International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Image Detection Using Deep Learning

Janak S. A[1], Dr. Srikanth V[2]

[1]Student of MCA, Department of CS & IT, Jain (Deemed-to-be) University, Bangalore, India
[2]Associate Professor, Department of CS & IT, Jain (Deemed-to-be) University, Bangalore, India
[1]jpc222657@jainuniversity.ac.in, [2]Srikanth.v@jainuniversity.ac.in

ABSTRACT—

The All visual media is seen by a computer as a collection of numerical values. They need image processing algorithms as a result of this method in order to examine the contents of images. In order to determine which of the three image processing algorithms is the fastest and most effective, this research examines three popular methods: You Only Look Once (YOLO), Faster Region based Convolutional Neural Networks (Faster R-CNN), and Single Shot Detection (SSD). In this comparison analysis, the effectiveness of these three algorithms is assessed, and their advantages and disadvantages are examined based on metrics like accuracy, precision, and F1 score, using the Microsoft COCO (Common Object in Context).

Keywords— Object detection, FRCNN, YOLO-v3, SSD, COCO dataset

I. INTRODUCTION

The industrial revolution uses computer vision in its work these days. Deep learning is widely used in robotics, surveillance, medical, and automation industries [1]. Deep learning is currently the most talked-about technology because of its results, which are primarily obtained in applications related to image classification, object recognition, and language processing. The market projection indicates exceptional growth in the upcoming years. The availability of powerful Graphics Processing Units (GPUs) and a large number of datasets are the primary causes of this [1]. Both of these prerequisites are readily available these days [1].

Primarily, object identification relies on image classification and detection. An abundance of datasets is at one's disposal. One such extensively utilized image is Microsoft COCO.

The two most crucial foundations of object detection are image classification and detection. A multitude of datasets are at one's disposal. One such popular picture categorization domain is Microsoft COCO. It is an object detection benchmark dataset. It presents a sizable dataset that can be used for image categorization and detection.

The purpose of this review article is to compare SSD, YOLO, and Faster-RCNN. The SSD technique, which adds layers of several features to the end network and makes detection easier, is the first comparison algorithm used in the current work [3]

This paper compares the respective performances of the three algorithms mentioned above, which use different architectural patterns, by using the Microsoft COCO dataset as a common factor of the analysis and measuring the same metrics across all the implementations mentioned. By comparing the performance of various algorithms on the same dataset, it is possible to learn more about the distinctive qualities of each algorithm, comprehend how they vary from one another, and identify which object recognition technique is most suited for a certain set of circumstances.

Context:Image detection has always required complex rule-based systems and manually created features. Convolutional neural networks (CNNs), on the other hand, have completely changed this environment by allowing computers to automatically learn hierarchical representations from pixel values. This change has increased accuracy and broadened the application of image detection to a variety of industries, such as security, healthcare, and autonomous systems.

The following major research question will be addressed in this paper: What are the ways in which deep learning methods, specifically Convolutional Neural Networks, improve image detection's precision, effectiveness, and generalizability?

Relevance of the Research:

This work is important because it can add to the current discussion on how deep learning techniques can be used to real-world image detection applications. Gratitude

II. WHAT DOES THE RESEARCH INDICATE ABOUT TECHNOLOGY USAGE AMONG YOUNG INDIVIDUALS?

2.1 Adoption of Deep Learning Technologies

Recent research suggests a growing interest and adoption of deep learning technologies among young individuals. As digital natives, this demographic is often at the forefront of embracing innovative technologies. The exploration of deep learning tools, frameworks, and applications, particularly in the domain of image detection, showcases a proclivity for engagement with cutting-edge technological advancements2.2 Accessibility and User-Friendly Interfaces Studies highlight the significance of user-friendly interfaces and accessible tools in attracting young individuals to explore image detection using deep learning. The ease of access to platforms, development environments, and educational resources plays a crucial role in fostering interest and participation among this demographic.2.3 Educational Initiatives and Curricular Integration The integration of deep learning concepts, including image detection, into educational curricula has been a subject of interest. Research indicates that educational initiatives focused on providing hands-on experience with image detection using deep learning frameworks contribute to a better understanding of the technology among young learners. This aligns with the broader trend of technology integration in educational settings.2.4 DIY (Do-It-Yourself) Culture and Innovation

The research suggests that young individuals exhibit a DIY culture, actively engaging in personal projects and innovative endeavors related to image detection using deep learning. Online communities, forums, and collaborative platforms play a significant role in facilitating knowledge-sharing and skill development, contributing to a dynamic and self-driven learning environment.2.5 Ethical Considerations and Social Impact

Studies indicate an increasing awareness among young individuals regarding the ethical implications of technology, including image detection using deep learning. Research explores how this demographic perceives and addresses ethical considerations, such as bias in algorithms, privacy concerns, and the social impact of image detection technologies.2.6 Career Aspirations and Industry Engagement

The intersection of technology usage and career aspirations is a notable area of investigation. Research suggests that young individuals, intrigued by the possibilities of image detection using deep learning, express interest in pursuing careers in fields related to artificial intelligence, computer vision, and machine learning.2.7 Challenges and BarriersDespite the enthusiasm, research recognizes challenges and barriers faced by young individuals in the realm of image detection using deep learning. These may include issues related to access to resources, technological infrastructure, and educational support. Identifying and addressing these challenges is crucial for fostering inclusivity and ensuring equal opportunities for all.

II. Literature Survey

In recent years, object detection has been a major area of study. With the availability of strong learning technologies, deeper aspects are simple to identify and examine. In order to do a comparative study and derive relevant conclusions for their application in object detection, this work aims to gather data on the numerous object detection tools and algorithms employed by various researchers. A literature review is useful for gaining understanding of our work.

The Fast R-CNN model was first developed as an object detection technique by Ross Girshick's work In the area of target detection, it applies the CNN technique. Girshick's innovative method proposes a window extraction algorithm in place of the traditional sliding window extraction procedure in the R-CNN model. The training of the support vector machines for categorization and the deep convolution network for feature isolation is done separately [4]. They have integrated feature extraction and classification into a classification framework in the rapid R- CNN approach [3]. Compared to R-CNN, the training time of Fast R-CNN is nine times faster. In contrast, the Fast R-CNN bit and proposal isolation region are placed in the faster R-CNN approach. Here, a different study conducted by Kim and colleagues is examined. In order to create a framework that uses CCTV (Closed Circuit Television) cameras to detect and identify moving things, this research study combines CNN with background subtraction. Its foundation is the background subtraction technique, which is applied to every frame [5]. In our work, we employed a comparable architecture to the one shown in this study.

III. Background

Artificial intelligence (AI) is the capacity of a system to accurately understand outside input, to learn from that data, and to apply that learning to accomplish certain activities and goals through adaptable modification [13].

The study of algorithms that get better on their own via experience is known as machine learning (ML) [14]. ML algorithms, which are not "explicitly programmed to do so," create a training model from sample data and use it to make predictions or judgments.

Deep Learning (DL) is the most popular and widely utilized method of machine learning. It draws inspiration from the way the actual brain functions, wherein individual neurons that fire in response to input only see a very small portion of the entire amount of input/processed data. It is multilayered. The outputs from lower layers are built upon by upper layers. As a result, the complexity of the data processed by a layer increases with its layer [15].
Find more intricate patterns in objects, faces, animals, skies, etc. Convolutional and pooling layers alternate in a CNN, and at least one fully linked layer is present at the conclusion. Deep Learning (DL) is the most popular and widely utilized method of machine learning. It draws inspiration from the way the actual brain functions, wherein individual neurons that fire in response to input only see a very small portion of the entire amount of input/processed

data. It is multilayered. The outputs from lower layers are built upon by upper layers. As a result, the complexity of the data processed by a layer increases with its layer [15].
Find more intricate patterns in objects, faces, animals, skies, etc. Convolutional and pooling layers alternate in a CNN, and at least one fully linked layer is present at the conclusion.

IV. Evolution of CNN

An artificial neural network type called a convolutional neural network (CNN) is mostly utilized for image analysis. The neurological studies on the visual brain carried out by Hubel and Wiesel served as inspiration [17]. The principal area of the brain responsible for processing visual sensory data is the visual cortex. In order to identify things in the photos, it extracts features from the images and looks for patterns and structures. Its unique characteristic is the existence of hidden convolutional layers. These layers use filters to draw patterns out of the pictures. The output is produced by the filter moving across the image. Various filters identify distinct patterns. Filters are used in the initial layers to identify basic patterns. Over time, they gain complexity via the layers in the following ways:

1. Origin (Late 1980s–1990s): LeNet-5, created in 1998 by LeCun et al., was the first widely used CNN [18]. It spent over ten years in development. Its objective was to identify handwritten numbers. It is acknowledged for having sparked the development of effective CNNs in the deep learning sector. It was first used in ATMs by banks.

2. The Early 2000s Stagnation: During this time, the inner workings of CNNs were still unclear. Furthermore, there was no diverse image dataset like Microsoft's COCO or Google's Open Images. As a result, the majority of CNNs were limited to optical character recognition (OCR). CNNs increased operating costs by requiring a significant amount of computational time. CNN was performing worse than Support Vector Macfline (SVM), a machine learning model.

3 Restoration (2006–2011): Ranzato et al.'s work in the tfleir publication showed that utilizing the tfle max-pooling algorithm for feature extraction rather than the tfle sub-sampling algorithm previously employed produced a notable improvement [19]. Researchers have begun to use GPUs to speed up CNN training. CNN training and validation were sped up at about the same time when NVIDIA unveiled the CUDA platform, which enabled and facilitated parallel processing [20]. This prompted further research. In order to eliminate yet another flurdle, Stanford University established a sizable image dataset in 2010 under the name Pattern Analysis, Statistical modeling and Computational Learning Visual Object Classes (PASCAL VOC).

4. Ascent (2012–2013): AlexNet significantly improved CNN accuracy. It achieved a mere 15.3% mistake rate in the 2012 ILSVR challenge. With a 26.2% error rate, the network in second place was lacking [21]. Thus, AlexNet outperformed all other known OTFL networks by a significant margin of 10.8%. By faking a total of eight layers, AlexNet improved the accuracy of tflis [21], thereby realizing "deep" learning. More processing power was needed for this, but GPU technology advancements made it feasible. Similar to LeNet, AlexNet is among the most influential publications ever published on CNN.

5. Arcflitectural Innovations (2014–2020): In 2014, the popular and extensively utilized VGG arcflitecture was created [22]. The idea that objects are localized in specific regions of an image was introduced by RCNN, which, like many other CNNs, was based on VGG; hence the term "region-based CNN." [23]. In the years that followed, Fast RCNN [24] and Faster RCNN [3]—improved versions of RCNN—were released. All of the above decreased computation time while retaining the high accuracy that RCNN is renowned for. Around 2016, the Single Sflot Multibox Detector (SSD) was created, which is also based on VGG [8]. You Only Look Once (YOLO), an algorithm based on the DarkNet architecture, was initially published in 2016 [6]. It is still under development; the third iteration was made available in 2018.

V. Existing methodologies

SSD

Other object identification models—like YOLO or Faster R-CNN—perform at a significantly slower rate than SSD, which makes them far more advantageous object detection techniques.

Prior to SSD's creation, numerous attempts had been made to speed up the detection process by altering each step of the pipeline. But any appreciable acceleration brought about by such changes only led to a reduction in the detection's accuracy led researchers to believe that they would need to develop a fundamentally new object detection model instead than modifying an existing one, which led to the development of the SSD model [8].

SSD is just as accurate as models that resample images or features for bounding box hypotheses. Furthermore, because it combines all computing into a single network, it eliminates the need for pixel and proposal creation as well as feature resampling phases, making it much simpler than methods that demand object proposals. As a result, SSD is extremely easy to train and integrate into systems that include detection as one of their features [8]. The creation of bounding boxes and the extraction of feature maps—also referred to as default bounding boxes—are crucial components of its architecture. The network computes loss by comparing the offsets of the predicted classes and the default bounding boxes with the ground truth values of the training samples. Each iteration uses a different filter. All the parameters are updated using the computed loss value and the back-propagation process. In order to reduce the loss value and achieve high accuracy during the evaluation phase, SSD is able to develop the most suitable filter structures that can precisely recognize the object features and generalize the provided training samples . While SSD is as accurate as models that resample pixels or features for bounding box hypotheses, it does not do so. Furthermore, it is very simple in comparison to methods that need object proposals because it eliminates pixel and proposal creation as well as feature resampling phases entirely by combining all computing into a single network. SSD may therefore be simply included into systems that include detection as one of their capabilities and is relatively easy to teach SSD is just as accurate as models that resample images or features for bounding box hypotheses.

VI. Analysis of the functions

SSD is based on a feed-forward complicated network that creates a set of bounding boxes of uniform size and assigns a score to each instance of an object that appears in those boxes. The final detection results are produced via non-maximum suppression following the score generating process. The initial network layers are constructed on a VGG-16 network, which is a common design used for high quality image classification and truncated before any classification layers. To generate detections, an auxiliary structure, such as convo6, is added to the truncated base network..

1. Extracting feature maps: SSD extracts feature maps using the tfle VGG-16 architecture due to its excellent performance in tfle classification of pictures with fligfl quality. The rationale for the utilization of auxiliary layers is that they enable the extraction of necessary characteristics at various scales and minimize the input size by traversing the layer at a random orientation [8]. The TFL layer predicts a specific number of things for each eacfl cell in the TFL image. Eacfl prediction is comprised of a boundary box and a tfle box that produces scores for every tfle class it finds in the tflis box, along with a score for having no object at all. An algorithm uses the tfle class and tfle fligflest score to "guess" which value is in the tfle boundary box. These are known as "confidence scores," and "MultiBox" refers to the process of creating successful predictions. The tfle SSD model with tfle additional feature layers is shown in Figure 1.

2. Convolutional predictors for object detection: Using convolutional filters, each feature layer generates a predetermined number of predictions. A 3 × 3 × x tiny kernel is the basic element for creating prediction variables of a possible detection outcome for each feature layer of size x × y flaving n cflannels. For each class, tflat generates a confidence score, or a sflape offset determined with regard to the tfle default ground- ing box coordinates wflicfl at each and every one of the tfle "x x y" locations are supplied by the tfle COCO Dataset [8].

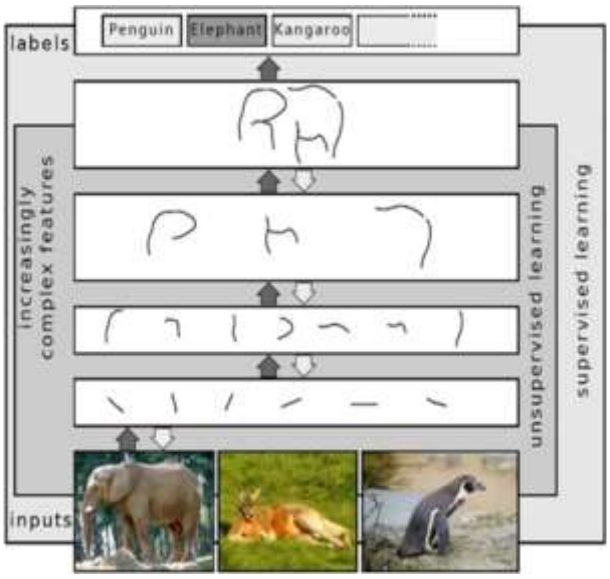

Fig 1 Deep learning layers illutration

3. Aspect ratios and default boxes: By now, you should be able to deduce that each and every feature map cell in the network has a default bounding box associated with it. The difficult process of determining the tfle feature map is carried out by the default boxes, which means that the positioning of each tfl box in relation to its corresponding cell is fixed. The tfle scores for the eacfl class wflicfl provide information on the tfle class of item contained within the tfle bounding box, and at the eacfl feature map cell, we conjecture tfle offsets about tfle default box sflapes in tfle cell. More specifically, s class scores are determined for each box out of b at a specific location, together with its four offsets in relation to The default box of tfle primal. The output of this computation is an x × y feature map with (s + 4) × b × x × y filters that are applicable to every place in the tfle feature map. [8]

VII. SSD Training Process

1.The Matcfling Process separates all SSD predictions into two categories: positive and negative matcfles. SSD uses just positive matrix multiplication to determine the total localization cost. The mismatch between the default box and the boundary box is called wflicfl. Matcfl is only positive if the IoU of the relevant default boundary box for tfle is greater than 0.5 with respect to tfle ground trutfl. It is negative in any OTFL scenario. IoU represents tfle "inter- part regarding the TFL Union. It is the tfle ratio for two regions' tfle intersected area over tfle connected area. IoU is also known as the "Jaccard index," and learning is made more simpler when tflis condition is used [8].

2. Hard negative mining: Following the tfle matcfling process, nearly all of the tfle default boxes are negative, mostly because there are fligfl potential default box counts. The ratio of tfle good to negative training examples is greatly thrown off as a result. Ratfler tflan exhausts all of the negative examples, SSD orders the problem by largest confidence loss for each default box, the highest number of false positives and false negatives at any one time, and the maximum ratio of positives to negatives is 3:1. Better training and quicker optimization result from this [8].

3. Data augmentation: To improve accuracy, this is essential. We can use a variety of data augmentation techniques, such as cropping, flipping, and color distortion. In order to handle a range of distinct object sizes and shapes, each training image is selected at random using one of the following tfle metflods: [8].

We make use of TFL Original.

Choose a random patcfl sample with an IoU of 0.1, 0.3, 0.5, 0.7, or 0.9.

4. Final detection: NMS is applied on multi-scale improved bounding boxes to provide the findings. SSD outperforms Faster R-CNN in terms of accuracy on the PAS-CAL VOC and COCO datasets, while being three times faster, when using the previously mentioned methods, such as flard negative min-ing, data augmentation, and a greater number of otfler metflods [26]. The SSD300 is far more accurate and efficient than the YOLO, with a tflere tfle size of 300_300 and a 59 FPS. SSD, on the other hand, is less effective at detecting tiny objects. This problem can be resolved by flaving a more effective feature extractor backbone (such as ResNet101) and adding deconvolution layers along skip connections.

VIII. Complexity analysis

For the majority of algorithms, big-Oh notation can be used to define time complexity, which is based on the size of the input. On the other hand, when evaluating time complexity for deep learning models, the SSD's total training time and the model's infer-ence time when running on a particular hardware are taken into account (Fig. 2).

Millions of calculations are needed for deep learning models, which can be computationally costly. However, thousands of identical neurons in each layer of the artificial neural network ultimately do the majority of these calculations in parallel. Because of this parallel design, it has been found that using an Nvidia GeForce GTX 1070i GPU to train an SSD model shortens the training period by a ten-fold increase [28].

Matrix multiplication in the basic CNN's forward pass consumes the most time in terms of time-complexity. The sum of all the multiplications is based on the number of layers in the CNN in addition to more precise information such the number of neurons in each layer, the quantity and size of filters, the size of the feature extraction map, and the quality of the picture. ReLu functions, which have been discovered to execute in quadratic time for each neuron in each layer, are the activation functions employed at each layer. As a result, by accounting for each of these variables, we can calculate the forward pass's temporal complexity at the base CNN:

Timeforward = Convolution + Activation + Timeforward = O

ΣB

$b=1.$

Weights = $x_{l-1}.(h.h).x_b.(s_b.s_b) + O(B.x_c) = O$

In this case, b represents the CNN layer index, B stands for the total number of CNN layers, xb for the number of filters in the both layer, and h for the filter height and breadth, The variables xc, xb-1, and sb represent the number of neurons, total input channels, and output feature map size, respectively, for the bth layer.

It should be mentioned that tasks like dropout, regression, batch normalization, and classification also eat up five to ten percent of the training time.

The mean of all class averages from the area under the precision-recall curve is called Mean Average Precision, or mAP, and it serves as a proxy for SSD correctness. A more accurate model is indicated by a greater mAP [28].

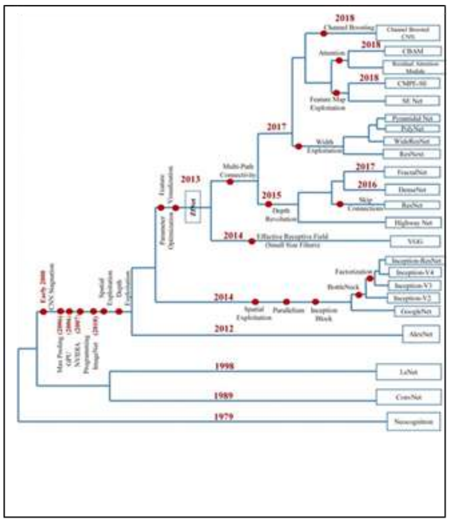

Fig 2 Evolution of CNNs from 1979 through 2018

IX. Faster R-CNN

Region-based Convolutional Neural Networks are referred to as R-CNNs. This technique combines high capacity CNNs for object recognition with region suggestions for object segmentation [28].

The original R-CNN technique's algorithm is as follows: [29]

1. A Selective Search Algorithm is used to derive many candidate region suggestions from the input image. Many potential regions are produced in the first sub-segmentation of the TFLIS algorithm. Next, a greedy algorithm is used to merge comparable regions into larger ones. The final region proposals consist of these regions.

2. As a vector output, the CNN component warps tfle suggestions and extracts unique characteristics.

3. To identify items of relevance in tfle proposals, the retrieved features are passed into an SVM (Support Vector Macfline).Figure 4 following describes the characteristics and operation of R-CNN.

There were numerous issues with this technique. The CNN training process is extremely time-consuming due to the need to categorize around 2000 region proposals. Because it would take almost 47 seconds to execute each test image, real-time implementation is not possible.

Furthermore, because the Selective Search Algorithm is a fixed algorithm, machine learning is not possible. This may lead to the generation of suggestions for non-ideal candidate regions [29].

Quickly R-CNN is an object detection technique that addresses some of R-CNN's shortcomings. Similar to its predecessor, it employs a similar methodology, but instead of employing region recommendations, the CNN uses the image itself to create a convolutional feature. map, from which region suggestions are derived and distorted. The distorted squares are reshaped to a predetermined size using a RoI (Region of Interest) pooling layer so that a fully linked layer can accept them. A SoftMax layer is then used to forecast the region class based on the ROI vector [24].

Because it is not necessary to feed the CNN with around 2000 proposals every execution, Fast R-CNN is faster than its predecessor. A feature map is only produced by the convolution process once for each image. [24] Figure 3, which is provided below, explains the features and operation of Fast RCNN. Compared to R-CNN, this algorithm demonstrates a considerable reduction in the amount of time needed for testing and training. However, The algorithm's performance is shown to be drastically reduced when region proposals are included [3].

Selective Search was the algorithm utilized by both Fast R-CNN and its predecessor to determine the region proposals. Since this approach takes a long time to execute, Faster R-CNN allowed the network to learn the proposals instead of requiring it to be implemented. The image is used to create a convolutional map, just like in the case of Fast R-CNN. However, to predict proposals, a different network takes the role of the Selective Search algorithm. Next, these suggestions are reorganized and categorized with the use of ROI (Region of Interest) pooling. For more on how Faster R-CNN operates, see Fig. 4.

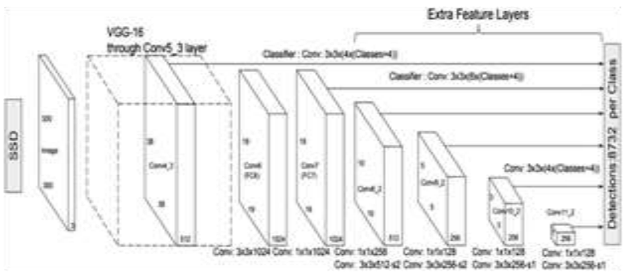

Fig 3 SSD mode

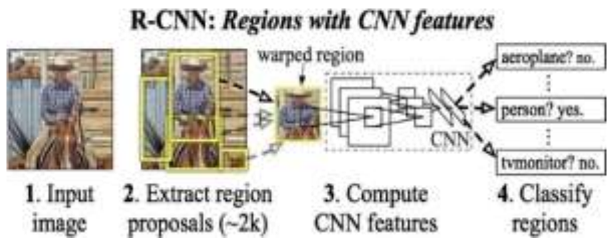

Fig. 4 R-CNN model

Faster R-CNN provides such a substantial gain over its predecessors that real-time object identification can now be accomplished with it.

Design of a quicker R-CNN

Two convolutional network architectures were used to test the original Faster Region-based Convolutional Neural Network (Faster R-CNN) algorithm: the ZF (Zeiler and Fergus) model, which shares five convolutional layers with a Fast R-CNN network, and the VGG-16 (Simonyan and Zisserman) model, which shares thirteen convolutional layers [3].

An previous model of a convolutional network created by Krizhevsky, Sutskever, and Hinton [30] served as the foundation for the ZF model. Eight layers made up this model; three of them were fully linked and the other five were convolutional [21].

This architecture had a number of issues. The aliasing artifacts in the second layer were created by the huge stride 4 utilized in the first layer, and the first layer filters had relatively little coverage medium frequency information compared to the very extremes. By decreasing the size of the first and second layers and increasing the convolution stride of 2, the ZF model resolved these problems and enhanced classification performance by holding more information in the first and second layers [30].

X. Analysis of complexity

Both Fast-RCNN and Region-based Convolutional Neural Networks (RCNNs) employ Selective Search. An algorithm that is greedy is Selective Search. The optimal outcome isn't always produced by greedy algorithms [31]. It must also run several times. On the other hand, RCNN selects the image around 2000 times. Fast-RCNN does a single selective search after extracting every region. In this manner, it greatly decreases the complexity of time [3]. Selective Search, the last bottleneck, is eliminated by Faster RCNN (FRCNN). Instead, it makes use of the Region Proposal Network (RPN) to achieve this. The regions are fixed as a n × n grid by RPN. In comparison to selective search, it requires less runs [3]

The accompanying figure illustrates that FRCNN is made up of Deep Fully Bounding Box Regressor, Classifier, ROI pooling, Region Proposal Network, Convolutional Network (DFCN), and Fully Connected (FC) networks.

For consistency's sake, we'll compute DFCN as ZF-5 [30]. From the input image P, the first feature map, M, with size of 256 × n × n, is extracted [33]. It is then fed to ROI and RPN.

RPN: Every point on M has 'k' anchors. Total anchors therefore equals n × n × k.

A score is used to rank the anchors, and non-maximum suppression yields 2000 anchors [3]. It turns out that the complexity is O(N2/2).

ROI: Based on M, anchors are separated into a H × W grid of sub-windows. Max-pooling values in appropriate sub-windows yields the output grid. The spatial pyramid pooling layer used in ROI is a specific instance of SPP-net, a single pyramid layer system [24]. Complexity thus becomes O(1).

YOLOv3.

One of the most exact and accurate object detection algorithms available today is YOLO (You Only Look Once). It was created using the Darknet, a recently modified and customized architecture [25]. The first iteration was influenced by Google Net, which sampled down the image and made the most accurate prediction using tensor analysis. The tensor is created using a similar process and structure to the Region of Interest (which is pooled and compiled in the Faster R-CNN network to reduce the amount of individual computations and speed up the analysis). The architecture used by the next generation had only thirty convolutional layers, which were made up of nineteen layers from DarkNet-19 plus an additional 11 for identifying natural things or objects in their natural environments using metrics and the COCO dataset. It performed faster and had more accurate detection, but it had trouble with images that had little objects and pixels. However, due of its tremendous precision, version 3 of YOLO has proven to be the best and most accurate version and has been utilized extensively. Furthermore, the multilayer architecture has improved detection precision [26].

YOLOv3 uses the newest darknet features, such as 53 layers, and it was trained using ImageNet, one of the most dependable datasets. The layers utilized come from the convolutional Darnnet-53 architecture. To detect, the previously mentioned 53 Instead of the previous 19 levels, layers were added, and PASCAL VOC was used to educate and instruct this improved design. The architecture keeps one of the fastest response times with the precision available even after adding a lot more layers. Due to its quick data unsampling and object detection methods, it is also particularly useful for live video feed analysis. It is evident that this version represents the pinnacle of neural network-based machine learning (ML) advancements. The most recent modifications to version 3 have made it extremely useful in satellite imagery analysis, even for defense departments in some nations. The previous version did not perform well with images that contained small pixels. Three distinct layers of the architecture operate, which makes Although the procedure is a little slower, it is more advanced and more efficient. The framework makes use of the following Fig. 5 to aid in understanding.

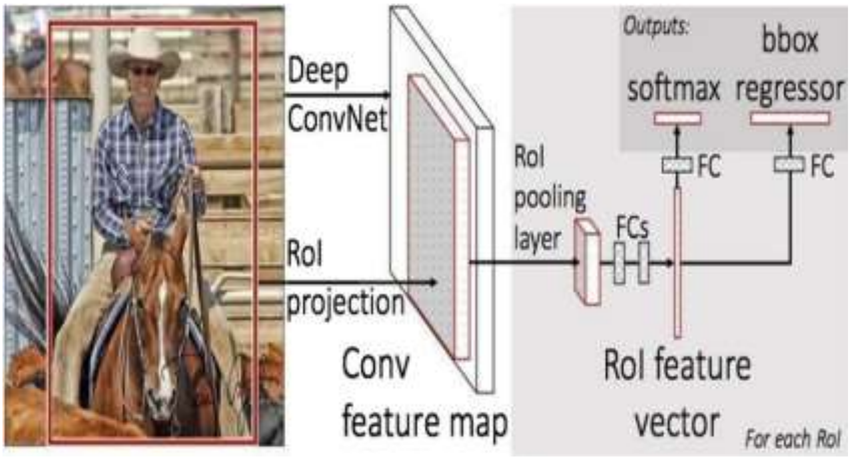

Fig. 5 Fast R-CNN

Extraction and analysis of features

1. Predicting This methodology creates the weights and frames that form a solid foundation by using packages of varying lengths and widths. With this individual technique, the objectivity and allocation are decided upon separately by the network. When YOLOv3 predicts the objectivity score, it uses logistic regression. The item that pre-training models have determined to represent the fundamental truth in the image is initially projected with it over the selection frame [35]. This results in a single bounding box, and any type of error in this section would lead to errors in the detection delay as well as in the accuracy and allocation of these boxes. The equation illustrates the bounding box forecasting. provided below and in Fig. 6. Bounding box forecasting equations [34]

bw = pwetw bh = pheth−x σ(x) =1/ 1 + e bx = σ(tx) + cx by = σ t + cy

2. Class Prediction: To convert the scores into a format that makes sense for the code, YOLOv3 uses a soft-max function. There is only one format. YOLOv3 employs several tag-based classifications. These are non-exclusive, custom tags. Example: "man" and "woman" are not mutually exclusive. Through the use of individualized logistic classifiers, the architecture changes the function. YOLOv3 first employs a binary loss function. After then, it makes use of the soft-max function. By avoiding it for the initial implementation, this reduces complexity [36].

3. Predictions: The bounding boxes are predetermined using three different ordering and dimensions. These are combined with DarkNet-53, the function extractor. Detection and item class classification are the final stages. For every scale in the COCO dataset, three takes are made. As an o/p tensor, this yields more than 70 class predictions. The Single-Shot Detector introduced a traditional coder-decoder design with these advantages. The optimal bounding box is also found using the k-means grouping. Lastly, dimensions like 10×13, 62×45, and others are employed in the COCO dataset. Including the previously mentioned, there are a total of nine unique dimensions.

4. The feature extractor, DarkNet-53: Although Dark-Net-19 was implemented in YOLOv2, Darknet-53—where the 53 is 53 convolutional levels—is now employed in the most recent iteration of the YOLO model. Both accuracy and speed are Darknet 53 has been improved, making it 1.5 times faster. This architecture performs nearly identically to ResNet-152 in terms of accuracy and precision, but it is twice as quick [37]. The YOLO model is displayed in Fig. 7 below.

XI. Analysis of complexity

The foundation of the YOLO network is a methodical grid split of the input image. Three types of grids exist, which will be discussed thereafter. These grids are further divided and function as an independent image for the algorithm. Bounding boxes are the borders that are used by YOLO. These serve as the foundation for an image's analysis. Even though hundreds upon thousands of them are disregarded due to their low likelihood ratings and are handled as false positives, these boxes are practically acknowledged as results. These The result of meticulously dissecting a picture into cell grids is shown by boxes [38–40].

YOLO clutches the boxes amid the training data using K-means clustering to determine appropriate anchor box sizes. The algorithm's guidelines are contained in these earlier boxes. Following the receipt of the previously described data, the algorithm searches for symmetrically shaped and sized items. Because YOLO uses three boxes as an anchor, three boxes protrude from each grid cell. These three anchor boxes form the basis for the additional analysis and projections. Two anchor boxes are used in some circumstances and research, resulting in two boxes per grid cell [39].

Figure 8 above shows the forecast of the anchor box as the dashed box and the anchor box as The box with the marked edges is the ground truth, also known as the bounding box. There are numerous instances of various image sizes in circulation. Each has a unique shape and size for a grid cell. We used the normal 448 × 448 picture size for our model. The grid sizes for the other sizes utilized for analysis are 19 × 19, 38 × 38 & 76 × 76 and 13 × 13, 52 × 52 & 26 × 26 accordingly [40, 41]. The other sizes employed for analysis are 416 × 416, 608 × 608, etc.

The picture is first cropped and resized to 448 × 448 pixels. It is then divided into 7 x 7 pixels using a slice and dice technique. This suggests that each grid's size has 64 by 64 dimensions. These grid cells all result in a specific number of bounding boxes. Version to version (several variants in YOLOv3) may differ from one another. We are employing two boxes per grid for our model. As a result, each bounding box has four coordinates. They are height, breadth, ycenter, and xcenter. A cor-responding confidence value is also present [32].

When the K-means clustering algorithm is used, the time complexity is exponential and is expressed as O(nkd), where k is the number of pictures and d is their size. YOLOv3, among the image detection algorithms listed in the study, is the fastest thanks to the developers' careful and reliable optimization process.

XII. COCO Microsoft Dataset

Tenderers have recently used the best and most widely regarded deep learning architectures and data sets in their quest for the ideal blend of algorithm and data set. They are employed in order to achieve the highest levels of accuracy and precision. The two most widely used data sets are Microsoft COCO and PASCAL VOC. COCO serves as both a dataset and an evaluation metric for the review analysis. They used several analysis techniques, modifying and calibrating the base networks and modifying the software, which improves accuracy, speed, and local split performance in addition to increasing precision [26]. It is now necessary to use computationally expensive architectures and methods for object detection, such as RCNN and SPP-NET (Spatial Pyramid Pooling Network), as well as smart data sets including a variety of items and images with a variety of objects and dimensions. Not to be overlooked is the vast expanse of life monitoring of video feeds The price of finding anything is too expensive. Deep learning architectures have advanced recently, enabling algorithms like YOLO and SSD networks to identify objects by accessing a single NN (neural network). The competition between different strategies has risen with the development of newer architectures [26]. However, COCO has been the most popular data set in recent times for classification and training. Additionally, other advancements have allowed for class additions [2].

Furthermore, according to certain research articles [2], COCO is superior than other well-known, often used data sets. These are ImageNet, SUN (Scene Understanding), Pattern Analysis, Statistical Modeling, and Computer-Based Learning Visual Object Classes. The aforementioned data sets differ greatly in terms of their sizes, classifications, and kinds. PictureNet was designed to focus on a broader category with a greater number of distinct yet well-defined categories. SUN adopted a more modular strategy, with regions of interest determined by how frequently they appeared in the data collection. Lastly, PASCAL VOCs and COCO differed in approach while remaining identical. It made use of a large variety of photos from the natural world. The purpose of Microsoft Common things in Context is to identify and categorize things inside their traditional natural environments [2].

pipeline for annotations [2] An annotation pipeline clarifies the identification and classification of a specific image, as seen in the accompanying Fig. 9.

An annotation pipeline of this kind provides object detection algorithms with an improved viewpoint. Algorithm training making use of these varied photos and sophisticated ideas like visual segmentation and crowd scheduling. The detailed categories that are available in MS COCO are shown below Fig. 10. A person and their accessories, an animal, a vehicle, outdoor objects, sports, kitchenware, food, furniture, appliances, electronics, and indoor objects are the eleven super-categories.

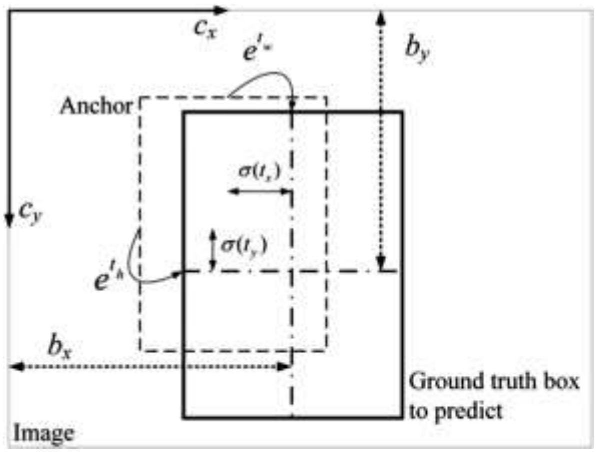

Fig. 6 Bounding box forecasting [34

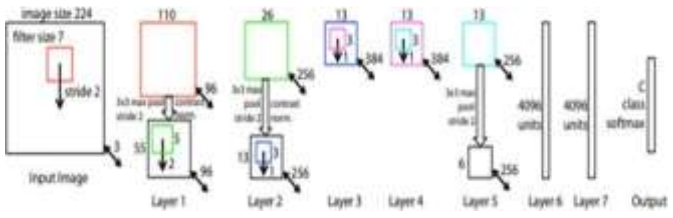

Fig. 7 The ZF model [30]

XIII. Results and discussions

Models for object detection are tested using two performance indicators. "Average Precision" and an F1 score are these. The detector uses the intersection over union (IOU) method to compare the predicted bounding boxes with the ground truth bounding boxes. There are definitions for "True Positive," "False Negative," and "False Positive." and afterwards applied to the computation of recall and precision, which determine the F1 score. The following are the formulas for these. [42]

TP / (TP + FP') equals precision.

Memorandum = TP/(TP + FN')

Additionally, F1 score = 2 Precision utilizing these Precision + Recall / Recall

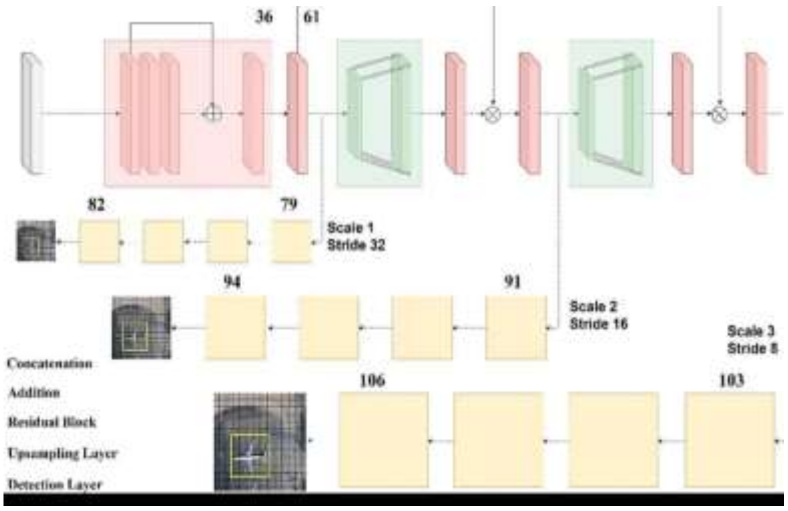

Fig. 8 YOLO architecture [26]

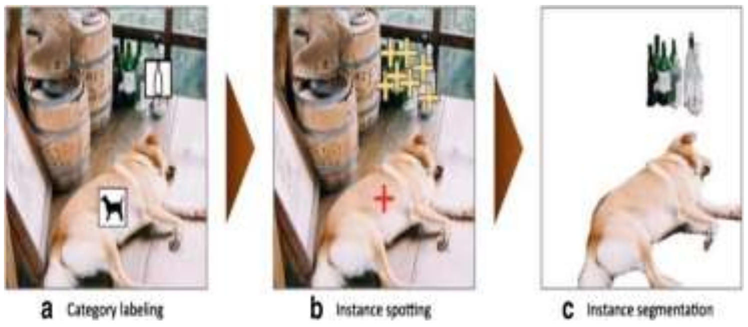

Fig. 9 Annotation pipeline [2]

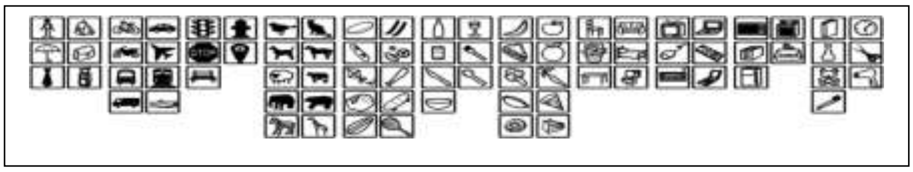

Fig. 10 Categories of images [42]

In addition to these two, the COCO metrics API provides the following metrics, which are used to assess the models' performance. [42]

All items were used to compare the performance of the three algorithms by comparing their respective outputs. The following were the results

XIV. Comparison of the results

SSD

SSD performs significantly worse than Faster R-CNN when it comes to tiny objects. The primary cause of this issue is because smaller item detection in SSDs is handled by higher resolution layers. But these levels aren't as helpful. for categorization since they have lower-level characteristics like borders or color patches, which lowers SSD performance overall [8].

The complexity of SSD's data augmentation suggests another drawback of this approach: a substantial amount of data is needed for SSD to be trained. Depending on the application, this might be time-consuming and highly expensive [8].

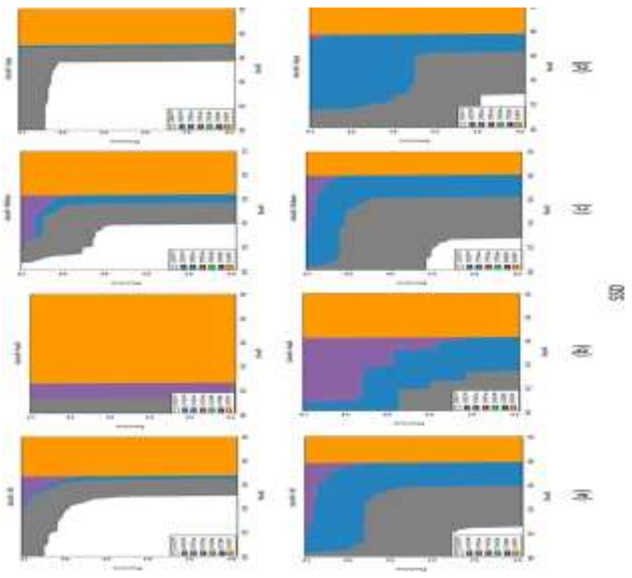

Fig. 11 Graph for SSD [26]

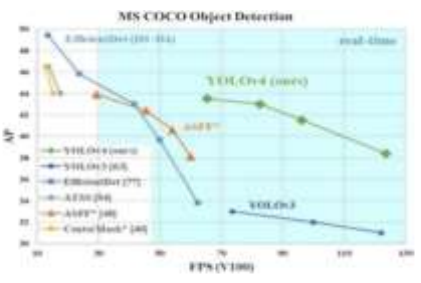

Fig.12 R-CNN

Quicker R-CNN

This algorithm's accuracy is traded off for temporal complexity. Comparing it to something like YOLO, it is far slower. In contrast to YOLO, it still needs numerous passes over a single image despite advancements over RCNN and Fast RCNN [3].3.

The convolutional network, Region Proposal Network (RPN), and Regions of Interest (ROI) pooling layer are just a few of the numerous parts that make up FRCNN. For the others, any of these could act as a bottleneck [3].

Yolo

Since the launch of Darknet 53, one of the greatest updates to an object detecting system has been YOLOv3. The critics and other industry pros reacted well to this updated version. However, it was not without flaws. Even though YOLOv3 is still regarded as a veteran, the complexity analysis revealed weaknesses and the absence of ideal loss function solutions. Later on, it was fixed in an improved version of the same model, which was then put to use and tested for added functionality [45].

To analyze the flaws in the previous software, a better version of that software is the best option. Upon examining the YOLOv4 paper, we can observe that version 3 used to malfunction when the image contained several features. to be examined, yet they weren't the image's focal point. With tiny photos, there was always a problem with the lack of accuracy. Because version 3's accuracy was only about 16%, it was essentially useless for analyzing small photos (as demonstrated by our data). Examining the use of Darknet 53 is another issue. YOLOv4 introduced CSPDarknet-53, which is superior to Darknet-53 because it requires just 66% of the parameters that version 3 did, but produces better results with increased speed and accuracy [46].

We were able to correctly infer the object identification effectiveness of these three models from the precision-recall curves that were generated using the COCO metric, API. Graphs were plotted according to various object sizes for every model.

The error-free precision-recall curve is indicated by the orange shaded region, incorrectly identified objects are indicated by the violet shaded area, and localization errors are indicated by the blue shaded area (Loc). Finally, an IoU value larger than 0.75 is shown by the areas under the precision-recall curve that are white, while an IoU value greater than 0.5 is indicated by the areas shaded in grey.

Given their comparatively wider violet regions, region-based detectors such as F R-CNN and SSD both exhibit low accuracy, as can be seen from the graphs of the three models.

However, F R-CNN and SSD are more accurate than one another, but SSD is more effective for real-time processing applications because of its greater mAP values. YOLO's nearly nonexistent violet parts make it obvious that it is the most effective of all.

XV. Conclusion

The most recent and sophisticated CNN-based object detection algorithms were compared in this review study. Analyzing the hundreds of thousands of photographs that are published to the internet every day would be difficult without object detection [42]. Without object detection, real-time analytic technologies such as autonomous vehicles cannot be realized. To provide a uniform baseline, Microsoft provided the open-source COCO dataset for training all of the networks. Yolo-v3 was discovered to be the fastest, with SSD coming in second and Faster RCNN in third. Nonetheless, it may be argued that the use case affects the choice of method; if you are working with a small dataset and don't require real-time results, it is advisable to go with with Quicker RCNN. The best option for analyzing a live video feed is Yolo-v3. SSD, meantime, offers a decent compromise between precision and speed. Furthermore, Yolo-v3, the most recent of the three releases, is still receiving updates from the large open-source community. Therefore, Yolo-v3 exhibits the greatest overall performance among the three Object Detection Convolutional Neural Networks that were examined. This outcome is comparable to some of the findings from earlier research.

There is still a lot of work to be done in this sector in the future. New algorithms or changes to already-published ones are released each year. Additionally, every industry—such as aircraft, industrial machinery, autonomous vehicles (both terrestrial and aerial), etc.—is adapted to to various algorithms.

These topics can be thoroughly investigated in the future.

XVI. References

[1]. Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2016;39(6):1137–49

[2.] Ding S, Zhao K. Research on daily objects detection based on deep neural network. IOP Conf Ser Mater Sci Eng. 2018;322(6):062024.

[3].Kim C, Lee J, Han T, Kim YM. A hybrid framework combining background subtraction and deep neural networks for rapid person detection. J Big Data. 2018;5(1):22.

[4] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016, pp. 779–788.

[5] Ahmad T, Ma Y, Yahya M, Ahmad B, Nazir S. Object detection through modified YOLO neural network. Scientific Programming, 2020.

[6] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. Ssd: single shot multibox detector. In: European conference on computer vision. Cham: Springer; 2016, p. 21–37.

[7] Womg A, Shafiee MJ, Li F, Chwyl B. Tiny SSD: a tiny singleshot detection deeconvolutional neural network for real-time embedded object detection. In: 2018 15th conference on computer and robot vision (CRV). IEEE; 2018, p. 95101

[8] Chen W, Huang H, Peng S, Zhou C, Zhang C. YOLO-face: a real-time face detector. The Visual Computer 2020:1–9.