An Overview of Different Object Detection Algorithms and Libraries

Nisha Bhadauriya Agarwal[OrcidID: 0009-0003-2422-7250], Ph.D. Scholar, Sage University, Indore, India & nishabhadauriya@yahoo.com

Dr. Deepak Kumar Yadav[OrcidID: 0009-0000-1226-487X], Associate Prof., Sage University, Indore, India & deepak_ku_yadav@outlook.com

Abstract — In the fields of computer vision and machine learning, object identification algorithms have undergone a swift and revolutionary transformation. One of the most difficult subjects in computer vision is its participation in both object location and object classification. In computer vision and artificial intelligence, object detection is a critical field that allows computers to identify and locate things in pictures and movies. They identify things and create bounding boxes around them using deep learning and machine learning models, such as Convolutional Neural Networks (CNNs). One of the main responsibilities of the deep learning discipline is computer vision. Identifying the presence of an object in a picture or video is known as object detection. There are numerous applications where object detection can increase work efficiency. Applications for object detection include industrial work, robotics, self-driving automobiles, sports, people counting, agriculture, traffic monitoring, military defense systems, and many more fields. Many methods, including YOLO, RetinaNet, Single Shot detector (SSD), R-CNN, Faster R-CNN, HOG, and others, can be used for object detection. Additionally, a summary of several object detection libraries is provided in this work.

Keywords — object detection, artificial intelligence, computer vision, you only look once, convolutional neural networks, CNN, YOLO

I. INTRODUCTION

An essential component of computer vision phenomena is object finding. By distinguishing between objects in images and videos according to their kinds, it can identify portions. One of the greatest achievements of deep learning and image processing is object detection, which locates and recognizes objects in images. Bounding boxes are a frequent tool used in the process of localizing things. An object detection model is flexible in that it may be trained to recognize and detect multiple distinct objects. Generally, object detection models are taught to identify particular things while they are present. The built models can be applied to pictures, movies, or real-time processes. Object identification has a wide range of applications, even before deep learning techniques and contemporary image processing technologies. In the field of object detection, there were comparatively few rivals, and certain methods (such as SIFT and HOG with their feature and edge extraction techniques) shown success.

Finding examples of a known class of real-world objects in photos or videos—such as vehicles, bikes, TVs, remote controls, mobile devices, flowers, fruits, and people—is known as object finding. It identifies, picks out the object region, and finds several things in an image, giving us a description of what's there. There may be a vast number of potential places of varying sizes at which the object can be found and that need to be located regardless of how few objects there are in the image. As object finding's needs grow and its performance improves daily, it has become one of the most important study topics being studied today. There are many methods which has been developed and used.

In the current generation, object detection has become considerably more prevalent with the development of convolutional neural networks (CNNs) and the adaptation of computer vision technologies. Seemingly limitless possibilities arise with the new wave of object detection using deep learning algorithms. item detection uses each class's distinct and special attributes to find the needed item. The object detection model can search for perpendicular corners that produce a square form with equal-length sides while searching for square shapes. The object detection model searches for central locations from which the formation of a specific round entity is feasible when searching for a circular object. These identification methods are applied to object tracking and facial recognition.

 DOI : 10.35291/2454-9150.2024.0318

II. WHERE IS OBJECT DETECTION USED?

We already use object detection in many aspects of our daily lives. For instance, when facial detection unlocks your smartphone. Or it detects suspicious activity in video monitoring of shops or warehouses. The primary uses of object detection include:

- **Number plate identification:** This technique identifies the alphanumeric characters on a car by combining object detection with optical character recognition (OCR) technology. Object detection is a useful tool for taking pictures and identifying cars in certain shots. The OCR technology begins translating the two-dimensional data into machine-encoded text as soon as the model recognizes the license plate.
- **Face detection and recognition** – As was previously mentioned, this is one of the main uses of object detection. We are able to identify human faces in an image or video by using contemporary algorithms. Thanks to one-shot learning techniques, it is now even feasible to identify faces from a single trained image.
- **Object tracking** – The ball may strike far away whether you're viewing a baseball or cricket match. It's a good idea to monitor the ball's motion and distance travelled in these scenarios. Object tracking can guarantee that we have real-time data on the ball's direction of travel for this reason.
- **Self-driving cars** – When operating an autonomous vehicle, it's imperative to pay attention to the various surroundings. The effective operation of autonomous cars depends on an object detection model that has been trained on several classes to identify various entities.
- **Robotics** – Robots carry out a lot of duties that need real-time work, such as pick-and-place operations, heavy lifting, and other tasks. Robots cannot detect objects or do tasks automatically without object detection.

III. RELATED WORK

Prior approaches to object detection relied on human expertise through the use of image operators and expert knowledge. However, with the development of convolutional neural networks (CNN), the task has become considerably more computationally efficient. CNN is a demonstration model that uses deep learning to locate objects inside images [25]. The findings of the deep learning object detection approaches outperform those of the previous methods. The VGG16 (Visual Geometry Group 16) is another name for this convolutional neural network (CNN) model. CNN layers are referred to as feature maps. This feature map features distinct pixel values and is a convinced mixed image. Applying pooling and filtering directs the layer's transformation. The filtering method supplies the values to a nonlinear function by transforming the obtained weights with the values of an approachable ground of blocks.

Adding regions with CNN features (R-CNN) improves the object detection methodology even further, and it works completely differently from earlier approaches. The area using CNN features approaches underwent significant changes, first with R-CNN, then with fast R-CNN, and finally with faster R-CNN [26]. SSD and YOLO are methods linked to regression classification. These techniques are intended to examine object detection difficulties and provide roughly accurate interpretations.

A computer-based, completely automated technique for identifying common weapons was developed by (Narejo et al., 2021). In order to identify weaponry, the YOLOv3 object detection model was used. The training results verified that YOLOv3 performed better than CNN and YOLOv2. YOLOv3, however, has significant localization issues.

A real-time WD system utilizing Scaled-YOLOv4 was presented by Ahmed et al. in 2022. TensorRT network on a popular edge computing device was used to improve the model. The study revealed that the model produced cheap cost and better Mean Average Precision (mAP) scores. False positives were prevalent, though. Conference Web E3S 391, 01071 (2023) [44] 10.1051/e3sconf/202339101071, available at https://doi.org DL and fog computing were combined with software-defined networking for WD in video surveillance systems by ICMED-ICMPC 2023 (Fathy & Saleh, 2022). The DL model was employed inside the artificial intelligence model to extract the video frames containing weapons. It was limited by expensive technology, even though it was able to identify concealed weapons.

The goal of (Ashraf et al., 2022) was to lower the quantity of WD false positives and false negatives. Area of Interest and YOLO formed the foundation of the structure. The outcomes demonstrated that YOLO-v5 was able to obtain a high detection speed and recall rate. However, in several situations, the high recall rate was unable to identify weaponry.

(Bhatti et al., 2021) concentrated on exploiting CCTV footage to identify dangerous weapons. The pistol class was used as the reference class for the binary classification, and the related confusion objects inclusion notion was put forth. The system was designed for metal identification; therefore, it cannot identify guns made of other materials.

A method for improving the brightness of knife detection in surveillance systems was proposed by (Galab et al., 2021). A criterion for assessing the brightness of the frame was suggested by the research. The outcomes validated the technique's efficacy in identifying knives.

But the study did not provide a thorough method for

 DOI : 10.35291/2454-9150.2024.0318

locating other dangerous items, such as firearms, handguns, etc.

A novel automated model was presented by (Kalla & Suma, 2022) for efficient WD in CCTV. For classification, a support vector machine was employed. The results demonstrated the great degree of accuracy of the model. To train on the data, the built-in model, however, required a significant amount of processing power.

IV. OBJECT DETECTION ALGORITHMS

The quality of algorithms used to solve object detection has been continuously improving since deep learning became popular in the early 2010s. We'll examine the most widely used algorithms while comprehending their advantages, working theory, and shortcomings in specific situations.

A. Histogram of Oriented Gradients (HOG)

One of the earliest techniques for object detection is the Histogram of Oriented Gradients. In 1986, it was initially made available. Although there were some advancements in the next ten years, the method did not become well known until 2005, when it began to be used to a wide range of computer vision problems. A feature extractor is used by HOG to locate items in a picture. The feature descriptor that HOG uses is a depiction of an area of an image from which we extract only the information that is absolutely required, discarding everything else. The purpose of the feature descriptor is to transform the image's overall size into an array or feature vector. The gradient orientation process is used in HOG to locate an image's most important areas.

1) What is HOG Method?

The histogram of oriented gradients method is a feature descriptor technique used in computer vision and image processing for object detection. It focuses on the shape of an object, counting the occurrences of gradient orientation in each local region. It then generates a histogram using the magnitude and orientation of the gradient [28].

2) Architecture Overview of HOG

The underlying architecture of Histogram of Gradients is as shown in Fig. 1 [8] below.

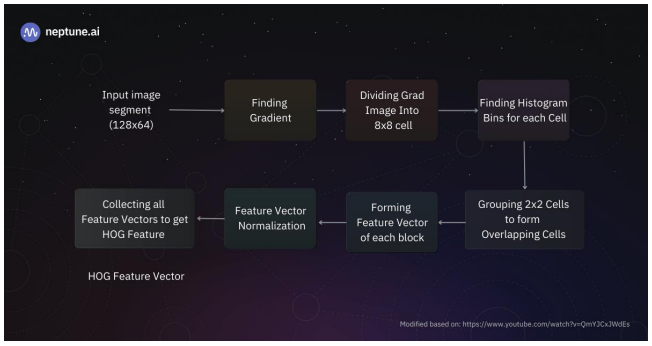

Fig.1 HOG – Object Detection Algorithm [8]

We need to know how HOG functions before we get a better understanding of its general architecture. The feature vectors for a given pixel in an image are obtained by computing the gradient's histogram, which takes into account both the vertical and horizontal values. By investigating the other entities in their horizontal and vertical surrounds, we may obtain a clear value for the present pixel with the aid of the gradient angles and magnitude.

We'll look at an image segment of a specific size, as indicated in the image representation above. The first step is to divide the full picture computation into gradient representations of 8x8 cells in order to identify the gradient. The 64 gradient vectors that are obtained allow us to divide each cell into angular bins and calculate the area's histogram. By using this method, 64 vectors are shrunk to a reduced size of 9 values. Upon determining the dimensions of the 9-point histogram values (bins) for every cell, we have the option to generate overlaps between the cell blocks. Forming the feature blocks, normalizing the resulting feature vectors, and gathering all of the feature vectors are the last steps in obtaining an overall HOG feature. [8] and [9].

3) HOG Accomplishments

1. The development of a feature descriptor that helps in object detection.
2. Sliding window effect creation for every position calculation.
3. The capacity can be used in conjunction with support vector machines (SVMs) to provide extremely accurate object recognition.

4) Considerations

a) Limitations

In the early days of object detection, the Histogram of Oriented Gradients (HOG) was a very novel technique, but it had many drawbacks. It is inefficient in some object detection scenarios with smaller spaces and highly time-consuming for complicated pixel calculation in photos.

b) When to use HOG?

When testing the performance of different algorithms for object detection, HOG should frequently be the first method employed. Nevertheless, HOG is useful for most object detection and can recognize facial landmarks with a respectable degree of accuracy.

c) Usage Examples

One of the popular use cases of HOG is in pedestrian detection due to its smooth edges. Other general applications include object detection of specific objects.

 DOI : 10.35291/2454-9150.2024.0318

B. Region-based Convolutional Neural Networks (R-CNN)

The object detection process is improved by the region-based convolutional neural networks compared to the earlier HOG and SIFT techniques. Using selective features, we attempt to extract the most important features (typically 2000 features) from the R-CNN models. A selective search algorithm capable of achieving these more substantial regional offers can aid in the computation of the process of choosing the most significant extractions.

1) CNN Architecture

ConvNets, short for Convolutional Neural Networks, are a particular kind of deep learning algorithm that are mostly used for tasks requiring object recognition, such as picture categorization, detection, and segmentation. CNNs are used in many real-world applications, including security camera systems and driverless cars, among others. The CNN's fundamental architecture is depicted in the following Fig. 2 [8].

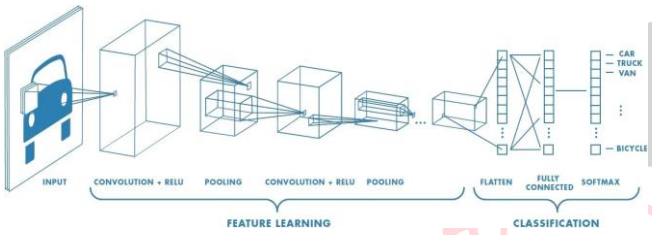

Fig. 2 CNN Architecture [15]

Fig. 3 below illustrates the layered architecture of the human visual cortex, which served as the model for convolutional neural networks. Despite these contrasts, CNNs have been essential in driving breakthroughs in computer vision since they imitate the human visual system but are simpler, lack the intricate feedback processes, and rely on supervised learning instead of uncontrolled learning.

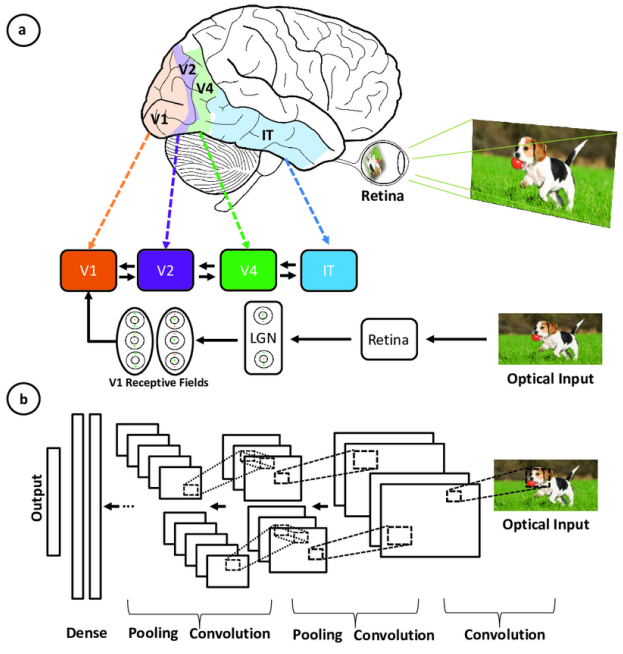

Fig. 3 CNN Architecture [45]

Robust frameworks such as Tensorflow, Pytorch, and Keras, which facilitate the training of convolutional neural networks and other deep learning models, are largely responsible for the explosive rise of deep learning.

2) R-CNN's operational mechanism

R-CNN's operational mechanism is shown below in Fig. 4 [10]. The *R-CNN* first extracts many (e.g., 2000) *region proposals* from the input image (e.g., anchor boxes can also be considered as region proposals), labeling their classes and bounding boxes (e.g., offsets).

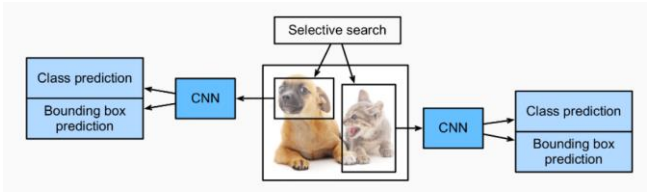

Fig. 4 R-CNN – Object Detection Algorithm [10]

The selective search method operates by first ensuring that you generate several sub-segmentations on a given image and then choosing the potential entries for your task in order to identify the most significant regional suggestions. Once the smaller segments have been appropriately combined into bigger segments, the greedy algorithm can be used to combine the effective entries for a recurrent process.

Extracting the features and generating the necessary predictions are our following steps after the selective search algorithm runs successfully. After that, we can present our top picks, and the convolutional neural networks will produce an output feature vector that is n-dimensional (2048 or 4096). We can easily complete the feature extraction task with the aid of a convolutional neural network that has already been trained. The R-CNN's last stage is to identify the corresponding bounding box by applying the proper predictions for the image. Each task's predictions are computed using a classification model in order to yield the best results, and the bounding box classification for the suggested regions is corrected using a regression model.

3) Problems with R-CNN

- Although the pre-trained CNN models yield good results for feature extraction, the present algorithms' extraction process is incredibly long when it comes to obtaining all of the region recommendations and, eventually, the best regions.
- The R-CNN model's high prediction time and slow training rate are two more significant drawbacks. Large computational resources are needed to solve the problem, which raises the process's overall viability. Therefore, it may be said that the architecture as a whole is rather costly..

 DOI : 10.35291/2454-9150.2024.0318

- The lack of improvements available in this specific step might sometimes lead to the selection of poor candidates at the outset. Several issues with the trained model may result from this..

4) Considerations

a) When To Use R-CNN?

The performance of the object detection models must first be tested using R-CNN, which is comparable to the HOG object detection technique. The most recent iterations of R-CNN are typically favoured since the processing time for object and image predictions can occasionally take longer than expected.

a) Usage Examples

R-CNN can be applied in a variety of ways to address various object identification task types. For instance, using a drone camera to track objects, identifying text in an image, and turning on object detection in Google Lens.

C. Faster R-CNN

Even while the R-CNN model could compute object detection and produce acceptable results, there were a few rather unimpressive aspects, chief among them being the model's speed. Therefore, in order to solve the challenges with R-CNN, faster approaches to solving some of these problems had to be developed. First, the Fast R-CNN was developed to address some of the R-CNN's earlier problems. Rather than taking into account each subsegment, the entire image is run through the pre-trained Convolutional Neural Network in the fast R-CNN approach. By using a selective search algorithm and two inputs from the pre-trained model, the region of interest (RoI) pooling technique creates an output for a fully connected layer. We will learn more about the Faster R-CNN network in this section, which is an advancement over the Fast R-CNN model.

1) Faster R-CNN's operational mechanism

An R-CNN's primary source of performance bottleneck is its ability to propagate CNN forward independently for each region proposal without sharing computation. Due to the frequent overlaps in these locations, separate feature extractions result in a high computing overhead. CNN forward propagation is limited to the full image, which is one of the primary advantages of the rapid R-CNN over the R-CNN (Girshick, 2015). Figure 5 [10] provides a description of the Faster R-CNN model.

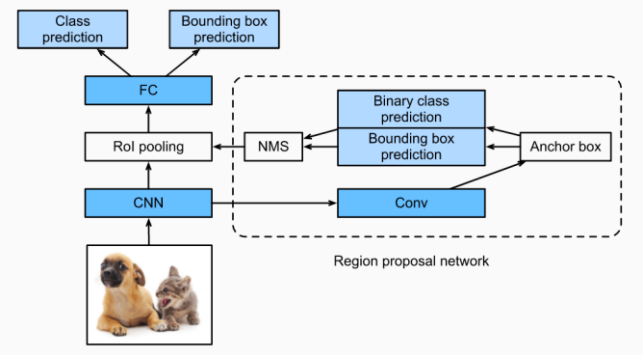

Fig. 5 Faster R-CNN – Object Detection Algorithm [10]

Faster R-CNN model is one of the greatest R-CNN models, which significantly increases performance speed over its predecessors. While the R-CNN and Fast R-CNN models compute the region proposals using a selective search algorithm, the Faster R-CNN approach substitutes an improved region proposal network for this current technique. To generate useful outputs, the region proposal network (RPN) computes images from a variety of sources and scales.

The margin computation time is decreased by the regional proposal network, often by 10 ms per picture. The convolutional layer of this network is what gives us access to each pixel's key feature mappings. We offer several anchor boxes with various sizes, aspect ratios, and scales for every feature map. We anticipate the specific binary class for every anchor box and create a bounding box for it. Since many overlaps are formed when constructing feature maps, the resulting data is then run through the non-maximum suppression to remove any unneeded data. After the region of interest is traversed by the non-maximum suppression output, the remaining steps and computations are comparable to how Fast R-CNN operates.

2) Considerations

a) Limitations of Faster R-CNN

The Faster R-CNN method's primary drawback is the length of time it takes to propose various things. Occasionally, the speed is dependent upon the kind of system being utilized.

b) When To Use Faster R-CNN?

When compared to other CNN techniques, the prediction time is faster. Although R-CNN typically requires 40–50 seconds to predict items in an image, Fast R–CNN only requires 2 seconds, and Faster R–CNN provides the best result in around 0.2 seconds.

c) Usage Examples

Use case examples for Faster R-CNN are comparable to those outlined in the R-CNN methodology. Faster R-CNN, on the other hand,

 DOI : 10.35291/2454-9150.2024.0318

allows us to complete these activities more efficiently and productively.

D. Single Shot Detector (SSD)

One of the quickest methods for completing object identification tasks in real time is the single-shot detector for multi-box predictions. Although the Faster R-CNN approaches can produce predictions with high accuracy, the entire process takes a long time and necessitates doing the real-time task at a rate of roughly 7 frames per second, which is not optimal. This problem is resolved by the single-shot detector (SSD), which increases the frames per second to nearly five times higher than the Faster R-CNN model. Instead of using the region proposal network, it uses default boxes and multi-scale characteristics.

1) Architecture Overview

The SSD method, exemplified in Fig. 6 below [12], relies on a feed-forward convolutional network to generate a fixed-size set of bounding boxes and scores for the existence of object class instances in those boxes. The final detections are then generated by a non-maximum suppression step. The early network layers (truncated before any classification layers) are based on a standard architecture used for high quality image classification. The SSD model extends a base network by adding many feature layers that forecast offsets to default boxes with varying sizes, aspect ratios, and confidence levels. A 300 x 300 SSD performs noticeably better than a 448 × 448 SSD. There are primarily three parts to the single-shot multibox detector architecture. All of the important feature maps are chosen during the feature extraction step, which is the first stage of the single-shot detector.

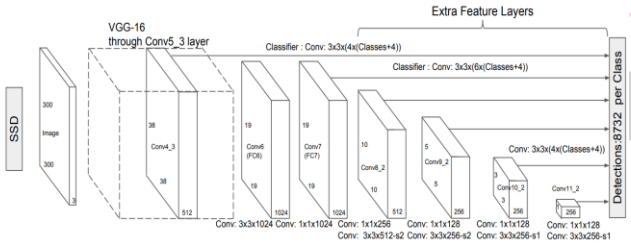

Fig. 6 SSD – Object Detection Algorithm [12]

This architectural zone has no further layers; only completely convolutional layers exist. Head detection is the next stage after extracting all of the important feature maps. Fully convolutional neural networks are also used in this step. Finding the images' semantic significance is not the goal of the second step of detection heads, though. Rather, the main objective is to provide the best bounding maps possible for each feature map. The last step is to run it through the non-maximum suppression layers to lower the error rate brought on by repeated bounding boxes after we have computed the two crucial phases.

2) Limitations of SSD

Even while the SSD greatly increases performance, the photos' resolution is lowered to a worse quality. For small-scale objects, the SSD architecture usually performs worse than the Faster R-CNN.

3) Considerations

a) When To Use SSD?

The single-shot detector is often the preferred method. The main reason for using the single-shot detector is because we mainly prefer faster predictions on an image for detecting larger objects, where accuracy is not an extremely important concern. However, for more accurate predictions for smaller and precise objects, other methods must be considered.

b) Usage Examples

The Single-shot detector can be trained and experimented on a multitude of datasets, such as PASCAL VOC, COCO, and ILSVRC datasets. They can perform well on larger object detections like the detection of humans, tables, chairs, and other similar entities.

E. YOLO (You Only Look Once)

One of the most widely used model architectures and object detection techniques is (YOLO). The YOLO architecture is typically the first idea that pops up when searching Google for object detection algorithms. We'll talk about the various iterations of YOLO in the parts that follow. One of the greatest neural network architectures is used by the YOLO model to achieve high processing speed and accuracy. The primary factor contributing to its appeal is its quickness and accuracy. To accomplish object detection, the YOLO architecture makes use of three main terms. To understand why this model performs so fast and accurately when compared to other object detection algorithms, it is important to comprehend these three strategies.

1) Operating mechanism of YOLO

YOLO improves detection performance by immediately training on complete photos. Comparing this unified model to other object identification techniques, there are significant advantages. Yolo is incredibly quick and easy. Since YOLO views the full image during training and testing, unlike sliding window and region proposal-based methods, it implicitly stores contextual information about classes in addition to their appearance. YOLO architecture can be seen in Fig. 7 below. Because it lacks context awareness, Fast R-CNN, a leading approach for object detection, misinterprets background patches in an image for actual objects. YOLO produces fewer than half of the

 DOI : 10.35291/2454-9150.2024.0318

background errors that Fast R-CNN does. YOLO acquires representations of objects that are generalizable. YOLO performs significantly better than leading detection techniques like DPM and R-CNN when trained on natural photos and evaluated on artistic creations. YOLO is less likely to malfunction when applied to new domains or unexpected inputs since it is extremely generalizable.

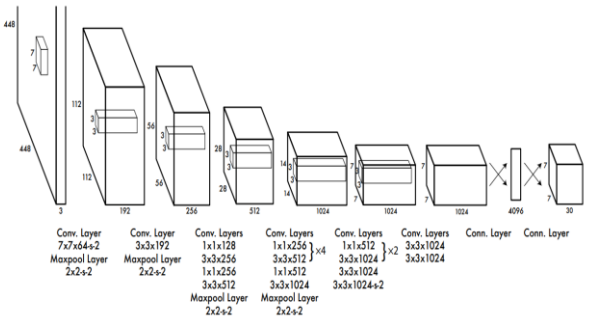

Fig. 7 YOLO – Object Detection Algorithm [13]

Residual blocks are the first idea in the YOLO model. In the first architectural design, grids were created in the specific image by using 7×7 leftover blocks. These grids all serve as focal centers, and specific predictions are drawn for each grid in accordance with that function. In the second method, while creating the bounding boxes, every central point for a given forecast is taken into account. Although the classification tasks are effective for every grid, the process of separating the bounding boxes for every forecast is more intricate. Utilizing the intersection of union (IOU) to determine the ideal bounding boxes for the specific item identification task is the third and last method.

2) Benefits of YOLO

- Compared to the majority of other training techniques and object identification algorithms, YOLO has a very high computing and processing speed, particularly in real-time.
- The YOLO architecture enables the model to learn and develop an understanding of various objects more efficiently.
- In addition to its quick processing speed, the YOLO algorithm manages to deliver an overall high accuracy with the elimination of background mistakes seen in other methods.

3) Limitations of YOLO

Inability to identify smaller things in a picture or video due to a decreased recall rate. Due to bounding box restrictions, it is unable to recognize two items that are extremely close to each other.

4) YOLO versions

Among the most popular and effective object detection techniques is the YOLO architecture. Following the release of the YOLO architecture in 2016, the YOLO v2 and v3 versions debuted in 2017 and 2018, respectively. 2019 saw no new releases, but 2020 saw three rapid releases: PP-YOLO, YOLO v5, and YOLO v4. Every subsequent iteration of YOLO represented a minor advancement over its predecessors. In order to guarantee that object detection could be supported on embedded devices, the small YOLO was also released.

5) Considerations

a) When To Use YOLO?

The YOLO architecture is one of the most recommended techniques for real-time object recognition, even though all of the previously stated methods work very well on photos and occasionally video analysis. Depending on the device you are using to execute the application on, it can accomplish most real-time processing jobs with a reasonable speed and number of frames per second while maintaining excellent accuracy.

b) Usage Examples

Aside from object identification on a variety of items, other common use cases of the YOLO architecture include person, animal, and vehicle detection.

F. RetinaNet

When it comes to single-shot object detection, the RetinaNet model, which was released in 2017, quickly rose to the top and was able to outperform other widely used object detection algorithms at the time. Upon the debut of the RetinaNet Architecture, the Yolo v2 and SSD models could not match its object detection capabilities. It was able to match the R-CNN family in terms of accuracy while keeping the same speed as these models. For these reasons, the RetinaNet model is widely used for object detection in satellite photography.

1) Architecture Overview

The design of the RetinaNet architecture somewhat balances out the drawbacks of earlier single-shot detectors to yield more effective and efficient outcomes. The focal loss in the preceding models is swapped out for the cross-entropy loss in this model design. The class imbalance issues in architectures such as SSD and YOLO are addressed by the focused loss. Three key components combine to form the RetinaNet model.

A feedforward ResNet architecture (a) is layered with a Feature Pyramid Network (FPN) [46] backbone to create a rich, multi-scale convolutional feature pyramid (b) in the one-stage RetinaNet network architecture depicted in Fig. 7. RetinaNet connects two subnetworks to this backbone: one (c) for classifying anchor boxes, and another (d) for regressing anchor boxes to ground-truth object boxes.

 DOI : 10.35291/2454-9150.2024.0318

Because the network architecture is purposefully kept simple, our work can concentrate on a novel focal loss function that, when implemented at a faster speed, closes the accuracy gap between our one-stage detector and the most advanced two-stage detectors, such as Faster R-CNN with FPN [46].

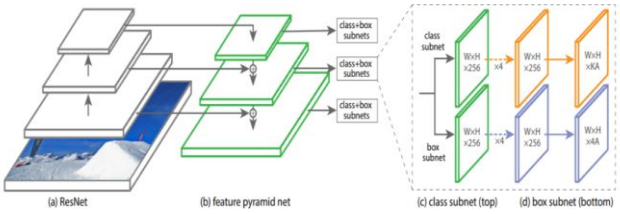

Fig. 8 RetineNet – Object Detection Algorithm [14]

Three components are used in the construction of RetinaNet: the focal loss, the feature pyramid network (FPN), and the ResNet model (ResNet-101). One of the best ways to address most of the issues with the prior architecture is to use the feature pyramid network. It facilitates the integration of the semantically weak aspects of higher resolution images with the semantically rich elements of lower resolution images. Like the other object identification techniques previously covered, we may construct both the classification and regression models using the final output. The regression network is designed to forecast the proper bounding boxes for the identified entities, while the classification network is utilized for suitable multi-class predictions.

2) Considerations

a) When to use RetinaNet?

RetinaNet is currently one of the best methods for object detection in a number of different tasks. It can be used as a replacement for a single-shot detector for a multitude of tasks to achieve quick and accurate results for images.

b) Usage Examples

There's a wide array of applications that can be performed with the RetinaNet object detection algorithm. A high-level application of RetinaNet is used for object detection in aerial and satellite imagery.

V. OBJECT DETECTION LIBRARIES

There are various object detection libraries are available and few of them are discussed here.

A. ImageAI

The goal of the ImageAI library is to give developers access to a wide range of deep learning techniques and computer vision algorithms for handling image processing and object detection tasks. The ImageAI library's main goal is to give developers of object detection projects a useful method for writing only a few lines of code [16] and [17]. Tensorflow, a well-known deep learning framework, and the Python programming language are used to write the majority of the available code blocks. This library uses a PyTorch backend as of June 2021 to compute image processing jobs.

Numerous operations related to object detection are supported by the ImageAI library, including custom object detection training and inference, custom image recognition training and inference, video object detection, video detection analysis, and image recognition. Up to 1000 distinct objects can be identified by the image recognition feature in a single image.

Eighty of the most frequent items observed in daily life can be identified with the use of the picture and video object detection challenge. Any specific object that is detected in a video or in real-time will be promptly analyzed with the aid of the video detection analysis. With this library, you can easily add own photos to train your own examples. Newer photos and datasets enable you train a lot more items for the object detection task.

B. GluonCV

One of the greatest library frameworks, GluonCV [18] and [19], has the majority of the most advanced deep learning algorithm implementations for a range of computer vision applications. This library's main goal is to assist those who are interested in this topic in producing useful findings more quickly. With a sizable collection of training datasets, implementation strategies, and thoughtfully created APIs, it boasts some of the best features.

You can use the GluonCV library framework to complete a significant amount of jobs. These projects include pose estimation to ascertain a specific body's pose, object detection tasks in image, video, or real-time, semantic segmentation and instance segmentation, picture classification, and action recognition to identify the kind of human activity being carried out. With these features, this library is among the finest for object detection to get faster results. This framework offers all the cutting-edge methods needed to do the aforementioned tasks. It provides a large selection of tutorials and extra help, and it supports both MXNet and PyTorch. With these, you can begin delving into a variety of concepts. It has a lot of training models that you may use to experiment and build the unique machine learning model you want to use for that activity.

You can begin the easy installation of this object detection library by installing PyTorch or MXNet in your virtual environment. For the library, you can select your own configuration. You may also use the Model Zoo, one of the greatest platforms for quickly deploying machine learning models, with it. With all of these characteristics, GluonCV is an excellent library for object detection.

 DOI : 10.35291/2454-9150.2024.0318

C. Detectron2

The Facebook AI research (FAIR) team created the Detectron2 framework, which is regarded as a next-generation library that covers the majority of cutting-edge segmentation algorithms, object identification approaches, and detection techniques. An object detection framework based on PyTorch is the Detectron2 library [20] and [21]. The library offers users a variety of excellent implementation strategies and methodologies and is very expandable and adaptable. Additionally, it supports a wide range of Facebook applications and production projects.

FaceBook's Detectron2 library, which was created on PyTorch, has a wide range of uses and can be trained on one or more GPUs to yield quick and efficient results. To get the finest outcomes, you can use a number of excellent object detection methods with the aid of this library. These state-of-the-art technologies and object detection algorithms supported by the library include DensePose, panoptic feature pyramid networks, and numerous other variations of the pioneering Mask R-CNN model family.

Additionally, users may easily train bespoke models and datasets with the help of the Detectron2 library. It has a very easy installation process. PyTorch and the COCO API are the only requirements needed for it. Once you meet these prerequisites, you may install the Detectron2 model and easily train a large number of models..

D. YOLOv3_TensorFlow

One of the effective applications of the YOLO series, which was introduced in 2018, is the YOLO v3 model. YOLO's third iteration enhances the earlier models. In terms of speed and precision, this model performs better than its predecessors. It can also operate reasonably well and precisely on smaller objects, in contrast to the other systems. The trade-off between speed and accuracy is the only significant issue when compared to other prominent algorithms.

One of the first applications of the YOLO architecture for object identification processing and computing is the YOLOv3_TensorFlow [22] library. In addition to many other features, it offers weight conversions, quicker training times, efficient outcomes and data pipelines, and incredibly quick GPU computations. The library is available on github, but PyTorch is now supported for this framework instead of this one, which, like most others, no longer receives support.

E. Darkflow

Darkflow is essentially a translation to fit the Python programming language and TensorFlow for making it accessible to a wider range of people. It is inspired by the darknet framework. Darknet is a preliminary implementation of a C and CUDA object detection library. This library has very straightforward installation and operation procedures. To get the best results in any situation, the framework also provides object detection tasks that are computed on both CPUs and GPUs.

To build the dark flow framework, a few fundamental requirements must be met. These prerequisites include OpenCV, Numpy, TensorFlow, and Python 3. You can easily begin computing tasks related to object detection with these requirements. You can accomplish a great deal of work with the dark flow library. You can get custom weights for a range of models, and the dark flow framework includes access to YOLO models.

Using the Darkflow framework for other similar applications, parsing annotations, designing the network according to a specific configuration, plotting graphs with flow, training a new model, training on a custom dataset, creating a real-time or video file, and finally, allowing you to save these models in the protobuf (.pb) format are some of the tasks that the Darkflow library helps you accomplish.

VI. Conclusion

Object detection is still one of the most essential deep learning and computer vision applications to date. We have seen a lot of improvements and advancements in the methodologies of object detection. It started with the algorithms like the Histogram of Oriented Gradients, introduced way back in 1986 to perform simple object detections on images with decent accuracy. Now, we have modern architectures such as Faster R-CNN, Mask R-CNN, YOLO and RetinaNet. Several object detection techniques like R-CNN, faster R-CNN, single shot detector (SSD), YOLO, RetinaNet etc. and the libraries for object detection have been discussed. From the discussions, it is found that as the model developed, the speed and accuracy has improved and increased. Fast R-CNN is improved than RCNN but Faster R-CNN is much improved than fast R-CNN. Also, single shot detector is better than faster R-CNN, while YOLO is better than single shot detector. YOLO model is beneficial as it can detect object directly and all objects are detected single time only in this model.

The development of the required model and resolution of the business challenge depend heavily on the theory and foundations of object detection. The hardest part of working with image data is figuring out how to identify things in photos so that the model can use them. One must assess a few duties when working with picture data, including object detection, bounding box calculations, determining the IoU value, and evaluation metrics. The limitations of object detection are not just confined to photos; they may also be applied to films and real-time recordings with remarkable accuracy. I hope that this article provides a helpful overview and aids in understanding or selecting the best object detection method based on the

 DOI : 10.35291/2454-9150.2024.0318

needs and dataset availability.

ACKNOWLEDGMENT

I thank Almighty God for his kindness in allowing me to write this survey paper. I would also like to thank my guide for all of his guidance and direction. I'm grateful to the professors and faculty at Sage University for providing me with all the guidance I required.

REFERENCES

[1] M. Pulipalupula, S. Patlola, M. Nayaki, M. Yadlapati, J. Das and B. R. Sanjeeva Reddy, "Object Detection using You Only Look Once (YOLO) Algorithm in Convolution Neural Network (CNN)," 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-4, doi: 10.1109/I2CT57861.2023.10126213.

[2] S. N. Bushra, S. A. Sibi, K. VijayaKumar and M. Niveditha, "Predicting Anomalous and Consigning Apprise During Heists," *2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF)*, Chennai, India, 2023, pp. 1-8, doi: 10.1109/ICECONF57129.2023.10084114.

[3] https://ijircst.org/DOC/40-effective-detection-of-weapons-in-video-surveillance.pdf

[4] K. S. Wong and L. P. Lin, "A Comparison of Six Convolutional Neural Networks for Weapon Categorization," *2022 International Conference on Electrical Engineering and Informatics (ICELTICs)*, Banda Aceh, Indonesia, 2022, pp. 1-6, doi: 10.1109/ICELTICs56128.2022.9932092.

[5] H. Jain, A. Vikram, Mohana, A. Kashyap and A. Jain, "Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications", Proc. Int. Conf. Electron. Sustain. Commun. Syst. ICESC 2020, pp. 193-198, 2020.

[6] G. K. Verma and A. Dhillon, "A Handheld Gun Detection using Faster R-CNN Deep Learning", Proc.7th Int. Conf. Comput. Commun. Technol., pp. 84-88, 2017.

[7] L. J. Halawa, A. Wibowo and F. Ernawan, "Face Recognition Using Faster R-CNN with Inception-V2 Architecture for CCTV Camera", Proc.3rd Int. Conf. Informatics Comput. Sci., pp. 2-7, 2019.M. Young, *The Techincal Writers Handbook.* Mill Valley, CA: University Science, 1989.

[8] https://www.youtube.com/watch?v=QmYJCxJWdEs

[9] https://www.youtube.com/watch?v=XmO0CSsKg88

[10] https://d2l.ai/chapter_computer-vision/rcnn.html

[11] https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

[12] https://arxiv.org/pdf/1512.02325

[13] https://arxiv.org/pdf/1506.02640

[14] https://arxiv.org/pdf/1708.02002

[15] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[16] https://imageai.readthedocs.io/en/latest/#

[17] https://github.com/OlafenwaMoses/ImageAI

[18] https://cv.gluon.ai/install.html

[19] https://github.com/dmlc/gluon-cv

[20] https://ai.facebook.com/blog/-detectron2-a-pytorch-based-modular-object-detection-library-/

[21] https://towardsdatascience.com/object-detection-in-6-steps-using-detectron2-705b92575578

[22] https://github.com/wizyoung/YOLOv3_TensorFlow

[23] https://neptune.ai/blog/object-detection-algorithms-and-libraries

[24] John, Anand, and Divyakant Meva. "A comparative study of various object detection algorithms and performance analysis." *International Journal of Computer Sciences and Engineering* 8.10 (2020): 158-163.

[25] Y. LeCun, Y. Bengio, G. Hinton, "Deep Learning", Nature, Vol.521, pp.436-444, 2015.

[26] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", IEEE Conference on Computer Vision and Pattern Recognition, Columbus, USA, pp. 580-587, 2014.

[27] Nguyen, Ngo-Doanh et al. "A Novel Hardware Architecture for Human Detection using HOG-SVM Co-Optimization." *2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)* (2019): 33-36.

[28] https://builtin.com/articles/histogram-of-oriented-gradients

[29] Ahmed, S., Bhatti, M. T., Khan, M. G., Lövström, B., & Shahid, M., Development and Optimization of Deep Learning Models for Weapon Detection in Surveillance Videos, Proceedings of Applied Sciences (Switzerland), 12(12). (2022).

[30] Ashraf, A. H., Imran, M., Qahtani, A. M., Alsufyani, A., Almutiry, O., Mahmood, A.,Attique, M., & Habib, M. Weapons detection for security and video surveillance using CNN and YOLO-V5s. Proceedings of Computers, Materials and Continua, 70(2), 2761–2775. (2022).

[31] Baig, M. S., & Khan, P. A., Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications. Proceedings of International Journal of Advanced Research in Science and Technology, 12(10), 127–135. (2020)

[32] Bhatti, M. T., Khan, M. G., Aslam, M., & Fiaz, M. J., Weapon Detection in Real-Time CCTV Videos Using Deep Learning. Proceedings of IEEE Access, 9, 34366–34382. (2021).

[33] Fathy, C., & Saleh, S. N., Integrating Deep Learning-Based IoT and Fog Computing with Software-Defined Networking for Detecting Weapons in Video Surveillance Systems. Sensors, 22(14). (2022).

[34] Galab, M. K., Taha, A., & Zayed, H. H., Adaptive Technique for Brightness Enhancement of Automated Knife Detection in Surveillance Video with Deep Learning, Arabian Journal for Science and Engineering, 46(4), 4049–4058. (2021)

[35] Gawade, S., Vidhya, R., & Radhika, R., Automatic Weapon Detection for Surveillance Applications. Proceedings of the International Conference on Innovative Computing and Communication, 1–6. (2022)

[36] Hashmi, T. S. S., Haq, N. U., Fraz, M. M., & Shahzad, M. Application of Deep Learning for Weapons Detection in Surveillance Videos. Proceedings of International Conference on Digital Futures and Transformative Technologies, October. (2021) 17 E3S Web of Conferences 391, 01071 (2023) https://doi.org/10.1051/e3sconf/202339101071 ICMED-ICMPC 2023

[37] Jain, H., Vikram, A., Mohana, Kashyap, A., & Jain, A Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications, Proceedings of the International Conference

 DOI : 10.35291/2454-9150.2024.0318

on Electronics and Sustainable Communication Systems, 193–198. (2020)

[38] Salido, J., Lomas, V., Ruiz-Santaquiteria, J., & Deniz, O. Automatic handgun detection with deep learning in video surveillance images. Applied Sciences, 11(13), 1-17 (2021)

[39] Hnoohom, N., Chotivatunyu, P., Maitrichit, N., Sornlertlamvanich, V., Mekruksavanich, S., & Jitpattanakul, A. Weapon Detection Using Faster R-CNN Inception-V2 for a CCTV Surveillance System. Proceedings of the 25th International Computer Science and Engineering Conference, 400–405. (2021).

[40] Ekmal, M., Quyyum, E., Haris, M., & Abdullah, L. Proceedings of the Multimedia University Engineering Conference. Atlantis Press International BV. (2023).

[41] Kaya, V., Tuncer, S., & Baran, A. Detection and classification of different weapon types using deep learning. Applied Sciences, 11(16), 1-13. (2021)

[42] Xu, S., & Hung, K, Development of an AI-based System for Automatic Detection and Recognition of Weapons in Surveillance Videos, proceedings of the IEEE 10th Symposium on Computer Applications and Industrial Electronics, 48–52. (2020).

[43] T Hamsini, Lokhande, H. V, NithisiriS, & L, R., A Review on Weapon Detection and Alert System Using Deep Neural Networks, proceedings of International Research Journal of Modernization in Engineering Today and Science, 4(06), 410–413. (2022).

[44] Raman Dugyala, M. Vishnu Vardhan Reddy, Ch. Tharun Reddy and G. Vijendar, Weapon Detection in Surveillance Videos Using YOLOV8 and PELSF-DCNN, E3S Web Conf., 391 (2023) 01071, DOI: https://doi.org/10.1051/e3sconf/202339101071

[45] https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns

[46] T.-Y. Lin, P. Dolla´r, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In*CVPR*,2017. 1,2,4,5,6,8

 DOI : 10.35291/2454-9150.2024.0318