DEEP LEARNING MODEL FOR OBJECT DEFECT DETECTION

Revolutionizing Quality Control with AI

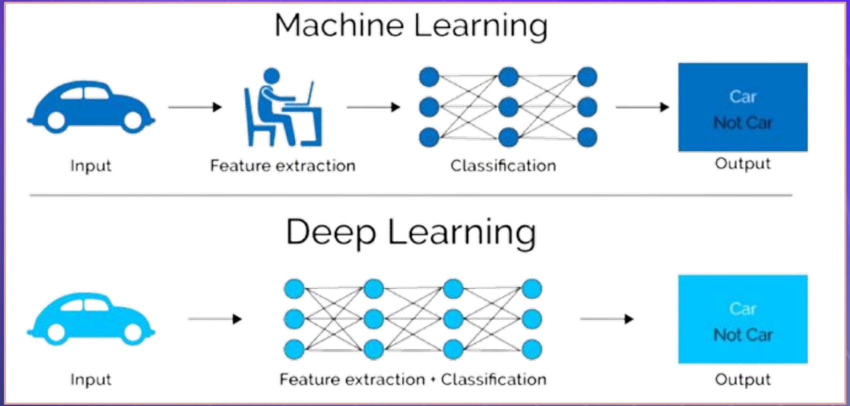
The Deep Dreamers
AIML
Ishan Naik
Kunal Pawar
Akshay Vennikkal
Anusha Yadav

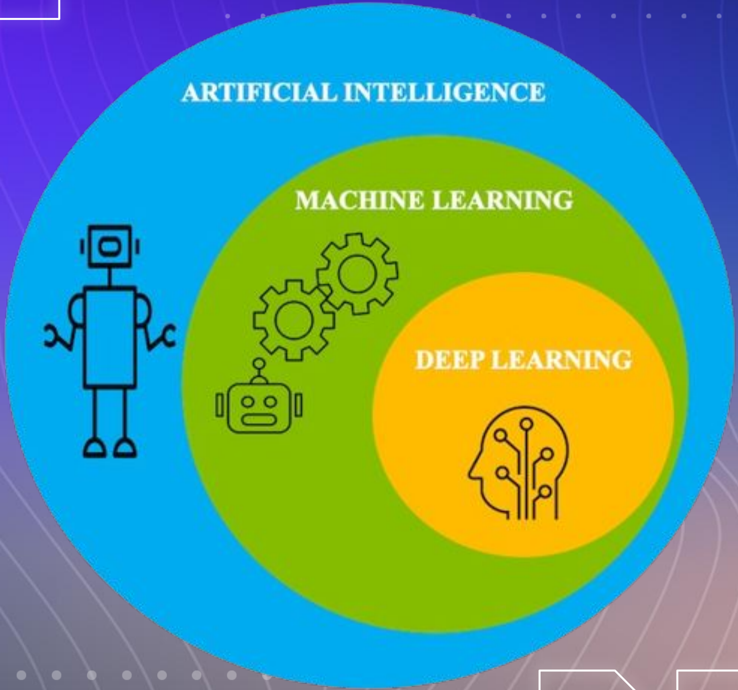
Introduction

- Deep Learning is a subfield of Machine Learning that focuses on artificial neural networks with multiple layers.
- These networks can automatically learn complex patterns and representations from large amounts of data.
- Deep Learning has gained popularity due to its ability to outperform traditional Machine Learning algorithms in tasks like image recognition, natural language processing, and speech recognition.

03

Overview of Deep Learning


Machine Learning
Car
Not Car
Input
Feature extraction
Classification
Output
Deep Learning
Car
Not Car
Input
Feature extraction + Classification
Output


ARTIFICIAL INTELLIGENCE
MACHINE LEARNING
DEEP LEARNING

Deep Learning

- Deep Learning algorithms are capable of automatically extracting features from raw data, unlike traditional Machine Learning algorithms which require manual feature engineering.
- This ability to learn features automatically has made Deep Learning algorithms very successful in tasks like object detection, where the features required to identify objects can be very complex and difficult to engineer manually.

04

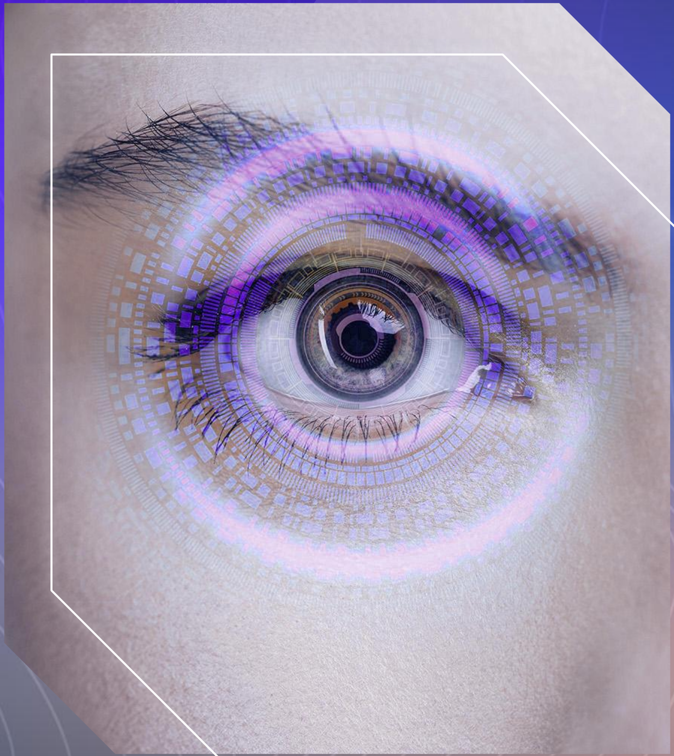
Introduction to YOLOv5

What is YOLO?

YOLO (You Only Look Once) is an object detection algorithm that can detect objects in an image in one pass. The YOLO algorithm consists of a convolutional neural network (CNN) that is trained on labeled images to learn representations of objects.

YOLO (You Only Look Once) a state-of-the-art object detection algorithm

- Unlike other object detection algorithms, YOLO divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell.
- YOLOv5 is an updated version of the YOLO algorithm that provides an easy-to-use interface for training and deploying custom object detection models.



05

YOLOv5 Architecture

YOLOv5 Architecture

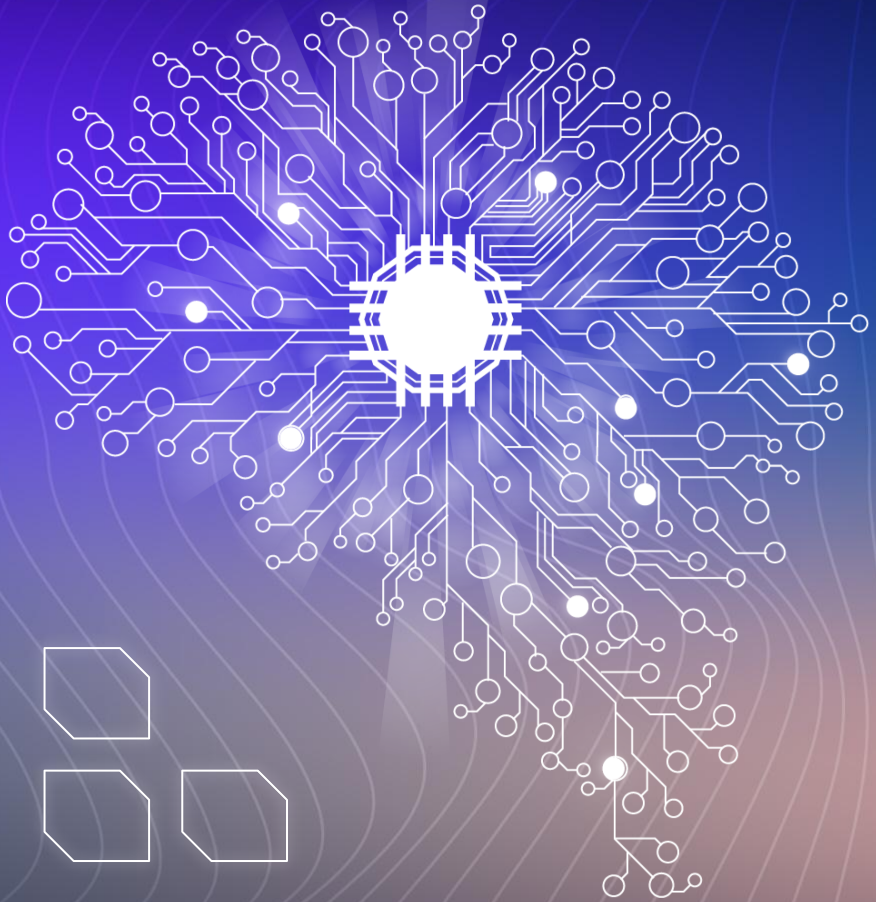
- YOLOv5 is built using PyTorch and supports various architectures like YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x.
- The architecture consists of multiple convolutional layers, residual blocks, and anchor boxes to detect objects at different scales and aspect ratios.
- Non-maximum suppression is used to refine the predicted bounding boxes and remove duplicates.

06

Model Training and Implementation

Implementation

- A custom dataset was created and labeled manually using the LabelImg tool.
- The dataset was used to train the YOLOv5 model with the specified training parameters, including image size, batch size, number of epochs, dataset file, pre-trained weights, and the number of worker threads.



Code Breakdown:

1. Installing Dependencies and Cloning Repositories:

```
In [ ]: !git clone https://github.com/ultralytics/yolov5

In [ ]: !cd yolov5 & pip install -r requirements.txt

In [2]: import torch
        from matplotlib import pyplot as plt
        import numpy as np
        import cv2
```

2. Loading the YOLOv5 Model:

```
In [87]: model = torch.hub.load('ultralytics/yolov5', 'custom', path='yolov5/runs/train/exp13/weights/best.pt', force_reload=True)
```

3. Training the YOLOv5 Model:

```
In [57]: !cd yolov5 && python train.py --img 640 --batch 16 --epochs 250 --data dataset.yml --weights yolov5s.pt --workers 8
```

4. Making Predictions on an Image:

```
In [88]: img = r"C:\Users\ishan\Downloads\TELEVISION NEW\s-1400.jpg"
         results = model(img)
         results.print()
```

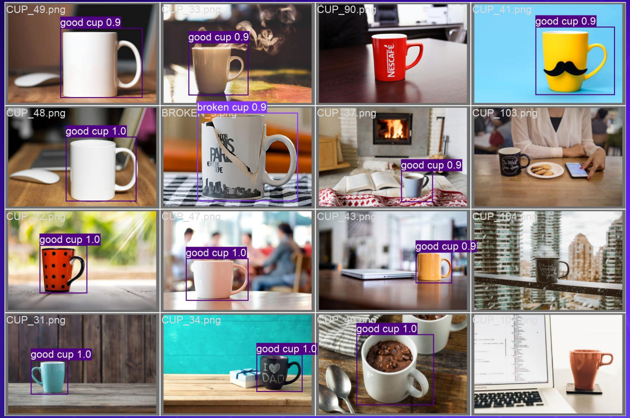
5. Real-time Object Detection on a Video Stream:

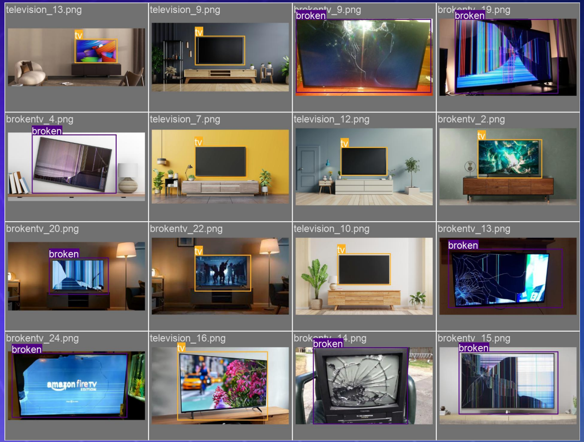
```
In [26]: cap = cv2.VideoCapture(2)
         while cap.isOpened():
             ret, frame = cap.read()

             # Make detections
             results = model(frame)

             cv2.imshow('YOLO', np.squeeze(results.render()))

             if cv2.waitKey(10) & 0xFF == ord('q'):
                 break
         cap.release()
         cv2.destroyAllWindows()
```

08 Results and Examples

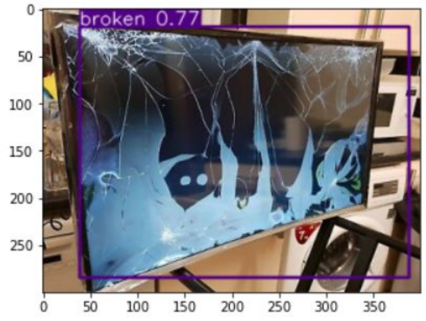
Results and Examples

- The YOLOv5 model achieved high accuracy in detecting defective cups and TVs, with little to no room for errors.
- The evaluation metrics, such as mAP (mean Average Precision) and IoU (Intersection over Union), demonstrate the model's performance.
- The model successfully detected defective cups in the provided images.
- Bounding boxes and confidence scores were displayed for each detected object





In [5]:
```
import os
os.environ["KMP_DUPLICATE_LIB_OK"]="TRUE" #QUICK FIX TO STOP KERNEL FROM DYING
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()
```



09 Conclusion

Conclusion

- The YOLOv5-based deep learning model successfully detected defective cups and TVs with high accuracy and speed.
- Its real-time object detection capability makes it suitable for a wide range of applications across various industries.
- While the model has limitations, it can be improved and adapted to meet specific requirements and challenges.

Thanks!

Do you have any questions?

ishannaik7@gmail.com
kunal4103@gmail.com
anushayadav6883@gmail.com
akshayvennikkal.av@gmail.com