Int. J. Sci. R. Tech., 2024 1(11)
A Multidisciplinary peer-reviewed Journal
www.ijsrtjournal.com [ISSN: 2394-7063]

Original Article

Comparison of Object Detection Algorithms CNN, YOLO and SSD

Ghansham More, Omkar Patil, Omkar More, Mihir More, Samadhan Suryavanshi, Manisha Mali

Dept. of Computer Engineering, BRACT's Vishwakarma Institute of Information Technology, Pune, India.

ABSTRACT

Since 2015, numerous studies have concentrated on object detection, a crucial element of computer vision, using convolutional neural networks (CNN) and their various architectures. Key methods for object detection done by "YOLO (You Only Look Once)", "CNN", and "SSD (Single Shot Multibox Detector)". This paper explores three representative series of methods based on "CNN, YOLO, and SSD", providing solutions to challenges like bounding box prediction in CNNs. The strength of these algorithms are measured in terms of accuracy, processing speed, and computational cost. YOLO models.we want to do comprehensive study of three models of object detection"(YOLO, CNN, SSD)".

Keywords: CNN, YOLO, SSD.

INTRODUCTION

In the fields of computer vision and image processing, object recognition is an essential approach that is used to analyze both still pictures and video streams. Natural images present complex challenges due to variations in color, shapes, and textures, making object detection in real-world scenarios a difficult task. Identifying and localizing items in an image by correctly categorizing them and comprehending their importance is the main objective of object detection. While humans can perform these tasks effortlessly, machines rely on specialized algorithms to achieve similar results. Object detection algorithms typically involve several key steps, such as identifying the object, centering a bounding box around it, and then classifying it. These techniques have various applications, including surveillance, vehicle detection, and object tracking. Recently, advancements have been made in image classification, video recognition, sound analysis, and face identification, with machine learning approaches like Convolutional Neural Networks (CNNs) leading the way. Earlier object detection methods, such as the "Scale Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG)", played a significant role in feature extraction. Techniques like Support Vector Machines (SVMs) were also used to improve recognition rates. But since deep learning techniques were developed, "CNNs" have taken over as the most used object recognition tool. Research by "AlexNet" demonstrated the power of "CNNs" in deep learning, marking a turning point for object detection accuracy. The "YOLO" "(You Only Look Once)" family of algorithms is examined in this work along with additional approaches such as Faster "R-CNN," which focuses on object identification methods based on "CNNs." There is also discussion on the developments and future paths of "YOLO" and other deep learning algorithms.

1. Related Work

Traditional The foundation of the early pill detection study was traditional machine learning. extracted feature vectors from pill imprint photos using invariant moments and Canny edge detection[11]. Analyzed images of pills from multiple angles to match unique features and identify the pills[12]. Similarly, employed Otsu's thresholding combined with noise reduction to extract pill imprints, achieving precision and recall rates over 57% for text detection on imprints[13],Neto et al. used color and shape-based feature extraction in a dataset of 1,000 images representing 100 different pill types, attaining over 99% accuracy[14]. A support vector machine (SVM) was employed in a different method by Dhivya et al. to identify text imprints on tablets [15]. However, traditional machine learning methods rely

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

heavily on manually designed feature extraction techniques, where detection accuracy is influenced by the chosen features and classifiers. This process often requires specific configurations for each type of pill, leading to significant manual effort, particularly in environments like China's centralized medicine bidding system, where pill types can vary annually. Such manual feature design is not resilient to diverse pill appearances and high-volume datasets, especially when imprints are missing, reducing recognition accuracy. The computational complexity of traditional object detection methods, such as the sliding window approach, further limits real-time performance, making more efficient solutions desirable.Deep learning, specifically convolutional neural networks (CNNs), provides a more effective approach to pill identification, utilizing multiple convolutional layers for feature extraction and detection of objects. By contrasting AlexNet, the winner of the ILSVRC 2012 competition, with more conventional machine learning techniques like random forests and k-nearest neighbors, it was shown to be superior. AlexNet outperformed these techniques with a top-1 pill recognition accuracy of 95.35%[16]. Although AlexNet is a relatively simple network with limited flexibility for more complex tasks, it marked a significant improvement over manual feature design.Swastika et al. proposed a network combining three CNN models, such as LeNet and AlexNet, to extract key pill features—shape, color, and imprint—achieving a remarkable 99.16% recognition accuracy using 24,000 images of eight different pill types[17]. Ou et al. developed a two-stage detection system using Xception for classification and ResNet for localization, achieving a top-1 accuracy rate of 79.4% with 1,680 images divided into 131 categories[18].These studies highlight the effectiveness of deep learning algorithms like LeNet, AlexNet, and ResNet in pill image classification and feature extraction. Despite this progress, CNN-based object detection architectures like "Faster R-CNN", "SSD", and "YOLO" typically used for target detection have not yet been applied to pill recognition. Additionally, there is a lack of research focused on real-time pill identification, which is critical in high-demand environments like pharmacies, where accuracy and speed are both essential.

2. "CNN RELATED ALGORITHM ANALYSIS"

2.1 "Convolutional Neural Network" "(CNN)"

A particular kind of multilayer perceptron called a "convolutional neural network" "(CNN)" is made especially for tasks requiring visual input, such picture identification and prediction. "CNNs" start by applying filters (also known as kernels) to the picture in order to learn a limited set of parameters. These filters create a saliency map that highlights how effectively certain features are detected at specific locations within the image. As the network delves deeper, the number of nodes increases while the size of the feature maps reduces. This reduction occurs without losing critical information, thanks to the network's pooling and convolutional layers, which condense the data while retaining important features..

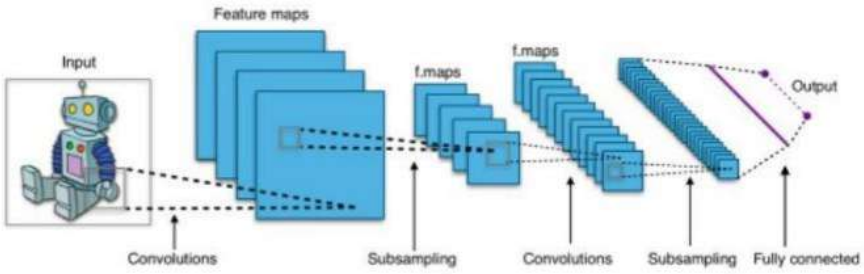

Figure 1: Architecture of *Convolutional Neural Network*

"CNNs" are composed of layers that gradually pick up increasingly intricate characteristics. The network recognizes simple edges and shapes in the first levels. The network can recognize more abstract patterns as the input moves through deeper levels, and in the last layers, it can identify things in different locations and situations. "CNNs" are very useful for vision-based applications because of their hierarchical nature, which enables them to perform effectively in a variety of visual tasks.

2.2 "Recurrent Neural Network" "(RNN)"

In order to forecast future events based on past inputs, "recurrent neural networks" (RNNs)" are made to identify patterns in sequential data. These networks

are widely used in deep learning and are modeled after the way neurons in the human brain process information. RNNs are particularly effective in tasks that require an understanding of temporal relationships, where the output is influenced by prior context or history. The capacity of "RNNs" to preserve a type of memory distinguishes them from other neural networks. They store past information, allowing them to use previous inputs when generating predictions, which is essential for sequential data processing. This phenomenon, often described as "natural cycles" or the ability to retain and utilize past information, helps “RNNs” improve the accuracy of their predictions. A notable example of ”RNN” application is in word embeddings, where the network predicts the following word in a phrase by considering the ones that come before it. Another creative use of “RNNs” is in text generation, where a network trained on literary works such as Shakespearean plays can generate text in a similar style. This form of computational creativity showcases how RNNs can replicate complex language patterns by learning from training data. The network’s ability to understand and generate text demonstrates AI’s growing role in creative fields like literature

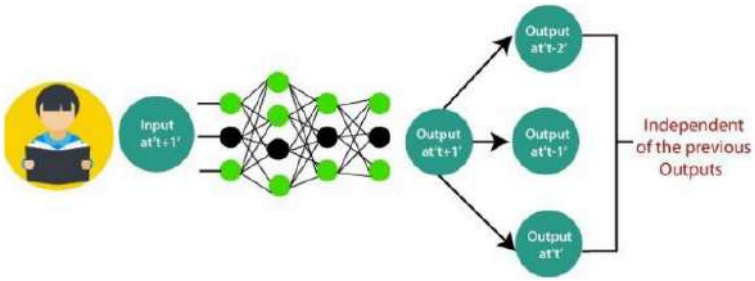

Figure 2: Architecture of RNN

2.3 “Region-based Convolutional Neural Network” “(R-CNN)”

The initial module operates independently of specific categories within the input image, generating potential detection regions that the subsequent module can analyze. This module identifies areas where the second component of the CNN can assess whether all relevant candidates are present. It extracts feature vectors of uniform length from the identified regions on three separate occasions. The second module then employs a “class-specific linear support vector machine (SVM)” to classify the objects within the identified zones.

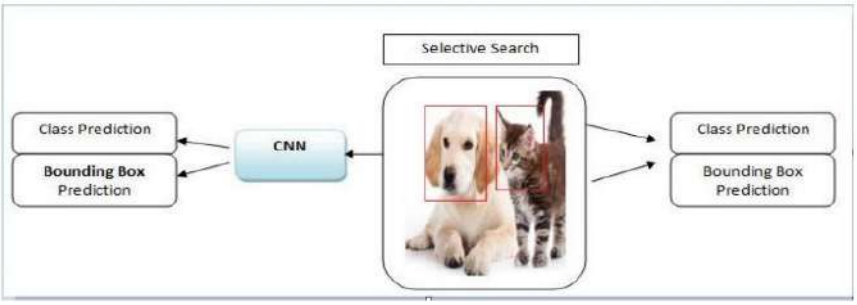

Figure 3: Recurrent Neural Network (RNN)

2.4 “Fast Recurrent Neural Network” “(RNN)”

“R-CNN” sets itself apart from “CNN”, “SVM”, and regression learning techniques; however, it struggles with long computation times. Fast R-CNN addresses this issue by utilizing the entire input image as a candidate region for CNN training. It trains the CNN by combining a single conventional feature map generated during the extraction phase [21]. "R-CNN” and “Fast R-CNN” differ primarily in their input methods": Fast R-CNN leverages functional maps for candidate regions, while R-CNN relies on pixel data from local detection areas.

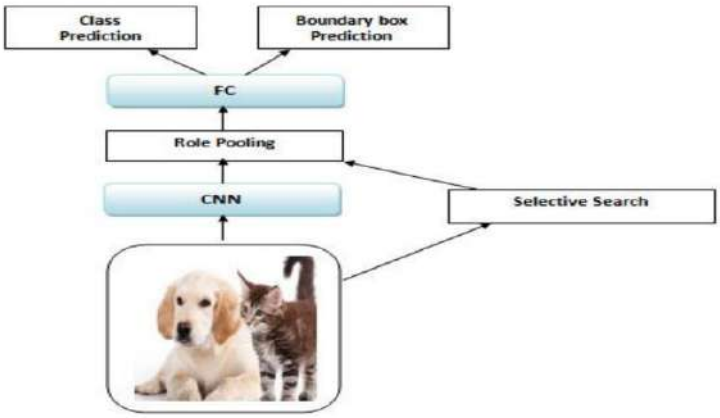

Figure 4: Architecture of FAST *Convolutional Neural Network*

CNNs collect local information around a pixel via convolution, enabling them to depict various objects and pinpoint their locations, although they do have some drawbacks. For data classification, "region-based convolutional neural networks" "(R-CNNs)" make use of deep learning regression techniques. A significant challenge has been resolved with the candidate region proposal method used by R-CNN, which is made up of three components and aims to detect objects in unfamiliar images.

2.5 "Faster R- Convolutional Neural Network"

Fast R-CNN's candidate region development module operates independently of CNN, which enhances both learning and execution speeds. However, an inefficiency issue arises when faster R-CNN is used for object posting and recognition within the same convolutional network. To address this, the Region Proposal Network (RPN) is employed to identify potential areas by estimating the resulting feature maps collectively, rather than relying on the Selective Search method[11]. This approach significantly improves feature map extraction comparing to previous "CNN models". The processes of feature map declaration and candidate region development occur within a series of networks when Compared to the input image, the feature map's declaration is smaller. In the "Fast R-CNN" and "Faster R-CNN" frameworks, various "CNN-based object detection systems", containing "SppNET", "R-CNN", and "CNN", have been analyzed to determine their effectiveness in generating candidate regions. This evaluation reveals a marked improvement in processing speed. Following the advancements made by Fast R-CNN, It is crucial to remember that total performance is significantly impacted by the development of candidate regions. Table 1 illustrates the differences in performance metrics for "R-CNN", "Fast R-CNN", and "Faster R-CNN", highlighting their respective speeds. The exploration and development of "Fast R-CNN" and "Faster R-CNN" further enhance the capabilities of "CNN" based object detection systems like "CNN", "R-CNN", and "SppNET".

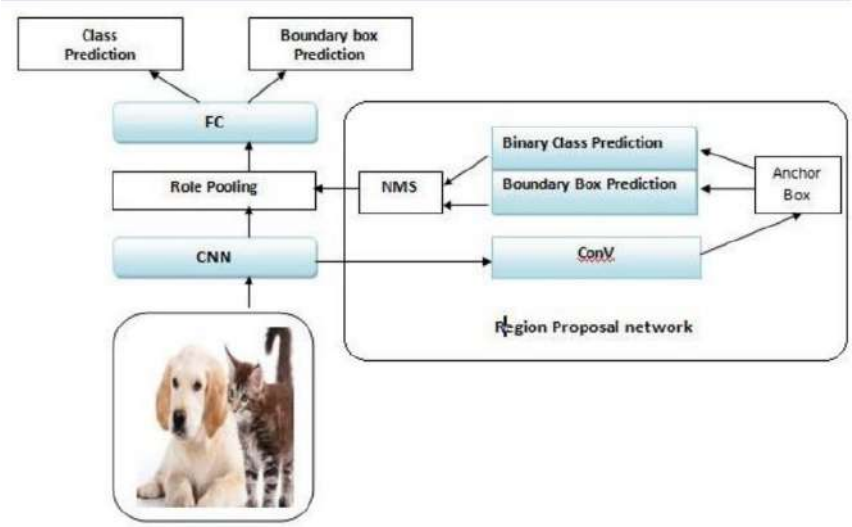

Figure 5: Architecture of FASTER R- *Convolutional Neural Network*

Table 1: Comparison of Speed for "Convolutional Neural Network"

Model	Region Proposal Method	Inference time (speed)	Key Characteristics
"R-CNN"	Selective Search (~2000 regions)	**~47 seconds per image**	- Separate CNN processing for each region - Very slow due to multi-stage feature extraction
"FAST R-CNN"	Selective Search	**~2 seconds per image**	- Shared feature extraction for the entire image - Uses ROI Pooling for proposals
"FASTER R-CNN"	Region Proposal Network (RPN)	**~0.2 seconds per image**	- End-to-end detection with RPN - Near real-time performance

3. YOLO RELATED ALGORITHM ANALYSIS

"YOLO (You Only Look Once)" is another approach for object detection [15]. This algorithm predicts objects and their locations based on a single view of the image. Using multidimensional separation and class probabilities, it handles the detection job as a regression issue rather than a classification one. The input image is represented as a grid of tensors using a "CNN". The technique "predicts bounding boxes for objects" and the associated class probabilities for each grid cell. One advantage of "YOLO" is that it can extract detection regions without the need for a separate network, which contributes to its enhanced processing speed and overall performance.

3.1 YOLO v1

The input picture is separated "into a grid of SS cells" in order to identify a particular object. The task of object detection is conducted by the grid cell whose center aligns with the midpoint of a lattice cell. Each grid is expected to predict bounding boxes, class probabilities, self-confidence scores, and associated grid cells. Given a limited number of bounding boxes, B, these predictions are organized into a tensor of dimensions SS* (5 + B). Here, C denotes the number of conditional classes associated with each cell. Equation 1 allows the model to estimate the probability of a bounding box having an item by assigning a score that represents the precision and confidence of the prediction.

$$CS = Pr(Obj) * IOU,$$

IOU stands for Intersection over Union. A cell's confidence score is 0 if it has nothing in it. The anticipated box and the ground truth are compared to determine the IOU value if an object is identified. The coordinates of each bounding box are "(x, y)", "width (w)", "height (h)", and a "confidence score." "Based on the conditional class" probabilities, all bounding boxes' class-specific confidence ratings are determined at any given moment. The probability that the bounding box contains an item is multiplied by the associated conditional class probability to determine these probabilities (as illustrated in Equation 2).

3.2 "YOLO v2"

"YOLO v2" employs a combined training algorithm that relies solely on classification data, allowing it to effectively utilize large datasets. However, Within this architecture, object detectors may also be trained. To improve both speed and accuracy, batch normalization was introduced to YOLO v1, incorporating a normalization layer that refined the initial learning process. Despite using high-resolution inputs, the size of the convolution anchor was optimized, and bounding box predictions were handled by a fully connected layer. Additionally, the methodology was thoroughly validated to enhance performance metrics. This process is executed within the anchor box, which facilitates an increase in output resolution while simultaneously compressing the network..

3.3 YOLO v3

"YOLO v4" attempts to solve the problem of developing an object detector with a smaller mini-batch size "that can be trained on a single graphics processing unit" "(GPU)". This development makes it possible to train an extremely accurate and efficient object detector with just one "1080 Ti or 2080 Ti GPU". "YOLO v4" solves this problem by allowing training with a lower mini-batch size on a single GPU. YOLO's one-stage design is often faster than two-stage detectors such as "R-CNN", "Fast R-CNN", and "Faster R-CNN," despite the latter's higher accuracy. Here, we will concentrate on the essential elements of a modern one-stage object detector.

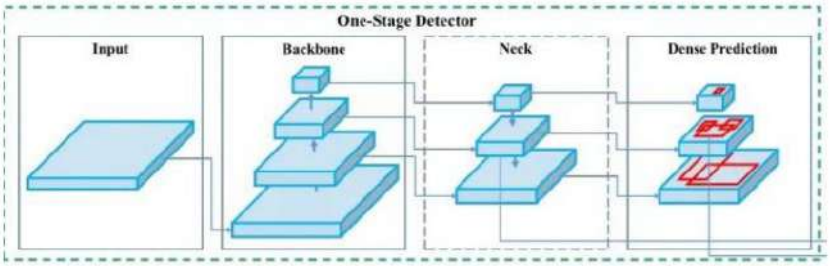

Figure 6: Architecture of YOLO v4

3.5 "YOLO v5"

"YOLO v5", which is maintained by "Ultralytics", was made available as an open-source project in 2020 by the group that created the original "YOLO" algorithm. With a number of improvements and new features, "YOLO v5" builds on the popularity of its predecessors. As seen below, it makes use of a more advanced architecture known as "EfficientDet", which is evolved from the "EfficientNet" concept. Because of its sophisticated design, "YOLO v5" can achieve higher accuracy and better generalization over a wider variety of object categories. Moreover, "YOLO v5" uses a contemporary strategy called "dynamic anchor boxes." The anchor boxes are used as the centroids of the "clusters created by first clustering" "the ground truth bounding boxes" "using a classification technique". Consequently, the anchor boxes better depict the dimensions and form of the detected objects.

3.6 "YOLO v6"

The "CNN" architecture that "YOLO v5" and "v6" use is one of the main distinctions between the different versions. In contrast to "YOLO v5"s "EfficientDet" design, "YOLO v6" uses "EfficientNetL2", a variation of the "EfficientNet" architecture, which provides a more efficient computational model with fewer parameters. This enables "YOLO v6" to attain state-of-the-art performance on a range of object identification tests. Furthermore, "YOLO v6" adds a brand-new function known as "dense anchor boxes." YOLO v7 beats other object detection algorithms in terms of accuracy, averaging "37.2% at an IoU threshold of 0.5" on the popular "COCO" "dataset", which is comparable to other leading object recognition technologies.

3.7 YOLO v8

The release of "YOLO v8," which has more features and better performance than previous iterations, was verified by "Ultralytics" at the time this article was published. While the framework still supports earlier versions of "YOLO", the new "API" in "YOLO v8" simplifies the inference and training procedures for "GPU" and "CPU" devices. "The development team is" now working on a scientific publication that will offer a thorough examination of the model's functionality and design.

4. "Single Shot multibox Detector" "(SSD)"

The "Single Shot Detector" "(SSD)" can recognize many items in a picture in a single step, unlike the "R-CNN" series and other techniques that use "regional proposal networks" "(RPNs)" to produce region proposals and identify objects inside those proposals in a two-step process. "SSD" is able to outperform two-step "RPN" based methods due to its efficiency. For example, "R-CNN" works at just "7 frames" per second" "(FPS)" yet achieves a better mean "Average Precision" "(mAP)" of "73.2%" than "YOLOv1", which gets "63.4% mAP at 45 FPS".

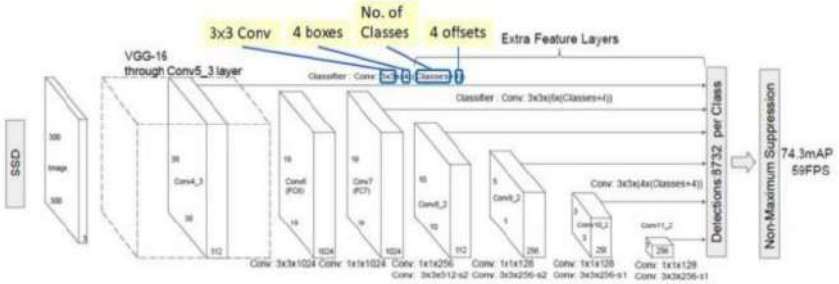

Figure 7: Architecture of SSD

The "SSD300" model achieves "74.3%" "mean Average Precision (mAP)" at "59 frames per second (FPS)", while the "SSD500" model reaches "76.9%" "mAP at 22 FPS", both outperforming previous models. The results presented below are based on training data from both the "PASCAL VOC" "2007" and "2012 datasets", with the "mAP" being calculated using the "PASCAL VOC 2012" testing set. The accompanying graph illustrates the performance of "SSD" with input images sized "300 × 300" and "512 × 512". Additionally, results for "YOLO" include images sized "288 × 288", "416 × 416", and "544 × 544". Generally, higher-resolution images lead to improved "mAP" for the same model; however, they also require more processing time for evaluation..

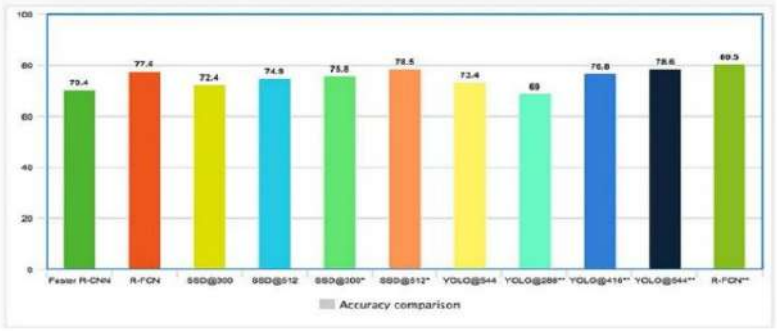

Figure 8: Accuracy comparison of three models(YOLO, CNN, SSD)

The choice of feature extractors and the resolution of input images significantly affect processing speed. The following data highlights the highest and lowest "frames per second" "(FPS)" recorded from relevant sources. However, these results may be heavily influenced due to testing conducted at different "mean Average Precision ""(mAP)" levels. With several models, "object detection" is a well-known field in computer vision. It's important to note that not all models are created equally. Although each model discussed in this video has its own strengths and weaknesses, our focus is on the most relevant ones. A comparison is made with a "Faster R-CNN" model from the Two Shot detector family, as well as "YOLO's" "single-shot" variations and "Single Shot Detectors" "(SSD)".

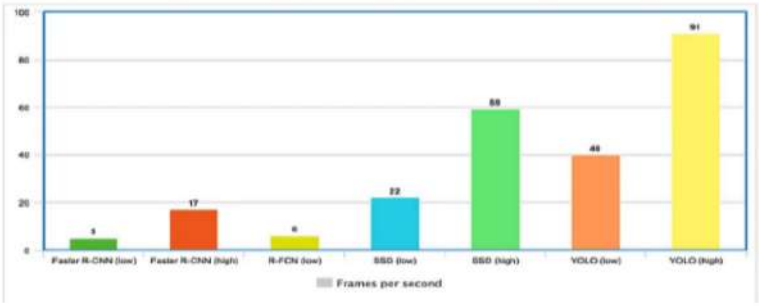

Figure 9: Speed of comparison of three models (YOLO, CNN, SSD)

In our comparison of models, we prioritized inference speed, specifically the number of frames each model could process per second. We assessed which model produced the greatest results in with respect to "accuracy" and "dependability". We also took into account the model's ease of usage, placing particular emphasis on the frameworks needed for implementation (such as "OpenCV", "PyTorch", or "TensorFlow") and the minimal amount of code necessary to enable the model's detection capabilities.

Table 2: "Comparison of" "Faster RCNN" & "SSD" & "YOLO"

	Speed	Accuracy	Ease of implementation
Faster RCNN	Bad	Good	Bad
SSD	Good	Good	Bad
YOLO	Good	Good	Good

CONCLUSIONS

We explored a "CNN" based object detection system that incorporates "YOLO". Compared to other classifiers, "YOLO" stands out as a suitable choice for access rooms due to its straightforward design and ability to learn from the entire image, making it practical for real-world applications. Unlike

traditional methods, "YOLO" enhances real-time object detection by optimizing processing time and directly improving detection performance during training with real functions. Throughout our investigation, we encountered challenges related to dynamic label assignment and issues with module replacement. To address these challenges, we propose enhancing object recognition accuracy by implementing a trainable bag-of-freebies approach. The application phase is a critical step that determines the program's effectiveness, necessitating evaluation alongside an independent algorithm.

REFERENCE

1. Asifullah Khan, Anabia Sohail, Umme Zahoora, AqsaSaeed Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks", Computer Vision and Pattern Recognition, Available at https://arxiv.org/ftp/arxiv/papers/1901/1901.06032.pdf [Accessed Mar. 13, 2020].
2. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich Feature Hierarchies for ccurate Object Detection and Semantic Segmentation", IEEE Conference on Computer Vision and Pattern Recognition, pp.580-587,
3. Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, Matti Pietikainen, "Deep Learning for Generic Object Detection: A Survey", International Journal of Computer Vision, vol.128, pp.261-318 2020.
4. Kaimin He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition", European Conference on Computer Vision, Part 3, pp.346-361,2014.
5. Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhizi Feng, Rong Qu, "A Survey of Deep Learning-based Object Detection", IEEE Access, vol.7, pp.128837-128868, , 2019.
6. David G. Lowe, "Distinctive Image Features from ScaleInvariant Keypoints", International Journal of Computer Vision, vol.60, pp.91-110, 2004.
7. Juan Du, "Understanding of Object Detection based on CNN Family and YOLO", Journal of Physics, Conference Series, vol.1004, issue.1, 2018.
8. Tsung-Yi Lin, Priya Goyal, Roos Girshick, Kaiming He, Piotr Dollar, "Focal Loss for Dense Object Detection", International Conference on Computer Vision, pp.2999- 3007, 2017.
9. Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "Speeded-Up Robust Features (SURF)", Computer Vision and Image Understanding, vol.110, issue.3, pp.346-359, 2008.
10. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Communications of the ACM, vol.60, no.6, 2017.
11. Yurong Yang, Huajun Gong, Xinhua Wang, Peng Sun, "Aerial Target Tracking Algorithm Based on Faster RCNN Combined with Frame Differencing", Aerospace, vol.4, no.32, 2017.
12. Kwanghyun Kim, Sungjun Hong, Baehoon Choi and Euntai Kim, "Probabilistic Ship Detection and Classification using Deep Learning", Applied Sciences, vol.8, no.6, 2018.
13. Rohith Gandhi, "R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms," 2018. Available at https://towardsdatascience.com/r-cnn-fast-r-cnnfasterr- cnn-yolo-object-detection-algorithms36d53571365e. [Accessed: Mar. 13, 2020].
14. N. Dalal, B. Triggs, "Histograms of Oriented Gradients
15. Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection" ,2020.
16. Mingxing Tan Ruoming Pang Quoc V. Le Google Research, Brain Team. "EfficientDet: Scalable and Efficient Object Detection". 2020
17. Chien-Yao Wang , Alexey Bochkovskiy, and Hong-Yuan Mark Liao, Institute of Information Science, Academia Sinica, Taiwan, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. 2022.
18. https://jonathan-hui.medium.com/object-detection-speedand-accuracy-comparison-faster-r-cnn-r-fcn-ssd-andyolo-5425656ae359.

HOW TO CITE: Ghansham More, Omkar Patil, Omkar More, Mihir More, Samadhan Suryavanshi, Manisha Mali, Comparison of Object Detection Algorithms CNN, YOLO and SSD, Int. J. Sci. R. Tech., 2024, 1 (11), 137-144. https://doi.org/10.5281/zenodo.14186397