iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.61533

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

Automating Helmet Usage Detection: A YOLOv8 Based Framework

Prof. Sumit Shevtekar[1], Sushant Langhi[2]
[1]Asst.Professor, [2]Student, Department of Computer Engineering Pune Institute of Computer TechnologyPune, India

Abstract: The biggest threat to Powered two-wheeler(PTW) riders is a head injury. With the humongous amount of daily commuters on such PTWs, the risk only increases substantially. In such cases, helmets reduce the risk of head injuries in accidents but not every citizen is keen on their use. In India, where the use of motorcycles is widespread, ensuring helmetuse remains a challenge. This study explores the effectiveness of YOLOv8(You Only Look Once) a state-of-the-art single-stage object detection algorithm for helmet detection on Indian roads [14]. The CNN(Convolutional Neural Networks) based technology implements many other efficient, reliable and quick algorithms to train, validate and predict the object's detection, segmentation and classification task.

We leverage the use of a publicly available Indian helmet dataset from Kaggle, containing 942 images with annotations. This an image dataset of real-life powered two-wheeler riders. The dataset has images and video frames for training on 5 different objects/classes along with annotations that specify those objects. This multi-class approach offers valuable insight into helmet usage patterns paves shows us the way for real-world applications for automated traffic monitoring. Real-time helmet and number plate detection can become a game changer in traffic monitoring and can strengthen basic safety laws.

Index Terms: Helmet Detection, YOLOv8 (You Only LookOnce), Deep Learning, Convolutional Neural Networks(CNN), Traffic Safety, Two-wheeler Riders, Automated Traffic Monitor-ing

I. INTRODUCTION

Ensuring proper helmet usage remains a critical but often neglected aspect of road safety for powered two-wheeler ridersin India. Despite increasing road accident fatalities where headinjuries are a leading cause of death, consistent helmet com- pliance continues to be a challenge [5]. This study investigates the potential of YOLOv8 (You Only Look Once), a cutting- edge image detection algorithm renowned for its single-stage object detection capabilities, to address this critical issue.

Head injuries sustained in motorcycle accidents are a major contributor to road fatalities in India [18]. Existing methods for enforcing helmet usage often rely on manual monitoring, which can be resource-intensive and prone to human error. This research explores the development of an automated helmet detection system using YOLOv8 to enhance road safety for Indian motorcycle riders. YOLOv8, developed by Ultralyt-ics, offers a promising solution for real-world applications due to its speed, accuracy, and single-stage detection approach. By leveraging the power of YOLOv8, we aim to develop a system that can accurately detect helmet usage in real-time traffic scenarios, paving the way for improved traffic monitoring and enforcement strategies.

This research focuses on utilizing YOLOv8 for automated helmet detection specifically tailored to the Indian road en- vironment. A critical aspect of this approach involves the creation and utilization of a curated dataset specifically de- signed for YOLOv8 training and validation. This "Indian Helmet Detection Dataset" comprises 942 high-resolution im- ages capturing the diverse real-world scenarios encountered by Indian riders, including variations in lighting conditions, road infrastructure, and rider demographics. The dataset is metic- ulously split into training (800 images) and validation sets(142 images). Object annotations within the dataset pinpoint the position and shape of the helmet, rider, or number plate. These annotations guide the model's training process, ensuring it learns to effectively detect these objects while maintaining the ability to perform well on unseen data [20].

The focus on an Indian context is crucial, as factors like lighting conditions, road infrastructure, and rider behavior can significantly differ across geographical regions. By incorpo- rating these regional variations into the training data, the model can be better equipped to handle the complexities of Indian road traffic. This study lays the groundwork for the development of a robust and generalizable automated helmet detection system using YOLOv8. Future research directions include:

Enriching the Indian Helmet Detection Dataset with a larger and more diverse set of images will further enhance the model's generalizability and ability to handle various real- world scenarios. Implementing data augmentation techniques can artificially increase the dataset size and introduce varia- tions in lighting, weather conditions, and image noise. This can improve the model's robustness to unseen data.

Exploring the feasibility of deploying the YOLOv8 helmet detection model on edge devices with limited computational resources will pave the way for its integration into practical traffic monitor- ing systems. Collaborating with traffic authorities or helmet manufacturers can provide access to real-world traffic camera data or controlled testing environments, further refining the model's performance. By addressing these future directions, this research has the potential to significantly contribute to improving road safety in India and potentially other regions facing similar challenges with helmet usage compliance.

II. RESEARCH GAPS AND OBJECTIVES

The primary objective is to develop a YOLOv8 model specifically trained for helmet detection on Indian roads, leveraging the Indian Helmet Detection Dataset. By utilizing a YOLOv8 variant known for its balance between accuracy and speed, we explore the feasibility of real-time helmet detection on resource-limited platforms. The evaluation of processing speed for each stage (pre-processing, inference, post-processing) provides valuable insights into the model's potential for deployment in practical scenarios. This research contributes to the field by:

Expanding the scope of helmet detection datasets: By in- corporating an Indian-specific dataset, we contribute to a more comprehensive understanding of helmet usage patterns across diverse geographical regions.

Investigating processing efficiency: The analysis of pro-cessing speed at each point highlights areas for potential optimization and shows us the way for further development ofthe model for devices with limited computational resources.

III. RELATED WORK

Powered two-wheeler(PTW) helmet usage remains a crucialsafety challenge for motorcyclists globally, particularly in regions with high motorcycle usage like India [7]. Automated helmet detection offers a great solution for improving road safety by enabling real-time monitoring and enforcement. The literature review explores the application of YOLOv8, a state- of-the-art object detection algorithm and other technologies like Faster RCNN(Faster Region-Convolutional Neural Net- work) [13] and others [16], for helmet detection tasks.

Several studies have demonstrated the effectiveness of YOLO models for helmet detection. Deng et al. (2022) [1] proposed a lightweight YOLOv3 variant specifically designed for safety helmet detection. Their model achieved promising results on a custom helmet dataset, highlighting the potential of YOLO architectures for this task. Gu et al. (2019) [2] also explored a deep learning approach for helmet detection, using the Faster RCNN model architecture. [17] [19]

The research by Jo¨nsson Hyberg & Sjo¨berg (2023) [3] investigated YOLOv8's performance in pedestrian detection, demonstrating its accuracy and efficiency. Lin (2024) [4] further explored YOLOv8 for helmet detection, proposing im- provements to the model architecture for better performance. These studies highlight the potential advantages of YOLOv8 for helmet detection, including:

Real-time processing: YOLOv8's single-stage architecture enables real-time object detection, crucial for practical appli- cations like traffic monitoring systems.

Accuracy: YOLOv8 models demonstrate high accuracy in object detection tasks.

Efficiency: YOLOv8 offers a good balance between ac- curacy and computational efficiency, making it suitable for deployment on resource-constrained devices.

Our research mainly focuses on the YOLOv8 algorithm performance in cities in India providing valuable insights into the model's performance in a specific geographical context thus, contributing to the development of robust and efficient helmet detection systems.

IV. KEY CONCEPTS TO UNDERSTAND

A. Computer Vision, Image and Object Detection

Computer vision is a multidisciplinary field that includes techniques from computer science, mathematics, and artifi- cial intelligence to enable machines to acquire a high-level understanding of visual data. It achievers tasks such as im- age processing, object recognition, scene understanding, and image-based modeling. Image detection is a fundamental con- cept in computer vision that encompasses object recognition, surveillance, and autonomous systems [9] [10].

Thus, the task of object detection is performed by the YOLOv8 model that involves the identification of the locationand class/type of an object in an image or video file. Its outputis a collection of bounding boxes that enclose the objects in theimage, along with their label tags and probabilities/confidencescores for each box displayed. This is used especially to identify objects of interest in a scene without knowing the shape, location, size of the object.

B. YOLOv8 Framework

The YOLOv8 model provided by Ultralytics is fast, accurate and easy to use is an excellent option for a wide range of object detection and tracking, instance segmentation and imageclassification. It provides a model that can be custom trained on image and video datasets as well as pre-trained models trained to detect, segment and pose on the COCO(Common Objects in Context) dataset while classified models are pre- trained on the ImageNet dataset. Post-installation it may be used directly in the Command Line Interface (CLI) with ayolo command or a as a PyTorch pre-trained *.pt models to create a model instance in Python.

The YOLOv8 AI framework is such that it can work in6 different modes each engineered to provide flexibility and efficiency for different tasks and use cases:

1) Train: Train your model on your custom dataset or pre-loaded datasets.
2) Val: Validate your model's performance post-training ofthe model.
3) Predict: Implement the model on real-world data forprediction based on your requirement.
4) Export: Deploy your model in various formats.
5) Track: Use your model for a real-time object or scene-tracking applications.
6) Benchmark: Analyze and check the speed and accuracyof your model in various deployment environments.

It can also perform 4 different computer vision tasks basedon different objectives and use cases:

a) Detection: This is the primary task for the model andinvolves detecting objects in an image or video frame and drawing boxes that contain the detected object. It further classifies these objects into different categories based on their features. It can detect multiple objects ina single frame simultaneously.
b) Segmentation: This includes segmenting the identified objects into different regions based on the content of the image. It is useful in image segmentation and medical imaging using the U-Net deep learning architecture.
c) Classification: Classifying the detected and segmented objects into different categories based on their content, environment and scenes.
d) Pose: It is the task of identifying specific points in an image or video frame. These key points are used to trackthe movement or pose estimation of the object. YOLOv8can detect such key points with high speed and accuracy.

YOLOv8 Implements various mathematical models rangingfrom:

- *Convolutional Neural Networks (CNNs):*
  - *Convolution Operation:* Involves capturing spatial relationships between pixels after sliding a filter that contains across the input image.

$$(f * g)(x) = \int_{-\infty}^{\infty} f(t)g(x - t)\,dt$$

where:

$f(t)$ and $g(t)$: are the functions being convolved.

x: the variable at which the convolution result is evaluated.

  - *Pooling Operations:* The dimensionality of the fea- ture maps extracted by convolutions are reduced by Max Pooling or Average Pooling.

$$\text{Max Pooling}(x, y) = \max_{i,j}((x + i, y + j))$$

- *mAP (Mean Average Precision):*
  - *Precision:* This metric represents the proportion of detection that are correct (do not include false positives).

$$Precision = TP/(TP + FP)$$

  - *Recall:* This metric represents the proportion of actual objects that were correctly detected by the model (doesn't include false negatives)

$$Recall = TP/(TP + FN)$$

where:

TP: True Positive FP: False Positive TP: True Positive TN: True Negative

The provided models by Ultralytics trained on the COCO(Common Objects in Context) dataset includes 80 pre- defined classes ranging from animate objects like a person, birds, cats and dogs to inanimate objects like bicycles, cars, motorcycles, traffic lights and many others.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

Different models provide effective use and are classified based on size(pixels), mAP^{val} (mean Average Precision value) ranging from 50-95 milliseconds, Speed A100 TensorRT (milliseconds), params (Million), FLOPs (Billion)The different models are as follows:

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Fig. 1. Models by Ultralytics

V. METHODOLOGY AND MODEL ARCHITECTURE

A. Dataset Description

The Indian Helmet Detection Dataset contains 942 images of Indian Road Traffic including riders and their powered two-wheelers [22]. The dataset has been further divided into train containing 800 images and valid including 142 images.

These images are captured in diverse locations across India, encompassing different lighting conditions, road infrastructure variations, and weather situations. Each image is accompanied by a corresponding annotation file in a standard format which specifies the following:

- 0: Number Plate.
- 1: Face with No Helmet.
- 2: Face with Good Helmet.
- 3: Face with Bad Helmet.
- 4: Rider.

The class distribution (number of images per category) is crucial information. An imbalanced dataset (unequal distribu- tion of images across classes) might require specific strategies during training to ensure the model performs well for all helmet usage categories.

B. Architecture

YOLOv8 stands out for its efficient single-stage architec- ture, making it well-suited for real-time object detection tasks [12]. Here's a concise overview of its key components:

1) *Backbone Network:* The foundation is a Convolutional Neural Network (CNN) backbone, often a modified Darknet variant (e.g., CSPDarknet53). This network extracts features from the input image at various levels of detail, providing a rich representation for object detection.
2) *Neck Networks:* Some YOLOv8 variants might include a neck network. This network refines the feature maps extracted by the backbone at different stages, combining them to create a more comprehensive image representation that incorpo- rates information from various resolutions.
3) *Head Network:* The head network takes the processed feature maps (from the backbone and optionally the neck) and performs the final detections. It typically consists of several convolu- tional layers followed by fully connected layers. These layers predict bounding boxes for potential objects and classify them into predefined categories.

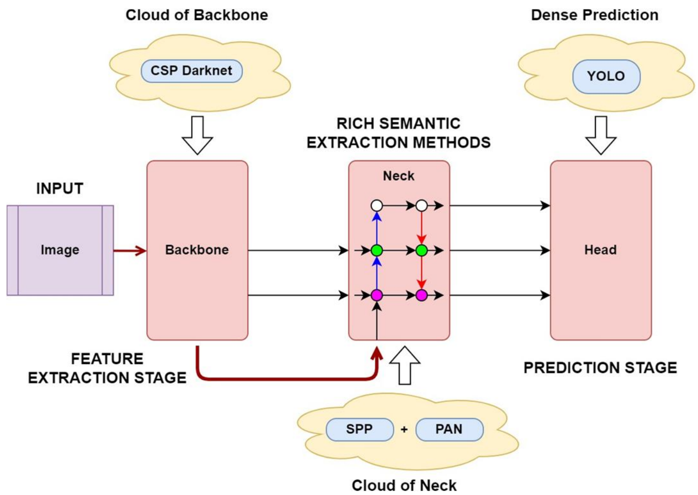

Fig. 2. Architecture

Thus, YOLOv8 takes an input image, feeds it through the backbone network for feature extraction optionally refines the features through the neck network, and finally utilizes the headnetwork to predict bounding boxes and classify objects.

VI. PROPOSED MODEL CREATION

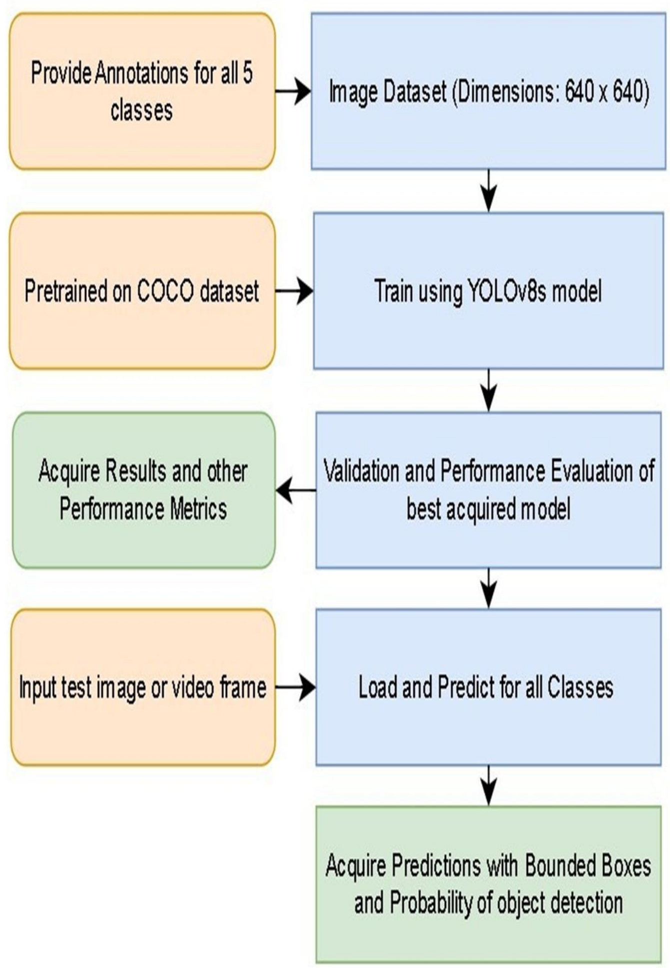

Fig. 3. Process Flow Diagram

A. Training

From the described dataset, 800 images were used for training a YOLOv8s pre-trained model. The dataset includes motorcycle riders with and without helmets.

Fig. 4. Train images

To increase model generalizability and prevent over-fitting, the image size was fixed. The model was trained with the parameters. The model was initially trained for 100 epochs but the best result was considered at the 48th iteration. Thus, to remove and redundancy and over-fitting introduced, the following train model was considered best:

epochs=100: This parameter sets the number of cycles the entire training dataset will be trained and validated throughthe model.

batch=16: This defines the number of images processed by the model in each training iteration.

imgsz=640: This specifies the size (resolution) to which thetraining images will be resized before feeding them into the model.

optimizer=SGD: This indicates the optimizer algorithm usedto update the model's weights during training. SGD (StochasticGradient Descent) is being used here.

lr=0.01: This is the initial learning rate, which controls the magnitude of updates to the model's weights.

momentum=0.937, weightdecay=0.001: These are addi- tional hyper-parameters used with the SGD optimizer to improve convergence and stability.

B. Experimental settings

Hardware

Workstation: Lenovo ThinkStation P620CPU: Powerful multi-core CPU

GPU: NVIDIA RTX A4000 GPU

Memory: 64GB GDDR6 memorySoftware Operating System: Linux distribution (e.g., Ubuntu)Deep Learning Framework:

PyTorch 1.12.1 Programming Language: Python 3.7

Libraries

PyTorch 1.12.1 (Deep Learning)NumPy (Numerical Computing) Scikit-learn (Machine Learning)

C. Experimental Results

The model's performance was evaluated on the 142 val- idation images and using metrics like mAP (mean Average Precision), which considers both how many objects were correctly detected and how well their bounding boxes localizedthe objects. This validation process helps identify any over- fitting issues. From which we concluded with the performancemetrics:

Overall mAP of 0.676: This indicates a good ability to detectobjects across all classes, which is a great point.

High mAP for riders (0.9) and number plates and good helmets(0.736 and 0.745): The model seems to be good at detecting these important objects.

Relatively fast processing speed: The model can process im-ages fairly quickly, which is crucial for real-time applications.

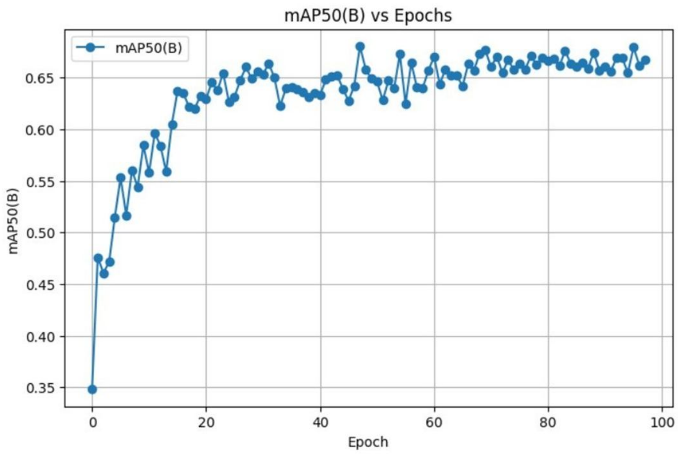

Fig. 5. Epoch VS Precision

D. Testing

To evaluate the generalizability and real-world applicabilityof the trained YOLOv8 model, we conducted testing on unseenimages not included in the training or validation sets. A single test image was processed, providing insights into the model’s processing speed for each stage:

Pre-processing (0.6ms per image): This initial stage involvestasks like image resizing and normalization. The relatively fastpre-processing time indicates efficient handling of image data preparation.

Inference (16.3ms per image): This core step encompasses passing the pre-processed image through the YOLOv8 model to generate detections.

Fig. 6. Validation Images

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

The inference speed is similar to the validation speed suggests potential consistency between controlled and real-world scenarios. Post-processing (98.3ms to under 2ms): This final stage varies depending on the image complexity and the specific post-processing tasks performed. It often involves decoding detections from the model's output format and applying non- max suppression (NMS) to remove redundant bounding boxes.

VII. RESULTS AND DISCUSSION

Now, we discuss the important factors deduced from our study and the use of our model. The model is optimized fora relatively small dataset and yet performs well for any given test image or video frame. The results are as follows:

1) *Confusion Matrix;* The model seems to perform well on classes like the "numberPlate", "rider", "faceWithGood-Helmet class has high values on the diagonal, indicatinga good number of correct detections for license plates, helmets and riders as required. Also, considering the fine line between identifying a good and bad helmet taking into consideration that the lighting conditions and other visual aspects affect the prediction as well as training and validation
2) *Precision Curve:* The mAP of 0.676 indicates a some- what balanced ability to detect objects across all classes in the dataset. The model performs well at detecting riders (mAP: 0.9) and number plates (mAP: 0.736), which are crucial for your helmet detection task. This suggests the model can effectively localize these objectsin the images.
3) *Precision-Recall Curve:* An ideal curve would be in the top left corner, indicating high precision (most detection are correct) and high recall (the model finds most helmets).

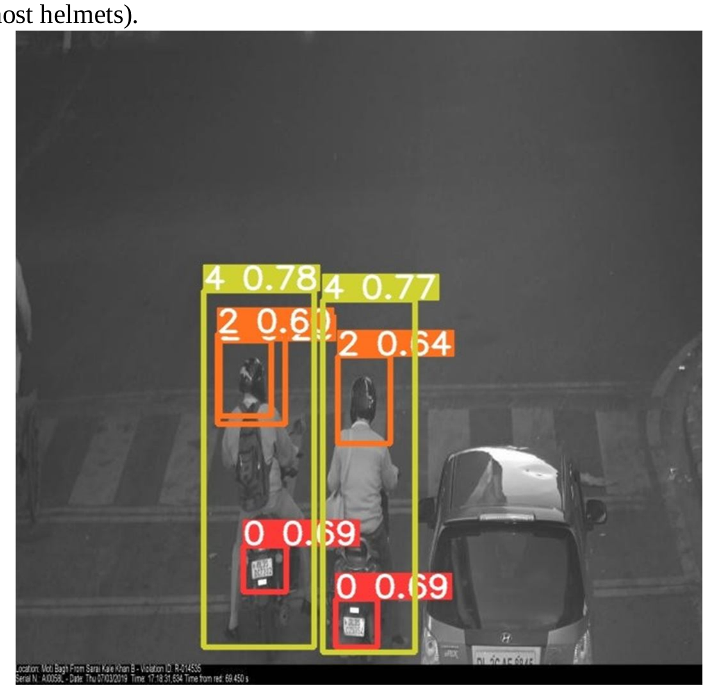

Fig. 7. Prediction image.

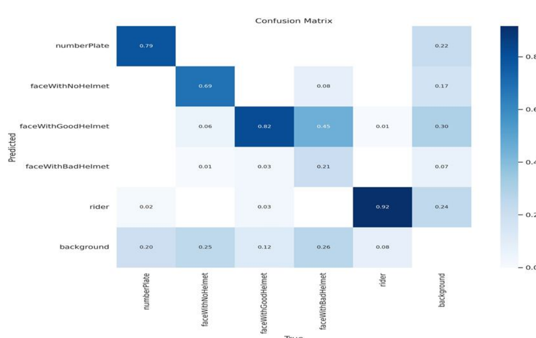

Fig. 8. Confusion Matrix

The curve suggests a good performance inhelmet detection. It starts at a high precision of 1.0and very gradually decreases as the model detects more images (increasing recall very slowly).

4) *Training and Validation Loss:* The training loss curve (blue) generally decreases over epochs, indicating the model is significantly learning from the training data. The validation loss curve (green) also seems to be de- creasing, suggesting the model is generalizing well and avoiding over-fitting to the training data.

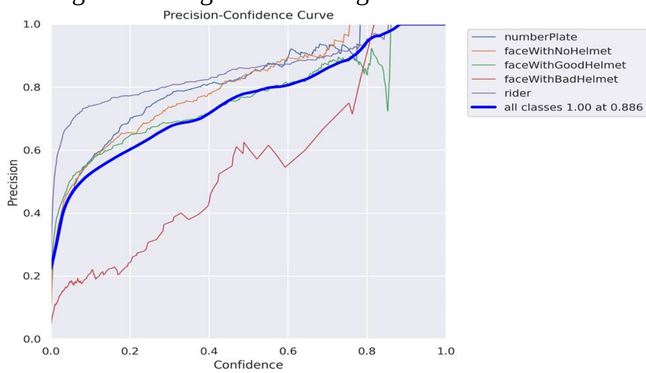

Fig. 9. Precision Curve

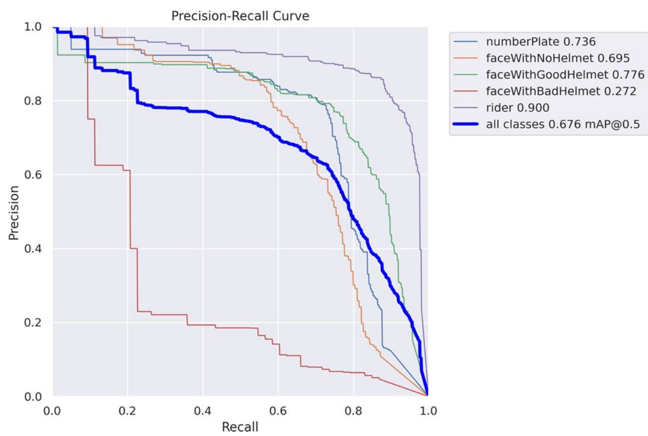

Fig. 10. Precision-Recall curve

The training mAP curve (orange) increases over epochs, as expected during our training. The validation mAP curve (red) also shows an increase, indicating improvement in general object detection across all classes in your dataset. Itwould be beneficial to see the numerical values on the y-axis to determine the specific mAP achieved.

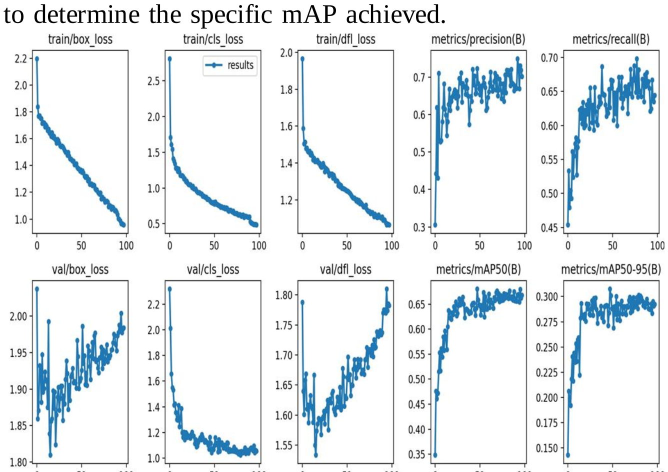

Fig. 11. Training and Validation Losses

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

Model	Accuracy (%)	Recall (%)	F1-score (%)
YOLOv8	70	64	66

Fig. 12. Accuracy, Precision and Recall

VIII. CONCLUSION

Thus general trends observed in the provided figures suggest that the YOLOv8s model is on a promising path for Indian helmet detection. The decreasing training and validation loss curves indicate successful learning from the provided training data. This is further influenced by the increasing mAP curves, suggesting improvement in object detection across various classes in the dataset.

The confusion matrix reinforces these positive trends. The high diagonal values illustrate a significant number of correct detections for crucial classes like ”numberPlate,” ”rider,” and ”faceWithGoodHelmet.” This accurate localization of riders and license plates is critical for effective helmet detection.

The precision-recall curve offers further insights into the model’s performance. This curve, ideally positioned in the top left corner, suggests a balance between high precision (most detections are correct) and high recall (the model finds most helmets). While the available portion of the curve indicates a good starting point, further optimization might be beneficial to improve both precision and recall.

It’s important to acknowledge the challenges associated with differentiating between ”good” and ”bad” helmets. Factors like lighting conditions and visual variations influences prediction accuracy. Addressing these challenges might involve adding techniques for illumination correction or data augmentation with diverse lighting scenarios within the training process.

Overall, the results suggest that the YOLOv8s model ex- hibits promising capabilities for Indian helmet detection. It demonstrates the ability to learn from the training data and improve its object detection performance, particularly for riders and license plates, which are key elements for helmet identification. Further optimization through techniques like hyperparameter tuning, data augmentation, and addressing class-specific challenges can potentially lead to even better performance and a more robust helmet detection system.

IX. FUTURE SCOPE

This study paves the way for further advancement in stud- ies for several conditions/constraints that affect the acquired image quality and image detection model updation.

1) *Low visibility helmet detection:* Since many motorcycle journeys occur during low-light conditions (dusk, dawn, poorly lit streets or adverse weather conditions), inves- tigating the model’s performance and potential improve- ments in low-light scenarios is crucial. Since many mo- torcycle journeys occur during low-light conditions (dusk, dawn, or poorly lit streets), investigating and improving the model’s performance and potential improvements in low-light scenarios are very important.
2) *Real-time Video Implementation:* While image-based de- tection offers great insights, real-time video processing is essential for practical, real-life applications like traffic monitoring systems. Adapting the model for real-time video analysis, potentially leveraging hardware accel- eration techniques would be a significant step towards deployment.
3) *Image Segmentation for Focus and Efficiency:* Traffic road pictures can contain a lot of background information that might not be relevant for helmet detection. Exploring image segmentation techniques to isolate the regions of interest (riders, helmets and number plates) could improve processing efficiency and thus, lead to better model performance. This could involve segmenting the entire image or focusing on specific areas where riders are likely to be present.

The future scope of this study does not end here and can be further extended to practical implementation for the betterment and enhancement of road safety and regulation laws/systems.

X. DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] Deng, Lixia, et al. ”A lightweight YOLOv3 algorithm used for safety helmet detection.” Scientific reports 12.1 (2022): 10981.
[2] Yang, Wenhan, et al. ”Advancing image understanding in poor visibility environments: A collective benchmark study.” IEEE Transactions on Image Processing 29 (2020): 5737-5752.
[3] Gu, Yuwan, et al. ”An advanced deep learning approach for safety helmet wearing detection.” 2019 International conference on internet of things (IThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData). IEEE, 2019.

[4] Jönsson Hyberg, Jonatan, and Adam Sjöberg. ”Investigation regarding the Performance of YOLOv8 in Pedestrian Detection.” (2023).
[5] Lin, Bingyan. ”Safety Helmet Detection Based on Improved YOLOv8.”IEEE Access (2024).
[6] Sagar, SM Krishna, et al. ”Pattern of Head Injuries in Deaths Dueto Two-Wheeler Road Traffic Accidents.” Prof.(Dr) RK Sharma 21.1 (2021): 866.
[7] Deshpande, Anjali, and Vinit Tribhuvan. ”A Novel Machine Learning- Based Approach for Analysis, Prediction and Prevention of Accidents on Indian National Highways.” 2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA). IEEE, 2023.
[8] Huang, Zhongjie, et al. ”Research on traffic sign detection based on improved YOLOv8.” Journal of Computer and Communications 11.7 (2023): 226-232.
[9] Voulodimos, Athanasios, et al. ”Deep learning for computer vision: A brief review.” Computational intelligence and neuroscience 2018 (2018).
[10] Guo, Yanming, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew. ”Deep learning for visual understanding: A review.” Neurocomputing 187 (2016): 27-48.
[11] Theofilatos, A. and Yannis, G., 2015. A review of powered-two-wheeler behaviour and safety. International journal of injury control and safety promotion, 22(4), pp.284-307.
[12] Jiang, Peiyuan, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. ”A Review of Yolo algorithm developments.” Procedia computer science 199 (2022): 1066-1073.
[13] Du, Juan. ”Understanding of object detection based on CNN family and YOLO.” In Journal of Physics: Conference Series, vol. 1004, p. 012029. IOP Publishing, 2018.
[14] C´ orovic´, Aleksa, et al. ”The real-time detection of traffic partici- pants using YOLO algorithm.” 2018 26th Telecommunications Forum (TELFOR). IEEE, 2018.
[15] Liu, Wenyu, Gaofeng Ren, Runsheng Yu, Shi Guo, Jianke Zhu, and Lei Zhang. ”Image-adaptive YOLO for object detection in adverse weather conditions.” In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 2, pp. 1792-1800. 2022.
[16] Srivastava, Shrey, Amit Vishvas Divekar, Chandu Anilkumar, Ishika Naik, Ved Kulkarni, and V. Pattabiraman. ”Comparative analysis of deep learning image detection algorithms.” Journal of Big data 8, no. 1 (2021): 66.
[17] Li, Zewen, et al. ”A survey of convolutional neural networks: analysis, applications, and prospects.” IEEE transactions on neural networks and learning systems 33.12 (2021): 6999-7019.
[18] D'Alconzo, Alessandro, et al. ”A survey on big data for network traffic monitoring and analysis.” IEEE Transactions on Network and Service Management 16.3 (2019): 800-813.
[19] M. R. Arun, M. Anto Bennet, S. D Satav, V. Manikandan, M. Gopi Anand and N. A. Dawande, ”Deep Learning Based Object Detection in Surveillance Video,” 2023 4th International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2023, pp. 1794-1801, doi: 10.1109/ICOSEC58147.2023.10276262.
[20] Laroca, Rayson, et al. ”A robust real-time automatic license plate recognition based on the YOLO detector.” 2018 international joint conference on neural networks (ijcnn). IEEE, 2018.
[21] Minaee, Shervin, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. ”Image segmentation using deep learning: A survey.” IEEE transactions on pattern analysis and machine intelligence 44, no. 7 (2021): 3523-3542.
[22] https://www.kaggle.com/datasets/aryanvaid13/indian-helmet-detection- dataset