International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 9 Issue 6 ǁ 2021 ǁ PP. 29-34

Object Detection – Trained YOLOv4

[1]Saish Kantak, [2]Prachi Kadam, [3]Kripa Sarvaiya, [4]Siddhi Keni and [5]Kalpesh Kubal.
[1,2,3,4]Student and [5]Lecturer.
[1,2,3,4,5]Thakur Polytechnic, Kandivali, Mumbai, Maharashtra, India

ABSTRACT: *Object detection is a pivotal ability required by most computer vision systems. The latest research in this field has been making tremendous development in many areas. Object detection and tracking have a variety of uses, this paper presents a general trainable framework for object detection in images and videos including live video. The detection technique we are using is based on YOLO. In this paper, we also discuss current and prospective applications of object detection in several fields. The results presented here suggest that this architecture can be further developed and used in face detection, face recognition, anomaly detection, crowd counting, security surveillance, etc.*

KEYWORDS- *object detection, YOLO, image processing, computer vision, tensor flow, machine learning, training models.*

Date of Submission: 25-05-2021 Date of acceptance: 07-06-2021

I. INTRODUCTION

During the last years, there has been a rapid and thriving expansion of computer vision research. Parts of this success have come from adopting and adapting machine learning methods, while others from the event of the newest representations and models for specific computer vision problems or from the event of efficient solutions. One field that has accomplished exceptional progress is object detection. Object detection is a technology affiliated with computer vision and image processing, this field deals with recognizing and identifying instances of specific objects of a chosen class (cars, humans, laptops, human faces, etc.) in digital images and videos. Object localization refers to identifying characteristics of one or more objects in an image or a video and drawing a bounding box around their extent. Object detection does the work of blending these two tasks and localizes and classifies one or more objects during a picture. When a user or practitioner refers to the term "object recognition ", they often mean "object detection" As we move towards more complete image understanding, having a more precise and detailed beholding becomes crucial. During this context, one cares not only about classifying images, but also about precisely estimating the category and site of objects contained within the photographs, a haul mentioned as object detection.

Object detection aims to detect all instances of objects from a known class, like people, cars, or faces during an image or a video. Generally, only a small number of instances of the object are present within the image, but there is a sizable number of possible locations and scales at which they're going to occur which require to somehow be explored. Each detection of the image is reported with the name of the object that's being detected, this is often as simple due to the position of the object, location, and scale, or the extent of the thing defined in terms of a bounding box. In different circumstances, the pose data is more detailed and holds the parameters of a linear or non-linear transformation. As an example, a face detector during face detection may compute the locations of the eyes, nose, and mouth, additionally to the bounding box of the face.

II. HISTORY OF OBJECT DETECTION

The earliest history of computer vision was way back within the 50s when two researchers, neuroscientists, Torsten Wiesel & David Hubel, published their work called "Receptive fields of single neurons within the cat's striate cortex". They conducted multiple experiments to understand how the mammalian brain functions. They took a cat which they did many experiments in this regard, they inserted electrons into a sedated cat and tried to work out how the cat's neurons fire concerning visual stimuli presented to the cat. Altogether the outcome of their early experiments was that easy, complex neurons exist. They won the award in 1981 for his or her understanding of the mammalian visual cortex. This was one of the earliest efforts in mammalian vision but played the base for computer vision. In 1959, there was another major development, which was by Russell Kirsch and his colleagues where for the first time they represented an image as a gaggle of 1s and 0s. Representing an image as a variety grid was an enormous achievement then, which are some things that we inherit today. Then between 1971 and 1978, there were lots of efforts that were attempted by researchers but

which didn't lead anywhere. This period was also noted as the "Winter of AI". Although, at that time many efforts were made on understanding and using shapes. Easily explained, trying to look at objects as a summation of parts. The parts were often solids, which were different types of skeletal parts of objects, which was a significant effort at that time. Importantly, there was also the world's first machine vision course offered by MIT's AI lab at that time within the 1970s. In the 1970s, collectively the first product of computer vision was developed, which was optical character recognition, developed by Ray Kurzweil, who was considered a visionary for the world of AI. It started with Eigenfaces for facial recognition which was a variant of Eigen categorization for doing face recognition. It happened in 1991 which was successful for face recognition with a minimum of detentions settings. There are also computational theories of object detection by Edelman that were proposed in 1997. Then came Perceptual grouping and Normalized cuts which was a milestone step for image segmentation methods that came in 1997. In 1998, Scale Invariant Feature Transform. Which were a vital image key point detector and depiction technique that was developed in the late 90s early 2000s. Then Viola-Jones face detection, again that came within the first 2000s. Conditional Random Fields which was an enhancement over Markov Random fields. Then Pictorial structures, the tactic proposed in 1973 was revisited in 2005 to further develop and improve upon, they came up with an improved statistical approach to be able to estimate the individual parts and their connections between parts. These were called pictorial structures at that time, they showed that they can train more and provides good performance for image matching. PASCAL VOC which can be an information set that's popular to the present day started in 2005 and around that time between 2005 to 2007, lots of methods for action recognition, vista recognition, site recognition also grew. Constellation models which were part-based probabilistic generator models also grew at that time to be able to again recognize objects in terms of parts and also the way the parts were put together within the entire. And deformable part models, an awfully popular approach, considered one of the key developments of the first decade of 2000 of the twenty-first century, came in 2009.

III. THEORETICAL WORKING

3.1 Tools.

Darknet- Darknet is an open-source neural network structure. It's a speedy and extremely specific, framework for real-time object detection where accuracy for custom trained model depends on training data, iterations, batch size, etc. The major reason it is quick because it is written in C and CUDA.

TensorFlow- TensorFlow is an end-to-end open-source platform for machine learning and numerical computation, the tactic of acquiring data, training models, serving predictions, and refining future results. TensorFlow brings together models and algorithms for Machine Learning and Deep Learning. It's originally based on C++ and uses Python as the front end.

TensorFlow is at the present the foremost popular software library. Multiple real-world applications of deep learning and Machine Learning make TensorFlow popular. Being an Open-Source library for deep learning and machine learning, TensorFlow finds a task to play in text-based applications, image recognition, voice search, and lots more. Deep Face, Facebook's image recognition system, uses TensorFlow for image recognition. it's employed by Apple's Siri for voice recognition. Every Google app that you simply use has made good use of TensorFlow to form your experience better.

OpenCV- OpenCV supports a good sort of programming languages like C++, Python, Java, etc., and is out there on different platforms including Windows, Linux, OS X, Android, and iOS. Interfaces for high-speed GPU operations supported by CUDA and OpenCL also are under active development.

In OpenCV, a video is often read either by using the feed from a camera connected to a computer or by reading a video file. the primary step towards reading a video file is to make a Video Capture object. Its argument is either the device index or the name of the input file to be read. In most cases, just one camera is connected to the system. So, all we do is pass '0' and OpenCV uses the sole camera attached to the pc. When quite one camera is connected to the pc, we will select the second camera bypassing '1', the third camera bypassing '2', and so on.

YOLO- YOLO is an algorithm that uses neural networks to supply real-time object detection. This algorithm is popular due to its speed and accuracy. it's been utilized in various applications to detect traffic signals, people, parking meters, and animals.

YOLO uses a special approach. YOLO may be a clever convolutional neural network (CNN) for doing object detection in real-time. The algorithm implements one neural network to the entire image, then breaks the image into sections and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the anticipated probabilities.

In YOLO, a CNN predicts multiple bounding boxes at one given time and probabilities for those boxes. It trains on real images and directly optimizes performance.

3.2 Working

Compared to other Region-Based Convolutional Neural Networks (fast RCNN) which perform detection on various regions and thus find yourself performing prediction multiple times for various regions in a picture or a video, Yolo's architecture is alike to FCNN hence YOLO passes the image (NXN) once through the FCNN and output (MXM) is the prediction. This architecture is splitting the input image in MXM grid and for every grid generation 2 bounding boxes and sophistication probabilities for those bounding boxes. Likely, the bounding box which represents the area of the detected object is larger than the calculated grid itself.

i. The YOLO Model:

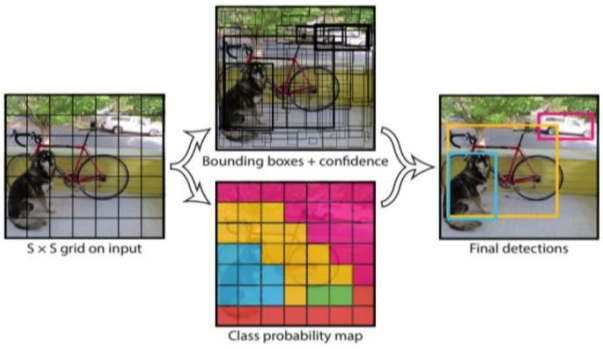

Figure 1: The Model.

YOLO treats object detection as an exclusive regression problem, straight from image pixels to bounding box coordinates and object probabilities. An individual convolutional network at one time predicts multiple bounding boxes and probabilities for those boxes. YOLO runs the detection on full images and undeviating optimizes detection performance. This joined model has various advantages over classical methods of object detection.

YOLO's system models detection as a regression problem. It divides the image into an X × X grid, for every grid cell predicts B bounding boxes, confidence for those boxes, and C object probabilities. Each grid cell predicts Bbounding boxes and confidence rates for these boxes. These confidence rates indicate how confident the model is that the box comprises an object and also how precise it thinks the box and the predicted objects are. Formally we define confidence as Pr(Object) ∗ IOU. If no object exists therein the cell, the arrogance scores should be zero. Otherwise, we might just like the arrogance score to equal the intersection over union (IOU) between the anticipated box and thus the bottom truth Each bounding box consists of 5 predictions: x, y, w, h, and confidence. The (x, y) coordinates represent the middle of the box relative to the bounds of the grid cell. What the width and height are, is predicted depending upon the entire image. Finally, the arrogance prediction represents the IOU between the anticipated box and any ground truth box. Each grid cell also predicts C conditional class probabilities, Pr(Classi |Object).|Each grid cell also predicts C conditional class probabilities, Pr(Classi |Object). These probabilities are transformed on the grid cell holding an object. We only predict one set of sophistication probabilities per grid cell, regardless of the quantity of boxes B.

ii. Network Architecture and Training:

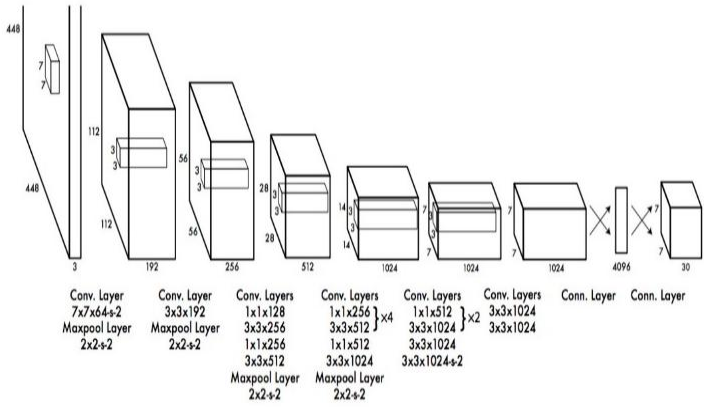

Figure 2: The Architecture

YOLO's interface has 24 convolutional layers followed by 2 entirely connected layers. It simply uses 1 × 1 reduction layers followed by 3 × 3 convolutional layers

FastYOLO practices a neural network with 9 convolutional layers instead of 24 and fewer filters in those layers. Leaving apart the volume of the network, all training and testing parameters are the same between YOLO and Fast YOLO.

YOLO is optimized for sum-squared error within the output of our model. It implements sum-squared error because it is easy to optimize, even though it doesn't align to maximize average precision. It weights localization error uniformly with classification error which is not prototypical. Also, in every image, many grid cells don't contain any object. This drives the "confidence" of many of those cells towards zero, often overwhelming the gradient from cells that do contain objects. this will cause model instability, causing training to diverge early. To change this, YOLO intensifies the loss from bounding box coordinate predictions and decreases the loss from confidence predictions for boxes that don't contain objects. YOLO uses two parameters, λcoord and λnoobj to achieve this. YOLO sets λcoord = 5 and λnoobj = .5.

The sum-squared error also equally weights errors in large boxes and small boxes. Its error metric should reflect that tiny deviation in large boxes matters but small boxes. To partially address this we predict the basis of the bounding box width and height instead of the width and height directly.

YOLO predicts multiple bounding boxes per grid cell. At the time of training, we only want individual bounding box predictors to be liable for each object. We assign one predictor to be "responsible" for predicting an object supported which prediction has the very best current IOU with the bottom truth. This results in specialization between the bounding box predictors. Each predictor gets more qualified at predicting specific sizes, aspect ratios, or classes of objects, improving overall recall.

IV. USE CASES.

i. Security surveillance

Surveillance is a fundamental element of security and safeguarding. Recent advances in laptop vision technology have junction rectifiers to the event of assorted automatic police work systems, but their effectiveness is adversely stricken by several factors, and that they aren't utterly reliable. Several police work cameras are put in however can't be closely monitored throughout the day. Since events are additionally doubtless to occur whereas the operator isn't looking, several vital events go undetected, even after they are recorded. Users cannot be expected to trace through hours of video footage, particularly if they are not positive or sure about what they are searching for.

ii. Crowd counting

Crowd tally is another valuable application of object detection. For densely inhabited areas like theme parks, malls, city squares, analyzing store performance or crowd statistics throughout festivals. These tend to be harder as folks move out of the frame quickly (also as a result of folk's area unit non-rigid objects). Object detection will facilitate businesses and municipalities a lot of effectively live completely different sorts of traffic—whether on foot, in vehicles, or otherwise.

iii. Online Examination

With the arrival of COVID-19, remote learning has blossomed. faculties and universities have stopped working however they switched to applications like Microsoft groups/teams to complete their tutorial years. However, there has been no resolution to examinations. Some have modified it to the associate degree assignment kind where students will simply copy and paste from the web, whereas some have simply cancelled them outright. If the method we tend to reside is to be the new norm there has to be some solution to this problem. So, as we all know online exams have their own set of challenges as well as observing every student whereas he/she is giving exams as there are not any during a controlled atmosphere this software package can solve that issue.

iv. Anomaly detection

Anomaly detection is applicable in an exceeding form of domains, like intrusion detection, fraud detection, fault detection, system health monitoring, event detection in sensor networks, detecting ecosystem disturbances, and defect detection in images using machine vision. As a result, manufacturing costs are reduced thanks to the avoidance of manufacturing and marketing defective products. Anomaly detection, in factories, could be a useful gizmo for internal control systems due to its features.

v. Face Detection and Face Recognition

Face detection and Face Recognition are widely employed in computer vision tasks. We noticed how Facebook detects our face once you upload a photo. This is often a straightforward application of object detection that we see in our everyday life. Face detection may be considered a selected case of object-class detection. In object-class detection, the task is to search out the locations and sizes of all objects in a picture that

belongs to a given class. Examples include upper torsos, pedestrians, and cars. Any facial feature variations within the database will overturn a similar process. Face recognition describes a biometric technology that goes way beyond recognizing when somebody's face is present. It attempts to determine whose face it's. Face-detection algorithms specialize in the detection of frontal human faces. It's comparable to image detection throughout which the image of an individual is matched bit by bit. Image matches with the image stored within the database. Any facial feature variations within the database will overturn a similar process.

V. PRACTICAL WORKING DEMONSTRATION.

We executed our program on various videos and images which are displayed the respective results from **Fig 3 - Fig 6**

Figure 3: Before and after we ran our project.

Figure 4: Before and after we ran our project.

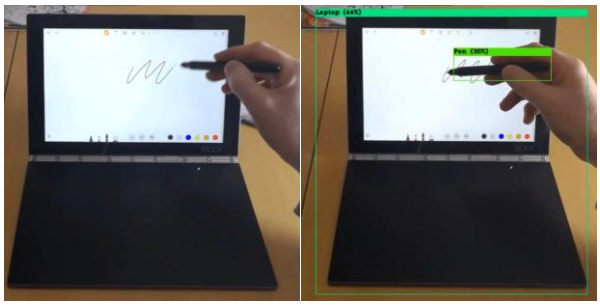

Figure 5: Before and after we ran our project.

Figure 6: Before and after we ran our project.

This project is python-based and evaluated on distinct video sequences and runs with steady FPS. It works in a way that it detects the classes which we trained beforehand. The input file is broken down into the total number of frames and passes each image to the object detector, if the input file is a video or an image it will be saved in a pre-decided folder, if the input is a live video input, the results will be displayed on the screen.

While conducting object detection, video is divided into multiple frames and each frame, as well as a video output, is saved with detection data obtained for each input video after using YOLO. The output is provided as bounding boxes with the class name and confidence scores above the bounding box.

VI. CONCLUSION

In this paper, object detection is done on videos and images by training detector for a custom dataset consisting of 10000 images for 12 specified classes. The object detection is done using YOLO. Accuracy and precision can be controlled by training the system for more iterations and fine-tuning the training dataset. For Future work, the system can be trained for more classes or more types of objects as it can be used for different domains of videos and different objects can be detected. Our detection system includes Book, Bottle, Car, Computer mouse, Human face, Laptop, Mobile phone, Pen, Person, Picture frame, Weapon, as a class/objects, this can be expanded to more multiple objects or can be dedicated for a specific object with a varying number of datasets.

REFERENCES

[1]. Scaled-YOLOv4: Scaling Cross Stage Partial Network
[2]. YOLO: Real-Time Object Detection- pjreddie.
[3]. You Only Look Once: Unified, Real-Time Object Detection- Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi
[4]. YOLO9000: Better, Faster, Stronger -Joseph Redmon, Ali Farhadi
[5]. Applications of Object Detection System -Abdul Vahab, Maruti S Naik, Prasanna G Raikar, Prasad SR
[6]. Anomaly Detection with Computer Vision -Heimer Rojas, Mia Morton, Abdel Perez, Ximena Andrade.
[7]. Visual Object Detection and Tracking using YOLO and SORT -Akansha Bathija, Prof. Grishma Sharma.
[8]. Object Detection Guide - fritz