2023 International Conference for Advancement in Technology (ICONAT)
Goa, India. Jan 24-26, 2023

Safety Helmet Detection: A Comparative Analysis Using YOLOv4, YOLOv5, and YOLOv7

Siddhi Chourasia
Department of Electronics Engineering
Shri Ramdeobaba College of Engineering and Management
Nagpur, India - 440013
Siddhichourasia07@gmail.com

Rhugved Bhojane
Department of Electronics Engineering
Shri Ramdeobaba College of Engineering and Management
Nagpur, India - 440013
rhugvedb@gmail.com

Prof. Lokesh Heda
Department of Electronics Engineering
Shri Ramdeobaba College of Engineering and Management
Nagpur, India - 440013
hedalm1@rknec.edu

2023 International Conference for Advancement in Technology (ICONAT) | 978-1-6654-7517-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICONAT57137.2023.10080723

***Abstract*—Safety helmets are of utmost importance to workers' lives as the most fundamental form of protection. However, safety helmets are frequently not worn as a result of a lack of safety awareness. Utilizing outdated manual inspection techniques and video monitoring to check if employees are wearing helmets results in missed inspections and poor punctuality. As object detection technologies advanced, the YOLO family of detection algorithms, which have extremely high speed and precision, were applied in multiple detection segments. In this paper, we compare and analyze the three models of the YOLO family, the YOLOv4, the YOLOv5, and the YOLOv7 for helmet detection. A publicly available dataset of 5000 images was collected and annotated. Our results have shown that the YOLOv7 accomplishes an mAP of 96.4% which is 1.36% better than the YOLOv5 and 3.00% better than the YOLOv4. The results also show that the YOLOv7 has an average detection time of 12.4 ms, outperforming that of the YOLOv4 and the YOLOv5. Both in terms of accuracy and speed, the YOLOv7 exceeds both models, making it possible for even greater real-time object detection.**

Keywords— Human Detection, Object detection, YOLOv4, YOLOv5, YOLOv7,

I. INTRODUCTION

In today's fast-paced work, construction activities have significantly increased. Accidents caused at construction sites due to a lack of strict safety regulations are occurring very often, causing harm to lives and property. Safety helmets are the most fundamental form of protection for a worker's life. However, safety helmets are not typically used by construction workers because of a lack of safety knowledge. Utilizing outdated manual inspection techniques and video monitoring to check if employees are wearing helmets results in missed inspections and poor punctuality. To avoid casualties, real-time detection of safety helmets is of prime importance.

In recent years, there have been massive improvements and developments in computer vision and deep learning. They offer a wide array of object detection applications. Redmon et al.'s[1] YOLO identification approach for target detection uses a single convolutional neural network instead of the candidate region's intermediary phases. The YOLO was further improved in the year 2018 with Redmon et al proposing the YOLOv3. In 2020, Alexey A.B.[6] further progressed and made efficient improvements based on YOLOv3. In YOLOv3, PANet is employed as a parameter assemblage approach. Additionally, YOLOv4 suggests the following novel enhancement techniques:

1) Mosaic and Adversarial training; a novel data augmentation technique.
2) Using GA, choose the ideal hyper-parameters.
3) SAM and PAN should be improved to make the current approaches more effective for training and inference.[6]

The latest features included WRC, CSP, CmBN, SAT, Mish activation, Mosaic data augmentation, CmBN, DropBlock regularization, and CIoU loss. A few of them were combined to engineer cutting-edge results: On the Tesla V100, 43.5% AP (65.7% AP50) for the MS COCO dataset [14] at a real-time speed of 65 FPS [18]. YOLOv4 maintains accuracy and detection speed while drastically improving detection results.

Glenn Jocher introduced YOLOv5 using the Pytorch framework not long after the release of YOLOv4. YOLOv5 gets the majority of its performance boost from PyTorch training procedures, while the model architecture remains similar to YOLOv4.[10] With the introduction of the adaptive anchor's calculation in YOLOv5, the object detection algorithm no longer has to build the anchors using k-means prior to training. This somewhat lessens the object-detecting method's complexity [3]. The backbone, neck, and output components make up the three parts of the YOLOv5 network structure. Similar to YOLOv4, it also adopts the CSP structure, and the neck segment acquires the FPN + PAN edifice and uses the newer focus technology[3]. The YOLOv5 is divided into five models YOLOv5(n,s,m,l,x.) For this research, we will be using the YOLOv5x model. The term COCO AP Val refers to metrics with mAP@0.5:0.95 measurements made on the 5000-image COCO val2017 dataset using different inference sizes ranging from 256 to 1536. The YOLOv5x achieves a mAP@0.5:0.95 of 50.7 and mAP@0.5 of 68.9. Experimental data has shown the speed of YOLOv5s reaching an impressive 110 FPS while the YOLOv5x has a speed of 21 FPS. Using the pre-training weight of the trainable target detector, YOLOv5x achieves an mAP of 94.7%, demonstrating the efficacy of helmet detection-based systems YOLOv5.[1]

For computer vision tasks, the quickest and most accurate real-time object recognition model is YOLOv7. YOLOv7's official article is titled "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors". Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao published it in July 2022. Cutting-edge real-time object detections can be effectively reduced by up to 40% of their parameters and by 50% of their computation using YOLOv7, which leads to faster inference times and better

detection precision.[11] On GPU V100, in comparison to all other true object detectors with 30 or more frames per second, YOLOv7 has the higher precision (56.8% AP) outperforming all other known object detectors in the 5-160 FPS range. The MS COCO dataset [14] is the only dataset used to train the YOLOv7 from scratch; no other datasets or pre-trained weights are used.[4]. The YOLOv7 also supports instance segmentation by integrating BlendMask and pose estimation by integrating YOLO-Pose in it.

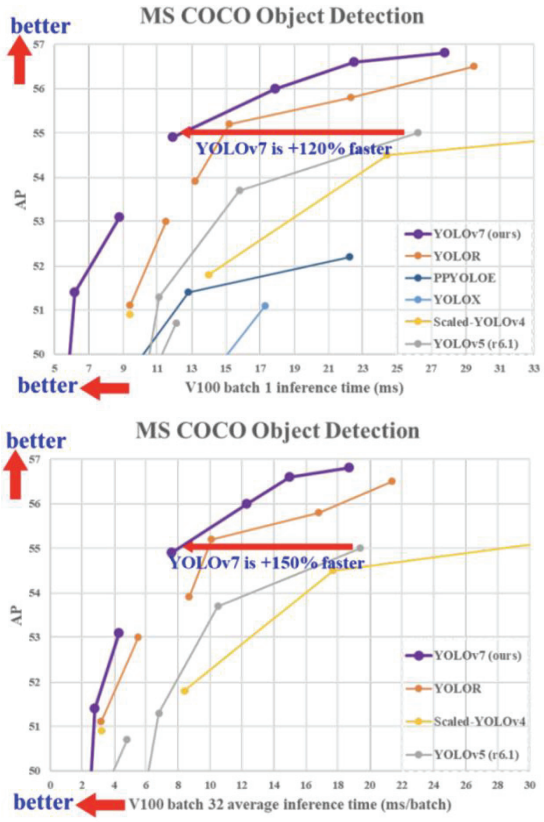

Fig. 1. Comparison of real-time object detectors [4]

The paper attempts to make the following contributions:

1. A dataset of 5000 images was downloaded from Kaggle (source:https://www.kaggle.com/datasets/andrewmvd/hard-hat-detection).The correct helmet was annotated as "helmet" on a head.
2. The YOLOv4, YOLOv5, and YOLOv7 models are trained and evaluated using the same performance metrics.
3. The three models are analyzed and compared in the end.

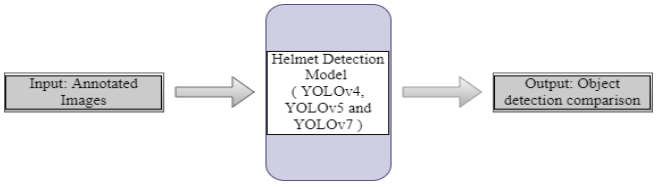

Fig. 2. The flow of the paper

The related work is introduced in the following segment on object detection algorithms and helmet detection.We then discuss the architecture of the models and their distinctive features.We then conduct the experiments, analyse and compare the results. Finally, we put a conclusion to our task.

II. Related Work

Current segment focuses on the related work done on various object detection methods, especially helmet detection.

Object detection is a task that combines object classification and localization. Current object detectors are classified into two types: networks that separate the tasks of determining object location and classification, such as Faster R-CNN, and networks that predict bounding boxes and class scores at the same time, such as the YOLO and SSD networks.

The two-stage algorithm is theoretically more precise than the one-stage approach. Faster-RCNN is a two-step target detection system that includes a convolutional layer, a region candidate network, a region of interest pooling layer, and a classification layer. The convolutional layer's fundamental convolutional layer, activation layer, and pooling layer are utilized to draw out features and build feature maps.[1]

Single-stage target detectors are quicker than two-stage target detectors. The YOLO series algorithms are truly remarkable in object detection and use single-stage target detection. The target is detected using regression and a category is determined while the target is positioned. The YOLO series uses only deep convolutional layers. The input images help make up the network. After performing convolution on the images, the target category and position are returned directly.

A. YOLOv4

With the help of the MobileNet network and the YOLOv4 algorithm, the helmet detection application proposed in this paper achieves both good detection accuracy and speed. The FPS and mAP values discovered in this paper using transfer learning and tuning parameters on the datasets are 27.3 6% and 94.47%, respectively, exceeding the research findings of some related works.

A YOLOv4 helmet detection application for mobile networks is also suggested in this paper by combining the YOLOv4 and MobileNetv3 networks.With mAP and FPS values of 91.47% and 42.58%, respectively, it meets the precision and real-time demands of the presently used hardware.[3]

The safety helmet detection model proposed by Deng Benyang, Lei Xiaochun, and Ye Miao makes use of YOLOv4 suggesting clustering the data set utilizing the K-means algorithm to extract the proper a priori frame dimensions center and more precise edge information. Using a multi-scale training technique, the model's capacity to adapt to diverse functional scales is improved. The experimental findings demonstrate that the model's detection accuracy and speed were enhanced compared to YOLO v4 in the helmet-wearing detection task, with the model's mAP value reaching 92.89% and the detection speed reaching 15 f/s.[6]

The YOLOv4 target detection algorithm is used in this study to identify workers wearing helmets in complex settings and on real-world construction sites. This article

includes a helmet-based human body model. The human body and the detected helmet form a one-to-one relationship as a result of training. According to the findings of the study, the model has a 93% accuracy rate.[4]

This paper combines the YOLOv4 algorithm with an image super-resolution at the end of the input to improve image quality and reduce noise. The remaining backbone network blocks are then replaced with the CSPDarknet53 architecture, resulting in less work and fewer network structure parameters. The model detects a picture at 416 416 in 3.0 ms and achieves 93.3% accuracy on average, a 7.8% improvement over the previous method.[10]

In terms of training, YOLOv4 uses Mosaic data enhancement, Label Smoothing, CIOU, and cosine annealing decay learning rate which are small tricks to make YOLOv4 better performance in target detection. The Mish activation function [3] is calculated as follows

$$f(x) = x^* \tanh(\ln(1 + e^x)) \quad (1)$$

B. YOLOv5

This paper introduces an enhanced YOLOv5 framework. By using the DIoU-NMS in place of the NMS, which ignores the overlap area and center distance of the two boxes, the projected bounding box is suppressed more accurately. Another improvement is adding a functionality detection scale to target smaller objects as small as 4*4 pixels. According to the experimental findings, the suggested approach greatly increases accuracy when contrasted to the network structure of the YOLOv5. It also has a detection speed of 98 frames per second, for real-time detection, which is quick enough. [2]

A feature map change of 160*160 is proposed in the YOLOv5 and used for detecting smaller objects.

To obtain a more suitable priori anchor box, the helmet data set is re-clustered using K-means clustering. According to the experimental findings, when compared to the original model, the accuracy of helmet recognition improved by 2.4%, reaching 94.6%, and the average precision of the upgraded YOLOv5 algorithm increased by 2.9%, reaching 95%.[5]

A human body posture estimate network OpenPose and an object detection network YOLOv5 are used to find safety harnesses. According to the experimental results, the YOLOv5 model can identify safety harnesses with an accuracy rate of 89%. This study's detection method can guarantee that the detection program is capable of detecting safety harnesses with accuracy while also lowering the output results' false alarm rate, which has an elevated application value.[9]

The suggested end-to-end helmet monitoring system in this research uses a super-resolution (SR) reconstruction-driven approach to identify helmets. Super-resolution reconstruction and detection make up the two modules that make up the monitoring system. The former generates high-resolution photos using the SR technique, while the latter recognizes helmets. The resolution and quality of the images are increased using a super-resolution reconstruction module. The detection module, which comprises of YOLOv5, is then given the processed photos to find helmets.The two modules are trained separately first, then tuned together. The proposed model achieves an mAP of 0.501 compared to other models fine-tuned to the YOLOv5 using interpolation and DRN.[11]

The Safety Helmet dataset with 5K images (SHEL5K) dataset, which is presented in this paper, is an improved version of the SHD dataset. There are six fully labelled classes in the suggested dataset: face, helmet, head, head with a helmet, person, person with a helmet, person without a helmet, and person.The suggested dataset was evaluated using many cutting-edge object identification models, including YOLOv3, YOLOv4, YOLOv5-P5, the Faster Region-based Convolutional Neural Network with the Inception V2 architecture, and YOLOR.[12]

C. YOLOv7

YOLOv7 derives its architecture from YOLOv4, Scaled YOLOv4, and YOLO-R. In order to create a new and superior YOLOv7, additional tests were conducted using these models as a foundation. Every YOLOv7 model is better than earlier object detectors in terms of speed and accuracy in the 5 to 160 frame-per-second range.[4] Although the YOLOv7 is new and has very little work done around it, here are some cases where the YOLOv7 has been discussed.

An autonomous pallet racking system is proposed in this study centered around the YOLOv7 algorithm. In order to address the problem of data shortages through the creation of representative data samples, a domain variance modeling mechanism is put forth. The model performed with an mAP of 0.911. The model achieved a speed of 19 FPS. [13]

Deep-SORT is enhanced with YOLOv7 as an object detection network, yielding YOLOv7-DeepSORT. Experiments reveal that this network outperforms YOLOv5-DeepSORT in tracking accuracy. Due to YOLOv7 and DeepSORT's excellent generalisation capabilities, the YOLOv7-DeepSORT can be used for any kind of target tracking operation. [20]

III. ARCHITECTURE

The way that the YOLO series detects objects is as follows. The input initially consists of an image and the actual values for the bounding boxes. The input image is segmented into a square grid, and the item is detected in the grid cell containing the object's center. B bounding boxes and the related confidence ratings will be predicted for each grid cell. This rating reflects the accuracy of the box and the model's belief that the object is inside. The IoU of the expected and actual values for the bounding box constitutes the confidence score. [15]

A. YOLOv4

The YOLOv4 Architecture consists of various parts as shown in figure 3

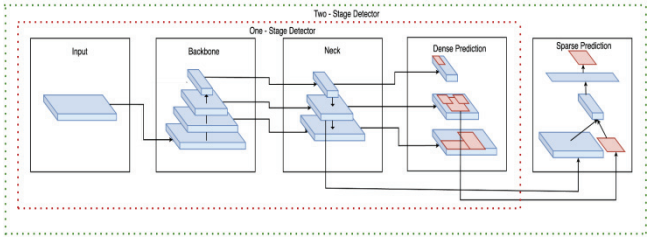

Fig. 3. YOLOv4 Architecture

The network used in the backbone is CSPDarknet53 CNN. It is built on the DenseNet architecture. Before entering the dense layers, the prior inputs and the current input are combined to create the dense connectivity pattern. CSPDarknet53 CNN is made up of two blocks, the first is a Convolutional Base Layer, and the second block is a Cross Stage Partial Block (CSP). The CSP splits the base layer's feature map into two sections and unites them employing a cross-stage hierarchy, allowing more gradient to flow across the layers and lessening the infamous "Vanishing Gradient" problem. The convolutional base layer is composed of the full-sized input feature map. The CSP block, stacked on top of the Convolutional Base layer, divides the input in half. The thick block will receive one half while the other half will be forwarded directly to the subsequent step. As more tightly linked convolutional layers may reduce detection efficiency, only the last convolutional block in the backbone network may extract more semantic data.[18]

The feature aggregation takes place in the neck. It is a cluster of bottom-up parts and top-down paths. The neck assists in gathering feature maps from several backbone stages and blending and combining them to get ready for the following stage. The first block of the neck of the CSPDarkNet is called Spatial Pyramid Pooling (SPP), and it is associated with the last layers of the densely connected convolutional layers. SPP has essentially no impact on network operation performance; it is used to expand the receptive area and distinguish the most important context features. Next is a Path Aggregation Network block (PANet). The main goal of PANet is to enhance instance segmentation by the preservation of spatial information, which supports precise pixel localization. When employing Adaptive Feature Pooling, the updated PANet concatenates adjacent layers rather than adding them.

The head is the last component, and its primary job is to locate bounding boxes and carry out classification. Scores and bounding box coordinates (x, y, height, and width) are detected.

The authors also introduce new terms called Bag of Freebies(BoF) and Bag of Specials (BoS). The BoFs help improve the network's performance without adding to the inference time, most of which are data augmentation techniques. CutMix and Mosaic data augmentation, DropBlock regularization, and Class label smoothing are the BoFs employed as the backbone, and CIoU-loss, CmBN, DropBlock regularization,

Mosaic data augmentation, Self-Adversarial Training, and CIoU-loss are the BoFs used for the detector. Using several anchors to establish one ground truth, eliminating grid sensitivity, a cosine annealing scheduler, optimal hyperparameters, and random training shapes are some of the other features. The BoS substantially boosts performance while only slightly increasing inference time. Mish activation, SPP-block, SAM-block, PAN path-aggregation block, and DIoU-NMS are the BoS used for the detector, whereas Cross-stage partial connections (CSP), Multi-input weighted residual connections (MiWRC), and Mish activation is the BoS used for the backbone.[18]

Fig. 4. YOLOv4 Flow Chart

B. *YOLOv5*

Yolov5 is similar to Yolov4 but for the following differences:

1. YOLOv4 was provided as part of the Darknet framework, it is in the C language. Yolov5 is built on top of the PyTorch framework.
2. YOLOv4 utilizes a .cfg configuration however Yolov5 utilizes a .yaml file for configuration.[16]

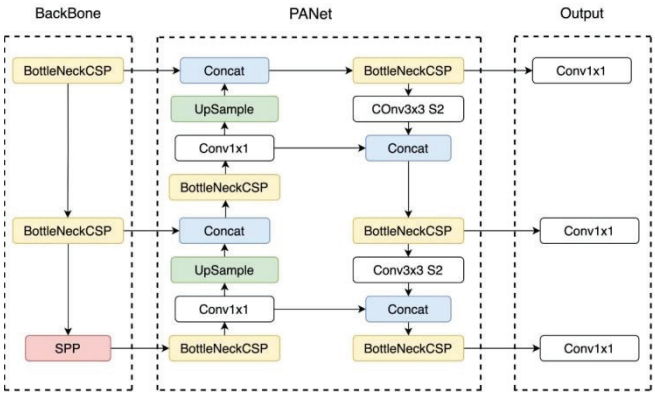

Fig. 5. YOLOv5 Architecture

To increase performance, various current methods like SAM-block, PANet, and CmBN have been modified in YOLOv5.

For feature extraction from images made up of cross-stage partial networks, the YOLOv5 Backbone utilizes CSPDarknet as the foundation. To combine the characteristics and send them to the Head for prediction, the Neck creates a feature pyramid network using PANet. The Head Layers produce predictions for object detection from the anchor boxes.

C. *YOLOv7*

The YOLOv4, Scaled YOLOv4, and YOLO-R were used to create the architecture. Using these models as a foundation, additional tests were carried out to create a new and enhanced YOLOv7.

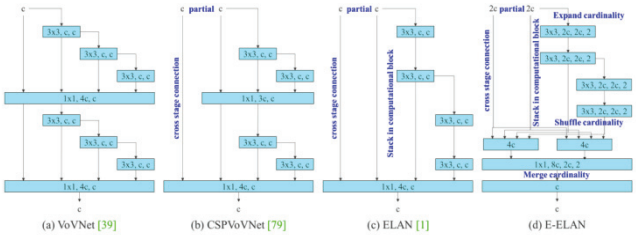

Fig. 6. Networks with prolonged effective layer aggregation. The suggested prolonged ELAN (E-ELAN) maintains the gradient transmission path of the authentic design but employs group convolution to raise the cardinality of the additional features and mix the characteristics of other groups in a way that shuffles and merges cardinality. This mode of functioning can increase the utilization of parameters and calculations as well as the features that are learned by various feature maps.[4]

The Computational block in YOLOv7 is E-ELAN (Extended Efficient Layer Aggregation Network). It is inspired by earlier network efficiency research. It was created by considering the following parameters that influence speed and accuracy. Expand, shuffle, and merge

cardinality are utilized in the continually projected E-ELAN to enhance the network's learning capabilities while maintaining the initial gradient route. E-ELAN only affects the computing block's design in terms of architecture; the transition layer's architecture remains unaffected. The E-ELAN design allows the framework to learn more effectively. It's built around the ELAN computational block. The plan is to increase the channel and cardinality of computing blocks by means of group convolution. The computational blocks of a computational layer are all subjected to a similar group parameter and channel multiplier. The set group parameter g will be used to randomly divide each computational block's feature map into g groups, which will then be concatenated together. Finally, g feature map groups are added to perform merge cardinality. E-ELAN can assist multiple a set of computational building pieces to learn more diversified properties while keeping the original ELAN design architecture.[4]

Model scaling is done to modify a few of the model's properties, build models at various scales to accommodate different inference speeds, and fit the models into various computing-platforms.

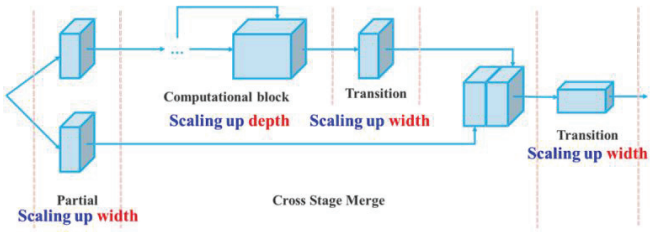

Fig. 7. Compound Scaling up depth and width for Concatenation-based Models [4]

Resolution (the input image's size), Width (the number of channels), Depth (the number of layers), and Stage (the number of feature pyramids) are taken into account when scaling a model size. NAS is a common model scaling technique (Network Architecture Search). Researchers use it to run over the parameters and identify the ideal scaling variables. But techniques like NAS carry out parameter-specific scaling. Scaling variables in this scenario are unrelated. According to the YOLOv7 paper, by using a compound model scaling technique, it may be made even better. For concatenation-based models, breadth and depth are scaled in coherence.[17] A computational block just has to be scaled in depth, when doing model scaling on concatenation-based models, and appropriate width scaling is used for the final transmission layer.[4]

The new trainable Bag of Freebies introduced in YOLOv7 are:

1. Re-parameterized convolution is anticipated.
2. Fine for lead loss and coarse for auxiliary
3. Supplementary trainable bag-of-freebies (The authors employed a few of the freebies utilized in training but they did not propose the concepts.)

Re-parameterization is a method for enhancing the model after training. It lengthens training duration while improving inference results.[17] A module-level re-parameterization is used in which the model training procedure is broken down into many components. To construct the proposed re-parameterized convolution's architecture, a RepConv without identity connection (RepConvN) is employed.

Deep network training typically makes use of the approach of deep supervision. YOLOv7 uses multiple heads instead of a single head. The Lead Head is responsible for producing the ultimate product, and the auxiliary head, in meantime is utilized to support the middle layers. An assistant loss is used to update the auxiliary heads' weights. It enables Deep Supervision and improves model learning. These ideas are intertwined with the Lead Head and the Label Assigner. Label Assigner is a technique that evaluates network prediction outcomes as well as ground truth before assigning soft labels. It generates smooth and coarse labels rather than hard labels. The YOLOv7 network's Lead Head forecasts the outcomes. Soft labels are created using these results in the end. The most important aspect is that the lead head and auxiliary head's loss estimates are made using identical soft labels. Ultimately, both heads are instructed to use the soft labels. The lead head guided label assigner generates two different types of soft labels: coarse and fine. The coarse label produced by enabling additional grids will be evaluated as potential targets by loosening restrictions placed on a successful sample assignment procedure. The need to prevent information loss and the fact that an auxiliary head's capacity for learning is less powerful than a lead head is the reasons behind this. As for the lead head's output, we can distinguish between high-precision and high-recall findings to produce the desired outcome.[4]

IV. EXPERIMENTS AND RESULTS

A. Experimental system details

The following describes the experiment's environment: CPU: Intel® Core(TM) i5-10210U CPU @ 2.11 GHz, RAM: 8.00 GB, GPU: NVIDIA® GeForce® MX230, Operating System: Windows 11, PyTorch was the deep learning framework utilized in this project, developed by Meta's AI research group. The Python programming language was used to write the software code. GPU acceleration was accomplished using cuDNN and NVIDIA CUDA. The file type used for configuration in YOLOv4 was .cfg, whereas YOLOv5 and YOLOv7 used .yaml file configurations.

B. dataset

The Safety Helmet Detection dataset, created by larxel and retrieved from Kaggle, was the one used in this research. The dataset consists of 5000 XML files that are bounding box annotated images in the PASCAL VOC format. Here are some sample training images:

Fig. 8. Few sample images from the dataset

C. Performance metrics

To accurately evaluate the experimental results of the models, assessment metrics including precision, recall, and mAP were used.

1) Precision:

The ratio of true positives to all positive predictions is known as precision. It represents the model's probability that the anticipated bounding box matches the ground truth box exactly. Precision can be calculated using the following equation (2):

$$\text{Precision} = (\text{TP})/(\text{TP} + \text{FP}) \quad (2)$$

2) Recall:

The ratio of true positives to all actual objects is known as recall. It shows the likelihood that real objects will be successfully detected. Recall is given by the following equation (3):

$$\text{Recall} = (\text{TP})/(\text{TP} + \text{FN}) \quad (3)$$

3) Mean Average Precision(mAP):

The mean Average Precision (mAP) is the result of comparing the detected bounding box to the actual ground truth bounding box. The detection is regarded as TP if the intersection is 50% or higher above the union score of both boxes. The mAP can be calculated as in equation (4):

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (4)$$

AP_k = The AP of the class

n = Number of classes

where TN, TP, FN, and FP refer to True Negative, True Positive, False Negative, and False Positive, respectively.

D. Experimental setup

To provide input to object detection models, annotated files were first converted to the YOLO format from the PASCAL VOC format. Training and testing sets were created by randomly dividing the intended dataset. 80% of the total images, or 4000 images, were in the training set, whereas only 20% of the dataset, or 1000 images, were in the testing set. To assess how well the various models performed, the mAP0.5 was employed. Throughout the research, the Intersection over Union (IoU) threshold value was held constant at 0.5. For optimal solution, the image sizes used for YOLOv4 and YOLOv5 were [416,416,3], whereas for YOLOv7 it was [640,640,3]. Some of the hyperparameters were also changed for the YOLOv4 during training for an optimal solution. The table contains the following hyperparameters used for the experiments.

TABLE I. HYPERPARAMETERS

Parameters	Value
LR: Learning Rate	0.01, 0.001*
LRD: Learning rate decay	0.99
Learning rate decay step	1
WRD: Weight rate decay	0.0005
Momentum	0.937, 0.949*
Size of Batch	16
Epochs	100

**: Used for YOLOv4*

E. Results

Three classes make up the dataset: Head, Helmet, and Person. This study concentrates more on the class helmet (helmet detection) than the other two classes. The mAP is the basic and the most commonly used to analyze the performance of the models. The results obtained from the experiments have been summarized in the tables below (TABLE II, TABLE III, TABLE IV, AND TABLE V). The other performance metrics used are Precision and Recall but for this experiment, we are mostly concerned with the mAP.

TABLE II. MODELS AND DETECTION TIME REQUIRED

Model	Detection Completion time (in Hrs.)
YOLOv4	6.54
YOLOv5	3.43
YOLOv7	5.49

TABLE III. MODELS AND AVERAGE TIME REQUIRED

Model	Average Detection Time (in ms)
YOLOv4	45.1
YOLOv5	24.7
YOLOv7	12.4

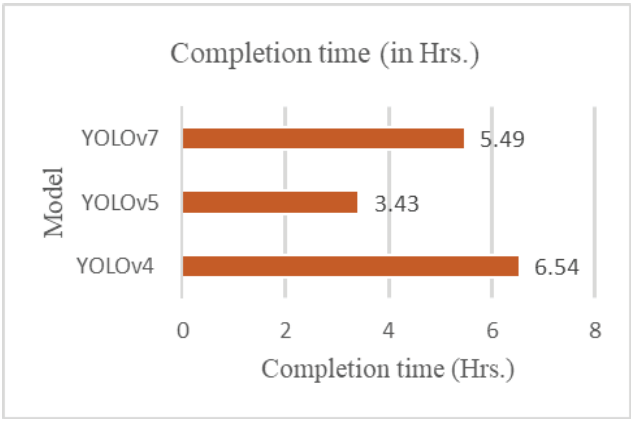

Fig. 9. Models vs. time required to complete the experiment

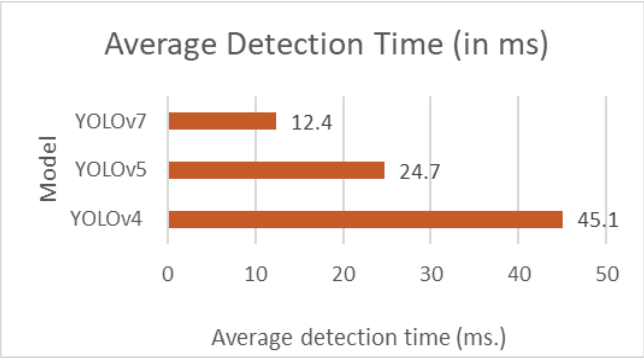

Fig. 10. . Models vs. average time required for detection

TABLE IV. PERFORMANCE OF MODELS FOR HEAD CLASS

Performance Metrics	Class: Head		
	YOLOv4	YOLOv5	YOLOv7
mAP	0.905	0.93	0.93
Precision	0.912	0.932	0.922
Recall	0.88	0.884	0.888

TABLE V. PERFORMANCE OF MODELS FOR HELMET CLASS

Performance Metrics	Class: Helmet		
	YOLOv4	YOLOv5	YOLOv7
mAP	0.936	0.951	0.964
Precision	0.88	0.955	0.937
Recall	0.88	0.89	0.921

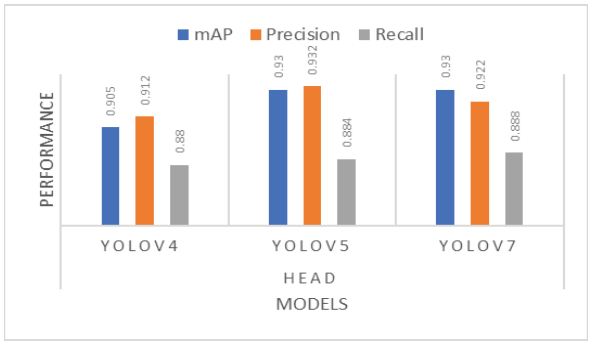

Fig. 11. Performance of different models for Head class

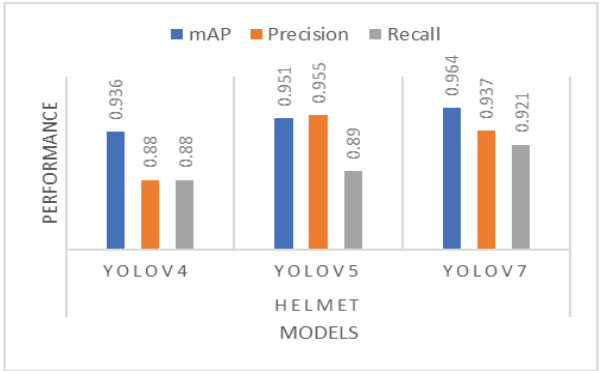

Fig. 12. Performance of different models for Helmet class

I I I I

Fig. 13. Images I-II are the images detected by YOLOv7.

III IV

Fig. 14. Images III-IV are the images detected by YOLOv5

The YOLOv4 achieves an mAP of 93.6% for the helmet, while the YOLOv5 achieves an mAP of 95.1% for the helmet. The YOLOv5 performed 1.6% better than the YOLOv4. The YOLOv7 model performed 3.0% better than the YOLOv4 whereas 1.36% better than the YOLOv5, giving the helmet an mAP of 96,4%. The YOLOv7 and YOLOv5 detected the class head with an mAP of 93.0% while the YOLOv4 gave an mAP of 90.5%. The Recall obtained for the YOLOv7 was 92.1% which was better than the Recall obtained by the YOLOv5 (89.0%) and the YOLOv4 (88.0%). The YOLOv5 (95.5%) outperforms the YOLOv7 (93.7%) and the YOLOv4 (88.0%) in terms of precision. The YOLOv7 transcends the other two models (YOLOv5: 24.7 ms, YOLOv4: 45.1 ms) in average detection time as well, with a staggering average detection time of 12.4 ms paving the way for even faster real-time detection that the YOLO models specialize in.

V. CONCLUSION

We investigated helmet detection using YOLOv4, YOLOv5, and YOLOv7 as target detection algorithms, using a publicly available dataset, and compared the results to analyze which model achieves the best results. The findings indicate that the YOLOv7 model achieves a higher average accuracy of 96.4% which is 1.36% and 3.0% better than the YOLOv5 (95.1%) and the YOLOv4 (93.6%) respectively. The YOLOv7 also detects helmets with an average time of 12.4 ms which is faster than that of the YOLOv5 (24.7 ms) and the YOLOv4 (45.1 ms). Although the YOLOv7 has been recently developed, it has very promising potential in the world of real-time object detection. The YOLOv7 not only provides better accuracy but is also very much faster compared to the other two models. We will look for ways to improve the YOLOv7 model's accuracy in the coming future.

REFERENCES

[1] Fangbo Zhou, Huailin Zhao, Zhen Nie, " Safety Helmet Detection Based on YOLOv5", 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)

[2] Shilei Tan, Gonglin Lu, Ziqiang Jiang, and Li Huang, " Improved YOLOv5 Network Model and Application in Safety Helmet Detection", 2021 IEEE International Conference on Intelligence and Safety for Robotics Nagoya, Japan, March 4-6, 2021

[3] Yongze Ji, Yu Cao, Xu Cheng, Qiong Zhang, "Research on the Application of Helmet Detection Based on YOLOv4", Journal of Computer and Communications, 2022, 10, 129-139 https://www.scirp.org/journal/jcc

[4] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", arXiv:2207.02696v1[cs.CV]

[5] Desu Fu, Lin Gao, Tao Hu, Shukun Wang, Wei Liu, "Research on Safety Helmet Detection Algorithm of Power Workers Based on Improved YOLOv5", Desu Fu et al 2022 J. Phys.: Conf. Ser. 2171 012006

[6] Deng Benyang, Lei Xiaochun, and Ye Miao, "Safety helmet detection method based on YOLO v4", 2020 16th International Conference on Computational Intelligence and Security (CIS)

[7] Haikuan Wang, Zhaoyan Hu, Yuanjun Guo, Zhile Yang, Feixiang Zhou, and Peng Xu, "A Real-Time Safety Helmet Wearing Detection Approach Based on CSYOLOv3", Appl. Sci. 2020, 10, 6732; doi:10.3390/app10196732, www.mdpi.com/journal/applsci

[8] Xitian Long, Wenpeng Cui, Zhe Zheng, "Safety Helmet Wearing Detection Based On Deep Learning", 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC 2019)

[9] Citation: Fang, C.; Xiang, H.; Leng, C.; Chen, J.; Yu, Q. "Research on Real-Time Detection of Safety Harness Wearing of Workshop Personnel Based on YOLOv5 and OpenPose". Sustainability 2022, 14, 5872. https://doi.org/10.3390/su14105872

[10] Citation: Ku, B.; Kim, K.; Jeong, J. "Real-Time ISR-YOLOv4 Based Small Object Detection for Safe Shop Floor in Smart Factories". Electronics 2022,11, 2348. https://doi.org/10.3390/electronics11152348

[11] Citation: Liu, Y.; Li, Z.; Zhan, B.; Han, J.; Liu, Y. "A Super-Resolution Reconstruction Driven Helmet Detection Workflow". Appl. Sci. 2022,12, 545. https://doi.org/10.3390/app12020545

[12] Otgonbold, M.-E.; Gochoo, M.; Alnajjar, F.; Ali, L.; Tan, T.-H.; Hsieh, J.-W.; Chen, P.-Y. "SHEL5K: An Extended Dataset and Benchmarking for Safety Helmet Detection". Sensors 2022, 22, 2315. https://doi.org/10.3390/s22062315

[13] Hussain, M.; Al-Aqrabi, H.; Munawar, M.; Hill, R.; Alsboui, T. "Domain Feature Mapping with YOLOv7 for Automated Edge-Based Pallet Racking Inspections". Sensors 2022, 22, 6927. https://doi.org/10.3390/s22186927

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollar, "Microsoft COCO: Common Objects in Context", arXiv:1405.0312v3 [cs.CV]

[15] https://iq.opengenus.org/yolov4-model-architecture/

[16] https://medium.com/analytics-vidhya/object-detection-algorithm-yolo-v5-architecture-89e0a35472ef

[17] https://learnopencv.com/yolov7-object-detection-paper-explanation-and-inference/#YOLOv7-Architecture

[18] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection", arXiv:2004.10934v1 [cs.CV]

[19] https://github.com/ultralytics/yolov5/issues/280#issue-650463630%20https://ieeexplore.ieee.org/abstract/document/9528256

[20] Feng Yang, Member, IEEE, Xingle Zhang and Bo Liu, "Video object tracking based on YOLOv7 and DeepSORT", arXiv:2207.12202v1 [cs.CV]

[21] Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", arXiv:2207.02696[cs.CV]

[22] Vung Pham, Du Nguyen, Christopher Donan, "Road Damages Detection and Classification with YOLOv7", arXiv:2211.00091 [cs.CV]

[23] Yange Li, Han Wei, Zheng Han , Jianling Huang, and Weidong Wang, "Deep Learning-Based Safety Helmet Detection in Engineering Management Based on Convolutional Neural Networks", Volume 2020, Article ID 9703560, https://doi.org/10.1155/2020/9703560

[24] Delin Wu , Shan Jiang, Enlong Zhao, Yilin Liu, Hongchun Zhu, Weiwei Wang and Rongyan Wang, "Detection of Camellia oleifera Fruit in Complex Scenes by Using YOLOv7 and Data Augmentation", Appl. Sci.2022, 12, 11318. https://doi.org/10.3390/app122211318

[25] Zhang Jin , Peiqi Qu, Cheng Sun, Meng Luo, Yan Gui, Jianming Zhang , and Hong Liu , "DWCA-YOLOv5: An Improve Single Shot Detector for Safety Helmet Detection", Volume 2021, Article ID 4746516, https://doi.org/10.1155/2021/4746516

[26] Kang Li, Xiaoguang Zhao, Jiang Bian, and Min Tan , "Automatic Safety Helmet Wearing Detection", arXiv:1802.00264 [cs.HC]