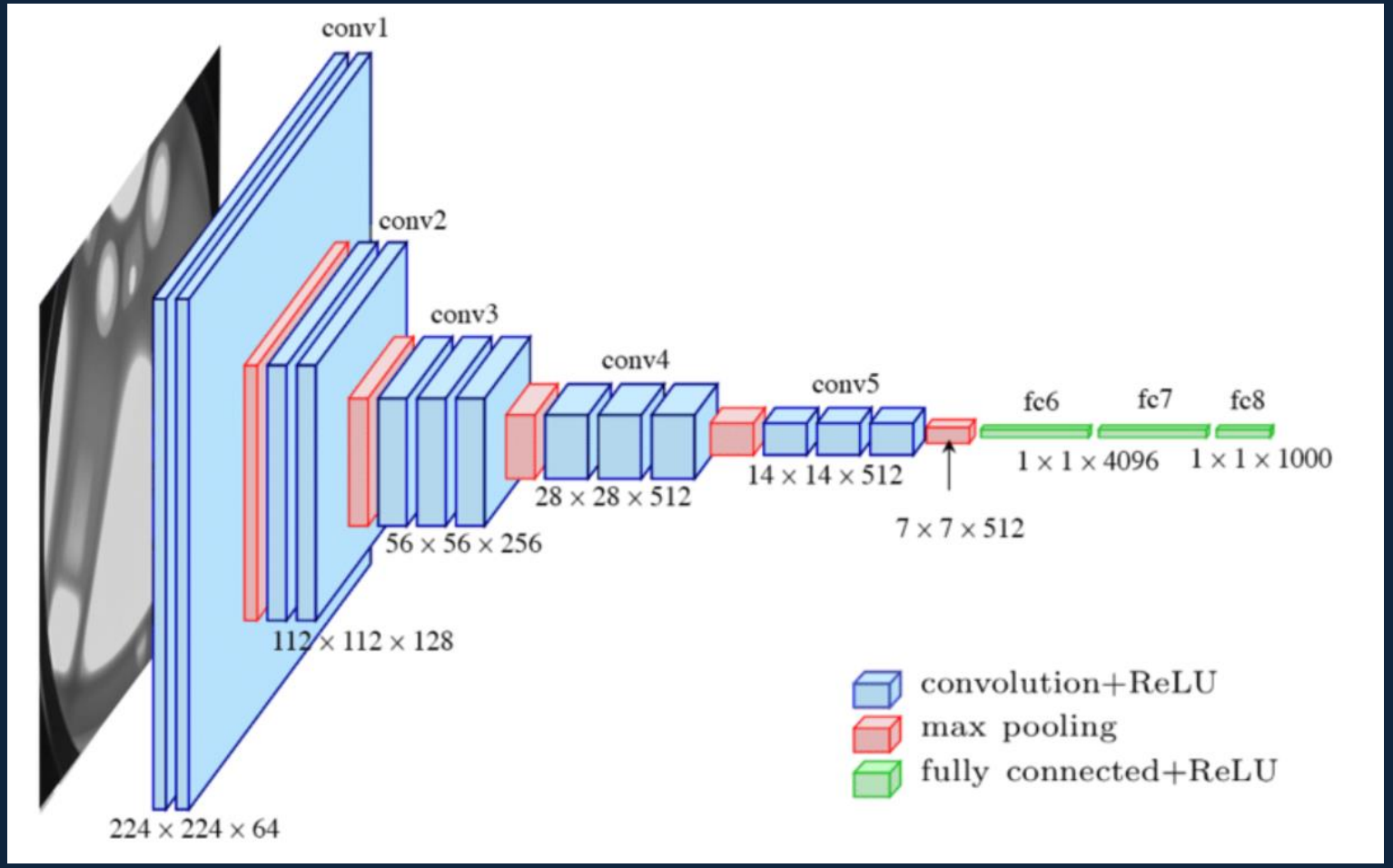
Transformers Neural Network

attention

Convolutional Neural Networks (CNNs): A Revolution in Computer Vision

Yann Lecun

Convolutional Neural Network Architecture



convolution

$$(1 \times 0) + (0 \times 1) + (1 \times 0) + (1 \times 1) + (0 \times 0) + (0 \times -1) + (0 \times 0) + (1 \times 1) + (1 \times 0) = 2$$

1x0	0x1	1x0	0	1
1x1	0x0	0x-1	1	1
0x0	1x1	1x0	0	0
1	0	0	1	0
0	0	1	1	0

*

0	1	0
1	0	-1
0	1	0

=

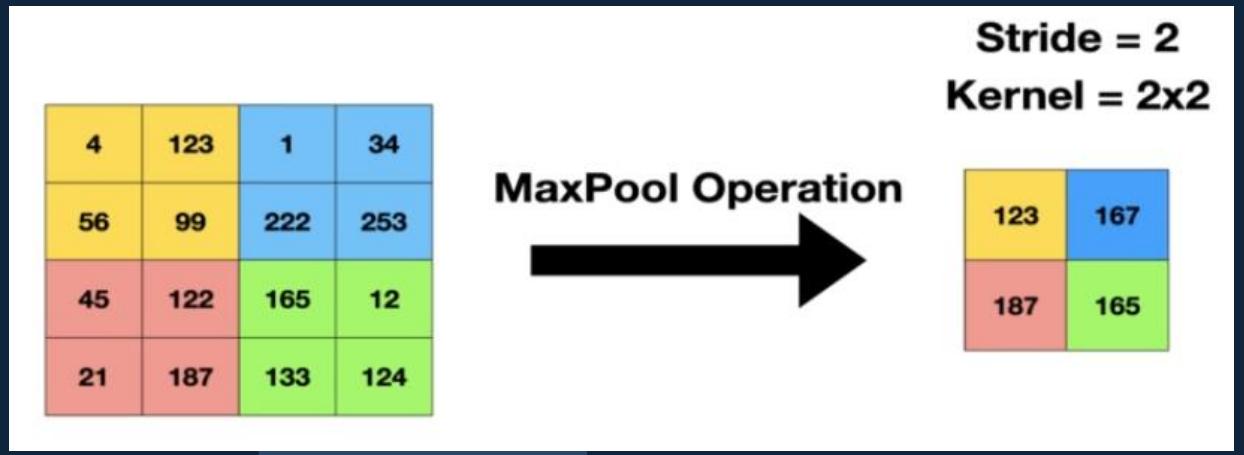
2		

Input Image | Filter or Kernel | Output or Feature Map

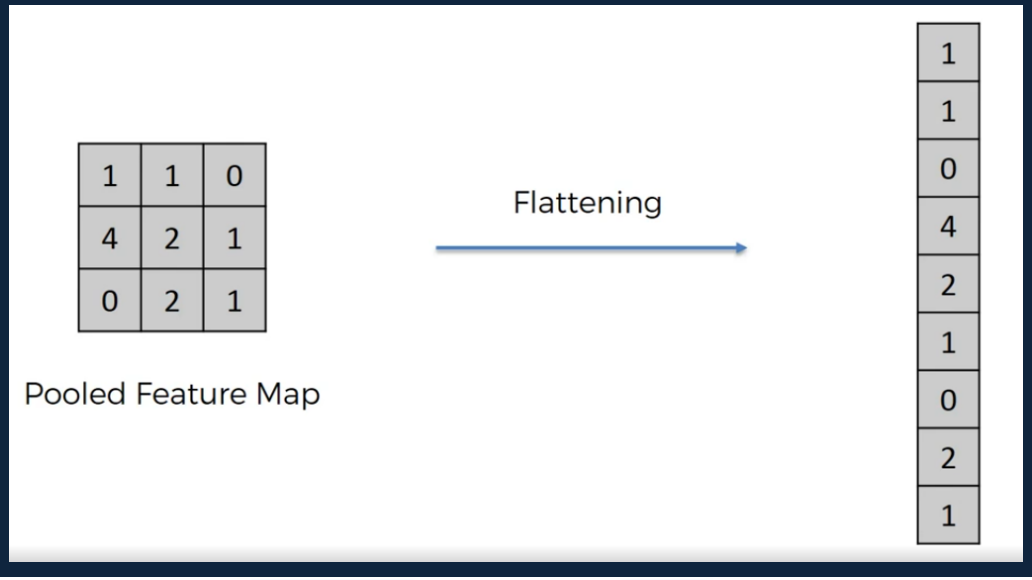
*

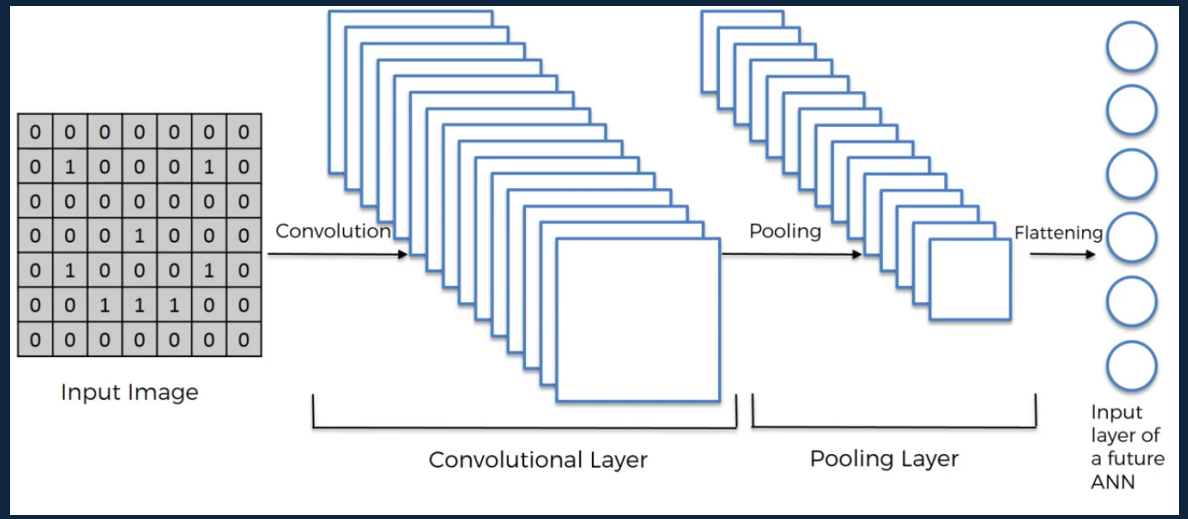
1	0	-1
2	0	-2
1	0	-1

Max Pooling

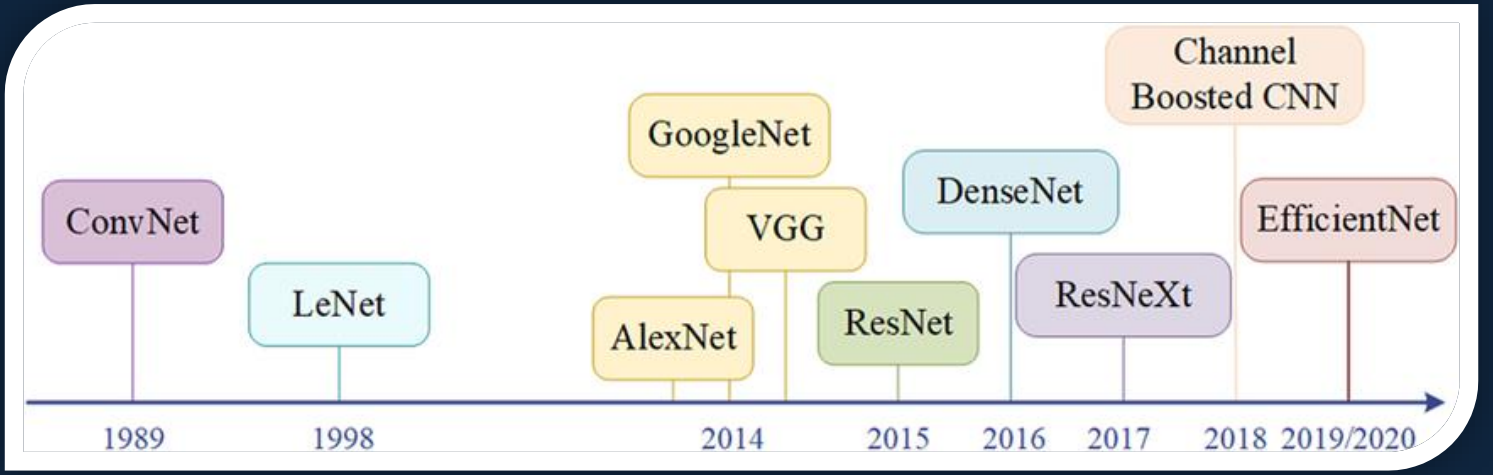

Stride = 2
Kernel = 2x2
4 123 1 34
56 99 222 253
45 122 165 12
21 187 133 124
MaxPool Operation
123 167
187 165

Flattening


1 1 0
4 2 1
0 2 1
Pooled Feature Map
Flattening
1
1
0
4
2
1
0
2
1

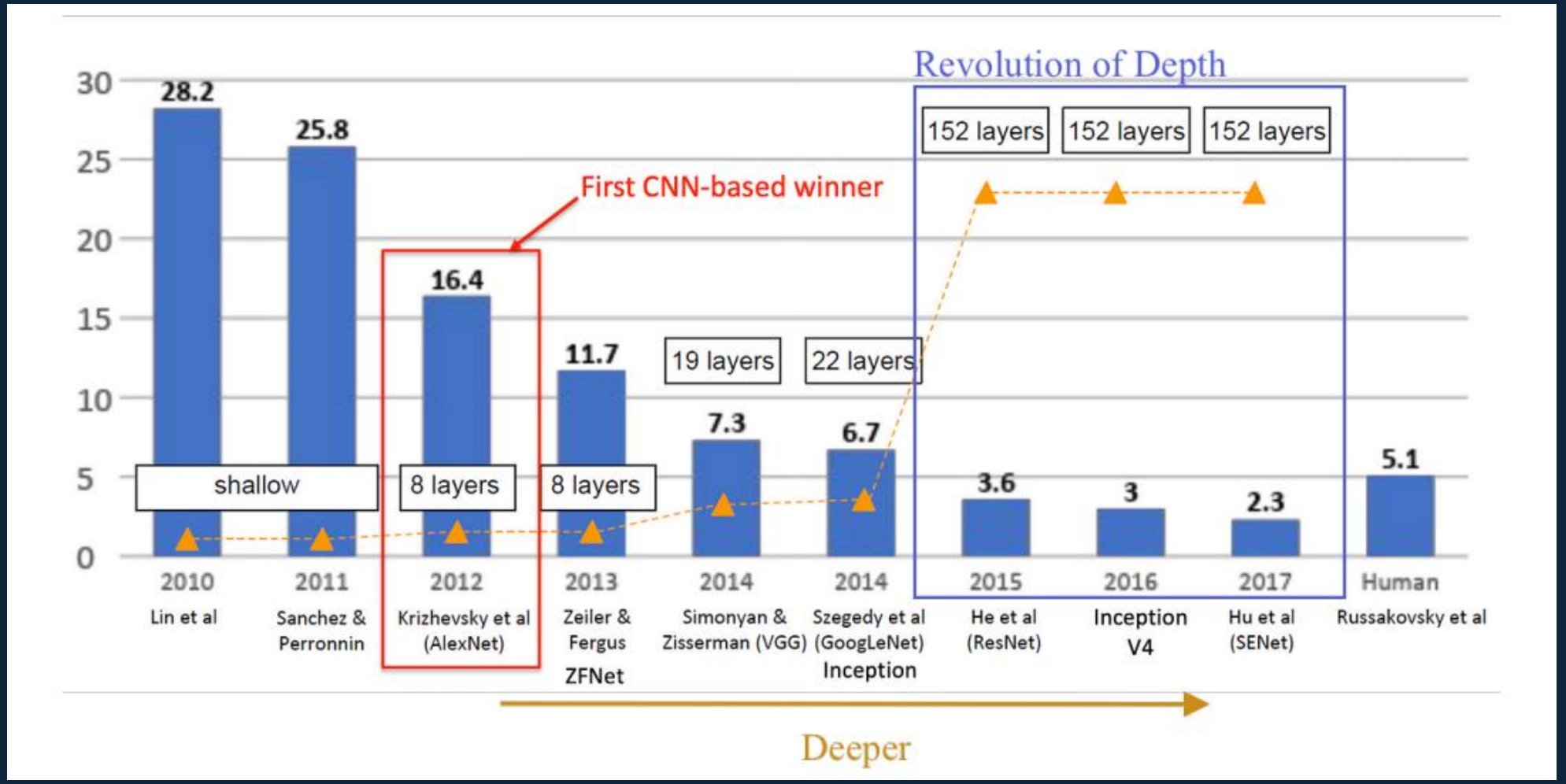

0 0 0 0 0 0 0
0 1 0 0 0 1 0
0 0 0 0 0 0 0
0 0 0 1 0 0 0
0 1 0 0 0 1 0
0 0 1 1 1 0 0
0 0 0 0 0 0 0
Input Image
Convolution
Pooling
Flattening
Convolutional Layer
Pooling Layer
Input layer of a future ANN

CNN History

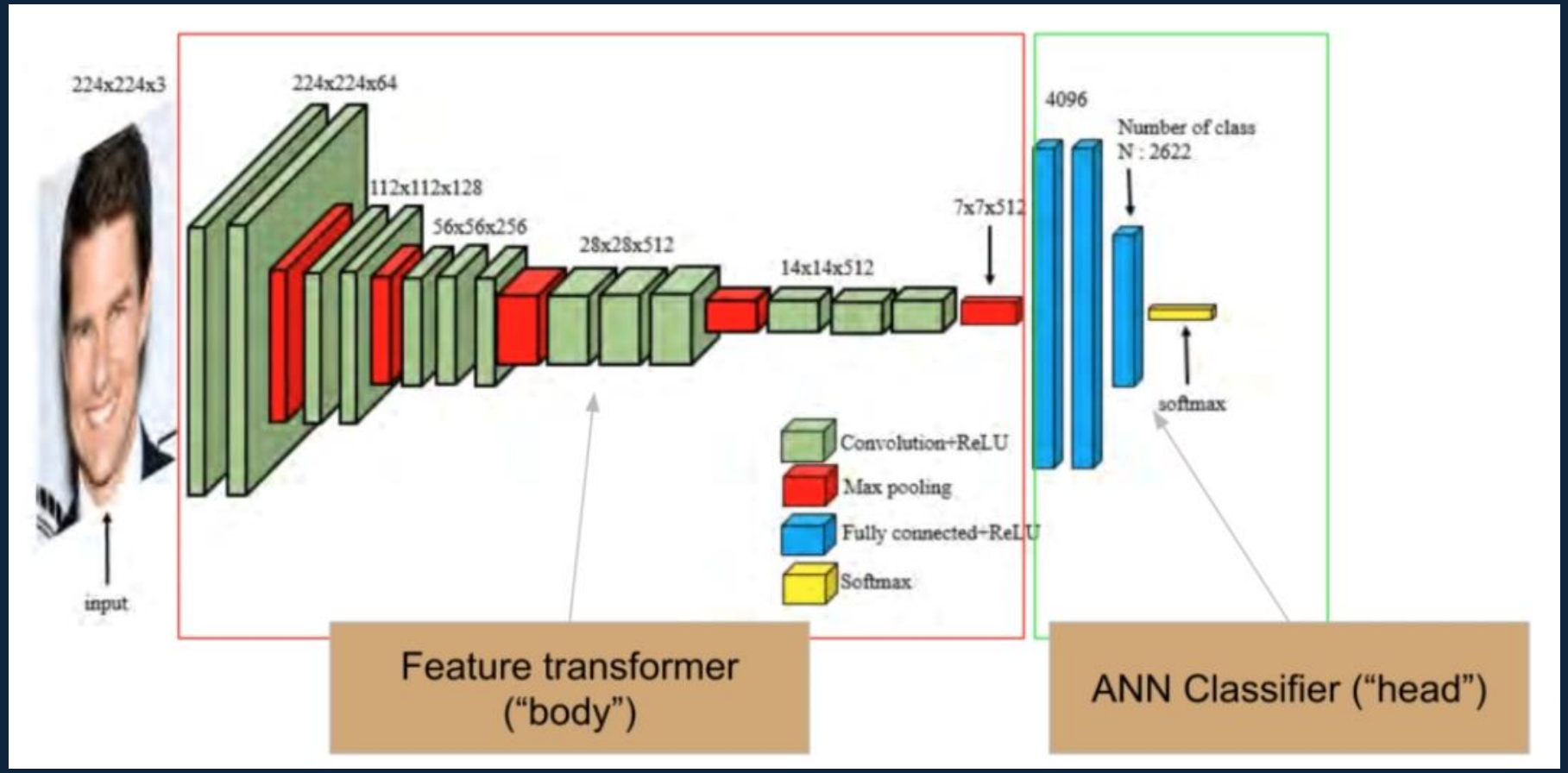

ConvNet
LeNet
AlexNet
GoogleNet
VGG
ResNet
DenseNet
ResNeXt
Channel Boosted CNN
EfficientNet
1989
1998
2014
2015
2016
2017
2018
2019/2020


I WAS WINNING IMAGENET
BUT THEN SOMEONE MADE A DEEPER NET

ImageNet


Revolution of Depth
First CNN-based winner
30
25
20
15
10
5
0
28.2
25.8
16.4
11.7
7.3
6.7
3.6
3
2.3
5.1
shallow
8 layers
8 layers
19 layers
22 layers
152 layers
152 layers
152 layers
2010
Lin et al
2011
Sanchez & Perronnin
2012
Krizhevsky et al (AlexNet)
2013
Zeiler & Fergus
ZFNet
2014
Simonyan & Zisserman (VGG)
2014
Szegedy et al (GoogLeNet)
Inception
2015
He et al (ResNet)
2016
Inception V4
2017
Hu et al (SENet)
Human
Russakovsky et al
Deeper

Transfer Learning and Fine Tuning



Task:

- Image Classification
- Image Manipulation
- Image Generation
- Object Detection
- Image Segmentation
- Object Tracking
- Pose Estimation
- Face Recognition

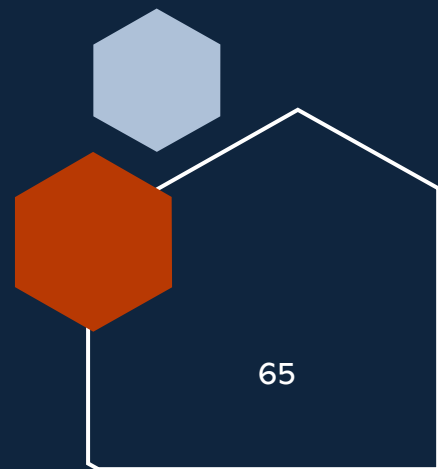
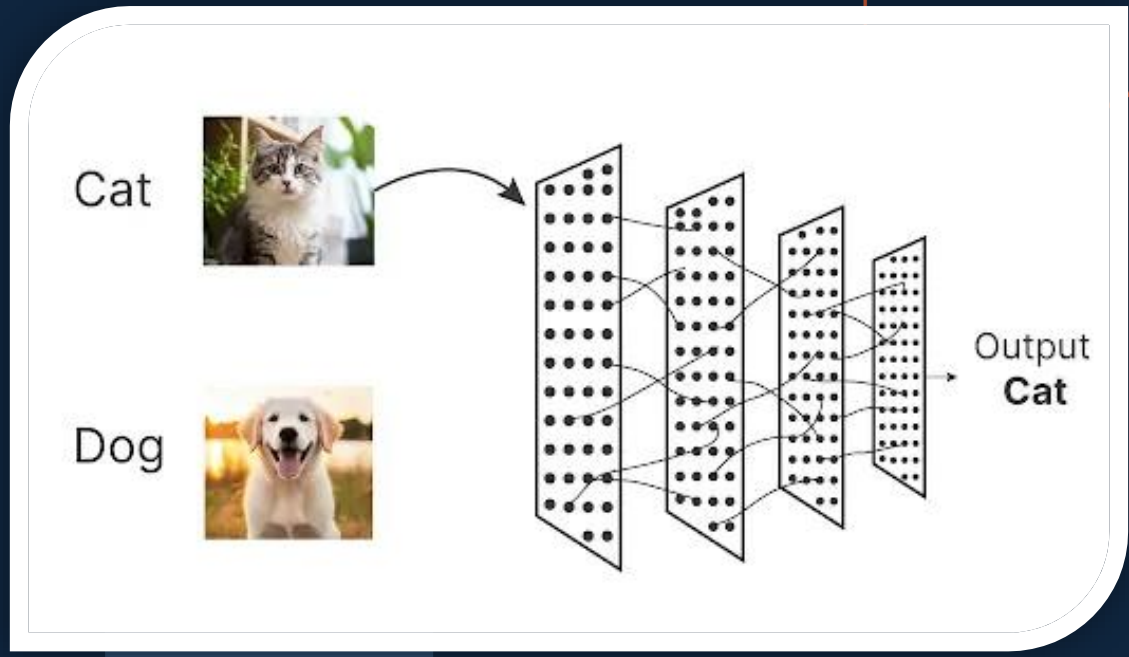


Image Classification



1 Project

About Dataset

LungVision

Contains PNG images of Normal, Pneumonia and Tuberculosis Lungs

The images are in PNG format. The resolution of the images are 512x512.

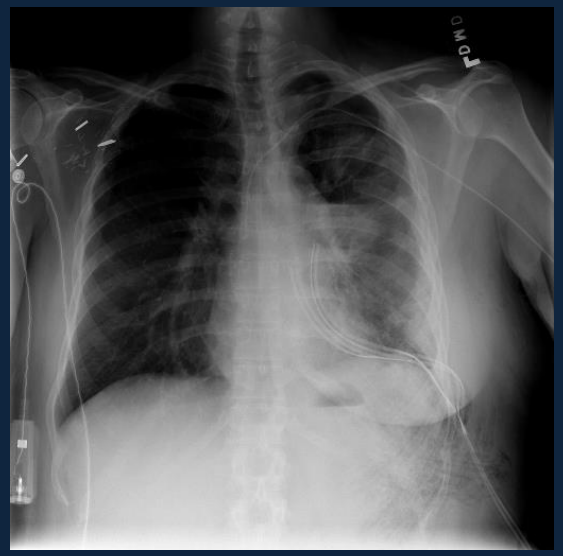
The dataset has been divided into train,test and val set.

The lung images dataset is a comprehensive collection of images used for the training and evaluation of deep-learning models for the diagnosis of lung infections. The dataset contains a total of 17,275 images, consisting of 10,406 normal images, 5,775 pneumonia images, and 1,094 tuberculosis infected images. The images were sourced from multiple locations, including RSNA, Montgomery County chest X-ray set, Shenzhen chest X-ray, Qatar University, Doha, Qatar, and the University of Dhaka, Bangladesh, and their collaborators. The original images were in the DCM format, and they were converted to the Png format to ensure compatibility with deep learning models. This dataset is an essential resource for researchers, clinicians, and data scientists working on lung infection diagnosis, and it provides a valuable tool for the development of advanced AI models for lung disease diagnosis.

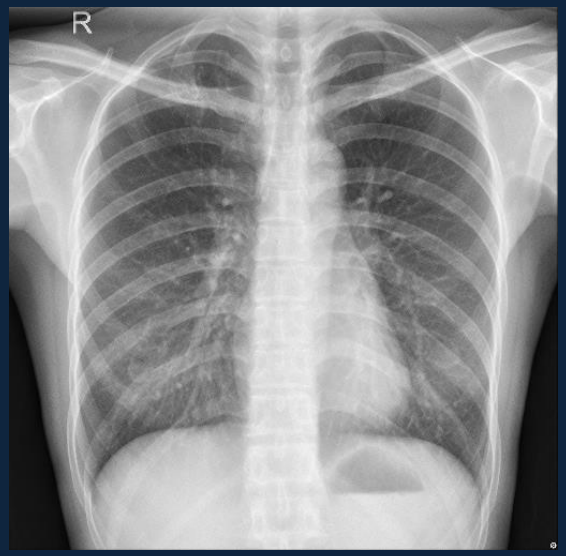
Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Image%20Classification/CnnWithKeras.ipynb

Dataset

Normal



Pneumonia

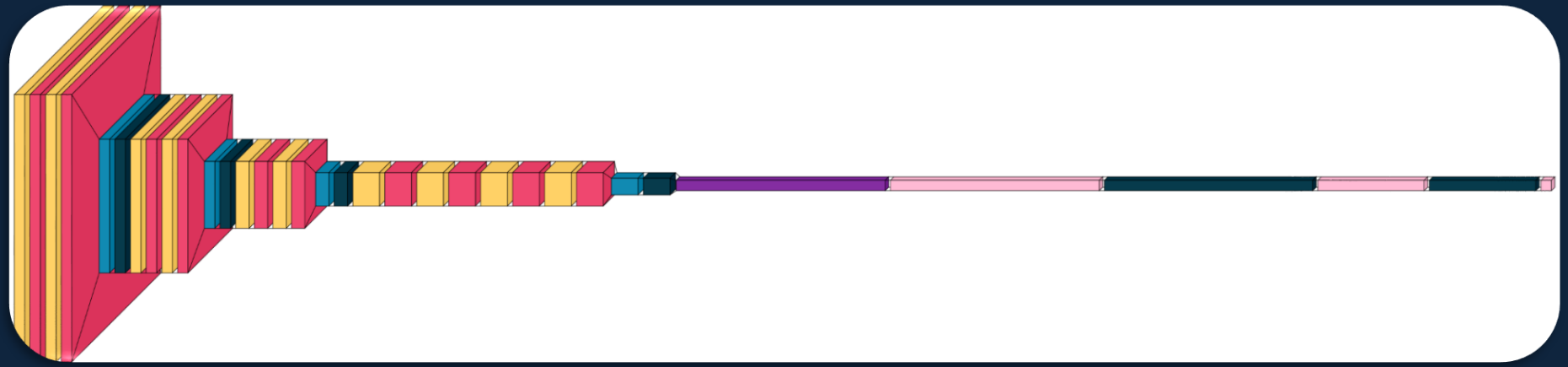


Tuberculosis

Total params: 152,530,497

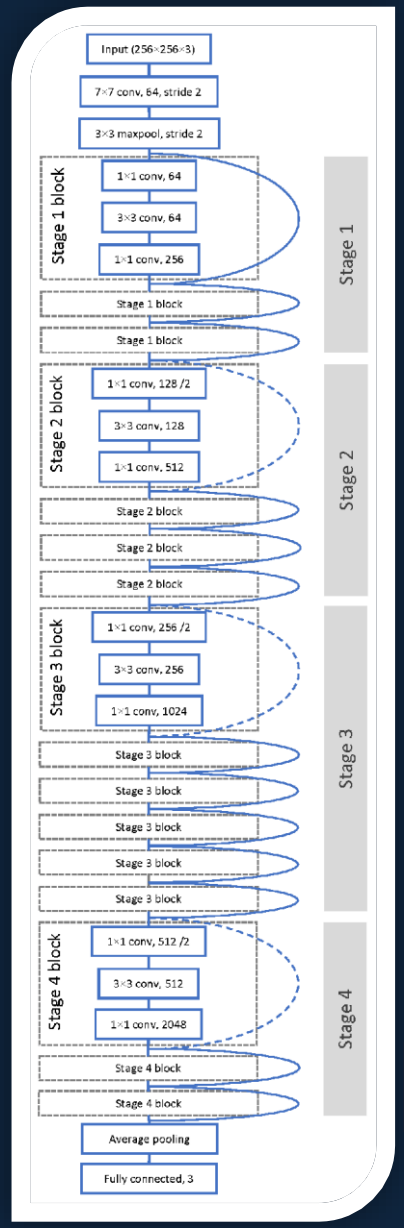
Trainable params: 152,524,609

Validation Accuracy: 0.7



2 Project

Using the Lungvision dataset, I decided to build a ResNet-inspired architecture from scratch, without using a pre-trained ResNet model.



Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Image%20Classification/ResidualNetworkWithPytorchLightning.ipynb

Total params: 51,043,971

Total layers: 104

Validation Accuracy: 0.812

Test Accuracy: 0.80

3 Project

About Dataset

Fundus Glaucoma

The Fundus Glaucoma Detection Data dataset available on the Kaggle platform is one of the key resources for detecting glaucoma, also known as glaucoma disease. This eye condition can lead to damage to the optic nerve, and if not diagnosed and treated in time, it may result in vision loss. In such datasets, eye images (more precisely, fundus images) are utilized to train artificial intelligence models for detecting glaucoma from these images.

Dataset Details:

Fundus Images: Fundus images, which specifically depict the back of the eye or retina, are the primary tool for glaucoma detection. These images are typically captured using specialized cameras like fundus cameras.

Labeling: In glaucoma detection datasets, each image usually comes with a label indicating whether the patient has glaucoma. These labels are binary (i.e., presence or absence of glaucoma):

0: No glaucoma

1: Glaucoma present

Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Image%20Classification/CnnWithPytorchLightning.ipynb

Dataset

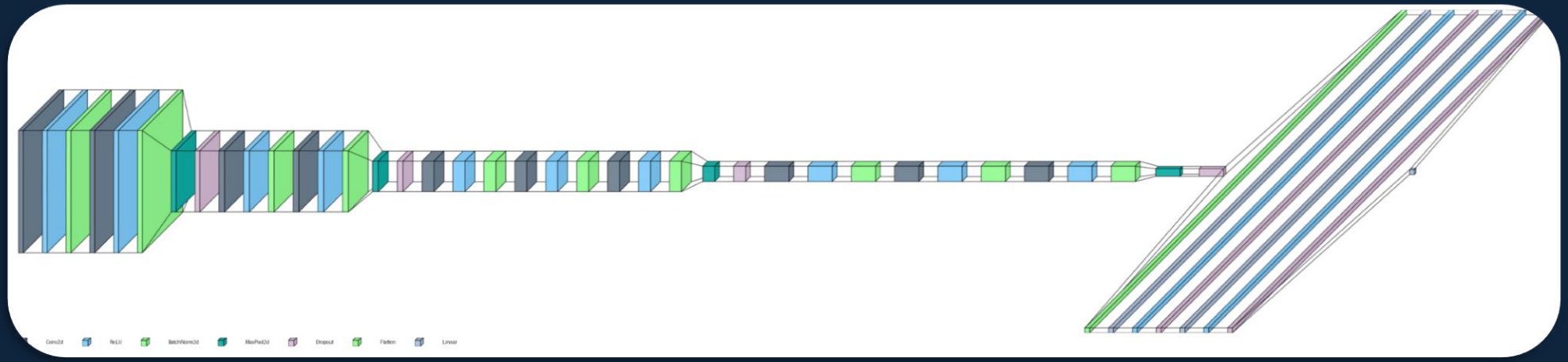
Glaucoma present

No glaucoma

Total params: 561,305,154

Train Accuracy: 0.6758

Validation Accuracy: 0.6730



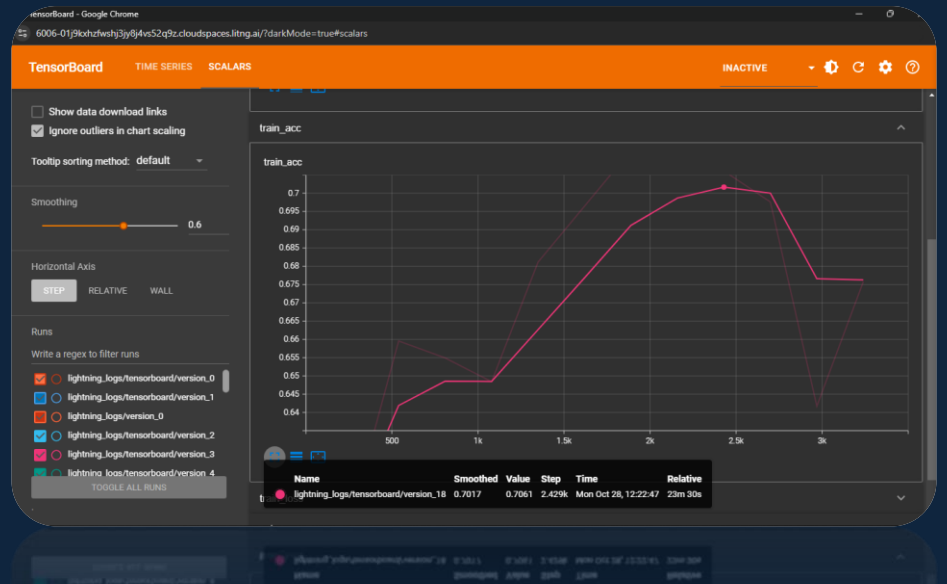
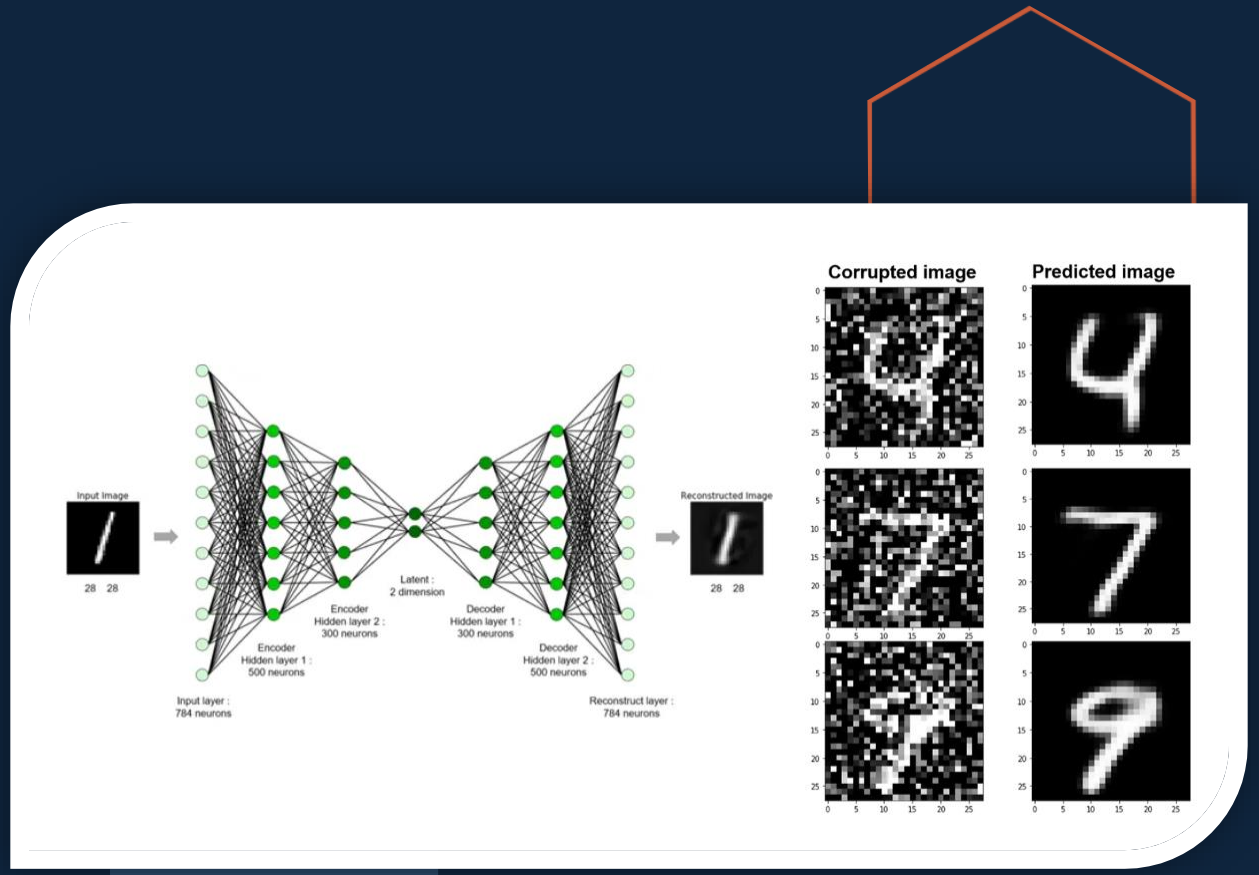
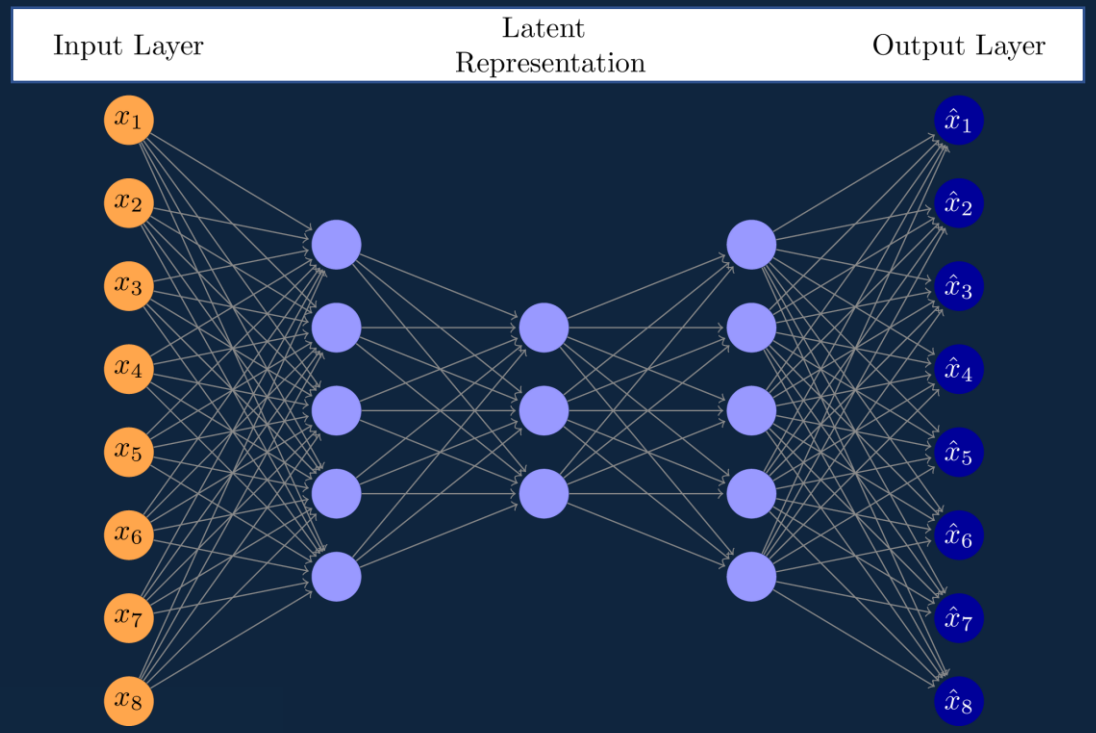


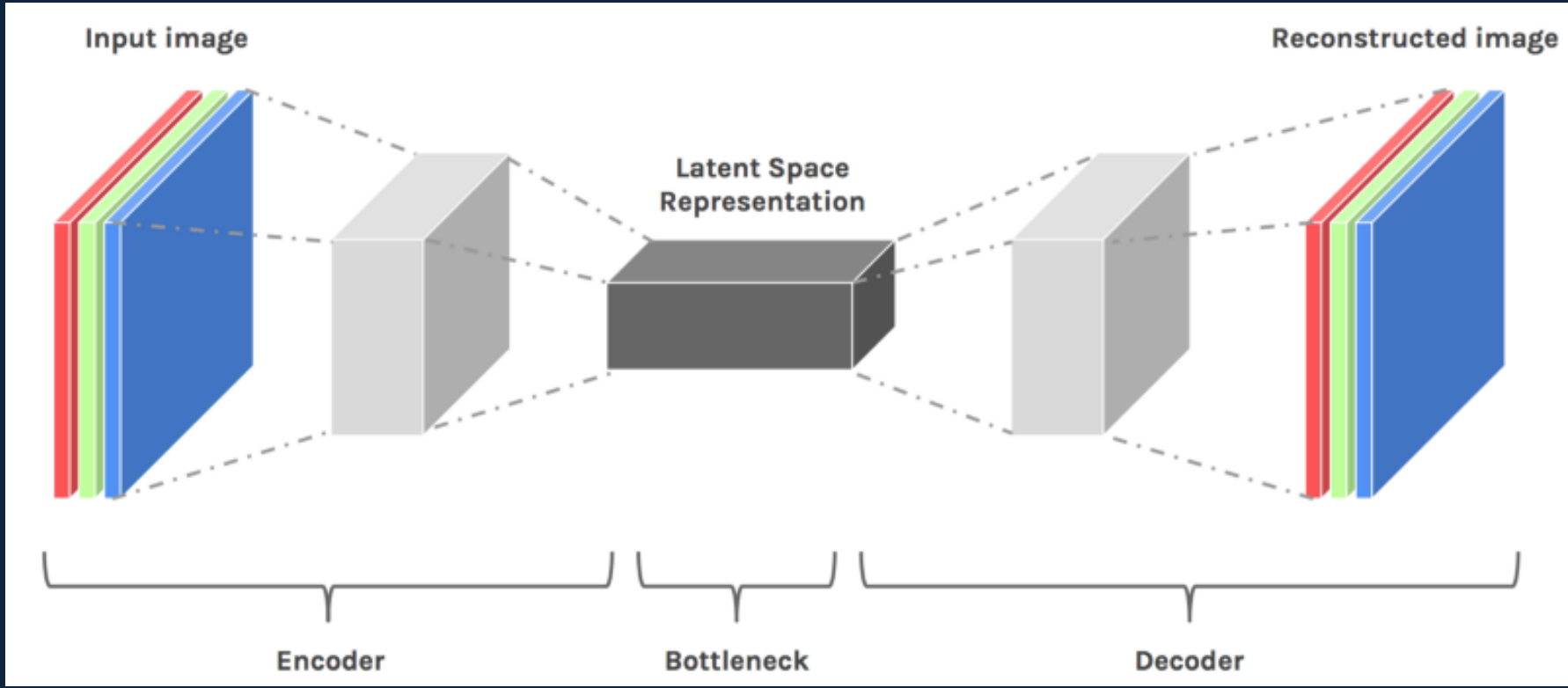
Image Manipulation



Autoencoder Architecture



Convolutional Autoencoder



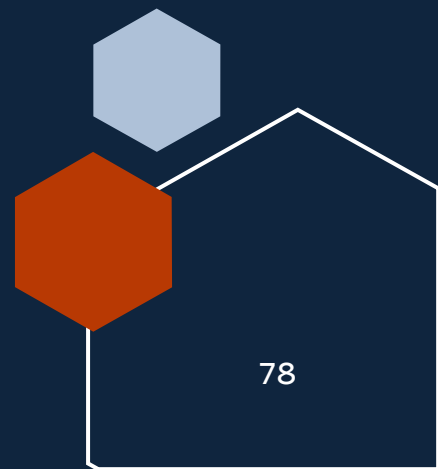
Project

About Dataset

Landscape Image dataset

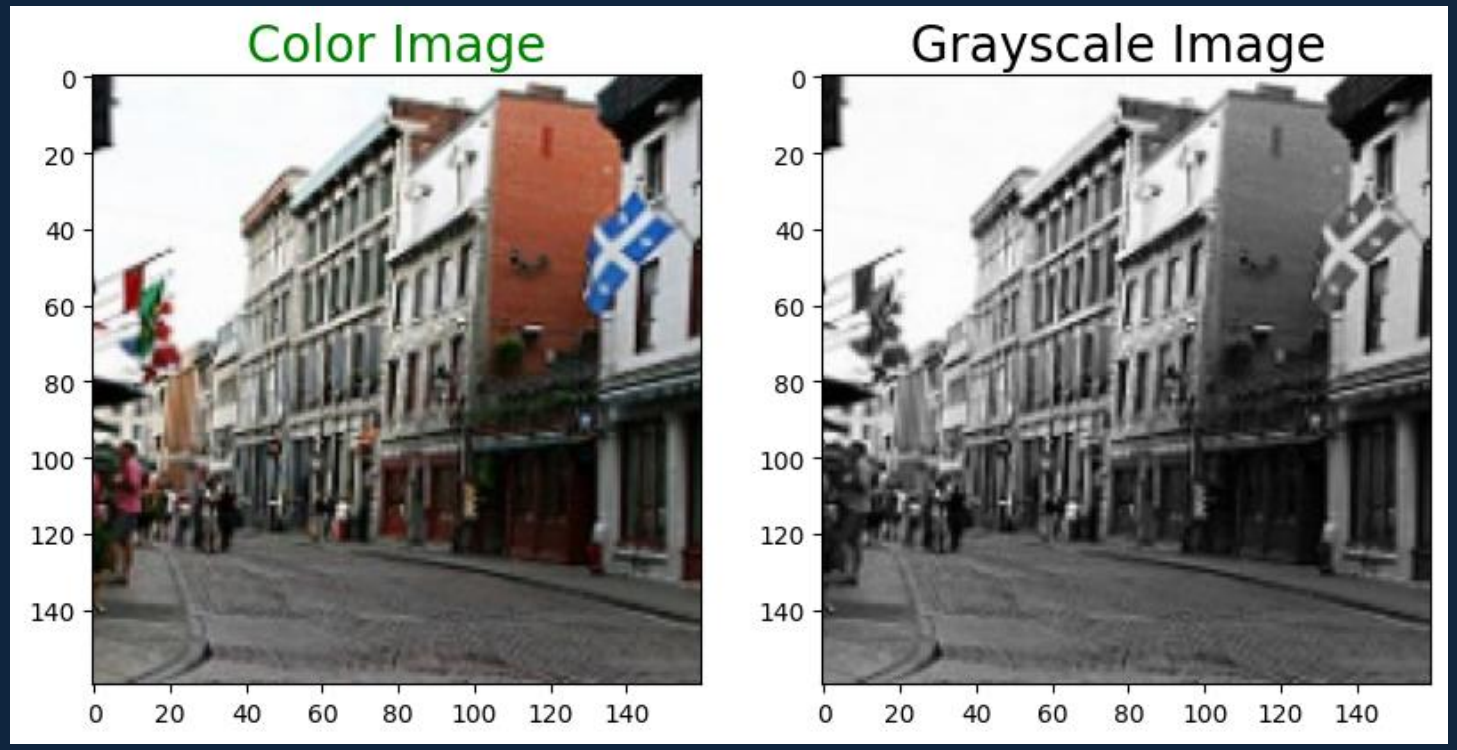
This dataset consist of street, buildings, mountains, glaciers ,trees etc and their corresponding grayscale image in two different folder.

The main objective of creating this dataset is to create autoencoder network that can colorized grayscale landscape images

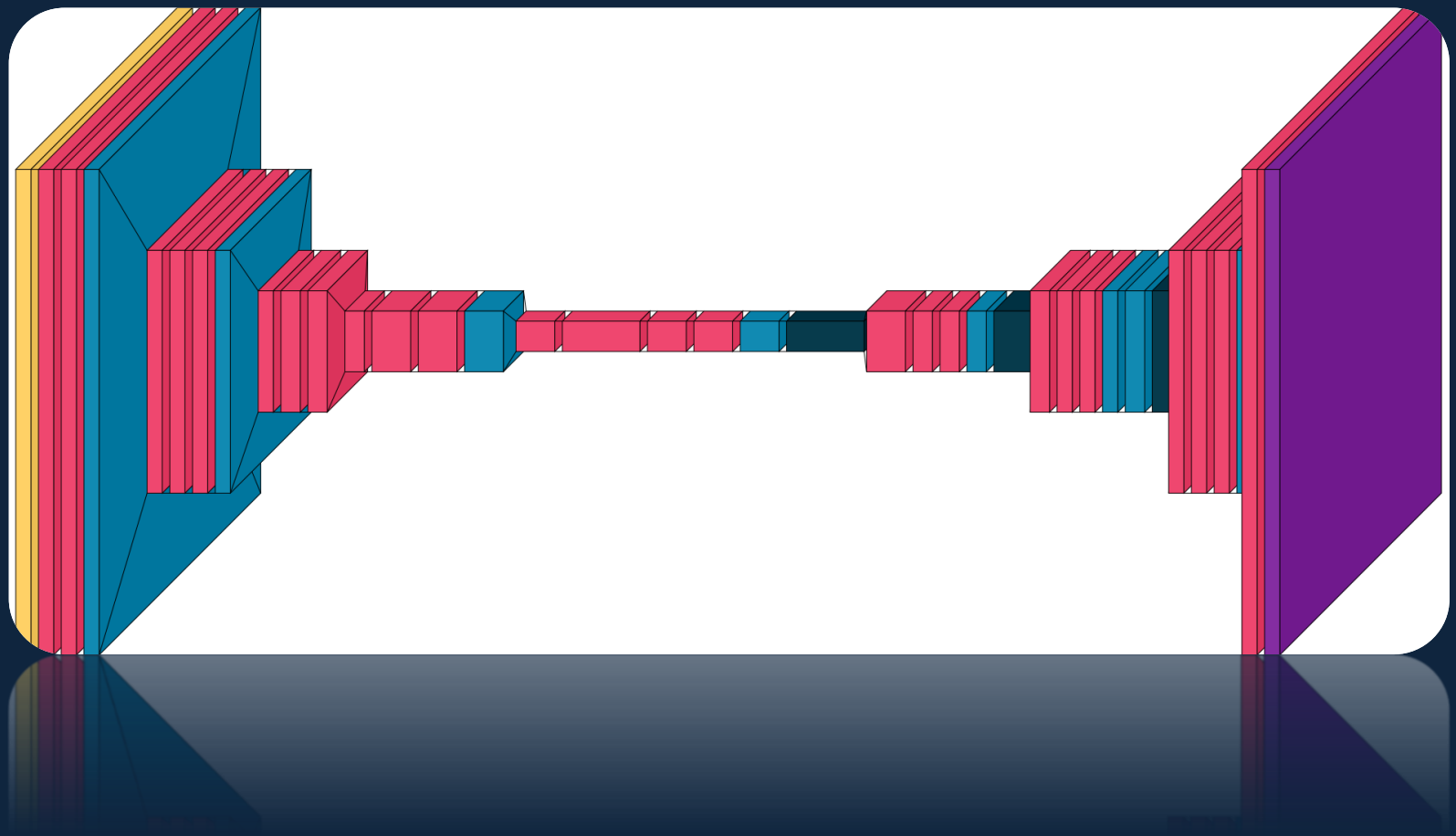


Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Image%20Manipulation/AutoEncoderWithKeras.ipynb

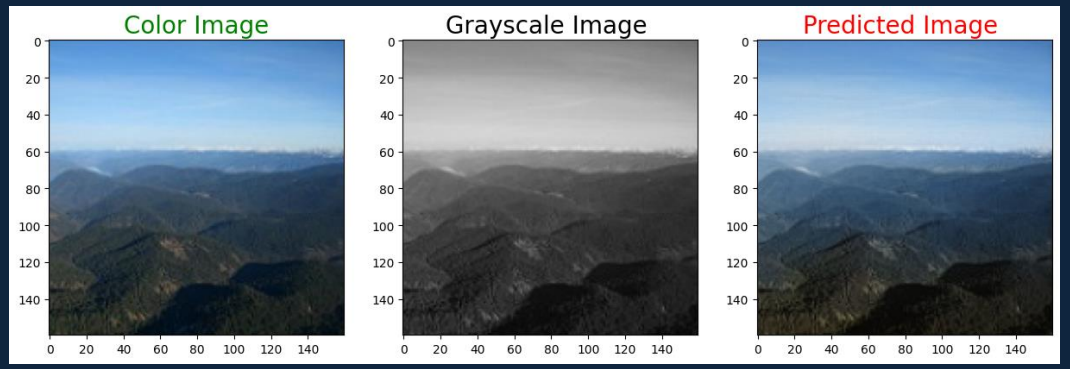
Dataset

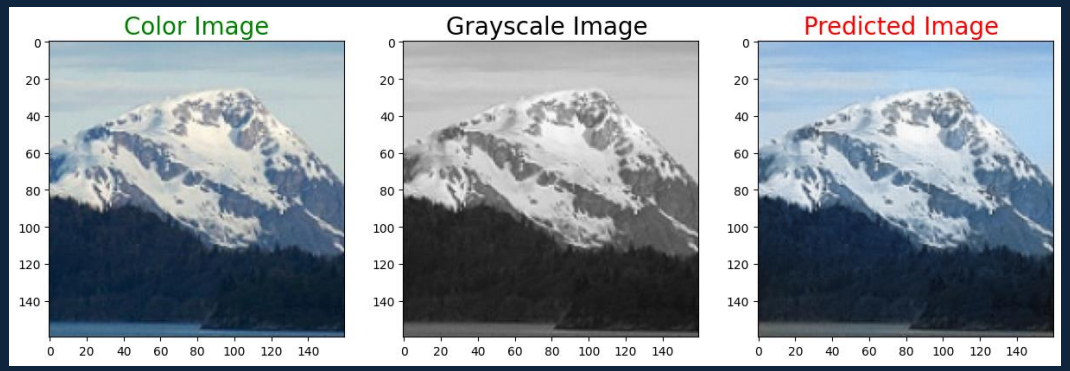


Total params: 28,384,835



Result


Color Image
Grayscale Image
Predicted Image


Color Image
Grayscale Image
Predicted Image

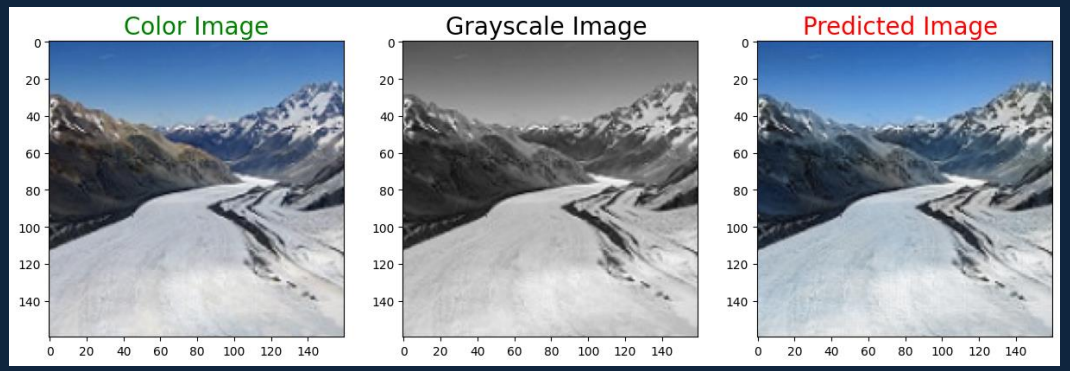
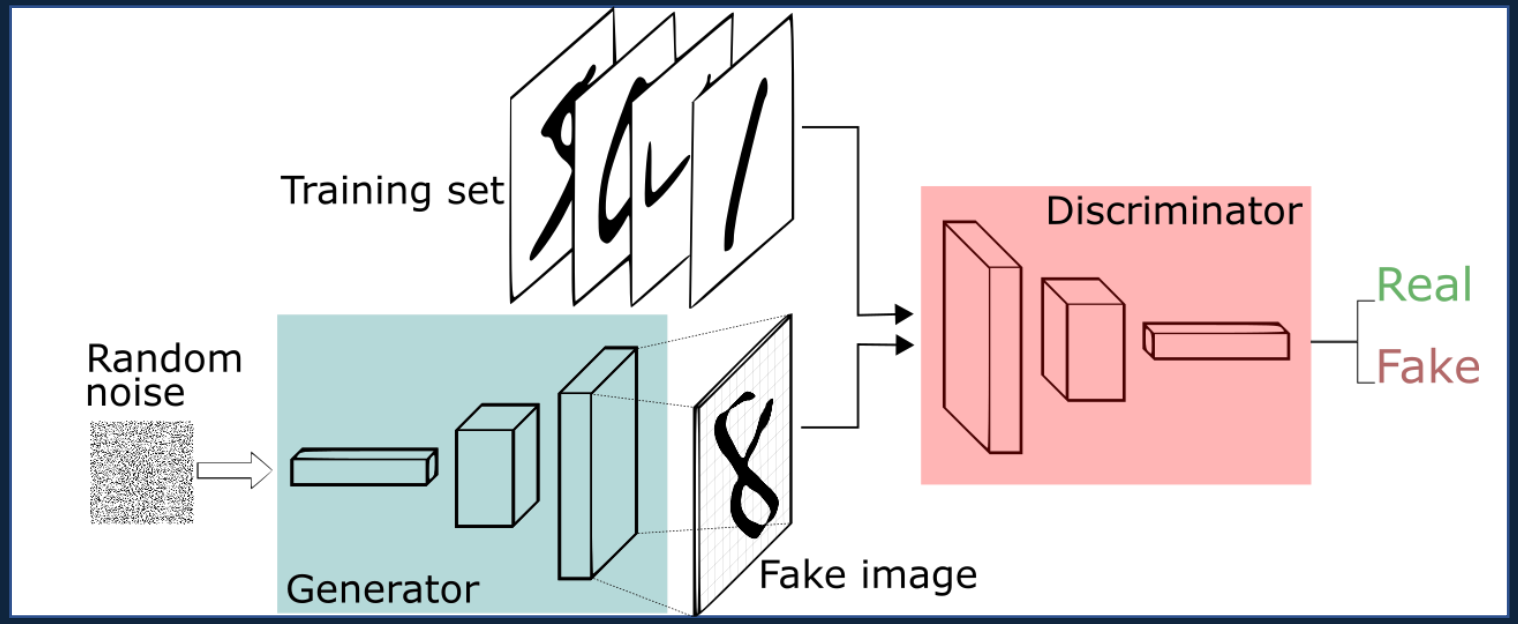

Color Image
Grayscale Image
Predicted Image

Image Generation

Adversarial Generative Networks (GAN)

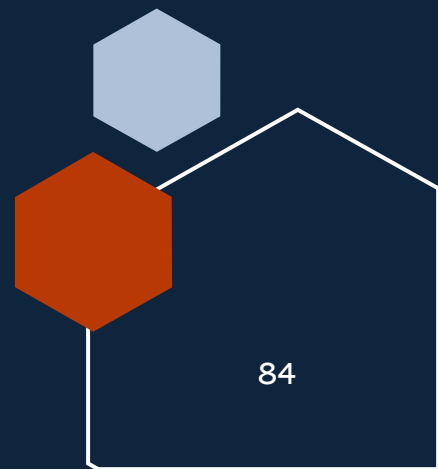


Project

About Dataset

MNIST dataset

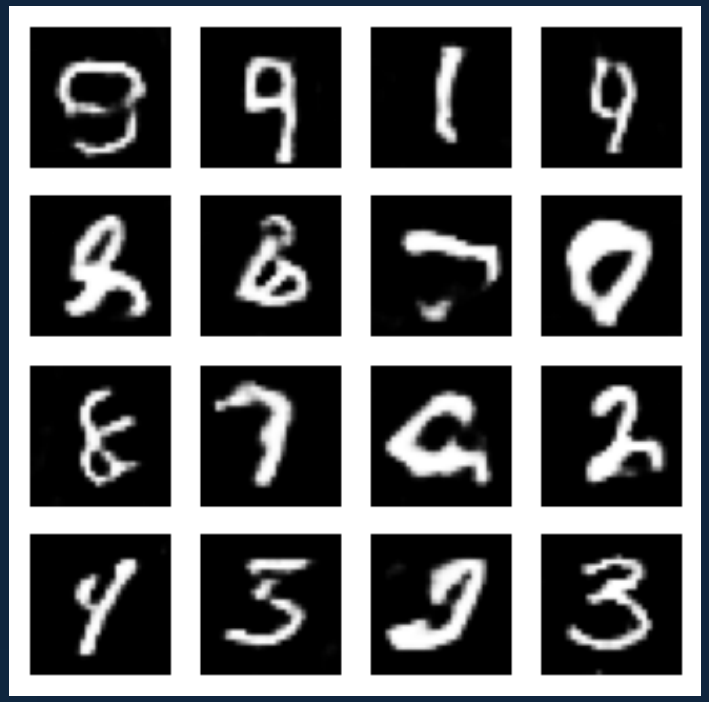
The MNIST dataset is a large database of handwritten digits that is commonly used for training image processing systems. It contains 70,000 grayscale images of handwritten digits, ranging from zero to nine. Each image is 28x28 pixels. The dataset is split into a training set of 60,000 images and a test set of 10,000 images. MNIST was created by "re-mixing" the samples from NIST's (National Institute of Standards and Technology) original datasets to be easier to use for machine learning and is now a widely recognized and frequently used resource for evaluating and comparing image classification methods.



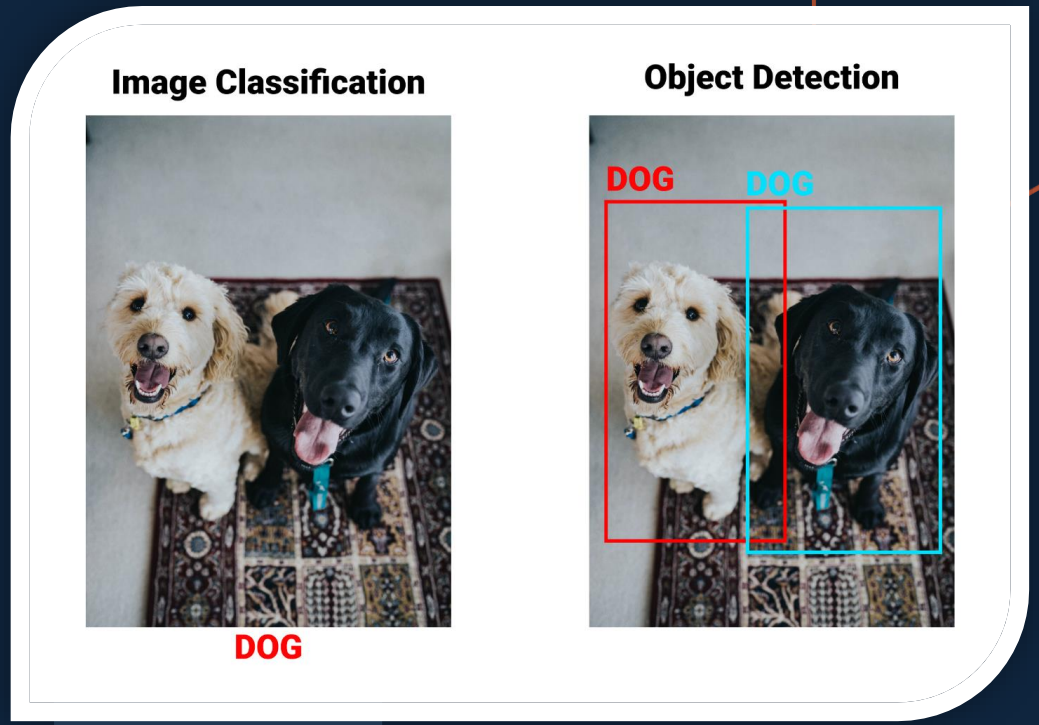
Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Image%20Generation/DCGANWithKeras.ipynb

Dataset

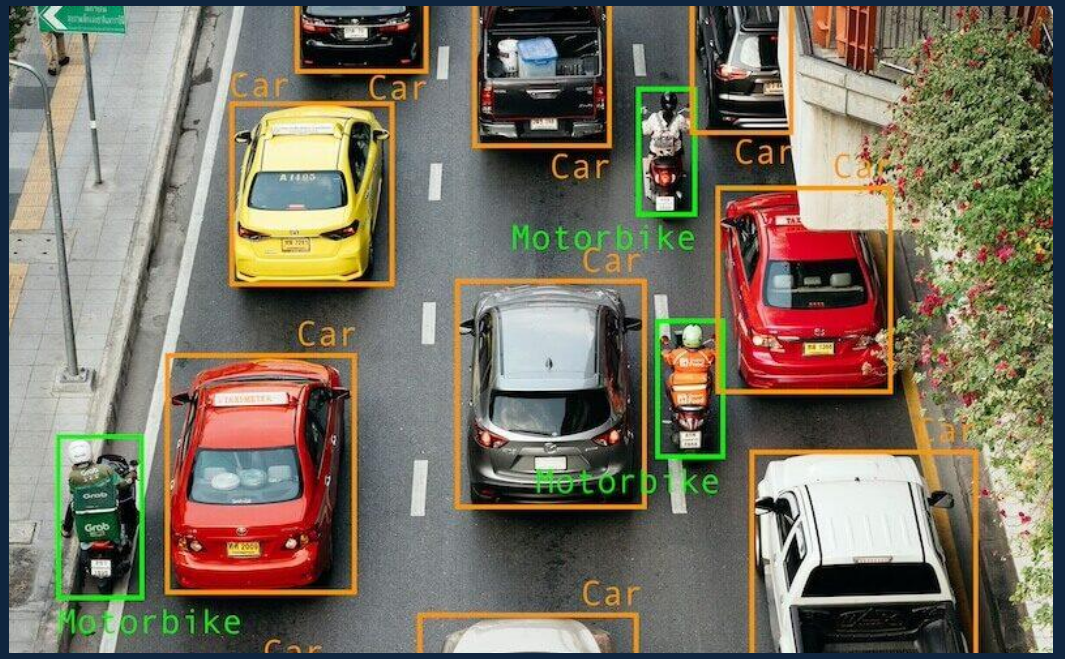
Result



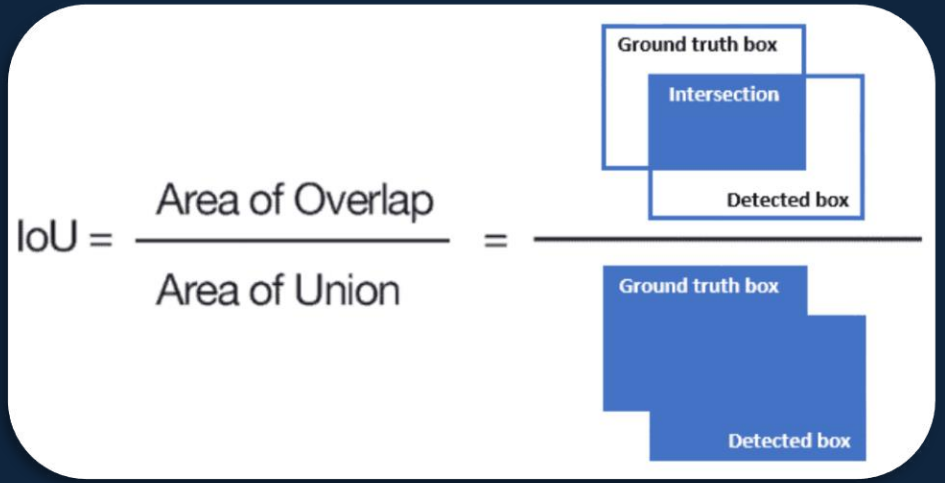
Object Detection



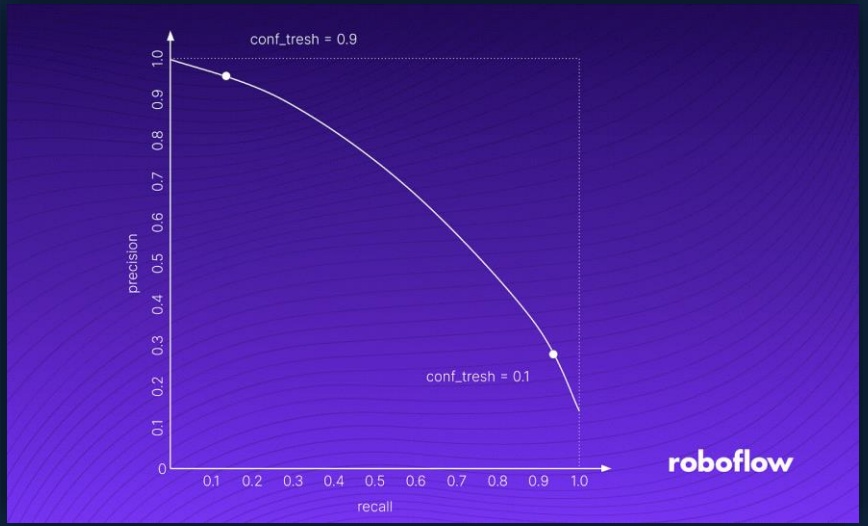
Bounding Box



Intersection Over Union
IoU



Mean Average Precision
mAP



Roboflow: Computer vision tools for developers and enterprises

Object Detection Models

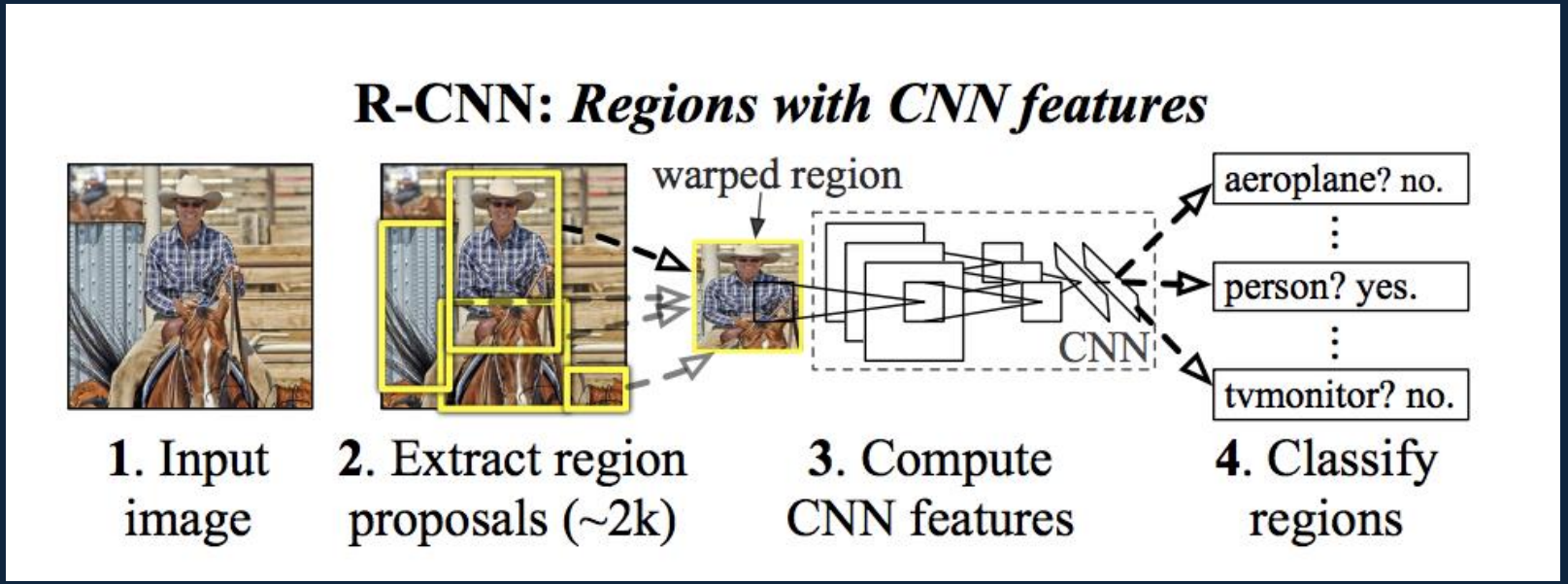
YOLO

SSDs

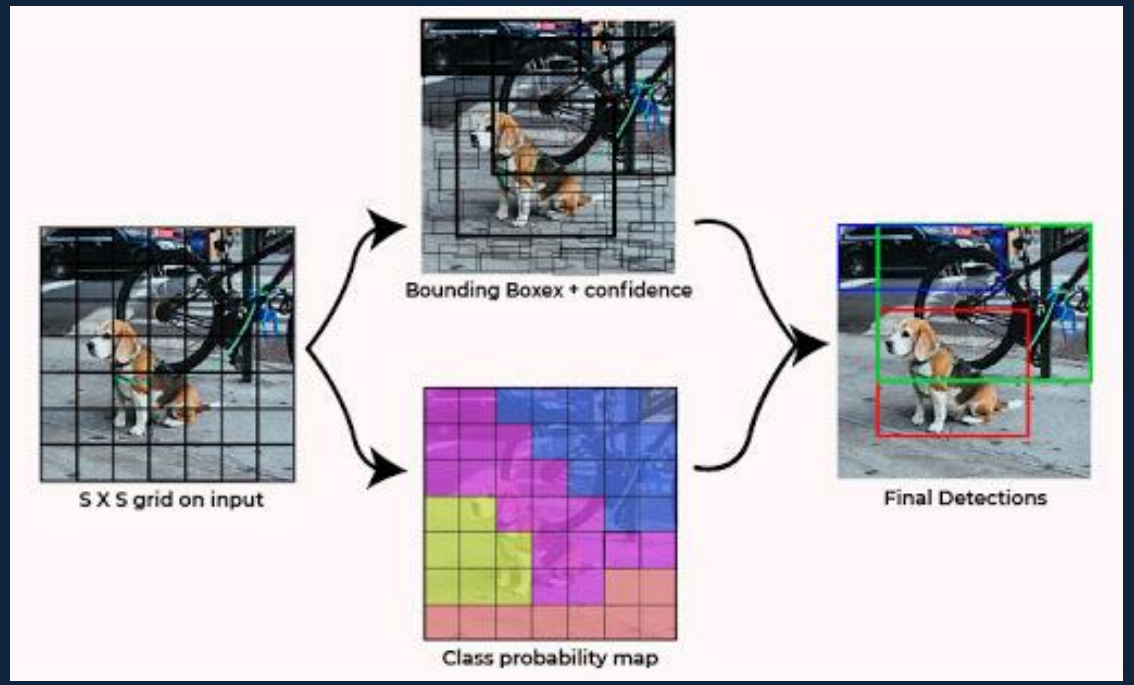
R-CNNs

Faster R-CNNs

R-CNN


R-CNN: *Regions with CNN features*
warped region
CNN
aeroplane? no.
person? yes.
tvmonitor? no.
1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

YOLO


Bounding Boxex + confidence
S X S grid on input
Class probability map
Final Detections

1 Project

Road Detection – YOLO v5

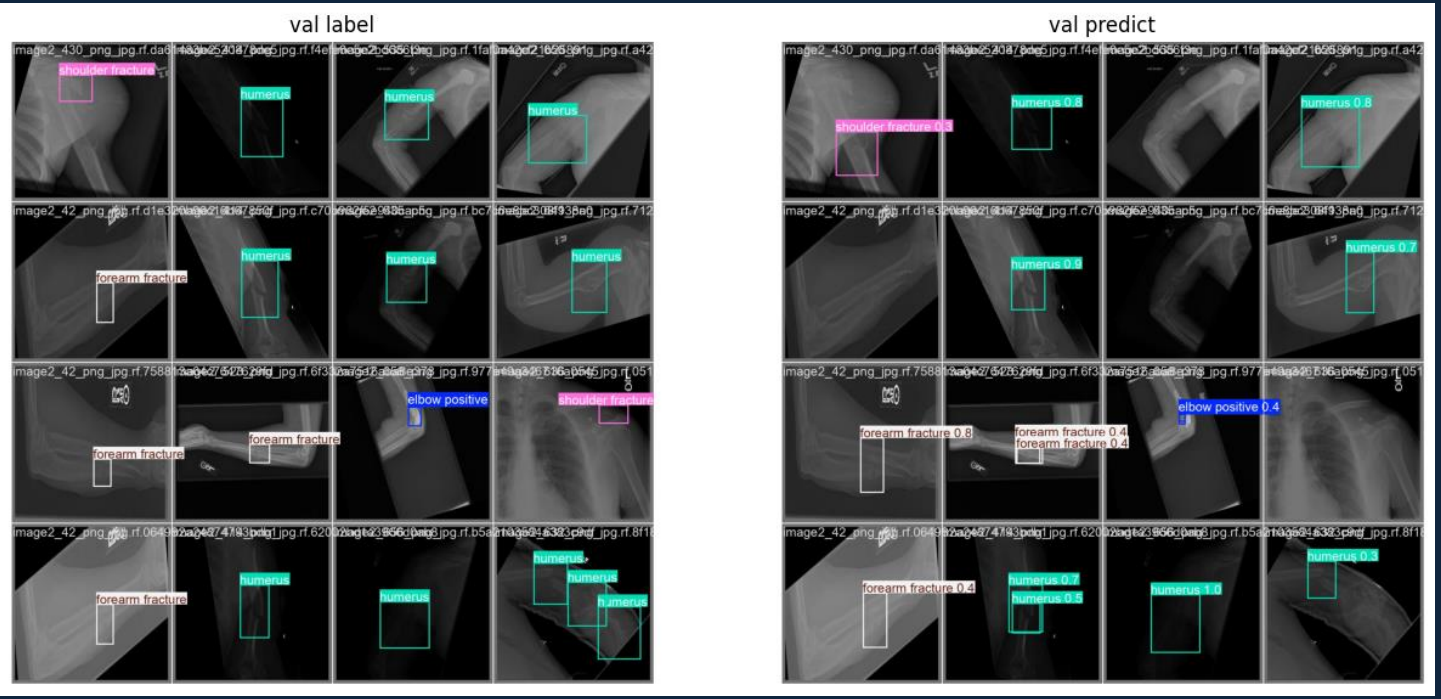
Result

Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Object%20Detection/ObjectDetectionUsingYOLOv5.ipynb

2 Project

Bone Fracture Detection – YOLO v8

Result

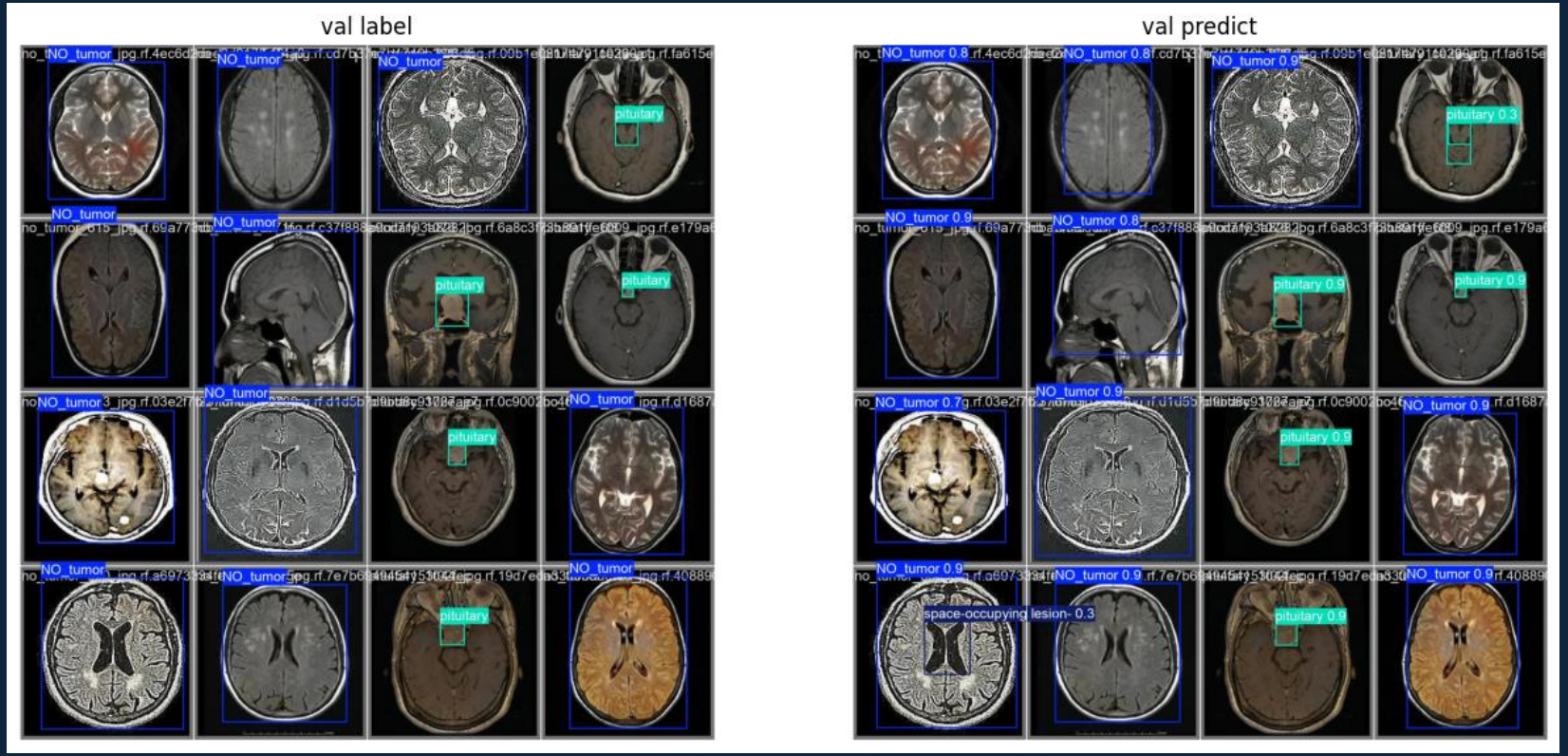


Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Object%20Detection/ObjectDetectionUsingYOLOv8.ipynb

3 Project

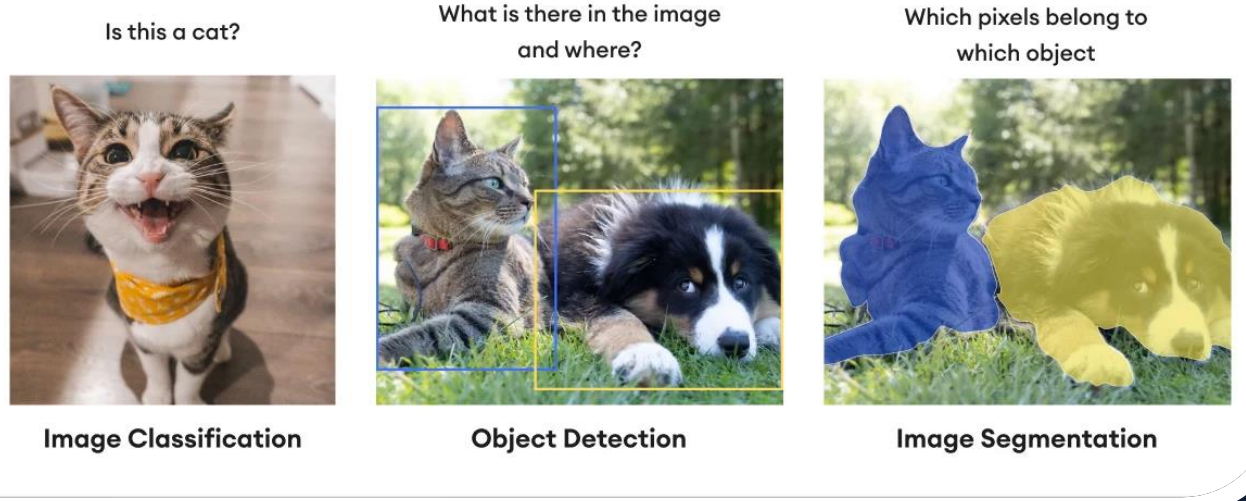
Tumor Detection – YOLO v11

Result



Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Object%20Detection/TumorDetectionUsingYolov11.ipynb

Image Segmentation



semantic vs instance segmentation

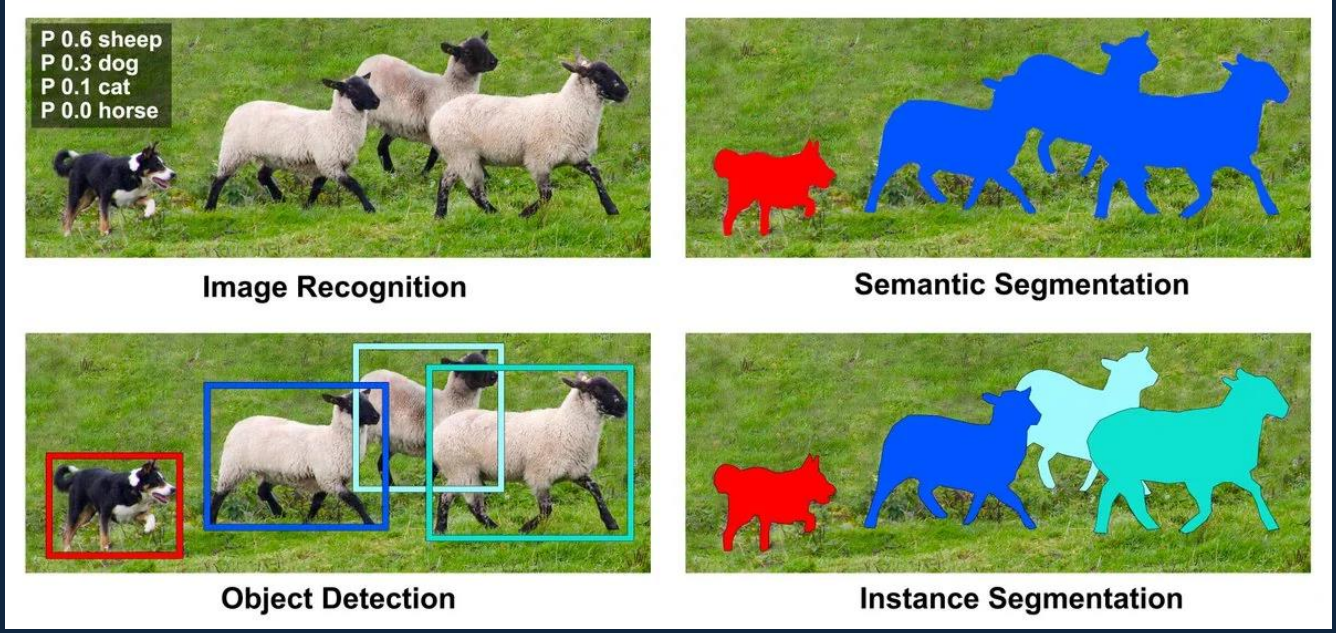


Image segmentation Models

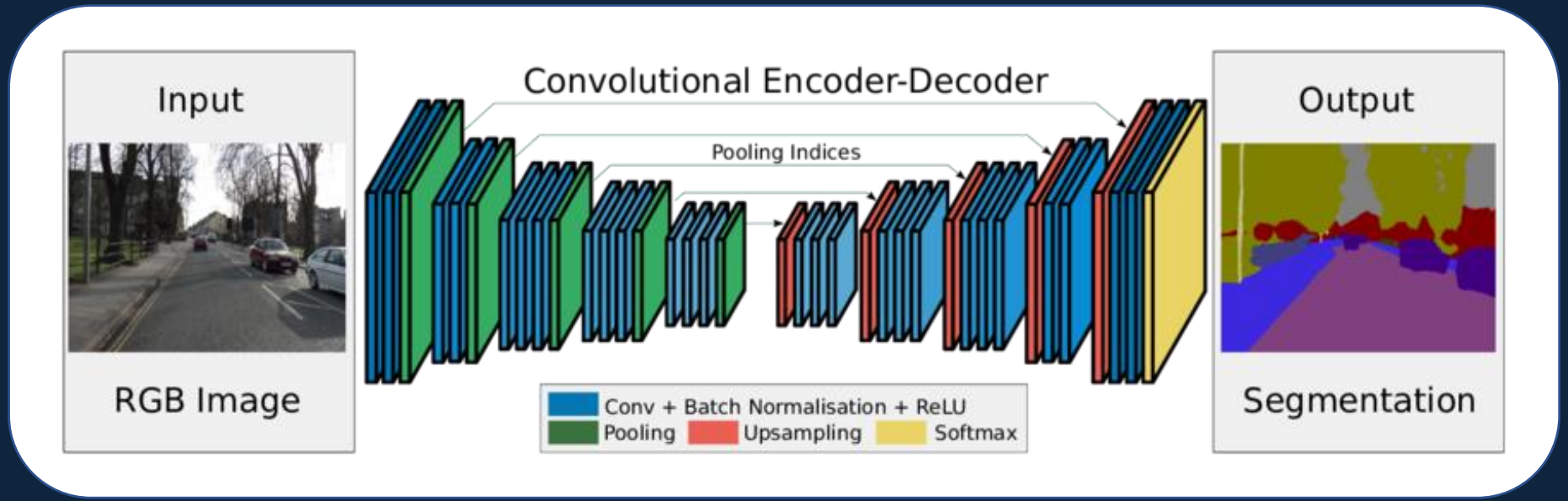
SegNet

U-Net

Mask R-CNN

YOLO

Segnet Architecture



1 Project

About Dataset

Chest X-ray Dataset for Tuberculosis Segmentation

Chest X-ray Organized Lung Segmentation Masks

This dataset consists of 704 chest X-ray images that have been curated from two sources: the Montgomery County Chest X-ray Database (USA) and the Shenzhen Chest X-ray Database (China). The images are used for training and evaluating machine learning models for tuberculosis (TB) detection.

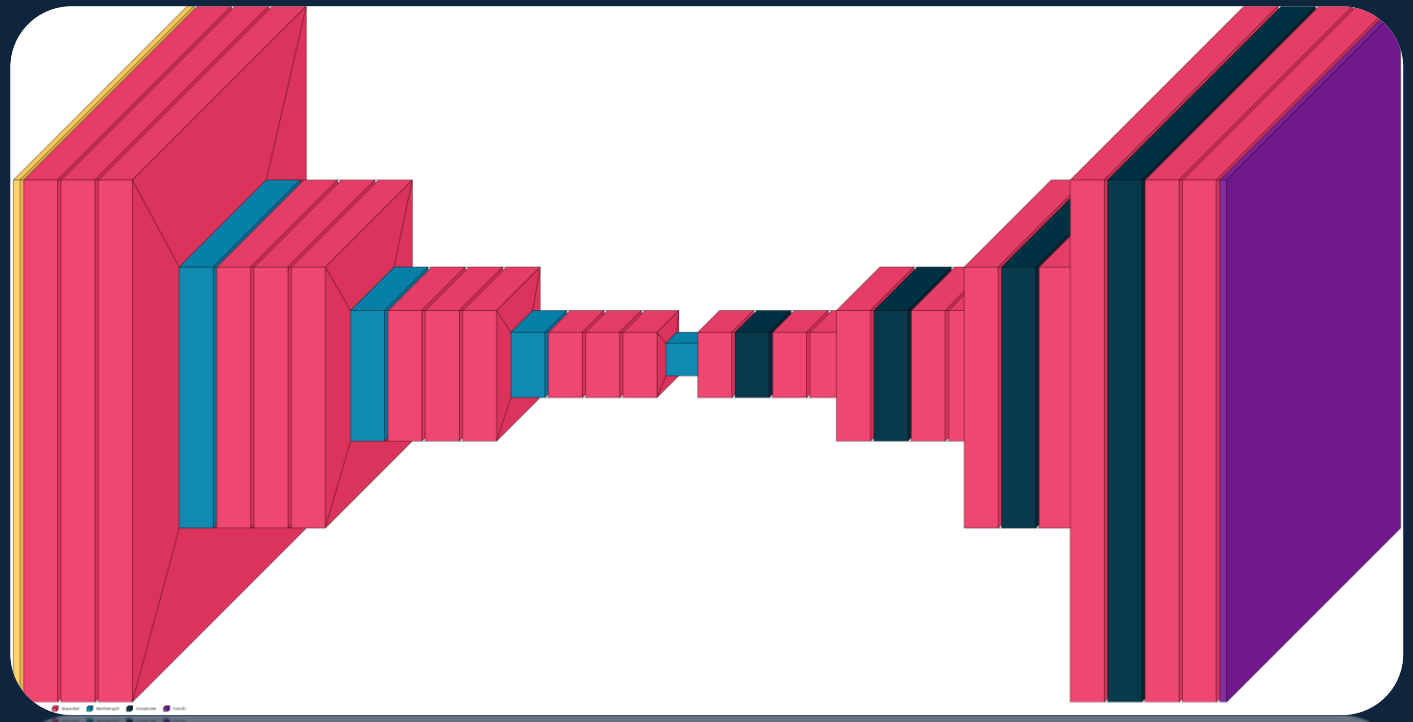
The dataset contains both tuberculosis-positive and normal chest X-rays, along with demographic details such as gender, age, and county of origin. The images are accompanied by lung segmentation masks and clinical metadata, which makes the dataset highly suitable for deep learning applications in medical imaging.

Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Image%20Segmentation/LungsSegmentationUsingU_Net.ipynb

Dataset

U-Net Architecture from Scratch

Total params: 18,981,633



Result

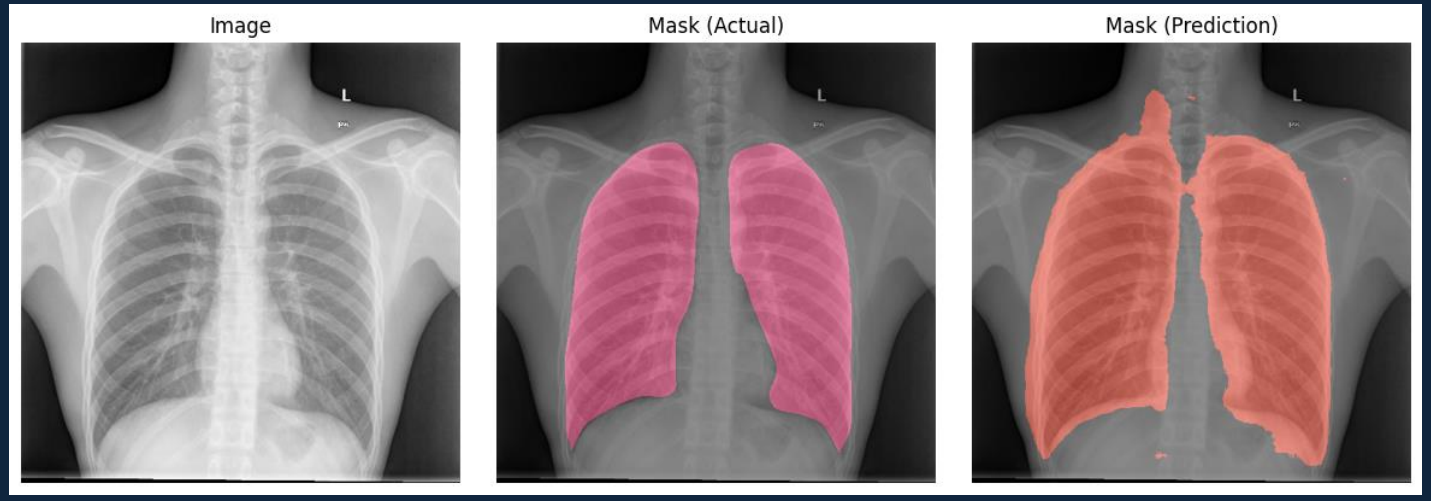
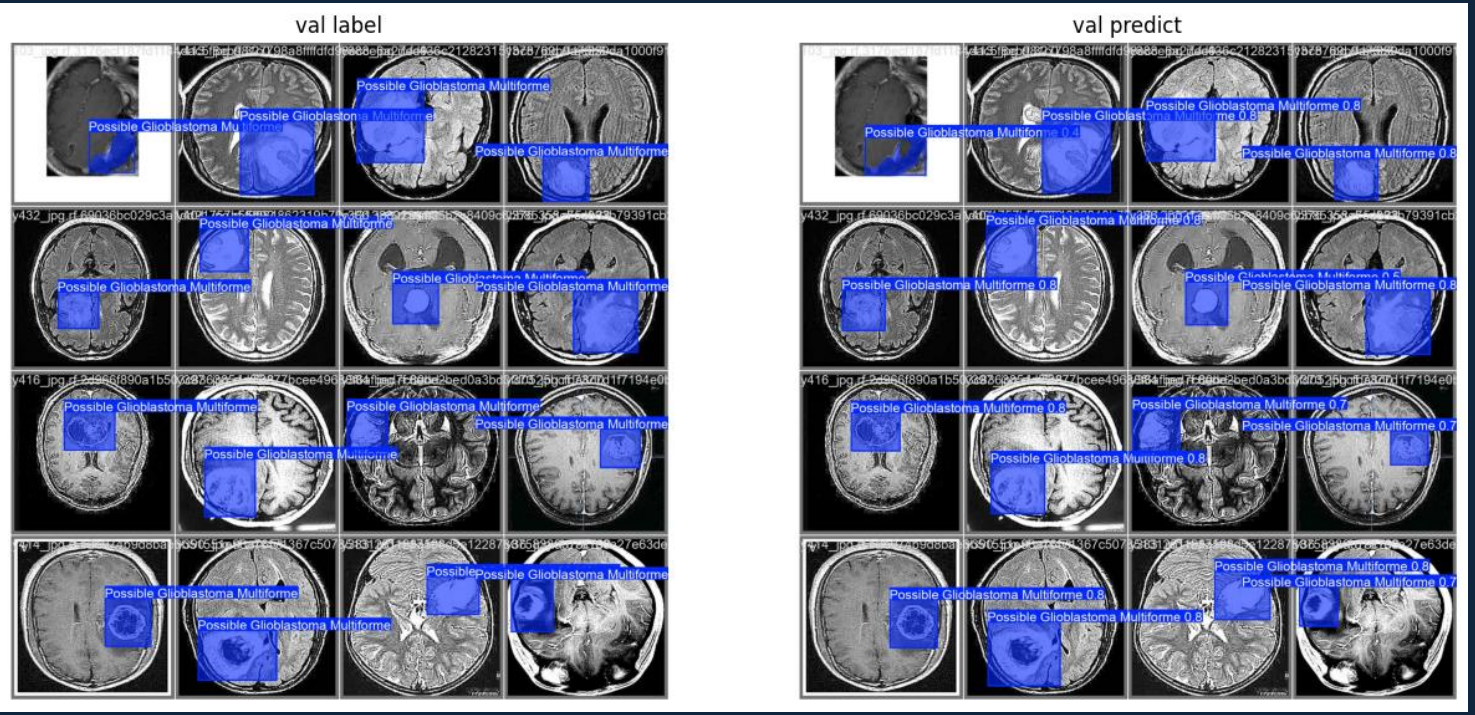


Image
Mask (Actual)
Mask (Prediction)
Image
Mask (Actual)
Mask (Prediction)

2 Project

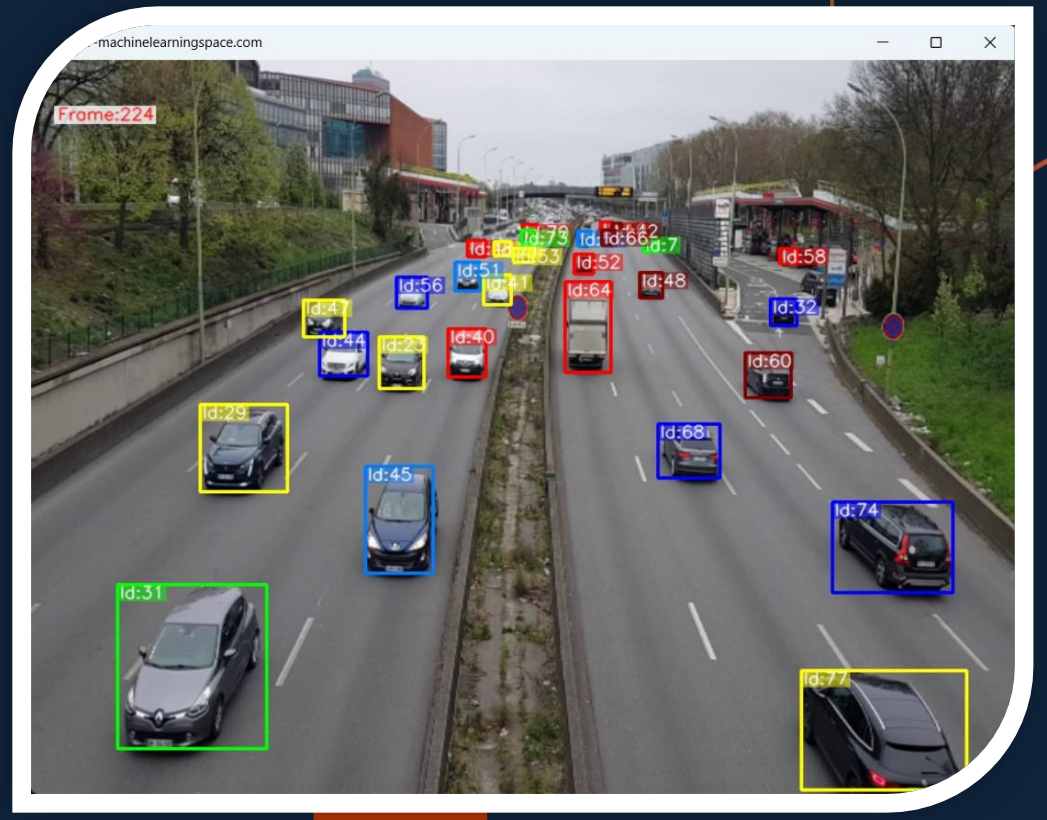
Tumor Segmentation – YOLO v11

Result



Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Image%20Segmentation/TumorSegmentationUsingYolo.ipynb

Object Tracking



1 Project

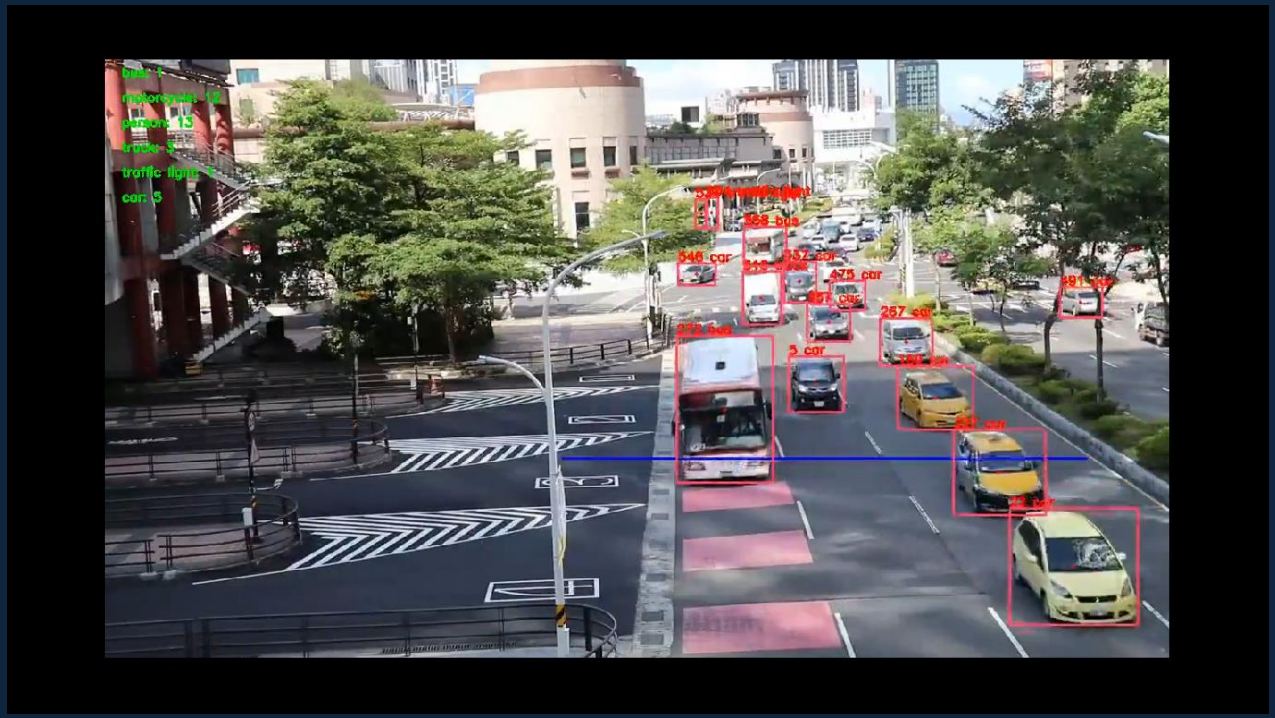
Tracking People in Video – YOLO v11

Download the result video: https://raw.githubusercontent.com/alirezasaharkhiz9/undergraduate-project-computer-vision/main/Modern%20Computer%20Vision/Object%20Tracking/ObjectTrackingWithYolo.avi

Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Object%20Tracking/ObjectTrackingWithYolo.ipynb

2 Project

Vehicle Tracking And Counting – YOLO v11

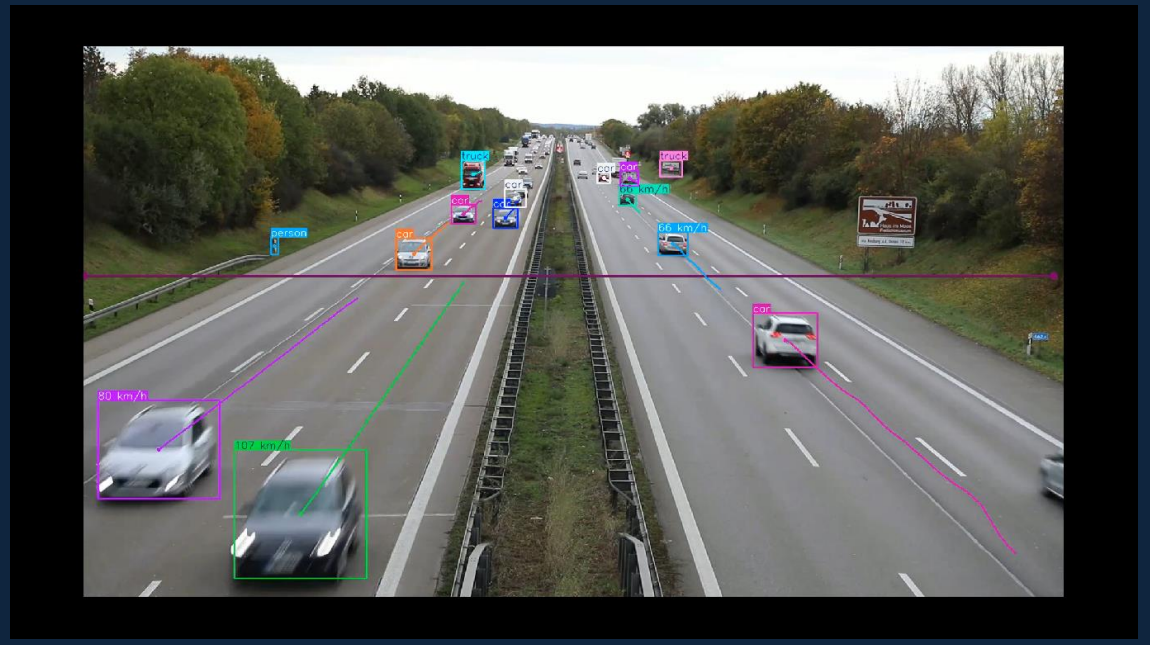


Download the result video: https://raw.githubusercontent.com/alirezasaharkhiz9/undergraduate-project-computer-vision/main/Modern%20Computer%20Vision/Object%20Tracking/TrackingAndCounting.mp4

Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Object%20Tracking/TrackingAndCounting.ipynb

3 Project

Speed Estimation – YOLO v11



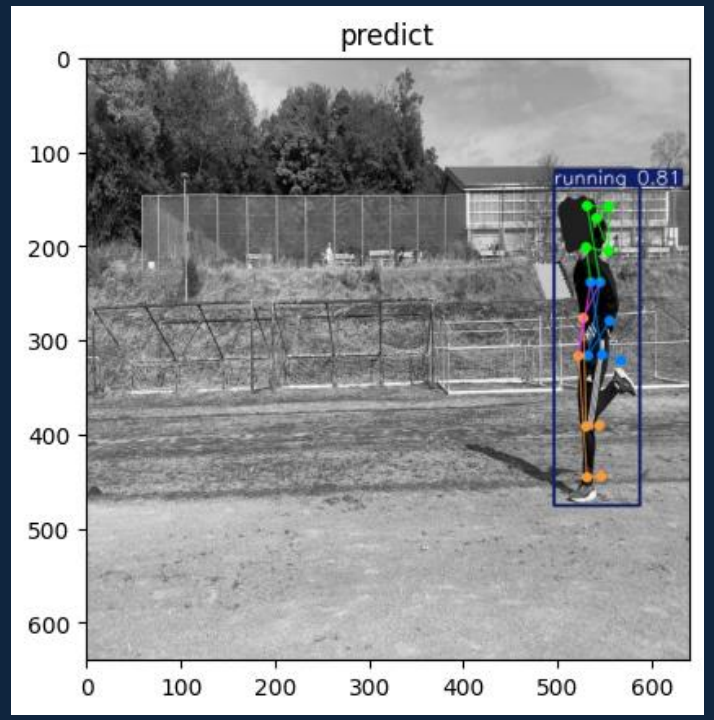
Download the result video: https://raw.githubusercontent.com/alirezasaharkhiz9/undergraduate-project-computer-vision/main/Modern%20Computer%20Vision/Object%20Tracking/SpeedEstimation.avi

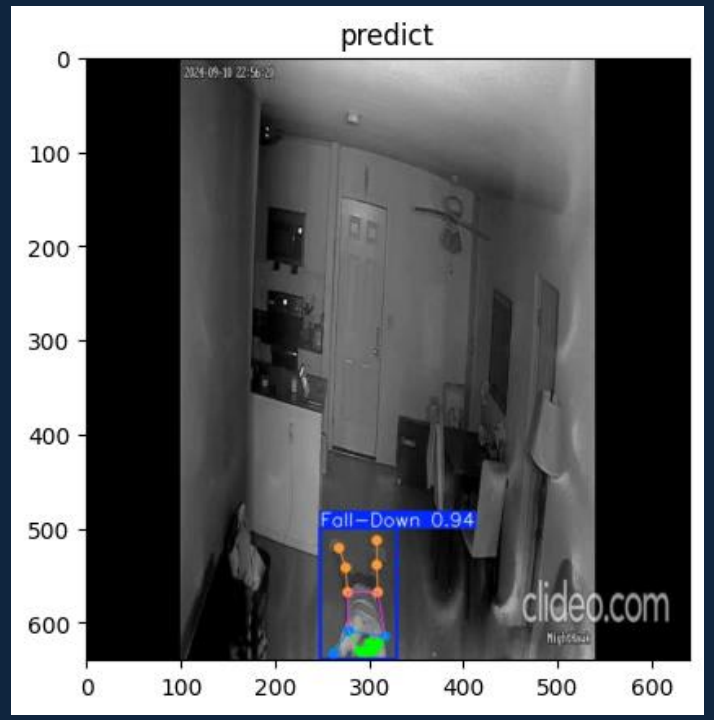
Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Object%20Tracking/SpeedEstimation.ipynb

Pose Estimation

Project

Pose Estimation – YOLO v11

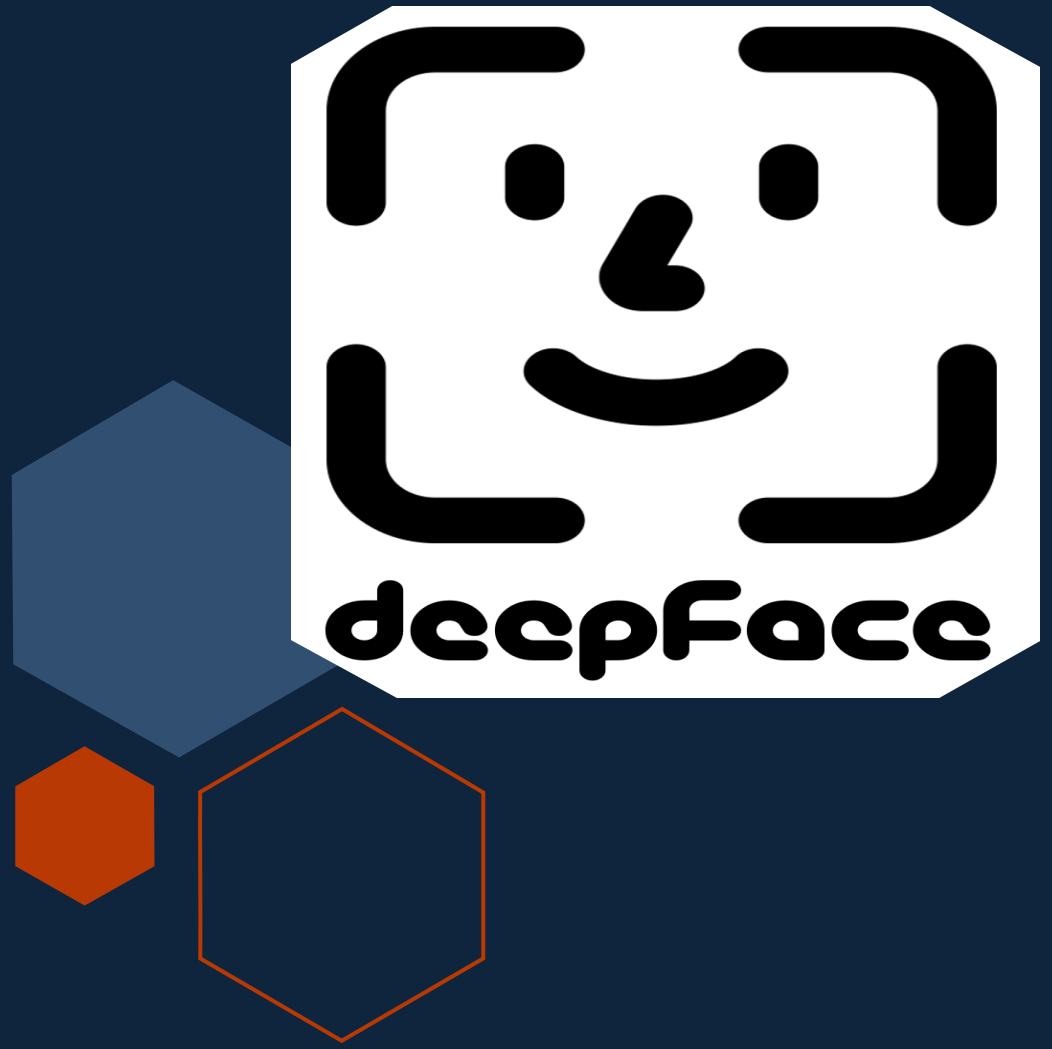




Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Pose%20Estimation/PoseEstimationWithYolo.ipynb

Face Recognition

DeepFace library



Source Code: https://github.com/alirezasaharkhiz9/Computer-Vision/blob/main/Modern%20Computer%20Vision/Face%20Recognition/FacialRecognitionWithDeepFace.ipynb

Thank you

Alireza Saharkhiz

 as.alirezasaharkhiz@gmail.com

 https://github.com/alirezasaharkhiz9