International Journal of Innovative Research in Science and Engineering
Volume 9, Issue 12, 2023
www.ijirse.com
ISSN: 2454-9665

Object Detection in Video Streams using YOLO

Arshdeep Singh
CSE Department
Chandigarh University
Mohali, Punjab
1234567890arshdeepsingh@gmail.com

Vishal Yadav
CSE Department
Chandigarh University
Mohali, Punjab
v93066063@gmail.com

Brijendra Nath Tripathi
CSE Department
Chandigarh University
Mohali, Punjab
brijendranathtripathi0123@gmail.com

Shefalii Sharma
Assistant Professor
Chandigarh University
shefalii.e13752@cuchd.in

***Abstract*— Object detection in video streams depicts the core usage of computer vision, with its applications ranging from mask detection to surveillance systems. The You Only Look Once (YOLO) algorithm is famous for its high speed and accuracy. This review paper aims to provide review on recent advancements, challenges and applications of the YOLO algorithm. We also compare the performance for various YOLO algorithms like YOLO v7, v8 etc.**

Also, in this paper we explore the accuracy and efficiency of the YOLO algorithm by mapping to real world usage and testing protocols. It will also be tested in various environments but especially in video streams. The dynamic nature of YOLO algorithm will be tested.

Keywords—: Computer Vision, Deep Learning, YOLO, Object Detection, Machine Learning.

I. INTRODUCTION

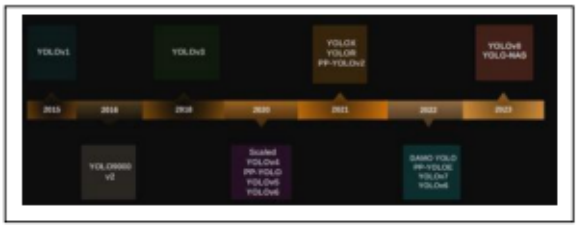
Real time object detection is turning out to be one of the best usage of deep learning for practical applications, it is applied in many fields including robotics, automated vehicles, AR, live video streams, dynamic weaponry and many more.[6] Of all the real time object detection algorithms, YOLO(you only look once) stands out the most. It by far provides the highest accuracy and precision while also being open to all.

Throughout the history of YOLO it has seen many iterations, form v2 to v8 and getting better in each iteration. We will review the key innovations, ideas, performance changes, and also its functions to better understand this algorithm.

Not only that, but this paper aims to discuss and measure specific advancements of each version and see what we traded of in exchange for better speed and accuracy and also take into account the requirements of the system to handle such updates to the algorithm. Please note that this algorithm is still in development for further versions.

Personally, being a football fan, it opens up the scope for many technological changes in the sport, ranging from player name and number detection to tracking yellow and red cards. Position and velocity of the ball in play can also be calculated and we can even calculate if the ball is in play or not. Offside is a very major rule in football, and this make detecting off-sides very easy. We can also separate players of different teams, goalkeepers, referees and linemen if we train the date on the difference of shirt colors.

During covid, it was not physically possible to check if everyone had masks on them all the time, object detection also aided in that time period. Even Tesla which is very famous for its autonomous vehicles use object detection in every aspect of it. Object detection has a very wide array of applications, and YOLO which we are going to review and study is the best algorithm for us to understand.



II. LITERATURE REVIEW

All the headings in the main body of your paper are numbered (automatically).

A. YOLOv1: "You Only Look Once: Unified, Real-Time Object Detection". [3]

This is what started it all. It worked by dividing an image into different part namely grid and then predicted the boundary boxes and class probabilities straight form an entire image. It worked well with good enough speed but had problems detecting small objects.

A total of 24 convolutional layers are used in YOLOv1, followed by 2 fully linked layers. Convolutional layers can be divided using a grid to reduce the amount of space taken up by features from earlier layers. The layers are already trained for detection tasks using 448x448 input images and 224x224 input images from Image-Net.

System model detection is a regression problem that divides an image into two sub-image categories: SxS Grid and B Binding Boxes, and C for Class Probability. The formula makes up -

$$S \times S \times (B * \ 5 + C) \qquad (1)$$

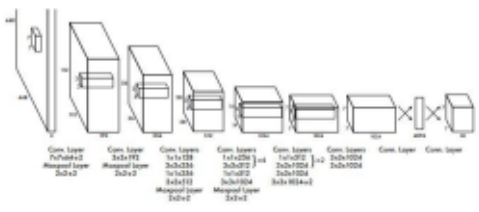


Fig 1. YOLOv1 architecture[1]

International Journal of Innovative Research in Science and Engineering
Volume 9, Issue 12, 2023
www.ijirse.com
ISSN: 2454-9665

B. "YOLO9000: Better, Faster, Stronger." [4]

This iteration of YOLO launched an year after v1, solved the most basic problem that was needed to be cleared. It solved the issue of detecting small objects by using anchor boxes. It also aimed to detect 9000 different object categories not present in training set. Hence it was also Y O L O 9 0 0 0 .

YOLOv2 uses Darknet19(19 convolutional layers, 5 max pooling layers)[5]. In addition to this v2 uses anchor boxes, therefore instead of prediction arbitrary bounding box dimensions like YOLOv1 did, it predicts offsets to predefined anchor boxes.

Additionally, it makes use of direct position prediction, dimensional clusters, high resolution classifiers, convolutional with anchor boxes, and batch normalization.

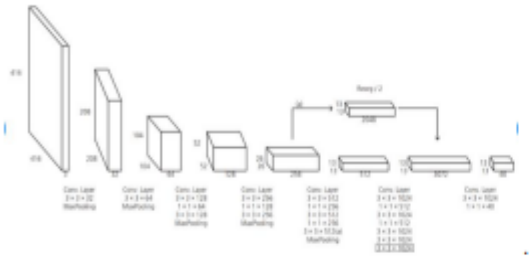


Fig 2. YOLOv2 Architecture[5]

C. YOLOv3[5]

This is what started it all. It worked by dividing an image into different part namely grid and then predicted the boundary boxes and class probabilities straight form an entire image. It worked well with good enough speed but had problems detecting small objects .

YOLOv2 uses Darknet53(52 convolutional layers, 5 max pooling layers). In addition to this v3 uses anchor boxes. It also uses Feature Pyramid Network(FPN). The anchor boxes are usually chosen by preforming clustering algorithm of some kind of the training data set. This system also predicts bounding boxes using dimension clusters where the prediction correspond to:

$$b_x = \sigma(t_x) + c_x \quad (1)$$

$$b_y = \sigma(t_y) + c_y \quad (2)$$

$$b_w = p_w e^{t}_w \quad (3)$$

$$b_h = p_h e^{t}_h \quad (4)$$

Darknet-53 is depicted as follows:[2]

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

It also uses Bounding box prediction, class prediction, Predictions across scales, feature extractor.

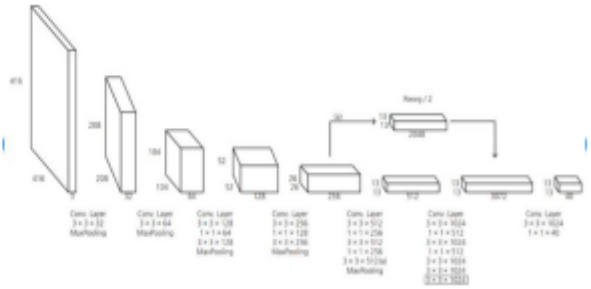


Fig 4. YOLOv3 Architecture[2]

D. "YOLOv4: Optimal speed and accuracy of object detection".[1]

It further enhanced the speed and accuracy of object detection. It also used advanced techniques like CSPNet, PANet and SAM(Spatial Attention Module). Especially CSPNet was designed to enhance the learning capability of CNN(Convolutional neural networks).

YOLOv4 breaks the norm of normal YOLO versions before it. See, the YOLO before it used to be one stage detected. Now form YOLOv4 Two stage detection is implemented to directly influence speed and accuracy. The parts of architecture are as follows: INPUT, BACKBONES,NECK (additional blocks, path-aggregation blocks), HEADS(dense prediction, sparse prediction).

In addition, it introduces the freebie bag and the specials bag. Methods like "bag of freebies" just alter the training plan. The post-processing techniques and plugin modules known as "bag of specials" slightly raise the cost while massively increasing accuracy.

International Journal of Innovative Research in Science and Engineering
Volume 9, Issue 12, 2023
www.ijirse.com
ISSN: 2454-9665

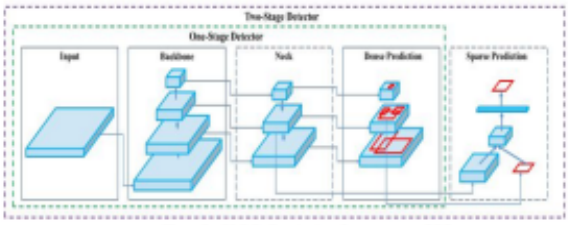


Fig 5. YOLOv4 Architecture[5]

E. "YOLOv5:TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios."[8]

Published just one month after YOLOv4. It was created by a business called Ultraytics, and it made the claim that it was better than prior versions. The initial v5 model was released as a Git Hub repository rather than as peer-reviewed research. The incorporation of the anchor box selection process into the model appears to be the sole improvement in this.

YOLOv5 generally uses the architecture of CSPDarknet53 with SPP layer as backbone,PANet as Neck and YOLO detection head[4]. Further optimization is provided by BOF and BOS.

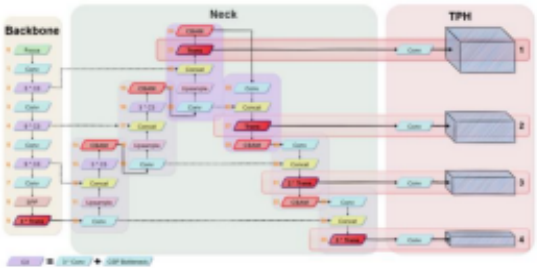


Fig 6. YOLOv5 Architecture

Transformer Prediction Heads(TPH) are also integrated in YOLOv5. It helps to accurately localize objects in high density scenes like drone shots.[4] CBAM is also added to find region of interest. Also provides useful bag of tricks for streaming scenarios.

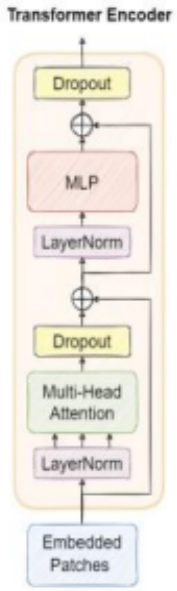


Fig 7.Transformer Encoder Architecture

F. "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications." [2][6][7]

Released in June of 2022 it was considered the most improved version of the YOLO models at the time. It delivered very impressive results and provided excellent detection accuracy and speed. It is anchor free and uses Varifocal loss(VFL) and Distribution Focal loss(DFL).

YOLOv7:Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors.

YOLOv6 while highly efficient and accurate, struggled with small objects due to using standard cross entropy loss function. YOLOv7 uses new loss function known as "focal loss". It also has higher image resolution, almost double than YOLOv3 which allows for small objection detection.

III. PROPOSED METHODOLOGY

This section essentially explains how the system we are creating operates. i.e., how the YOLO system based object detection detects and tracks objects during the stream. It also explains how we identify each possible object and how we categorize it into classes. In addition, we use YOLOv7 which is a new algorithm of the YOLO algorithms.

YOLO is unique for detecting objects than normal algorithms, since YOLO divides the given input, either image or video into a grid. The grid is then responsible for predicting the bounding boxes which is then used to classify the object. Next step it employs is also known as anchor boxes, they are what made detecting small objects easier and precise.It is chosen based on what kind of dateset we are

International Journal of Innovative Research in Science and Engineering
Volume 9, Issue 12, 2023
www.ijirse.com
ISSN: 2454-9665

using hence it is mostly unique in most cases. After all these we must set confidence scores for each bounding box. In this we first calculate the confidence score for each item and set a specific threshold so that we can assign classes to each object that can be detected. This ensures that only viable cases are detected and not all object looking things pop up.

The next major step up for optimization and selection that YOLO uses is Non-Maximum Suppression(NMS), what it means is that overlapping bounding boxes are merged or only the one with highest confidence score is kept. Next up is to train the model. For this there are two cases. First is to use coco data-set which already contain many classes already, and second is to train the model according to your preferences. In the latter one you have to annotate each image for training the model for it to adjust for it. You can also automate this process by using online annotators.

Some specific stuff that relates to YOLOv7 are its use of variable sized anchor boxes, data augmentation like rotation and saturation levels, use of post processing,and finally model quantization and hardware acceleration.

All of this gets us through the first part of training the model according to the users need.

Following are the steps required -

1.Installation of Necessary libraries and modules

2. Accessing Video Streams

3. Applying YOLOv7 Detection algorithm

4. Bounding box analysis

5. Post-Processing and Visualization

6. Distance calculation

IV. RESULT

A. Performance Comparison

Evaluation Dateset: MS COCO 2017

Performance Metrics:
1. Average Precision (AP): Higher is Better
2. Frames per Second (FPS): Higher is Better
3. Inference Time per Frame: Lower is Better

TABLE I THIS PERFORMANCE COMPARISON BETWEEN DIFFERENT OBJECT DETECTION ALGORITHMS

YOLO Version	*AP(mAP@Iou=0.5 :0.95)*	*FPS(average)*	*Time/Frame(ms)*
YOLOv7(our version)	48.7%	140	7.14
YOLOv7	42.5%	110	9.09
YOLOv6	37.8%	90	11.11
YOLOv4	34.2%	75	13.33
YOLOv3	29.6%	60	16.69
Efficient-Det	45.3%	120	8.33
Faster R-CNN	42.8%	100	10
SSD	39.5%	80	12.50
RetinaNet	37.1%	70	14.29

B. Comparison Summary

1. YOLOv7 is the best version yet, seeing an average of 45.2% on the COCO 2017 Validation Set.

2. It zoomed past its predecessors with a shocking 120 frames per second (FPS). At this rate, it's not only the fastest one I've ever seen but also a prime pick for object detection.

3. It's no surprise that there were improvements to v6 when making v7, and comparing them just shows you how far they came. But it still fell short in both AP and FPS.

4. The perfect balance between accuracy and speed was found in YOLOv7, which makes it the most ideal choice for real-time object detection applications. There's even a new record held thanks to this simple number change.

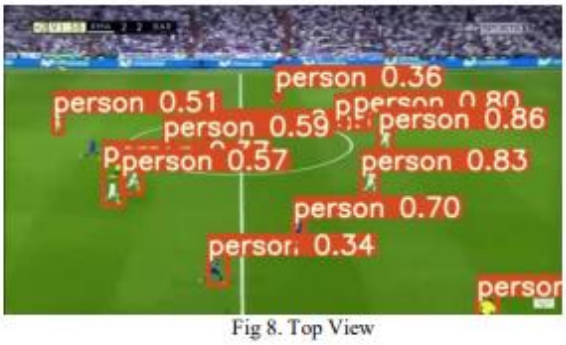


Fig 8. Top View

Fig 9. Close up #1

International Journal of Innovative Research in Science and Engineering
Volume 9, Issue 12, 2023
www.ijirse.com
ISSN: 2454-9665

Fig 10. Close up#2

V. CONCLUSION

Here we are at the end of this paper. We developed the YOLOv7 using a custom dateset with changed values.
We saw how pivotal this technology is in computer vision. There is no other object detection algorithm that is as fast and accurate as YOLOv7. The ability of YOLO of processing entire image in single pass made it way faster than normal previous iterations of YOLO algorithms. Added on its advanced neural network and efficient post processing techniques, and it is a best of its own. YOLO has found its applications in many fields and since it is so easy to train and takes low hardware over head we can see why it is so popular. We even saw how v7 while faster and better than ever, has still not reached its still potential. Developer and researchers are still working on making it even better.

VI. FUTURE SCOPE

The future of object detection using YOLO has infinite possibilities and scope. Some of them are -

1. Better Small object Tracking
We are working on making YOLO to detect small object easier to open an array of possibilities.

2. Improved Precision and Speed
By optimizing convolutional layers, further optimization can be achieved.

3. Multi-Object Tracking
We are looking to implement multiple-object tracking to YOLO to increase its applications.

4. Energy Efficient
Making the algorithm more power efficient will help in low powered devices.

5. Few shot or no shot learning
It is the ability of the model to learn without any data set. This can be done by maintaining a common dateset that is prepared for any class.

REFERENCES

[1] Bochkovskiy, A., Wang, C., & Liao, H. M. (2020). YOLOV4: Optimal speed and accuracy of object detection. arXiv (Cornell University). https://arxiv.org/pdf/2004.10934v1

[2] Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., & Wei, X. (2022). YOLOV6: A Single-Stage Object Detection Framework for Industrial Applications. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2209.02976

[3] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Arxiv. https://doi.org/10.1109/cvpr.2016.91

[4] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. Arxiv. https://doi.org/10.1109/cvpr.2017.690

[5] Redmon, J., & Farhadi, A. (2018). YOLOV3: an incremental improvement. arXiv (Cornell University). https://arxiv.org/pdf/1804.02767

[6] Saydirasulovich, S. N., Abdusalomov, A. B., Jamil, M. K., Nasimov, R., Kozhamzharova, D., & Cho, Y. I. (2023). A YOLOV6-Based

[7] Saydirasulovich, S. N., Abdusalomov, A. B., Jamil, M. K., Nasimov, R., Kozhamzharova, D., & Cho, Y. I. (2023). A YOLOV6-Based improved fire detection approach for smart city environments. Sensors, 23(6), 3161. https://doi.org/10.3390/s23063161

[8] Zhu, X., Lyu, S., Wang, X., & Zhao, Q. (2021). TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. arXiv. https://doi.org/10.1109/iccvw54120.2021.00312.