YOLOv7 Curriculum | Full Stack Object Detection Course |

MODULE 1 - INTRODUCTION
1.1 Introduction to YOLOv7
1.2 Run YOLOv7 on Windows 10/11
1.3 Run YOLOv7 on Ubuntu
1.4 Research Paper
MODULE 2 - Training Custom YOLOv7 [Using Roboflow Platform]
2.1 Training Process of YOLOv7 PPT
2.2 Training your Custom YOLOv7 Model
2.3 Data Collection
2.4 Data Annotation
2.5 Dataset Bias & Health
2.6 Data Pre-processing
2.7 Data Augmentation
2.8 Roboflow train
2.9 Training Prerequisites
2.10 How to tune your training parameters

2.11 How to analyze your model performance
2.12 Active learning for faster automated annotation
2.13 Training Model
2.14 Deploying Model
MODULE 3 - YOLOv7 + Tracking Intro
3.1 Multi-Object Tracking (MOT)
3.2 Integrating YOLOv7 with Byte Track Windows 10/11
3.3 Integrating YOLOv7 with Nor-Fair Windows 10/11
MODULE 4 - Flask Integration
4.1 How to connect Flask with Yolov7
4.2 Fundamentals of Computer Vision with Front-End WebApp
4.3 Integration of YOLOv7 with front end
4.4 Widget and Buttons
4.5 Minimalistic Front End Design
MODULE 5 - YOLOv7 Apps
App 1 - Security with Dashboard
App 2 - Mining Safety Check (Hard hat, Glasses, and Vest)

App 3 - Retail Heat Maps - Check heatmaps density
App 4 - Agriculture Count Fruit and display on Widget
App 5 - Traffic, Vehicle Counting, Tracking, and speed calculation
App 6 - License Plate Recognition with Dashboard - Easy OCR
App 7 - Gaming Aimbot with YOLOv7
App 8 - Whiteboard Drawings to Web page generation using YOLOv7
Docker Video
MODULE 6 - Model Conversion
6.1 Pytorch->ONNX->TensorFlow
6.2 Pytorch->ONNX->CoreML
6.3 Pytorch->ONNX->OpenVino
6.4 Pytorch->ONNX->TensorRT
6.5 Pytorch->ONNX->DeepSparce
MODULE 7 - Pose Estimation
7.1 Getting started with YOLOv7 Pose Estimation
7.2 Bicep Curl Application
7.3 Yoga Pose Identification

7.4 Fall Detection

YOLOR Curriculum | Detection, Tracking & Build Apps |

MODULE 1 - INTRODUCTION
1.1 Introduction to YOLOR
1.2 Run YOLOR on Windows 10/11
1.3 Run YOLOR on Ubuntu
1.4 Run YOLOR on Ubuntu on Colab
MODULE 2 - Training Custom YOLOR
2.1 Data Collection
2.2 Data Pre-processing
2.3 Data Augmentation
2.4 Dataset Bias & Health
2.5 How to tune your training parameters
2.6 How to analyze your model performance
2.7 Active learning for faster automated annotation
MODULE 3 - YOLOR + Tracking
3.1 Multi-Object Tracking (MOT)
3.2 What is DeepSORT and How it works.

3.3 Integrating YOLOR with DeepSORT
MODULE 4 - StreamLit Web-App
4.1 Why StreamLit?
4.2 StreamLit Basics
4.3 StreamLit + YOLOR Integration
4.4 YOLOR StreamLit on Images
4.5 YOLOR StreamLit on Video
4.6 YOLOR StreamLit Analytics UI
MODULE 5 - YOLOR App Development
App 1 - Card Counting
App 2 - Intrusion Detection
App 3 - Visual Code Generation
App 4 - Weed/Plant Detection
App 5 - Apple Center Stage
App 6 - Text Detection and OCR
App 7 - Push Up Detection
App 8 - Hard Hat Detection

YOLOX Curriculum | Computer Vision Dashboard |

MODULE 1 - Object Detection - YOLOX Basics
1.1 Introduction to YOLOX
1.2 YOLOX Paper Review
1.3 Loss Function
1.4 Evaluation Metrics
1.5 Intersection Over Union
1.6 Non-Max Suppression
1.7 Mean Average Precision
1.8 YOLOX on Video/Image
1.9 YOLOX Live Inference
MODULE 2 - Training Custom YOLOX
2.1 Data Collection
2.2 Data Augmentation
2.3 Dataset Bias & Health
2.4 How to tune your training parameters
2.5 How to analyze your model performance

2.6 Active learning for faster automated annotation
MODULE 3 - YOLOX + Tracking
3.1 Object Tracking Basics
3.2 SOTA exploration and Paper Review
3.3 Implementing Basic Tracking Architecture
3.4 Multi-Object Tracking (MOT)
3.5 Integrating YOLOX with SORT
MODULE 4 - YOLOX App Development
4.1 App 1- Detect & Track Vehicles
4.2 App 2 - Count & Sort Vehicles
4.3 App 3 - Identify Direction Trajectories
4.4 App 4 - Speed Calculation
4.5 Track from LIVE RTSP Camera
MODULE 5 - Traffic Flow Dash App - Build a Full-Stack Vision Analytics Dashboard
5.1 Dashboard Basics
5.2 Plotly Dash fundamentals
5.3 Dash User Interface Planning

5.4 HTML Elements
5.5 Dash Core Components
5.6 Visualizations - UI Widgets
5.7 Visualizations - Graphs
5.8 Visualizations - Pie Charts & Sunburst Charts
5.9 Bootstrapping Dash
5.10 Layout Design & Styling
5.11 YOLOX + Tracking Inference Integration in Dashboard
5.12 Enhancing the Dashboard
5.13 Deploy Solution

YOLOv5 Curriculum | Chess Piece Detection Course |
MODULE 1 - YOLOv5 Detection Strategy
MODULE 2 - Dataset Collection
MODULE 3 - Data Augmentation
MODULE 4 - Exporting the Dataset in YOLO Format
MODULE 5 - Training YOLOv5 AutoML
MODULE 6 - Training Host Web API
MODULE 7 - Custom YOLOv5 Deployment

YOLOv4 Curriculum | DarkNet, Training + PyQt App Development |
MODULE 1 - Introduction
1.1 Introduction to the Course
1.2 How to Take This Course
1.3 YOLOv4 Theory
1.4 YOLOv4 Prerequisites: Installations of Anaconda Python, Open CV etc.
1.5 YOLOv4 Object Detection on Image and Video
1.6 Detect Objects on Images and Video
1.7 Darknet Code Explanation YOLOv4 on Webcam
1.8 Social Distancing Monitoring App
1.9 Count Parked Cars
1.10 DeepSORT Intuition - How DeepSORT Object Tracking Works
1.11 Robust Tracking with YOLOv4 and DeepSORT
1.12 YOLOv1 - YOLOv3 Evolution
MODULE 2 - YOLOv4 Trainers Course
2.1 Introduction to Data Annotation
2.2 YOLOv4 format for Image Labelling

2.3 YOLOv4 Labelling Tools
2.4 Web-scaping Data
2.5 Annotating Images with LabelImg
2.6 Labeling on Video using LabelImg
2.7 Labeling on Video Using Darklabe
2.8 Annotation Summary
2.9 Data Annotation Key Take-away
2.10 Introduction How to Create Custom Dataset
2.11 Toolkit for Downloading Image Datasets
2.12 Downloading Images from Specific Classes
2.13 Converting Downloaded Files to YOLOv4 format
2.14 Data Augmentation using Rotational Transform
2.15 Summary - Key Takeaways for Custom Datasets
2.16 Introduction to Training YOLOV4 with DarkNet Framework
2.17 Step 1 - Configuring the files for Training
2.18 Step 2 - Creating the obj.names file
2.19 Step 3 - Dataset Placement for Training

2.20 Step 4 - Train Test metafiles
2.21 Step 5 - Training YOLOv4
2.22 Trained YOLOv4 Execution on Image and Video for Mask Detection
2.23 When to Stop Training
2.24 Summary - Key Takeaways
MODULE 3 - YOLOv4 PyQT Course
3.1 Lecture 1: Introduction to Object Detection with PyQt
3.2 Lecture 2: Installing PyQt
3.3 GUI Layout using PyQt Designer
3.4 Integrating PyQt with YOLOv4
3.5 Code Explanation
3.6 Adding GUI Widgets - Counting Objects
3.7 Adding Widgets - Slider Threshold
3.8 Adding Widgets - Class Filter using Checkbox Widget
3.9 Adding Widgets - Real-Time Live Plot Graph Widget
3.10 Social Distancing in PyQt Activity
3.11 Conclusion

3.12 Facial Recognition Attendance GUI - PyQt_Course

YOLOv3 Curriculum | Object Detection Course |

MODULE 1- Introduction
1.1 Introduction
1.2 How to Take This Course
MODULE 2 - Deep Learning Basics
2.1 Artificial Neural Network - Intuition
2.2 Convolutional Neural Network - Intuition
MODULE 3 - The Quickest Way to get YOLOv3 Up and Running!
3.1 Why Yolo is Better, Stronger & Faster! - Theory
3.2 The Very Brief Theory of YOLOv3
3.3 Execute YOLOv3 Like a Boss!
3.4 Need Help with Supervise.ly?
YOLOv3 Quiz
MODULE 4 - How to Train YOLOv3 - Training&Workflow
4.1 4 Steps to Setting up a Supervisely Deep Learning Cluster
4.2 How to Web Scrape Images for your Dataset like a PRO!
4.3 The Best Way to Annotate your Dataset

4.4 How to let the AI Annotate your Dataset for you - Human in the Loop Annotation
4.5 Got Little Data? No Problem! Data Augmentation to the Rescue
4.6 How to Train a YOLOv3 Network
4.7 A Quick and Easy Method Deploying your Custom Object Detector after Training
MODULE 5 - Post-Processing and Maintenance
5.1 How to Record video, change bounding box color and add confidence percentage
5.2 Cleaning up you Supervisely Cluster and Cluster Maintenance
MODULE 6 - Conclusion
Bonus Section
XR Developers Podcast with Ritesh Kanjee