2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST) | 978-1-6654-8663-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/GECOST55694.2022.10010490

# Safety Helmet Detection Using Deep Learning: Implementation and Comparative Study Using YOLOv5, YOLOv6, and YOLOv7

Nigel Dale Then Yung
*Dept. of Electrical and Computer Engineering*
*Curtin University Malaysia*
Miri, Malaysia
nigeldalethen@gmail.com

W. K. Wong
*Dept. of Electrical and Computer Engineering*
*Curtin University Malaysia*
Miri, Malaysia
weikitt.w@curtin.edu.my

Filbert H. Juwono
*Computer Science Program*
*University of Southampton Malaysia*
Johor, Malaysia
F.Juwono@soton.ac.uk

Zee Ang Sim
*Dept. of Electrical and Computer Engineering*
*Curtin University Malaysia*
Miri, Malaysia
zeeang.sim@gmail.com

***Abstract*—The safety of construction site personnel is highly dependent on the adherence of personal protective equipment (PPE) wearing. Safety helmet monitoring has become a popular topic in recent years as a result of the success in the field of image processing. Deep learning (DL) is widely used in object detection tasks due to its ability to create features based on raw data. Constant improvements in the DL models have led to numerous successful outcomes in the implementation of safety helmet detection tasks. The performance of different DL algorithms from previous studies will be assessed and studied in this review paper. The YOLOv5s (small) model, YOLOv6s (small) model, and the YOLOv7 model will be trained and evaluated in this paper.**

***Index Terms*—deep learning, safety helmet detection, YOLOv5, YOLOv6, YOLOv7**

## I. Introduction

Artificial Intelligence (AI) can be considered as a "prediction technology" that utilizes stored data to produce new information [1]. A machine that is equipped with AI is capable of performing the works of human by learning from human's intelligence and experiences. AI is also known as a General-Purpose Technology (GPT) as it can be applied in various applications to perform different types of tasks [2]. AI has contributed in the data production of various applications such as advanced computing, statistical modelling, and computer coding [3]. The success of AI has several major subfields such as Machine Learning (ML), Neural Network (NN), Deep Learning (DL), Robotics, and Computer Vision (CV).

ML enables software applications to perform tasks such as experience learning and future prediction with more accuracy without the need of specific programming [4]. On the other hand, DL utilizes ML algorithms to assist a system in understanding complicated tasks with improved accuracy [4]. DL is a process whereby technologies operate on Artificial Neural Network (ANN) to perform training and inferring of data [5]. In particular, it is able to perform feature generation from raw data [6]. In the training phase, data are labelled according to their characteristics and newly found data are labelled and processed based on the knowledge gained from previous data in the inferring phase. The efficiency of DL training process increases when the data volume increases, this is achieved by the constant collection of learning algorithms by ANNs [4]. DL produces high-level features from low-level features and allows a computer to analyse data characteristics with reduced parameters without altering the original data features [7].

Object detection is a task to perform clear image understanding by focusing on image classification and object location estimation found in an image. A semantic understanding on the important information of images and videos can be obtained through the process of object detection [8]. The success in NN developments has positively impacted object detection techniques as learning systems [9]–[12]. It is a challenge to precisely localize objects when object detection tasks are being performed due to the constant changing of factors such as viewpoints, obstructions, positions, postures, and lighting. In recent years, object detection has been a popular topic among studies and researches with the aim to create object detection models that are capable of performing useful area selection, feature extraction, and categorization tasks on images and videos [8]. Annotating bounding box is a key application in object detection as it serves as an object classification and localization tool [13] and enables the location prediction feature in object detection tasks [14]. Moreover, machines are able to identify objects in an annotating bounding box and the data obtained from annotated images can be used in the process of DL [13].

The problem of safety helmet detection, which is a form of object detection, using DL approaches is addressed in this paper. A safety helmet is an important personal protective equipment (PPE) that is commonly used in places where head injury accidents may occur. Construction sites, coal mines, and substations are examples of high-risk working environments as safety accidents often occur due to the carelessness and unsafe behaviours of workers. As an example, a total of 2,032 out of 3,014 accidents recorded in China are due to falling from elevated places and falling objects [15]. Unfortunately, manual supervision is inefficient and inaccurate due to the large number of workers [16]. A safer working environment in construction sites can be achieved by monitoring the presence of safety helmets equipped on construction site personnel through the application of image processing. Safety helmet detection is considered a popular research topic in the image processing field as it involves object detection tasks. To date, various types of DL technology are available to perform image processing by applying the target detection algorithm [15].

The rest of the paper is organised as follows. Section II summarizes the methods used in other safety helmet detection projects. Section III introduces the algorithms used in this study. Section IV discusses the method of evaluation conducted on the DL models. The tests results are presented in Section V. Finally, Section VI gives the conclusion and future work.

## II. Literature Review

### A. Deep Neural Network

A Deep Neural Network (DNN) is formed by adding a number of hidden layers between the input and output layers of an ANN. DNNs run in one direction which is from the input layer to the output layer without any looping process. Meanwhile, a Convolutional Neural Network (CNN) is a multilayer neural network derived from the concept of animal visual cortex. CNNs have developed and progressed well since the first proposed model from LeCun et al. Note that CNN is the most widely used DNN which is commonly used in image processing and recognition [4], [17].

Object detection through DL methods can be divided into four different categories, which include one-stage detectors, two-stage detectors, anchor-based detector, and anchor-free detectors. In one-stage methods, only a CNN is needed to directly perform target categorization and localization prediction on the given images. Two-stage methods on the other hand requires the generation of candidate regions in order to classify and locate them. Hence, two-stage methods have a more complicated network architecture than one-stage methods. By having a complex structured network, the detection accuracy is high with a slower detection process [15]. Real-life industrial applications require a fast detection speed of machines, therefore one-stage methods have a better overall performance in object detection compared to two-stage methods.

The architecture of a CNN-based object detection method that uses the two-stage approach is formed by multiple different layers. Some examples of DL model that falls under the two-stage detector category are RCNN, VFNet, and CenterNet2. The convolutional layer is the main part of a CNN which identifies features such as edges and colours of different classes of objects which are used to generate a feature map [13]. The feature map contains multiple image channels that carries distinct image information [14]. The next layer is called the pooling layer. The pooling process compresses the features, analyses and maximises the receptive fields, and lowers the overall computation cost [4], [14], [16]. The convolution and pooling process may be done alternately to achieve a better image understanding. In the activation layer, nonlinear activation functions such as Rectified Linear Unit (ReLU) activation functions are used to further improve a neural network's expressing ability in solving nonlinear problems. The final layers are called the fully connected layers where the combination of object data features and output of feature values are done [16]. The output layer utilizes backpropagation algorithms to detect features from different images [4].

### B. Safety Helmet Detection Related Work

In [15], the authors added an additional feature map size of 104 $\times$ 104 to improve feature learning, utilized CIoU to determine coordinate regression loss, and applied pixel feature statistics on the anchor boxes of the You Only Look Once, Version 3 (YOLOv3) network. The improved YOLOv3 network outperformed other algorithms in terms of mean Average Precision (mAP) and frames per second (FPS). We also note that the majority of the safety helmet detection techniques performs poorly when applied in environments with high requirements which includes human postures such as lying, bending, and squatting. It is also a challenge to detect safety helmets due to factors such as high fluctuation rate and environmental interventions.

The Single Shot Multibox Detector (SSD) algorithm has been widely used in safety helmet detection tasks as shown in [13], [16], [18]. It is found that the SSD detection model could not detect more than one person at a time when it is applied using a camera in real time. The performance of the SSD algorithm is also highly dependent on the quality and variations of the training image dataset. In order to improve the functionality of the algorithm, training images should include a variety of angles and sizes as well as a complex background.

The authors in [19] conducted a study on four different YOLOv5 variant models to determine the best performing safety helmet detecting algorithm. The models were formed by modifying the size of the BottleneckCSP module that is found in the neck of the model's architecture. All four YOLOv5 models had a similar mAP value but the YOLOv5s model stands out among the models for its outstanding speed of 110 FPS. It was also proved in [19] that YOLOv5 models with pre-training weights achieved an increased mAP value of 0.9 to 1.3. Another study on the YOLOv5 model by [20] showed an improved YOLOv5 model in detecting small targets. A functionality detection scale and the DIoU-NMS were introduced in this method to enhance the bounding box prediction task. The trained model was able to perform with great results of 95.7% mAP with a speed of 98 FPS.

In [21], the K-means++ algorithm was utilized as the clustering method to improve the sizing functionality of the anchor box. The improved YOLOv5 had better performance in target detection and resistance to background interference after the newly designed Depthwise Coordinate Attention (DWCA) mechanism was combined with the backbone of the original YOLOv5 network. The summary of the existing work is shown in Table I.

TABLE I: Performance summary of different DL algorithms for safety helmet detection [13], [15], [16], [18]–[20].

| Author | Algorithm | mAP (%), IoU=0.5 | FPS |
|---|---|---|---|
| Huang et al. [15] | SSD | 77.2 | 46 |
| | Faster R-CNN | 94.3 | 4 |
| | YOLOv3 | 82.3 | 58 |
| | Improved YOLOv3 | 93.1 | 65 |
| Li et al. [16] | SSD | 36.8 | - |
| Kamboj et al. [13] | | 96.0 | 25 |
| Long et al. [18] | | 68.5 | 21.6 |
| Zhou et al. [19] | YOLOv5s | 93.6 | 110 |
| | YOLOv5m | 94.3 | 64 |
| | YOLOv5l | 94.4 | 37 |
| | YOLOv5x | 94.7 | 21 |
| Tan et al. [20] | YOLOv5 | 92.1 | 106 |
| | Improved YOLOv5 | 95.7 | 98 |

## III. YOLOv5, YOLOv6, and YOLOv7 Algorithms

The YOLOv5 object detection algorithm is a one stage anchor-based object detection method [22]. The structure of YOLOv5 is divided into the backbone, the neck, and the head. The backbone of YOLOv5 is formed by the combination of cross stage partial network (CSPNet) and Darknet, forming CSPDarknet. An increased speed and accuracy during the inference process can be achieved with CSPNet as it is able to decrease the parameters that are present in the training model. YOLOv5 utilizes path aggregation network (PANet) as the neck of its architecture to increase the efficiency of information transfer within the model. It improves the object localization tasks by focusing on the feature transfer from lower layers with the Feature Pyramid Network (FPN). Lastly, the head of the YOLOv5 model is made of the Yolo layer. Multi-scale object prediction can be done as feature maps of 3 different sizes will be generated by the Yolo layer. With this complete architecture, the YOLOv5 model is capable of learning the characteristics and features of images from a given dataset hence performing object detection according to the features obtained from the training dataset. YOLOv5 has shown incredible results when it is applied on datasets such as Pascal VOC and Microsoft COCO [23].

The YOLOv6 and YOLOv7 algorithms were recently released in June and July 2022, respectively. YOLOv6 was published by Meituan while YOLOv7 was released by WongKinYiu and AlexeyAB [24], [25]. YOLOv6s has achieved a mAP of 43.1% on the COCO val2017 dataset, outperforming the YOLOv5s algorithm which has a mAP of 37.4%. The YOLOv7 algorithm has a higher inference speed and accuracy when compared with other algorithms such as YOLOR, PP-YOLOE, YOLOX, Scaled-YOLOv4, and YOLOv5 (r6.1) [25].

## IV. Preliminary Evaluation Using DL Approach

The Google Colaboratory notebook is chosen as the platform to run different DL models throughout this study. The Tesla T4 graphics processing unit (GPU) with 16GB of GPU memory is virtually connected through the Google Colaboratory to be used in all of the tests in this study. The platform also provides 12GB of random-access memory (RAM) which allows faster deep learning training. The pre-installed TensorFlow software library is used in Google Colaboatory to produce the graphs and confusion matrices shown in this paper. The Kaggle hard hat dataset [26] used to train the DL models contains 5,000 images in the PASCAL VOC format which are labelled with bounding box annotations. Three classes are defined in the original Kaggle hard hat dataset [26]: helmet, person, and head. The Mendeley dataset [27] is an improved version of the Kaggle dataset [26] that contains seven classes, including face, helmet, person, head, head with helmet, person with helmet, and person no helmet. To further improve the quality of the Mendeley dataset [27], a third hard hat dataset is produced by removing the classes: face, helmet, and person, while the class named head is renamed to head no helmet. The image dataset in all the tests is divided into 3,500 images for training set, 1,000 images for validation set, and 500 images for testing set. Three different tests were conducted to evaluate and compare the performances of the deep learning models. The main indicator used to evaluate the performance of the DL models is mean Average Precision (mAP). This indicator is used in many other safety helmet detection studies to show the precision and recall trends of various training models [18].

The precision rate (Pr), recall rate (Re), and F1-score (F1) are used as evaluation indicators which can be mathematically expressed by

$$Pr = \frac{TP}{TP + FP}, \quad (1)$$

$$Re = \frac{TP}{TP + FN}, \quad (2)$$

$$F1 = \frac{2 \times Pr \times Re}{Pr + Re}. \quad (3)$$

These indicators depend on the values of True Positive (TP), False Positive (FP), and False Negative (FN) samples. TP is the case where an object is predicted correctly as a positive sample. A FP case is where an invalid object is detected as a positive sample. Whereas in the case of a FN, a valid object is detected as a negative sample.

The expressions to calculate AP and mAP are respectively given by

$$AP = \int_0^1 Pr(Re)\ dRe, \quad (4)$$

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k, \tag{5}$$

where $n$ is the number of classes and $AP_k$ is the average precision of class $k$.

## V. Results and Discussion

Table II shows the training time of 100 epochs for YOLOv5, YOLOv6, and YOLOv7. It can be seen that YOLOv5 has the shortest training time compared to YOLOv6 and YOLOv7. Three tests are performed in this paper to observe the performances of three different deep learning algorithms in detecting safety helmets. Six trial runs of the YOLOv5s model are carried out as part of Test 1, with trials 1 and 2 utilizing data from the Kaggle platform, trials 3 and 4 from the Mendeley platform, and trials 5 and 6 from the Edited Mendeley dataset. Two trial runs are performed for Test 2 with Edited Mendeley dataset and YOLOv6s model. Finally, Test 3 uses Edited Mendeley dataset as well but with YOLOv7 model. Additionally, in this paper, the Intersection over Union (IoU) threshold is set to a score of 0.5. IoU is a value used to show the overlapping percentage of the predicted bounding box and the actual bounding box. When the IoU has a score greater than 0.5, it is considered a good prediction and the output will be true, otherwise the output will be considered false.

TABLE II: Algorithm comparison of YOLOv5, YOLOv6, and YOLOv7

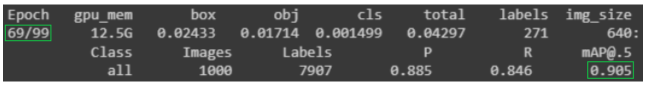
| Algorithm | Developer | Training time (100 epochs) |
|---|---|---|
| YOLOv5 | Glenn Jocher [28] | 1.115 hours |
| YOLOv6 | Meituan [24] | 3.266 hours |
| YOLOv7 | WongKinYiu, AlexeyAB [25] | 6.092 hours |

TABLE III: Tests performance results

| Test | Trial | Dataset | Algorithm | Epoch | Batch | mAP (%) |
|---|---|---|---|---|---|---|
| 1 | 1 | Kaggle [26] | YOLOv5s | 100 | 16 | 60.0 |
| | 2 | | | 200 | 32 | 61.2 |
| | 3 | Mendeley [27] | | 100 | 16 | 66.6 |
| | 4 | | | 200 | 32 | 68.1 |
| | 5 | Edited Mendeley | | 100 | 16 | 82.7 |
| | 6 | | | 200 | 32 | 83.7 |
| 2 | 1 | | YOLOv6s | 100 | 16 | 83.5 |
| | 2 | | | 200 | 32 | 82.1 |
| 3 | 1 | | YOLOv7 | 100 | 16 | 89.6 |

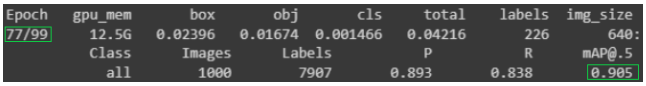
Table III shows an overview of the tests conducted in this study. As seen in the 5th trial run of Test 1, an improvement in the mAP of the YOLOv5s is achieved as a result of the dataset improvement done on the Mendeley dataset [27]. In [29], it was found that the YOLOv5s model performs best when an epoch value of 200 and batch size of 32 is set during the training stage. After performing transfer learning on the YOLOv5s model, we produce the same trend as expected from the study in [29]. From Test 2, it is found that the YOLOv6s has the best result with epoch value of 100 and batch size of 16 in the training stage. In Test 3, we observe that the highest mAP achieved from the YOLOv7 algorithm is 90.5%, which is at epoch number 69 and number 77 as shown in Figs. 1 and 2.

The graphs showing the mAP trend obtained from each test are shown in Figure 3. Among all the tests conducted in this study, Test 3 produces the highest mAP of 89.6%. The confusion matrix and detection performance matrix produced in Test 3 are shown in Figs. 7 and 8, respectively. By comparing the confusion matrices of the two best performing algorithms in Test 3 and Test 1, it is concluded that the YOLOv7 has an overall better performance in detecting safety helmets.



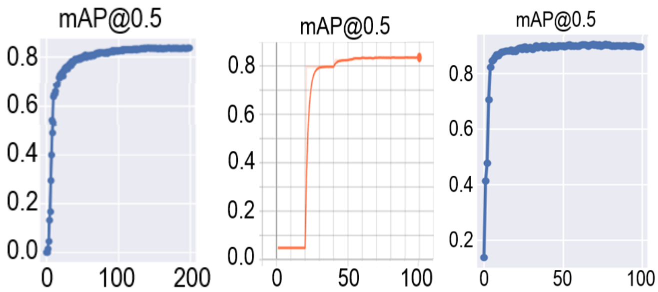
| Epoch | gpu_mem | box | obj | cls | total | labels | img_size |
|---|---|---|---|---|---|---|---|
| 69/99 | 12.5G | 0.02433 | 0.01714 | 0.001499 | 0.04297 | 271 | 640: |
| | Class | Images | Labels | P | R | mAP@.5 | |
| | all | 1000 | 7907 | 0.885 | 0.846 | 0.905 | |

Fig. 1: mAP of YOLOv7 at epoch number 69.



| Epoch | gpu_mem | box | obj | cls | total | labels | img_size |
|---|---|---|---|---|---|---|---|
| 77/99 | 12.5G | 0.02396 | 0.01674 | 0.001466 | 0.04216 | 226 | 640: |
| | Class | Images | Labels | P | R | mAP@.5 | |
| | all | 1000 | 7907 | 0.893 | 0.838 | 0.905 | |

Fig. 2: mAP of YOLOv7 at epoch number 77.




(a) YOLOv5s (b) YOLOv6s (c) YOLOv7

Fig. 3: mAP obtained from Test 1 to Test 3.

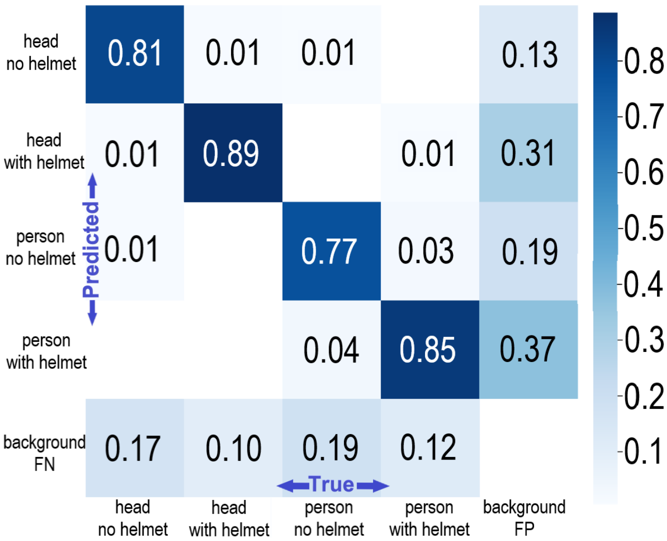



Fig. 4: Confusion matrix of the YOLOv5s model from Test 1.

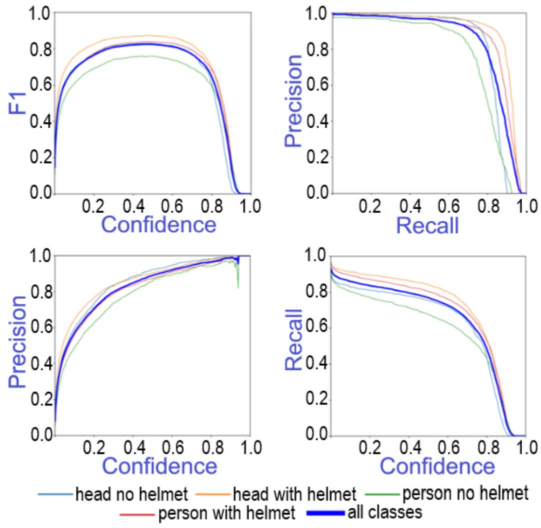



Fig. 5: Detection performance metrics of the YOLOv5s model from Test 1.

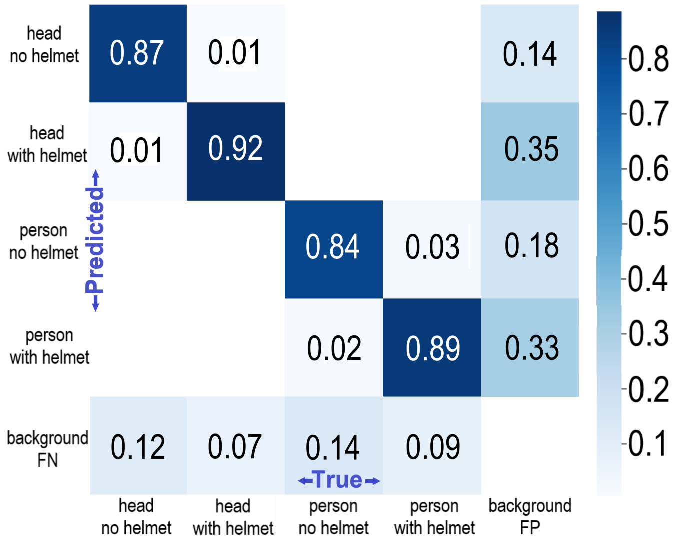



Fig. 7: Confusion matrix of the YOLOv7 model from Test 3.

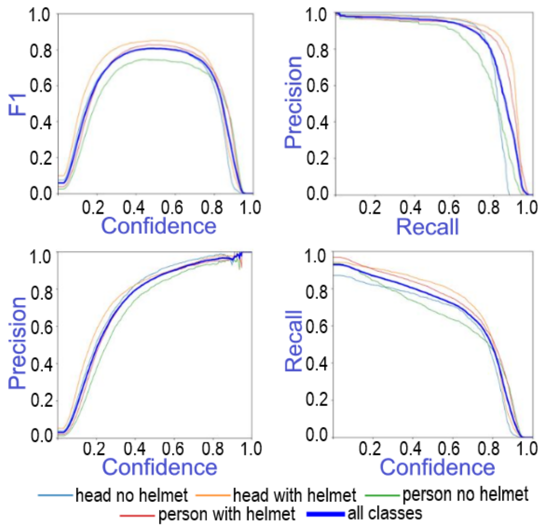



Fig. 6: Detection performance metrics of the YOLOv6s model from Test 2.

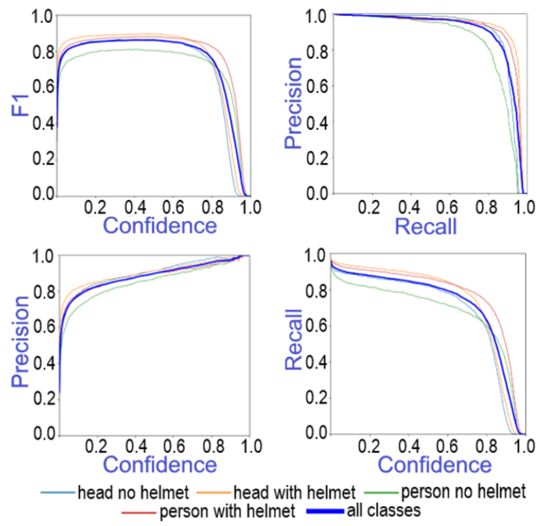



Fig. 8: Detection performance metrics of the YOLOv7 model from Test 3.

Figs. 4 and 5 show the confusion matrix and detection performance matrix of the YOLOv5s model from the 6$^{th}$ trial run in Test 1. The confusion matrix summarizes the performance of the trained model in doing object classification task for four different object classes. The detection performance matrix shows the trends of the F1-score, precision rate, and the recall rate of the trained model. Fig. 4 shows the confusion matrix of the model in detecting four different classes. By analyzing the confusion matrix, it can be seen that the trained model is capable of detecting the presence of safety helmets in images with a fine accuracy score. In particular, the model is able to detect the presence of safety helmets with accuracy scores of 0.89 and 0.85 for head with helmet and person with helmet, respectively. Whereas, the accuracy scores of detecting the absence of safety helmet are 0.81 and 0.77 for head no helmet and person no helmet, respectively. The model has a low false prediction score ranging from 0.01 to 0.37. From the graph of F1 against confidence that is shown in Fig. 5, it can be seen that when the trained model reaches a confidence level of 0.48, a F1-score of 0.83 is achieved for all of the object classes (blue curve). Note that the F1-score is a value used to represent a balance between both the precision rate and recall rate of the model. Referring to the plotted graphs of precision vs recall, precision vs confidence, and recall vs confidence displayed in Fig. 5, it can be concluded that the confidence level of the trained YOLOv5s model increases when its precision rate increases while its recall rate decreases. Hence, the precision rate and recall rate are inversely proportional to each other.

These trends are further proven to be valid in the detection performance matrices of the YOLOv6s and YOLOv7 models shown in Figs. 6 and 8, respectively.

Samples of the testing results are shown in Fig. 9. Some examples of the testing results obtained after training the YOLOv5s, YOLOv6s, and YOLOv7 models are displayed in Figs. 10, 11, and 12, respectively. The YOLOv6s and YOLOv7 models perform better in low lights conditions when compared to the YOLOv5s model. However, we observe that all three models have failed to differentiate a normal cap from a safety helmet. Both the YOLOv5s and YOLOv6s models struggle to detect the absence of the safety helmet on a person with dark skin. The YOLOv7 model is able to detect the absence of safety helmets on people with dark skin and in low light conditions but with low confidence levels of 60% to 80%.

Fig. 9: Sample testing images from the dataset.

Fig. 10: Testing results of YOLOv5s from Test 1.

Fig. 11: Testing results of YOLOv6s from Test 2.

Fig. 12: Testing results of YOLOv7 from Test 3.

## VI. Conclusion and Future Work

In this paper, we have evaluated the use of YOLO models to detect safety helmets. It has been shown that the YOLOv7 model outperforms the YOLOv5 and YOLOv6 algorithms. It is also found that the epoch value and batch size to produce the best results vary for each DL model. The test results are highly dependent on the image dataset as observed in the simulations. For future work, modifications and improvements will be done on the training image dataset, YOLOv7 model, and training settings such as the epoch value and batch size. Further studies will be done to solve the problem faced by DL models in differentiating regular caps and safety helmets. Another field of interest that will be studied and analyzed is the effect of model pruning on the performance of DL models.

# References


[1] A. Agrawal, J. Gans, and A. Goldfarb, *Prediction machines: the simple economics of artificial intelligence*. Harvard Business Press, 2018.

[2] A. Goldfarb, J. Gans, and A. Agrawal, *The Economics of Artificial Intelligence: An Agenda*. University of Chicago Press, 2019.

[3] J. Shaw, F. Rudzicz, T. Jamieson, A. Goldfarb *et al.*, "Artificial intelligence and the implementation challenge," *Journal of medical Internet research*, vol. 21, no. 7, p. e13659, 2019.

[4] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: a new paradigm to machine learning," *Archives of Computational Methods in Engineering*, vol. 27, no. 4, pp. 1071–1092, 2020.

[5] S. Gu, L. Ding, Y. Yang, and X. Chen, "A new deep learning method based on alexnet model and ssd model for tennis ball recognition," in *2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA)*. IEEE, 2017, pp. 159–164.

[6] B. A. Lease, W. Wong, L. Gopal, and C. W. Chiong, "Pixel-level weed classification using evolutionary selection of local binary pattern in a stochastic optimised ensemble," *SN Computer Science*, vol. 1, no. 6, pp. 1–13, 2020.

[7] C. Zhen, X. Jingbo, B. Jun, and X. Min, "Feature extraction algorithm based on evolutionary depth learning," *Computer Science Image*, vol. 42, no. 11, p. 11, 2015.

[8] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[9] A. Dundar, J. Jin, B. Martini, and E. Culurciello, "Embedded streaming deep neural networks accelerator with applications," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 7, pp. 1572–1583, 2016.

[10] R. J. Cintra, S. Duffner, C. Garcia, and A. Leite, "Low-complexity approximate convolutional neural networks," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 12, pp. 5981–5992, 2018.

[11] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3573–3587, 2017.

[12] A. Stuhlsatz, J. Lippel, and T. Zielke, "Feature extraction with deep neural networks by a generalized discriminant analysis," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 4, pp. 596–608, 2012.

[13] A. Kamboj, N. Powar *et al.*, "Safety helmet detection in industrial environment using deep learning," in *CS & IT Conference Proceedings*, vol. 10, no. 5. CS & IT Conference Proceedings, 2020.

[14] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, 2020.

[15] L. Huang, Q. Fu, M. He, D. Jiang, and Z. Hao, "Detection algorithm of safety helmet wearing based on deep learning," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 13, p. e6234, 2021.

[16] Y. Li, H. Wei, Z. Han, J. Huang, and W. Wang, "Deep learning-based safety helmet detection in engineering management based on convolutional neural networks," *Advances in Civil Engineering*, vol. 2020, 2020.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[18] X. Long, W. Cui, and Z. Zheng, "Safety helmet wearing detection based on deep learning," in *2019 IEEE 3rd information technology, networking, electronic and automation control conference (ITNEC)*. IEEE, 2019, pp. 2495–2499.

[19] F. Zhou, H. Zhao, and Z. Nie, "Safety helmet detection based on yolov5," in *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, 2021, pp. 6–11.

[20] S. Tan, G. Lu, Z. Jiang, and L. Huang, "Improved yolov5 network model and application in safety helmet detection," in *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, 2021, pp. 330–333.

[21] Z. Jin, P. Qu, C. Sun, M. Luo, Y. Gui, J. Zhang, and H. Liu, "Dwca-yolov5: An improve single shot detector for safety helmet detection," *Journal of Sensors*, vol. 2021, 2021.

[22] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2778–2788.

[23] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, p. 217, 2021.

[24] "Mt-yolov6 pytorch object detection model." [Online]. Available: https://models.roboflow.com/object-detection/mt-yolov6

[25] "Yolov7 pytorch object detection model." [Online]. Available: https://models.roboflow.com/object-detection/yolov7

[26] "Hard hat dataset," 2020. [Online]. Available: https://makeml.app/datasets/hard-hat-workers

[27] G. Munkhjargal, "Safety helmet wearing dataset," 2021. [Online]. Available: https://data.mendeley.com/datasets/9rcv8mm682/4

[28] "Yolov5 pytorch object detection model." [Online]. Available: https://models.roboflow.com/object-detection/yolov5

[29] N. Kwak and D. Kim, "A study on detecting the safety helmet wearing using yolov5-s model and transfer learning," *International Journal of Advanced Culture Technology*, vol. 10, no. 1, pp. 302–309, 2022.