Yolo Algorithm - Lab

Original Source:

YOLOv8 & YOLO11: Custom Object Detection & Web Apps 2024 (Udemy Course)

What is YOLO

- YOLO is the state-of-the-art object detection algorithm & it is so fast that it has become a standard way of detecting objects in the field of Computer Vision.
- Previously, people were using Sliding Window Object Detection, then more faster versions were invented which include RCNN, Fast RCNN, Faster RCNN.
- YOLO was invented in 2015
- YOLO outperforms previous object detection algorithms.

3

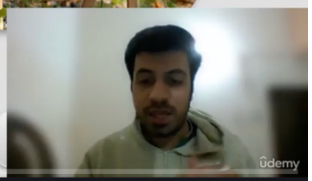
Image Classification

Is this a Deer or a Person in the Image?

Neural Network Output

Deer = 1

Person = 0



4

Object Localization

In the object localization, we are not only telling what class it is but we are also telling about bounding box or the position of the object in the image

- Where exactly is the Deer in the image?

Object Localization

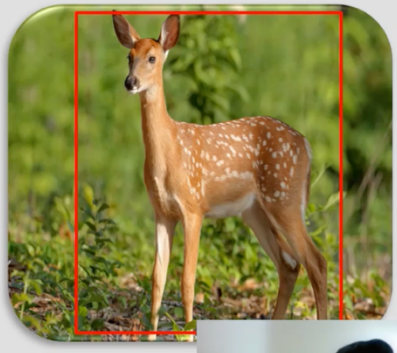
Neural Network Output:

- Deer = 1

Person = 0

+

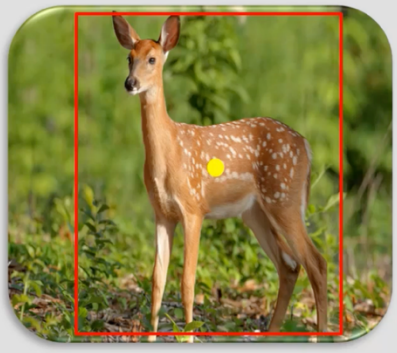
Bounding Box



Object Localization

In Neural Network, we have an output like this: Vector Size = 7

$$\begin{bmatrix} P_c \\ B_x \\ B_y \\ B_w \\ B_h \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 50 \\ 70 \\ 60 \\ 70 \\ 1 \\ 0 \end{bmatrix}$$



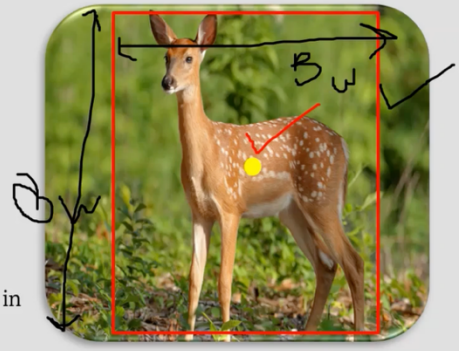
P_c = Probability of Class
B_x, B_y = Represents coordinate for center which is indicated in yellow color
C_1 = Represents class 1 which is for Deer
C_2 = Represents class 2 which is for Person
B_w, B_h = Represent the width and height of the Red Box

P_c = 1, if there is any object in the image, if there is no object in the image P_c = 0, and in that case the rest of the values does not matter

Object Localization

In Neural Network, we have an output like this: Vector Size = 7

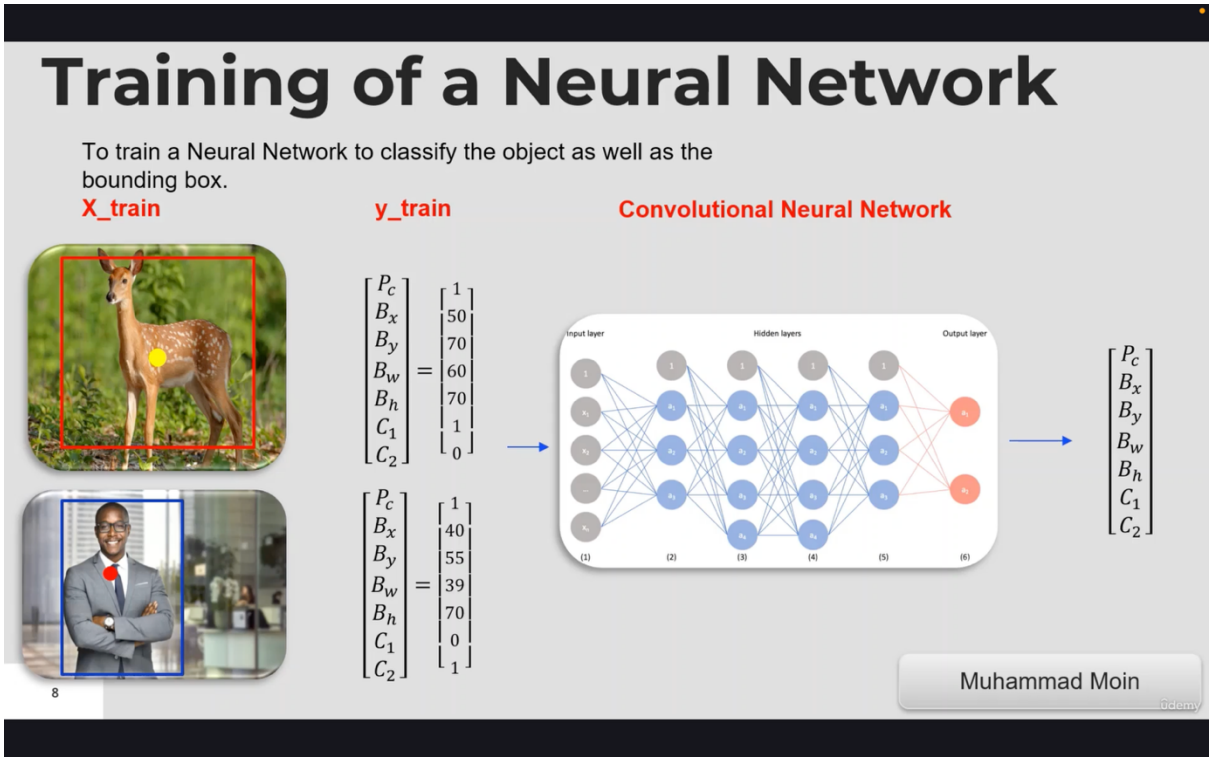
$$\begin{bmatrix} P_c \\ B_x \\ B_y \\ B_w \\ B_h \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 50 \\ 70 \\ 60 \\ 70 \\ 1 \\ 0 \end{bmatrix}$$



P_c = Probability of Class
B_x, B_y = Represents coordinate for center which is indicated in yellow color
C_1 = Represents class 1 which is for Deer
C_2 = Represents class 2 which is for Person
B_w, B_h = Represent the width and height of the Red Box

P_c = 1, if there is any object in the image, if there is no object in the image P_c = 0, and in that case the rest of the values does not matter

7



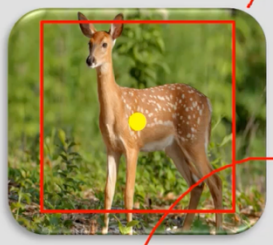
Neural Network only understand numbers so there is a 7-value vector for each image.

We have to train a neural network on our input images (large number of images)

Training of a Neural Network

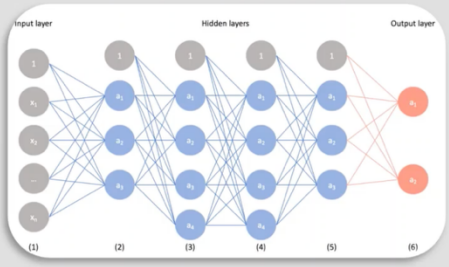
To train a Neural Network to classify the object as well as the bounding box.

X_train **y_train** **Convolutional Neural Network**



$$\begin{bmatrix} P_c \\ B_x \\ B_y \\ B_w \\ B_h \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 50 \\ 70 \\ 60 \\ 70 \\ 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} P_c \\ B_x \\ B_y \\ B_w \\ B_h \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 40 \\ 55 \\ 39 \\ 70 \\ 0 \\ 1 \end{bmatrix}$$

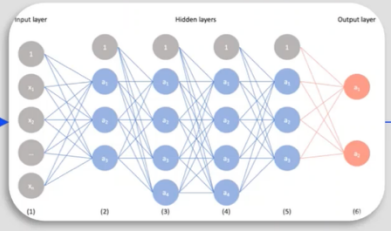


$$\begin{bmatrix} P_c \\ B_x \\ B_y \\ B_w \\ B_h \\ C_1 \\ C_2 \end{bmatrix}$$

Training of a Neural Network

We can train a neural network in such a way, if we enter a new image, it will tell us about the particular vector

Convolutional Neural Network

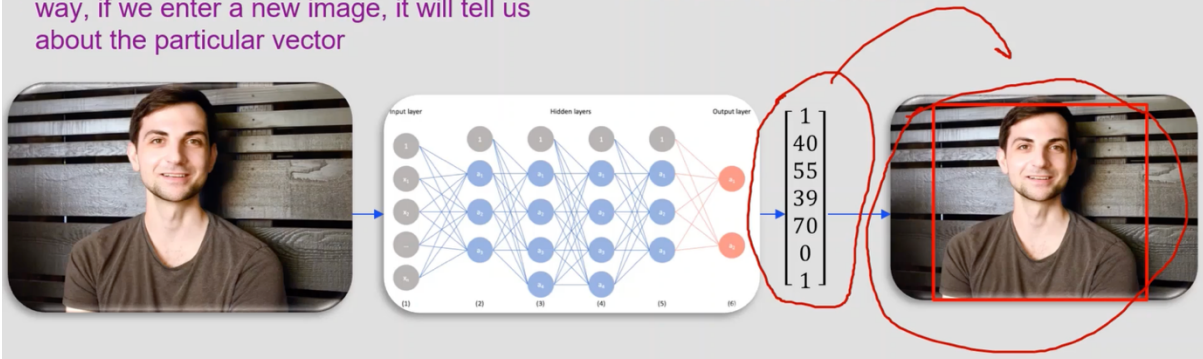


$$\begin{bmatrix} 1 \\ 40 \\ 55 \\ 39 \\ 70 \\ 0 \\ 1 \end{bmatrix}$$

Training of a Neural Network

We can train a neural network in such a way, if we enter a new image, it will tell us about the particular vector

Convolutional Neural Network



Issue: Why can't we use this approach for object detection and Why do we need YOLO?

Because neural network works well only for a single object.

What about multiple objects in an image?

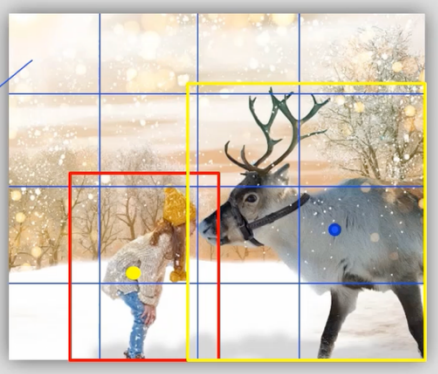
In case of multiple objects in an image for example 5 persons, 1 dog. For example, if we have n number of objects in an image, then determining the size of neural network is very hard.

For example if we have 10 objects in an image then the size of the vector will be 10 x7 = 70

So we need to have some thing else

YOLO Algorithm

YOLO Algorithm will divide the image into grid cells. I am using 4 x 4 grid here, there can be 19 x 19 or 3 x 3 grid.



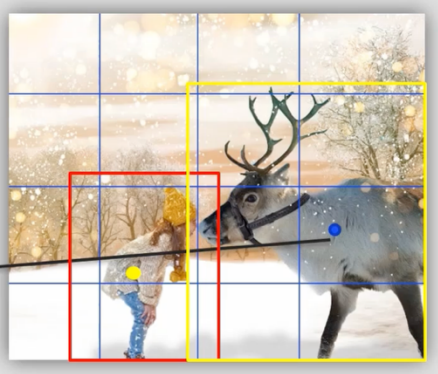
As we have no bounding object in this grid cell, so probability of class Pc will be zero and the rest of the values doesnot matter

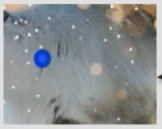
$$\begin{bmatrix} P_c = 0 \\ B_x \\ B_y \\ B_w \\ B_h \\ C_1 \\ C_2 \end{bmatrix}$$

Muhammad Moin

YOLO Algorithm

As we can see that the Cow is expanding to multiple gird cells, we try to find the central place of the Cow, and the cow belong to that grid cell as shown below





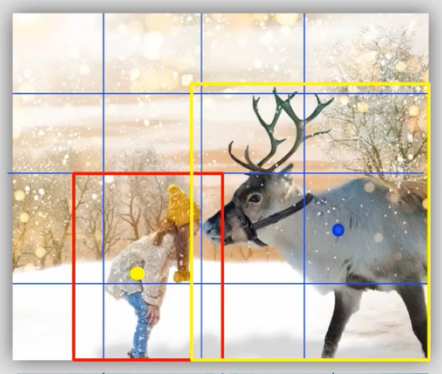
C_1
$= Represents\ class\ 1\ which\ is\ for\ Cow$

C_2
$= Represents\ class\ 2\ which\ is\ for\ Girl$

$$\begin{bmatrix} 1 \\ 0.05 \\ 0.3 \\ 2 \\ 3 \\ 1 \\ 0 \end{bmatrix}$$

2 → Represents the width as it is covering 2 cells
3 → Represents the height as it is covering 3 cells

Training

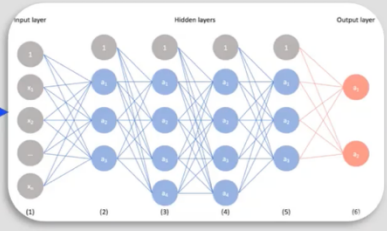


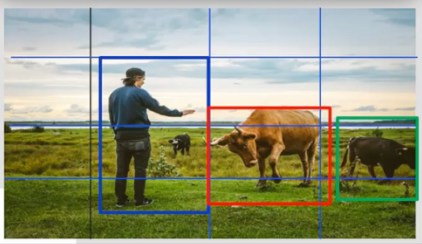
16 such vectors

4 by 4 grid cell Each vector has size 7

4 by 4 by 7

$$\begin{bmatrix} P_c \\ B_x \\ B_y \\ B_w \\ B_h \\ C_1 \\ C_2 \end{bmatrix}$$





16 such vectors

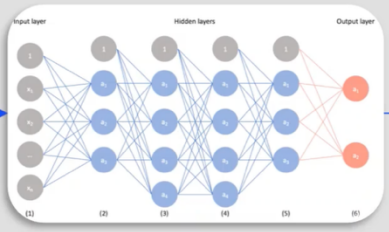
4 by 4 grid cell Each vector has size 7

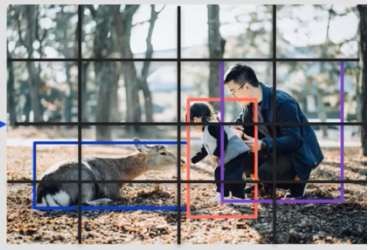
4 by 4 by 7

$$\begin{bmatrix} P_c \\ B_x \\ B_y \\ B_w \\ B_h \\ C_1 \\ C_2 \end{bmatrix}$$

Muhammad Moin

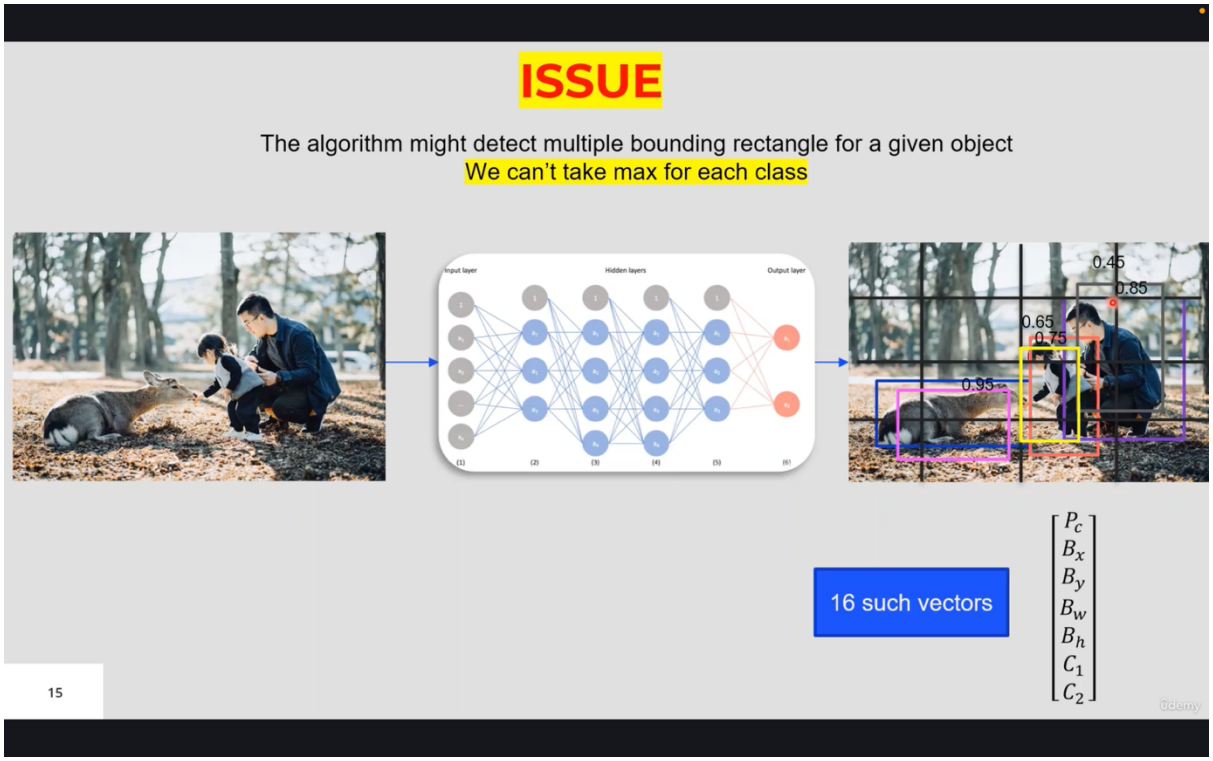
Prediction



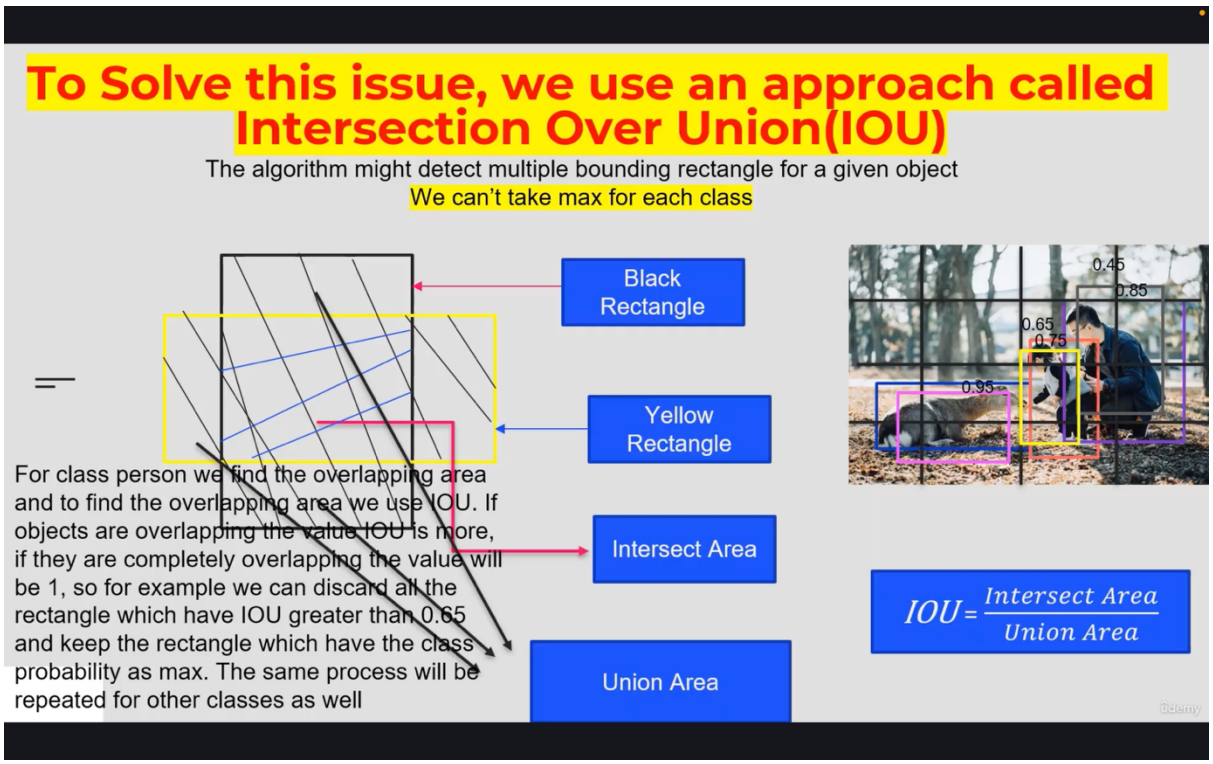


16 such vectors

$$\begin{bmatrix} P_c \\ B_x \\ B_y \\ B_w \\ B_h \\ C_1 \\ C_2 \end{bmatrix}$$



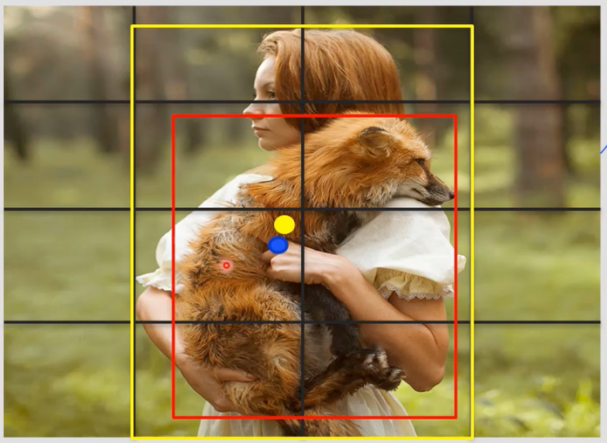
Bounding Box issue: Person with a confidence score of 0.85 and 0.45 similarly two bounding boxes for child and two bounding boxes for deer (0.95 and 0.35). We can't take max bounding box by removing those bounding boxes that have less confidence score.



To Solve this issue, we use an approach called Intersection Over Union(IOU)

So, this technique is also called Non-Max Suppression. After detecting all objects, we apply Non-Max Suppression, and we get unique bounding boxes/

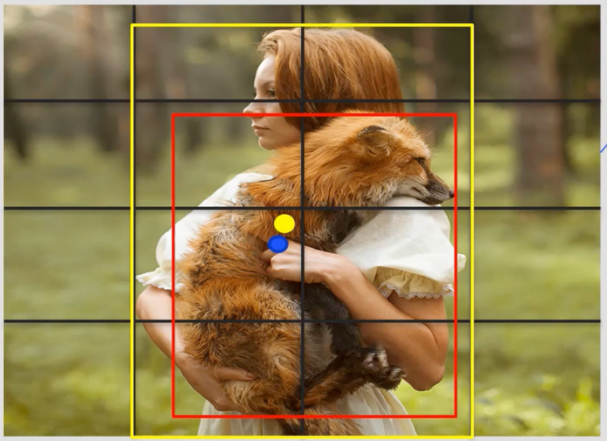
What if One Grid Cell has Center of Two Objects



Now this grid cell above can only represent one class. How can we represent two classes

$$\begin{bmatrix} 1 \\ 0.05 \\ 0.3 \\ 2 \\ 3 \\ 1 \\ 0 \end{bmatrix}$$

What if One Grid Cell has Center of Two Objects



So, in such case instead of having a vector of size 7 we have a vector of size 14.

$$\begin{bmatrix} 1 \\ 0.08 \\ 0.9 \\ 3 \\ 3 \\ 0 \\ 1 \\ 1 \\ 0.05 \\ 0.3 \\ 2 \\ 3 \\ 1 \\ 0 \end{bmatrix}$$

Thank You

Yolo-V8

Introduction to YOLOv8

1

Udemy

Objectives

- What is YOLO
- What is YOLOv8
- Key Features of YOLOv8
- What are the reasons for using YOLOv8

2

Udemy

What is YOLO

- Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi introduced YOLO (You Only Look Once).
- YOLO gained popularity because of its accuracy while maintaining a small model size.
- From versions 1-4 YOLO was maintained in a C code in a custom deep learning framework written by Redmon called Darknet.
- In the last two years YOLOR, YOLOX, YOLOv6 and YOLOv7 emerged around the world, out of their own PyTorch based implementations. Each model has brought new SOTA techniques that continue to push model accuracy and efficiency.

*state-of-the-art (SOTA)

What is YOLOv8

- The latest version of YOLO, YOLOv8 was released on January 10, 2023, claiming advancements in structure and architectural changes with better results.
- YOLOv8 is the newest state of the art YOLO model that can be used for object detection, image classification and instance segmentation tasks.
- In YOLOv8 the issue of prolonged training is somewhat addressed.
- The tradeoff between training time and precision is achieved more in YOLOv8.

Roboflow – No Code AI

Roboflow is a popular tool and platform designed to simplify the process of building custom computer vision models. It provides an intuitive, user-friendly interface for labeling, augmenting, and managing datasets, and then training models on those datasets for specific use cases like object detection, image classification, and segmentation. Roboflow helps streamline the entire pipeline from data collection to model deployment, making it easier for developers and data scientists to work with machine learning models for computer vision.

Key Features of Roboflow:

Dataset Management

Labeling: Roboflow provides tools for manually annotating images for object detection, classification, and segmentation tasks. You can label your data quickly and efficiently within the platform.

Dataset Hosting: You can upload, organize, and store your datasets securely in the cloud, making it easy to access them when needed.

Data Augmentation

Image Augmentation: Roboflow offers built-in augmentation techniques like rotation, flipping, scaling, color adjustments, and more. This helps increase the diversity of your dataset without collecting additional data.

Custom Augmentation Pipelines: You can create custom augmentation pipelines to apply specific transformations to images to improve model generalization.

Model Training

Pretrained Models: Roboflow allows you to start with pretrained models (like YOLO, EfficientDet, etc.) and fine-tune them on your custom dataset.

Custom Training: You can train models from scratch or transfer learning based on the specific use case.

Framework Support: Roboflow supports popular deep learning frameworks like TensorFlow, PyTorch, and others, so you can easily integrate your training process.

Model Deployment

Easy Deployment: Once the model is trained, Roboflow enables you to export it to multiple formats for deployment in different environments, including TensorFlow, PyTorch, CoreML, and ONNX.

Edge Deployment: The platform supports deploying models on edge devices, which is especially useful for IoT devices and real-time applications.

API Integration

Inference API: You can deploy models to Roboflow's cloud and run inference via a simple API, which can be integrated into your applications.

Python Integration: Roboflow also provides Python APIs, making it easy to integrate model training and inference into your code.

Version Control

Dataset Versioning: Roboflow automatically tracks the versions of your datasets, making it easier to keep track of changes, especially if you have multiple versions of data used for training.

Model Versioning: The platform also provides versioning for models, so you can keep track of your training experiments and deploy the best performing models.

Collaboration

Roboflow allows teams to collaborate on labeling datasets and managing projects. You can share datasets and models with others, and work on projects together, which is especially useful in team-based environments.

Steps to Use Roboflow:

Create an Account

Sign up for a free or paid account on the Roboflow website.

Upload Your Dataset

After logging in, you can create a new project and upload your dataset. Roboflow supports various formats such as JPG, PNG, and JSON annotations.

You can also use pre-labeled datasets or search for datasets within Roboflow's database.

Label Your Data

Use Roboflow's labeling tool to annotate images for the task you want to solve (e.g., bounding boxes for object detection). This can be done manually or with the help of automated tools, depending on your dataset.

Apply Data Augmentation

Once your data is labeled, you can apply various data augmentation techniques (flipping, rotation, etc.) to increase the variety of your training data and improve your model's robustness.

Train Your Model

Choose a model architecture (e.g., YOLO, EfficientDet, etc.) from Roboflow's pre-trained options, and then train the model on your dataset.

You can monitor the training process, view metrics like loss and accuracy, and make adjustments as needed.

Evaluate the Model

After training, evaluate the model's performance using Roboflow's built-in testing tools. You can track precision, recall, and mAP (mean average precision) for object detection tasks.

Export the Model

Once you are satisfied with the performance of your model, you can export it in various formats, such as TensorFlow, PyTorch, or ONNX, to deploy it in your application.

Deploy and Integrate

Roboflow also provides an API for you to run inference on new images. You can integrate this API into your app or system to make predictions with your trained model.

Common Use Cases for Roboflow:

Object Detection

Detect and classify objects in images or videos for applications like autonomous vehicles, surveillance, and manufacturing.

Image Classification

Classify images into predefined categories, useful for tasks like medical image analysis or quality control in manufacturing.

Segmentation

Segment specific objects in images, such as in medical imaging or aerial photography for land usage analysis.

OCR (Optical Character Recognition)

Recognizing text in images or documents for automation purposes, like invoice processing or license plate recognition.

Activity Recognition

Detect and classify human activities in video streams, like monitoring fitness activities or detecting abnormal behaviors in security footage.

Roboflow in a Nutshell

No-Code Interface: A lot of the process can be done with no code, which is great for people who don't have much programming experience.

Flexible: It can be integrated with code for more advanced workflows.

Collaborative: Teams can work together on labeling and project management.

Powerful: Roboflow's tools support advanced capabilities like custom augmentations and cloud deployment.

https://lu.ma/roboflow (Weekly Webinars)

YOLO v12: https://www.youtube.com/watch?v=fksJmIMIfXo

Colab:

https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolov12-object-detection-model.ipynb?ref=blog.roboflow.com

https://cocodataset.org/#download

Open Images Datasets

https://storage.googleapis.com/openimages/web/index.html

Key Features of YOLOv8

- The key feature of YOLOv8 is its Extensibility, it is designed as a framework that supports all previous versions of YOLO, making it easier to switch between different versions and compare their performance.
- YOLOv8 includes a new back bone network, a new anchor free detection head and a new loss function.
- YOLOv8 is highly efficient and can run on variety of hardware platforms, from CPU's to GPU's.
- Anchor free detections are faster and more accurate than previous versions.

5

What are the reasons for using YOLOv8

Here are the few reasons why you should consider using YOLOv8 for your Computer Vision Project.

- YOLOv8 has a high rate of accuracy measured by COCO and Roboflow (COCO is the bench mark that people usually use in object detection)
- YOLOv8 outperforms other YOLO models in terms of speed and accuracy, the Mean Average Precision of 53.7 has been marked.
- YOLOv8 can be installed in two ways from the source and via pip. This is because it is the first iteration of YOLO to have an official package.
- New backbone network, new anchor free detection head and new loss function makes things much faster.
- YOLOv8 doesnot predict based on bounding box anchors which is what the other models used to do.

6

Yolo V8 is faster and more accurate than YOLOV7

Yolo-v8 in Colab:

Lab Tasks

YOLOv8 Use Cases

Sr. No.	Use Case	Challenge	YOLOv8 Application
1	Pothole Detection	Identifying potholes in various lighting and road conditions	Detect potholes from street images or video feeds
2	PPE (Personal Protective Equipment) Detection	Detecting different types of safety gear	Real-time monitoring of PPE compliance in workplaces
3	Pen and Book Detection	Detecting stationary or moving pens and books	Ensuring students have the required exam materials
4	Mobile Phone Usage during Exams	Identifying mobile phone usage in exam environments	Real-time detection of mobile phone usage during exams
5	Traffic Sign Detection	Detecting traffic signs under various weather conditions	Automated traffic sign detection for navigation systems
6	Vehicle Number Plate Recognition	Detecting number plates in poor lighting conditions	Automatic vehicle number plate recognition
7	Animal Monitoring in Forests	Detecting animals in dense forests	Tracking and monitoring wildlife using camera traps
8	Human Fall Detection	Recognizing sudden falls in elderly individuals	Real-time detection for safety and healthcare applications

9	Face Mask Detection	Detecting masks in crowded public places	Enforcing mask-wearing regulations in public spaces
10	Littering Detection	Identifying littering behavior in public areas	Monitoring public spaces for cleanliness violations
11	Fire Detection	Detecting fire or smoke in images or video feeds	Early detection of fire hazards for safety alerts
12	Intruder Detection in Homes	Identifying unauthorized people in restricted areas	Security surveillance to detect intruders
13	Crowd Counting	Estimating crowd density in public places	Real-time crowd monitoring for events or public safety
14	Mask Detection in Healthcare Settings	Ensuring masks are worn in hospitals and clinics	Compliance enforcement in healthcare environments
15	Smart Parking Detection	Identifying available parking spaces in real-time	Automated parking management systems
16	Dog Detection for Animal Shelters	Identifying dogs for adoption or health checks	Managing animal shelters through automated tracking
17	Fraudulent Activity Detection in Retail	Identifying shoplifting or fraud attempts	Real-time monitoring of retail stores for theft
18	Object Tracking in Sports	Tracking athletes or equipment in sports arenas	Real-time tracking of players and objects during sports
19	Construction Site Safety Monitoring	Identifying safety violations like unmarked hazardous areas	Detecting unsafe conditions on construction sites

20	Smart City Surveillance	Detecting suspicious activities in city environments	Monitoring city streets for criminal activities
21	License Plate Recognition for Toll Systems	Detecting vehicle plates in fast-moving traffic	Automated toll collection system based on vehicle plates
22	People Counting in Retail Stores	Accurately counting customers in stores	Optimizing store layouts and staffing
23	Inventory Management in Warehouses	Detecting missing or misplaced inventory	Automating inventory tracking and stock management
24	Autonomous Drone Navigation	Identifying obstacles and landmarks for drone navigation	Enabling autonomous navigation for drones
25	Industrial Equipment Monitoring	Detecting faulty equipment or machinery	Real-time fault detection in industrial equipment
26	Animal Poaching Detection	Identifying poachers in wildlife reserves	Monitoring wildlife reserves for illegal activity
27	Blood Donation Detection	Ensuring donors are eligible and safe for blood donation	Monitoring donor eligibility in medical centers
28	Package Detection in Logistics	Detecting packages and sorting them accurately	Automated package sorting in warehouses
29	Helmet Detection in Work Zones	Ensuring safety gear compliance in industrial environments	Monitoring workers for helmet usage in hazardous zones
30	Building Code Compliance Detection	Identifying safety violations in buildings	Enforcing building safety regulations

31	Human Activity Recognition in Videos	Recognizing various human activities (walking, running)	Activity recognition for surveillance and analysis
32	Food Quality Inspection in Factories	Detecting defects or contamination in food products	Automated quality control in food production lines
33	Smart Home Automation	Recognizing specific objects to trigger home automation	Detecting household objects for smart home systems
34	Road Damage Detection	Identifying road cracks or damage	Automated assessment of road conditions
35	Inspection of Airplanes	Identifying cracks or faults in airplane bodies	Automated inspections for aviation safety
36	Farm Crop Health Monitoring	Detecting pest infestations or disease in crops	Real-time monitoring of crop health in agriculture
37	Real-Time Object Detection in AR/VR	Detecting and overlaying objects in virtual environments	Enhancing AR/VR experiences with real-time object detection
38	Water Pollution Detection	Identifying contaminants in water bodies	Monitoring water bodies for pollution
39	Anti-Cheating Systems in Online Education	Detecting cheating behaviors during online exams	Monitoring student behavior to prevent cheating
40	Audio-Visual Equipment Detection	Identifying faulty or malfunctioning audio-visual equipment	Ensuring smooth operation of AV setups
41	Glove Detection in Healthcare	Ensuring medical personnel wear gloves during procedures	Monitoring PPE compliance in healthcare facilities

42	Construction Material Detection	Identifying construction materials on-site	Automating material management at construction sites
43	Emergency Vehicle Detection	Detecting emergency vehicles in traffic	Improving traffic management for emergency vehicles
44	Counterfeit Product Detection	Identifying counterfeit goods in markets	Real-time counterfeit product identification
45	Facial Recognition for Security	Identifying individuals for access control	Security systems for building or event access
46	Vehicle Accident Detection	Identifying accidents or collisions on the road	Automatic detection and alert system for accidents
47	Building Fire Escape Route Monitoring	Detecting blocked emergency exits	Ensuring fire safety in buildings
48	Hazardous Gas Detection	Detecting hazardous gases in industrial environments	Real-time detection of harmful gases in factories
49	Crop Harvesting Automation	Identifying ripe crops for harvest	Automating the harvesting process in agriculture
50	Smart Classroom Monitoring	Detecting student behavior and engagement in classrooms	Monitoring classroom activity to improve engagement