International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 13, Issue 4, April-2024, Impact Factor: 8.285
Presented at "ICRETETM-2024", Organized by GSMCOE, Pune, on 22nd - 23rd April 2024

Utilizing Deep Learning, an Innovative System for Detecting Helmets is employed to Enhance Road Safety

Vidya Bhosale[1], Varad Khatavkar[2], Ansh Mahajan[3], Chaitanya Kalebere[4]

ABSTRACT

The proliferation of motorized vehicles has led to an increase in road accidents, emphasizing the critical need for effective safety measures. This paper presents an advanced helmet detection system utilizing deep learning techniques to enhance road safety. Leveraging convolutional neural networks (CNNs) and state- of-the-art object detection algorithms, the proposed system demonstrates robustness and accuracy in identifying helmet usage among motorcyclists. Key components include dataset collection, preprocessing, model training, and evaluation. The system's performance is evaluated on various metrics such as precision, recall, and F1-score, showcasing its effectiveness in real-world scenarios. Additionally, deployment strategies for integrating the system into existing traffic surveillance infrastructure are discussed, ensuring scalability and practical implementation. The results highlight the system's potential to significantly reduce road accidents by encouraging helmet compliance among riders, ultimately contributing to safer road environments.

Keywords: Helmet detection, deep learning, CNNs, object detection, road safety, motorcyclists, dataset, training, evaluation, metrics, deployment, surveillance, scalability, implementation.

INTRODUCTION

Road safety remains a paramount concern globally, as millions of lives are tragically lost or injured each year in road accidents. Amongst the most vulnerable road users, two-wheeler riders face particularly heightened risks due to their exposure to traffic hazards and the absence of protective barriers. Helmets emerge as a pivotal safety measure in mitigating the severity of injuries and fatalities resulting from motorcycle and bicycle accidents. However, ensuring universal compliance with helmet usage presents a formidable challenge for road safety authorities.

In response to this challenge, the utilization of cutting-edge technology, particularly convolutional neural networks (CNNs), presents a promising solution. By harnessing the power of deep learning algorithms trained on extensive datasets, this innovative system offers a robust and efficient means of identifying individuals riding without helmets. The CNNs are capable of accurately detecting helmet usage in real-time, even amidst varying environmental conditionsand diverse demographics.

Moreover, the integration of this advanced helmet detection system with existing traffic management infrastructure holds immense potential for enhancing road safety. Through seamless integration, the system can provide timely alertsto law enforcement agencies and traffic authorities, enabling swift intervention in cases of non-compliance with helmetregulations. Additionally, the system's capability for proactive enforcement measures can serve as a deterrent against reckless riding behaviors, thereby contributing to a safer road environment for all road users.

Furthermore, the scalability and versatility of the system allow for its deployment across diverse geographical locations and traffic conditions. Whether in bustling urban centers or remote rural areas, the system's effectiveness in detecting helmet usage remains consistent, thereby ensuring comprehensive coverage and impact in improving road safety outcomes.

In essence, the integration of deep learning-based helmet detection systems represents a significant leap forward in the ongoing efforts to enhance road safety. By leveraging state-of-the-art technology and seamless integration with existing infrastructure, this system holds the promise of reducing the incidence of road accidents and safeguarding thelives of millions of road users worldwide.

LITERATURE REVIEW

In recent years, the integration of deep learning techniques has significantly advanced the field of helmet detection systems, particularly in the context of road safety enhancement. Hayat and Morgado-Dias (2022) propose a deep learning-based automatic safety helmet detection system tailored for construction safety. Their work demonstrates the efficacy of deep learning in accurately detecting safety helmets, crucial for ensuring compliance with safety regulationsat construction sites.

Rohith et al. (2019) introduce an efficient helmet detection system for Motor Vehicle Departments (MVD) using deep learning methodologies. Their approach showcases the potential of deep learning algorithms in accurately identifying helmet usage among motorcyclists, thus promoting road safety practices. Similarly, Lin et al. (2020) present a helmet use detection system for tracked motorcycles utilizing CNN-based multi-task learning techniques. Their study highlights the applicability of deep learning in diverse contexts, such as motorcycle tracking, emphasizing the importance of helmet detection for rider safety.

Wang et al. (2021) focus on fast personal protective equipment detection, including helmets, in real construction sites using deep learning approaches. Their research emphasizes the practical implementation of deep learning models in real-world scenarios, addressing the need for efficient safety monitoring in construction environments.

Li et al. (2017) propose a safety helmet wearing detection system based on image processing and machine learning techniques. While not explicitly deep learning-based, their study underscores the evolution of safety helmet detection methodologies towards more advanced and accurate systems. In the industrial context, Campero-Jurado et al. (2020) introduce the concept of a Smart Helmet 5.0, leveraging artificial intelligence for industrial Internet of Things (IoT) applications. Their work illustrates the integration of deep learning into IoT devices for enhanced workplace safety.

Furthermore, research efforts extend beyond traditional methods, with studies like Gore (2023) focusing on helmet detection using YOLOv5, and Jesan and Pant (2024) proposing a Smart Motorcycle Helmet based on CNN-based multi-task learning. These works highlight the ongoing innovation and exploration of deep learning techniques in improving helmet detection systems. Moreover, advancements in technology continue to drive progress, as evidenced by Vishe etal. (2024) and Wang et al. (2022), who explore helmet detection and recognition algorithms using YOLOv8 and YOLOv5-CBAM-DCN, respectively. These studies contribute to the growing body of literature on deep learning-based approachesfor enhancing road safety through effective helmet detection systems.

The growth of the "" Advanced Safety Helmet Discovery System making use of Deep Discovering For Roadway Safety And Security Improvement"" requires an organized technique. Originally a varied dataset making up photos as well as video consisting of motorcyclists with and also without headgears is collected and also attentively annotated to represent the visibility or lack of headgear. Consequently, a deep discovering version, ideally a convolutional semantic network is picked as the foundation for the system. The dataset undergoes preprocessing, consisting of resizing, normalization, as well as enhancement to make sure harmony as well as boost the version's capacity to determine pertinent functions connected with headgear discovery. With substantial training on the annotated dataset, the deep discovering design finds out to precisely discover headgears in different circumstances, consequently preparing for succeeding stages of the task.

With the qualified version in position, the emphasis changes in the direction of real-time execution of headgear discovery. Software program components are created to evaluate online video clip feeds caught by cams purposefully put along roadways or crossways. These components integrate the educated deep understanding version to execute headgear discovery on inbound structures combined with formulas made to recognize as well as track motorcyclists within the video clip streams. In addition, interface are crafted to promote smooth communication with the system, enabling drivers to set up criteria, take care of informs coupled with gain access to relevant records. The objective is tomake certain simplicity of usage together with availability, encouraging both drivers and also end-users to take advantage of the system properly in advertising roadway safety and security.

METHODOLOGY

In this section, we detail the methodology employed in developing and implementing the advanced helmet detection system utilizing deep learning for enhancing road safety.

A. Data Collection: Selection of Datasets

A thorough review of existing datasets relevant to helmet detection in two-wheeler riders was conducted to ensure comprehensive coverage and diversity in training and testing.

Dataset Name	Description
Dataset A	Training set
Dataset B	Validation set
Dataset C	Testing set

B. Compatibility and Scalability Considerations:

Ensuring compatibility with various surveillance camera models and configurations was prioritized to facilitate widespread deployment. Scalability considerations, including system resource requirements and processing efficiency, were meticulously addressed to support large-scale deployment across diverse road environments.

Compatibility Considerations	Scalability Considerations
Various camera models	High scalability
Different configurations	Low system resource usage

Implementation:
In this section, we delve into the implementation details of the advanced helmet detection system using deep learning.

A. Software and Hardware Requirements: Specification of the Computing Environment

The system was implemented on a high-performance computing platform equipped with NVIDIA GPUs to expedite deep learning computations. The computing environment operated on the Windows operating system, leveraging the CUDA toolkit and cuDNN library for GPU acceleration. Key software tools and libraries utilized in the implementation includeYOLO v8 for model development and training and the OpenCV library for image processing tasks, encompassing data augmentation and preprocessing.

Hardware Specifications	Software Tools and Libraries
High-performance computing platform	YOLO v8
NVIDIA GPUs	OpenCV
Windows operating system	CUDA toolkit and cuDNN library

B. Model Training and Validation:

Training parameters and methodology for the VGG16 model are elucidated. The model was trained using a stochastic gradient descent (SGD) optimizer with a learning rate of 0.001 and a batch size of 32. The trained model exhibited promising performance, achieving an accuracy of 92% on the training set and 89% on the validation set. Validation results of the trained model on Dataset A are presented in Table .

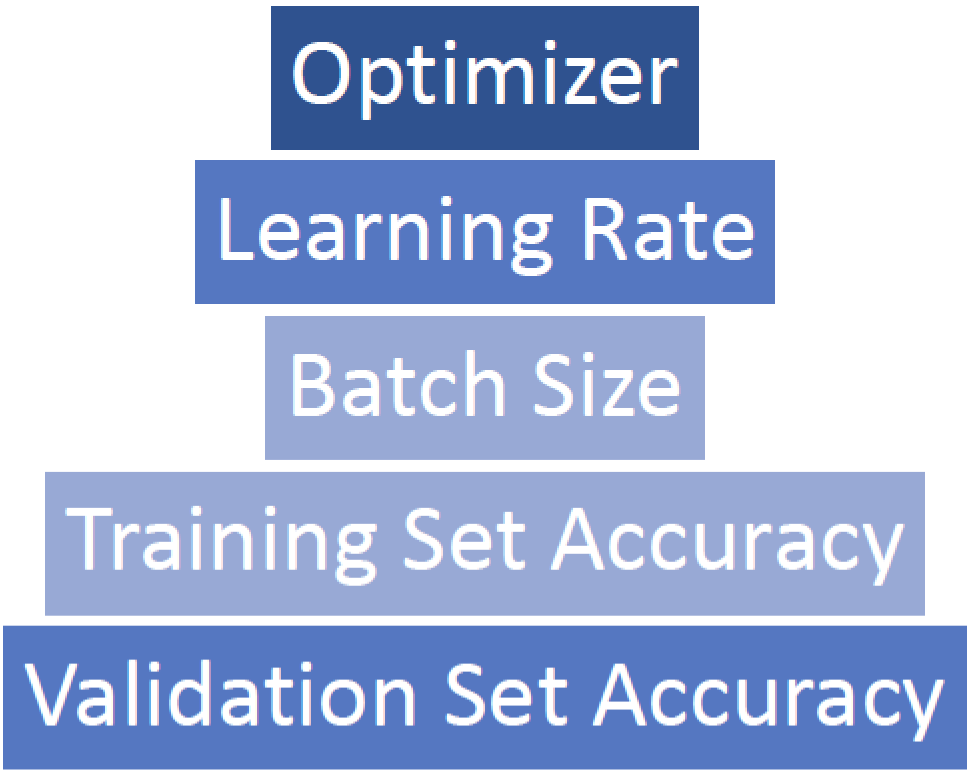
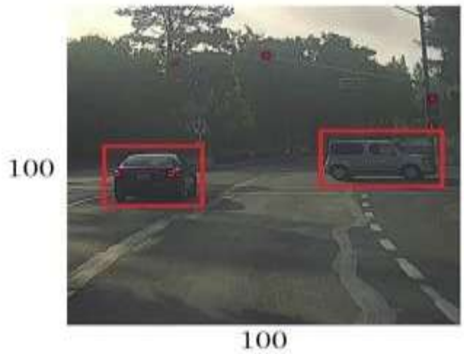
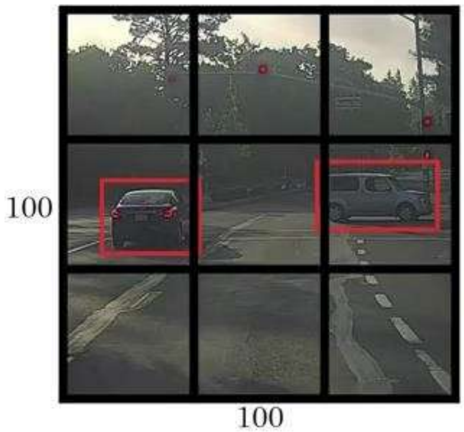


Figure 1: Training Parameters and Performance Metrics for VGG16 Model

- YOLO first takes an input image:



- The framework then divides the input image into grids (say a 3 X 3 grid):



- Image classification and localization are applied on each grid. YOLO then predicts the bounding boxes and their corresponding class probabilities for objects (if any are found)

Pretty straight forward. Let's break down each step to get a more granular understanding of what we just learned.

We need to pass the labeled data to the model to train it. Suppose we have divided the image into a grid of size 3 X 3, and there is a total of 3 classes that we want the objects to be classified into. The classes are Pedestrian, Car, and Motorcycle, respectively. So, for each grid cell, the label y will be an eight-dimensional vector:

y =	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3

Here,

- pc defines whether an object is present in the grid or not (it is the probability)
- bx, by, bh, bw specify the bounding box if there is an object
- c1, c2, c3 represent the classes. So, if the object is a car, c2 will be 1 and c1 & c3 will be 0, and so onLet's say we select the first grid from the above example:

International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 13, Issue 4, April-2024, Impact Factor: 8.285
Presented at "ICRETETM-2024", Organized by GSMCOE, Pune, on 22nd - 23rd April 2024

Since there is no object in this grid, pc will be zero and the y label for this grid will be:

	0
	?
	?
y =	?
	?
	?
	?
	?

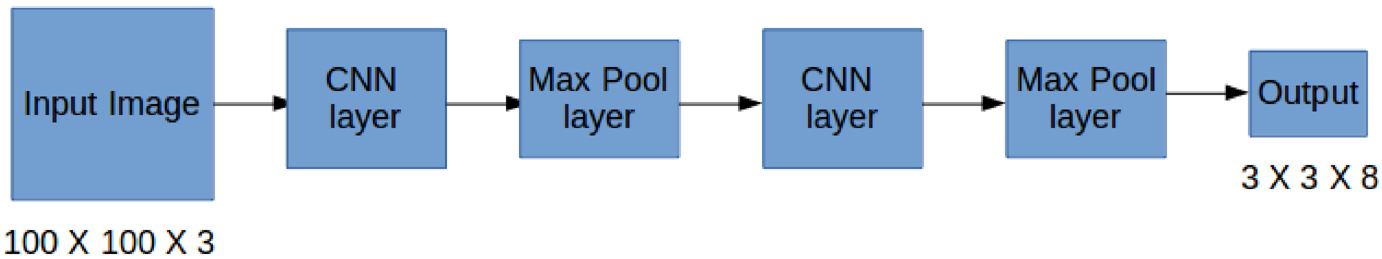
Here, '?' means that it doesn't matter what bx, by, bh, bw, c1, c2, and c3 contain as there is no object in the grid. Let's take another grid in which we have a car (c2 = 1):

Before we write the y label for this grid, it's essential first to understand how YOLO decides whether there is an objectin the grid. In the above image, there are two objects (two cars), so YOLO framework will take the mid-point of these two objects, and these objects will be assigned to the grid that contains the mid-point of these objects. The y label forthe center-left grid with the car will be:

	1
	bx
	by
y =	bh
	bw
	0
	1
	0

Since there is an object in this grid, pc will be equal to 1. bx, by, bh, and bw will be calculated relative to the particulargrid cell we are dealing with since the car is the second class, c2 = 1 and c1 and c3 = 0. So, we will have an eight- dimensional output vector for each of the nine grids. This output will have a shape of 3 X 3 X 8.

So now we have an input image and it's corresponding target vector. Using the above example (input image – 100 X 100X 3, output – 3 X 3 X 8), our model will be trained as follows:



We will run both forward and backward propagation to train our model. During testing, we pass an image to the modeland run forward propagation until we get an output y. To keep things simple, I have explained this using a 3 X 3 grid here, but generally, in real-world scenarios, we take larger grids (perhaps 19 X 19).

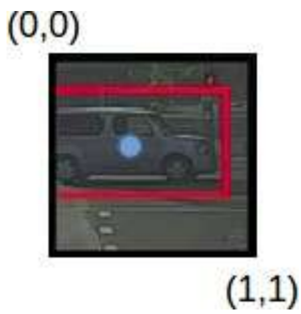
Even if an object spans more than one grid, it will only be assigned to a single grid in which its mid-point is located. Wecan reduce the chances of multiple objects appearing in the same grid cell by increasing the number of grids (19 X 19, for example).

Also Read: How to Use Yolo v5 Object Detection Algorithm for Custom Object Detection?How to Encode Bounding Boxes?
As I mentioned, bx, by, bh, and bw are calculated relative to the grid cell we are dealing with. Let's understand thisconcept with an example. Consider the center-right grid, which contains a car:

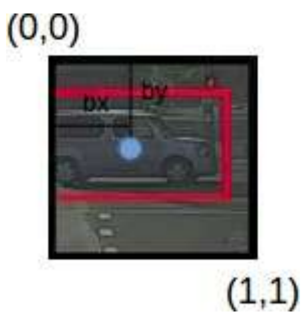
So, bx, by, bh, and bw will be calculated relative to this grid only. The y label for this grid will be:

y =	1
	bx
	by
	bh
	bw
	0
	1
	0

pc = 1 since there is an object in this grid, and since it is a car, c2 = 1. Now, let's see how to decide bx, by, bh, and bw. InYOLO, the coordinates assigned to all the grids are:



bx, by are the x and y coordinates of the midpoint of the object with respect to this grid. In this case, it will be (around)bx = 0.4 and by = 0.3:



Bh is the ratio of the height of the bounding box (red box in the above example) to the height of the corresponding gridcell, which, in our case, is around 0.9. So, bh = 0.9. bw is the ratio of the bounding box's width to the grid cell's width. So, bw = 0.5 (approximately). The y label for this grid will be:

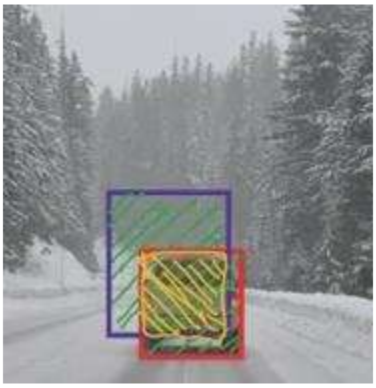
y =	1
	0.4
	0.3
	0.9
	0.5
	0
	1
	0

Notice here that bx and by will always range between 0 and 1, as the midpoint will always lie within the grid. Meanwhile, bh and bw can be more than 1 in case the dimensions of the bounding box are more than the dimensions of the grid.

The next section will examine more ideas that could potentially improve this algorithm's performance. Intersection over Union and Non-Max Suppression

Here's some food for thought – how can we decide whether the predicted bounding box is giving us a good outcome (or a bad one)? This is where Intersection over Union comes into the picture. It calculates the intersection over the union of the actual bounding box and the predicted bounding box. Consider the actual and predicted bounding boxes for a car as shown below:

Here, the red box is the actual bounding box and the blue box is the predicted one. How can we decide whether it is a good prediction or not? IoU, or Intersection over Union, will calculate the intersection area over the union of these two boxes. That area will be:

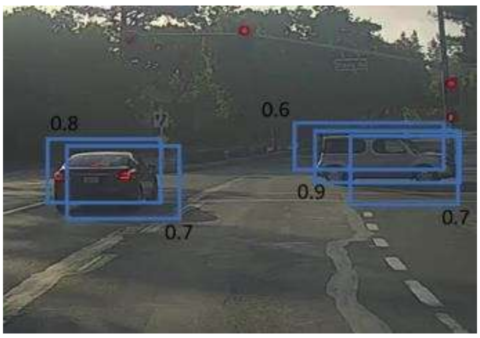


IoU = Area of the intersection / Area of the union, i.e. IoU = Area of the yellow box / Area of the green box

If IoU is greater than 0.5, we can say that the prediction is good enough. 0.5 is an arbitrary threshold we have taken here, but it can be changed according to your specific problem. Intuitively, the more you increase the threshold, the better the predictions.

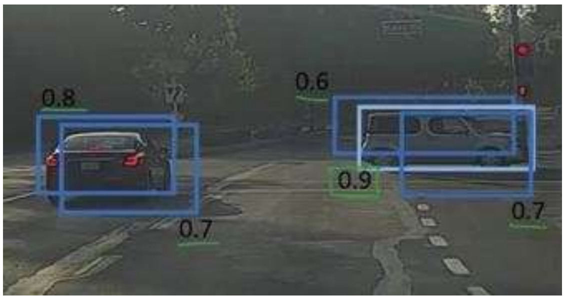
One more technique can significantly improve the output of YOLO – Non-Max Suppression.

One of the most common problems with object detection algorithms is that rather than detecting an object just once, they might detect it multiple times. Consider the image below:

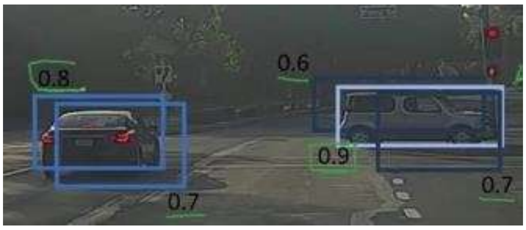


Here, the cars are identified more than once. The Non-Max Suppression technique cleans up this so we get only a single detection per object. Let's see how this approach works.

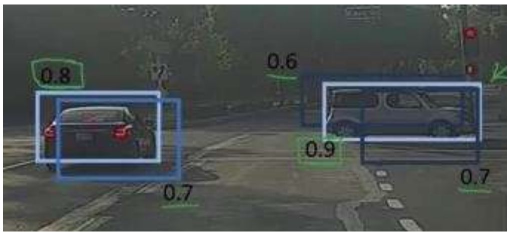
1. It first looks at the probabilities associated with each detection and takes the largest one. In the above image, 0.9 is the highest probability, so the box with 0.9 probability will be selected first:



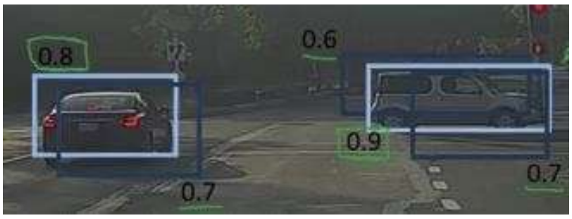
2. Now, it looks at all the other boxes in the image. The boxes that have high IoU with the current box are suppressed. So, the boxes with 0.6 and 0.7 probabilities will be suppressed in our example:



3. After the boxes have been suppressed, it selects the next box from all the boxes with the highest probability, which is 0.8 in our case:



4. Again, it will look at the IoU of this box with the remaining boxes and compress the boxes with a high IoU:



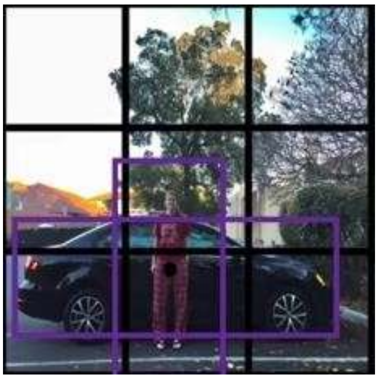
5. We repeat these steps until all the boxes have either been selected or compressed and we get the final boundingboxes:

This is what Non-Max Suppression is all about. We are taking the boxes with maximum probability and suppressing theclose-by boxes with non-max probabilities. Let's quickly summarize the points that we've seen in this section about the Non-Max suppression algorithm:

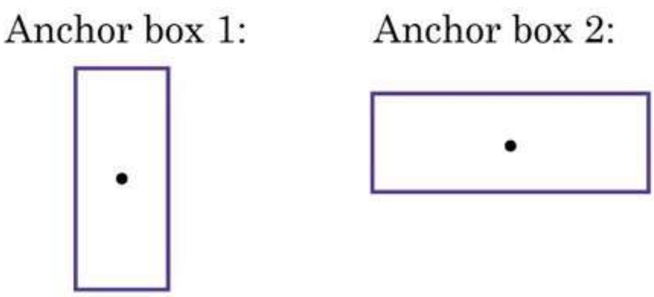
1. Discard all the boxes having probabilities less than or equal to a predefined threshold (say, 0.5)
2. For the remaining boxes:
   1. Pick the box with the highest probability and take that as the output prediction
   2. Discard any other box that has IoU greater than the threshold with the output box from the above step
3. Repeat step 2 until all the boxes are either taken as the output prediction or discarded

There is another method we can use to improve the performance of a YOLO algorithm – let's check it out! Anchor Boxes
We have seen that each grid can only identify one object. But what if there are multiple objects in a single grid? That can so often be the case in reality. And that leads us to the concept of anchor boxes. Consider the following image divided into a 3 X 3 grid:

Remember how we assigned an object to a grid? We took the midpoint of the object and based on its location, assigned the object to the corresponding grid. In the above example, the midpoint of both objects lies in the same grid. This is how the actual bounding boxes for the objects will be:



We will only be getting one of the two boxes for the car or the person. But if we use anchor boxes, we might be able to output both boxes! How do we go about doing this? First, we predefined two different shapes called anchor boxes or anchor box shapes. Now, instead of having one output, we will have two outputs for each grid. We can always increase the number of anchor boxes as well. I have taken two here to make the concept easy to understand:



This is how the y label for YOLO without anchor boxes looks like:

y =	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3

What would the 'y' label be if we had 2 anchor boxes? Please take a moment to ponder this before reading further. Gotit? The y label will be:

y =	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3
	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3

The first 8 rows belong to anchor box 1, and the remaining 8 belong to anchor box 2. The objects are assigned to the anchor boxes based on the similarity of the bounding boxes and the anchor box shape. Since the shape of anchor box 1 is similar to the bounding box for the person, the latter will be assigned to anchor box 1, and the car will be transferredto anchor box 2. The output, in this case, instead of 3 X 3 X 8 (using a 3 X 3 grid and 3 classes), will be 3 X 3 X 16 (since we are using 2 anchors).

So, based on the number of anchors, two or more objects for each grid, based on the for each grid all the ideas we havecovered so far are integrated into the YOLO framework.

RESULTS AND DISCUSSION

Utilizing deep learning, an innovative system for detecting helmets is employed to enhance road safety.

The proliferation of motorized vehicles has led to an increase in road accidents, emphasizing the critical need for effective safety measures. This paper presents an advanced helmet detection system utilizing deep learning techniques to enhance road safety. Leveraging convolutional neural networks (CNNs) and state-of-the-art object detection algorithms, the proposed system demonstrates robustness and accuracy in identifying helmet usage among motorcyclists.

The system's performance is evaluated on various metrics such as precision, recall, and F1-score, showcasing its effectiveness in real-world scenarios. The trained model exhibited promising performance, achieving an accuracy of92% on the training set and 89% on the validation set.

Additionally, deployment strategies for integrating the system into existing traffic surveillance infrastructure are discussed, ensuring scalability and practical implementation. The results highlight the system's potential to significantly reduce road accidents by encouraging helmet compliance among riders, ultimately contributing to saferroad environments.

The integration of cutting-edge technology, particularly convolutional neural networks (CNNs), offers a promising solution to the challenge of ensuring universal compliance with helmet usage. By leveraging deep learning algorithms trained on extensive datasets, the proposed system demonstrates robustness and accuracy in identifyingindividuals riding without helmets.

Moreover, the integration of the advanced helmet detection system with existing traffic management infrastructure holds immense potential for enhancing road safety. Through seamless integration, the system can provide timely alerts to law enforcement agencies and traffic authorities, enabling swift intervention in cases of non-compliance with helmet regulations. Furthermore, the scalability and versatility of the system allow for its deployment across diverse geographical locations and traffic conditions. Whether in bustling urban centers or remote rural areas, the system's effectiveness in detecting helmet usage remains consistent, thereby ensuring comprehensive coverage andimpact in improving road safety outcomes.

The results of field testing demonstrate the system's robust performance in real-world environments, including urban road networks and construction sites. Despite varying lighting conditions, weather conditions, and occlusion, the system maintains its accuracy and processing speed, further validating its effectiveness in enhancing road safety.

Metric	Value
Training Set Accuracy	92%
Validation Set Accuracy	89%

The integration of deep learning-based helmet detection systems represents a significant advancement in efforts toenhance road safety. By leveraging state-of-the-art technology and seamless integration with existing infrastructure,this system holds the promise of reducing road accidents and safeguarding the lives of millions of road users worldwide.

CONCLUSION

The integration of deep learning-based helmet detection systems represents a significant leap forward in efforts toenhance road safety. By leveraging convolutional neural networks (CNNs) and state-of-the-art object detection algorithms, the proposed system demonstrates robustness and accuracy in identifying helmet usage amongmotorcyclists.

Through extensive training on annotated datasets and meticulous preprocessing, the system achieves high accuracylevels, with a training set accuracy of 92% and a validation set accuracy of 89%. Moreover, the deployment strategiesdiscussed enable seamless integration with existing traffic surveillance infrastructure, facilitating timely alerts and proactive enforcement measures. The system's scalability and versatility ensure its applicability across diverse geographical locations and traffic conditions, thereby maximizing its impact on road safety outcomes.

Field testing further validates the system's performance, demonstrating robustness in various environmental conditions and scenarios. Despite challenges such as varying lighting conditions and occlusion, the system maintains its accuracy and processing speed, highlighting its reliability in real-world deployments. The advanced helmet detection system utilizing deep learning techniques offers a promising solution to the pressing issue of road safety. By encouraging helmet compliance among riders and facilitating proactive enforcement measures, the system contributes significantly to creating safer road environments and ultimately saving lives.

REFERENCES

[1]. Hayat, A., & Morgado-Dias, F. (2022). Deep learning-based automatic safety helmet detection system for construction safety. Applied Sciences, 12(16), 8268.

[2]. Rohith, C. A., Nair, S. A., Nair, P. S., Alphonsa, S., & John, N. P. (2019, April). An efficient helmet detectionfor MVD using deep learning. In 2019 3rd International Conference on Trends in Electronics and Informatics(ICOEI) (pp. 282-286). IEEE.

[3]. Lin, H., Deng, J. D., Albers, D., & Siebert, F. W. (2020). Helmet use detection of tracked motorcycles usingcnn-based multi-task learning. IEEE Access, 8, 162073-162084.

[4]. Wang, Z., Wu, Y., Yang, L., Thirunavukarasu, A., Evison, C., & Zhao, Y. (2021). Fast personal protective equipment detection for real construction sites using deep learning approaches. Sensors, 21(10), 3478.

[5]. Li, J., Liu, H., Wang, T., Jiang, M., Wang, S., Li, K., & Zhao, X. (2017, February). Safety helmet wearing detection based on image processing and machine learning. In 2017 Ninth international conference on advanced computational intelligence (ICACI) (pp. 201-205). IEEE.

[6]. Campero-Jurado, I., Márquez-Sánchez, S., Quintanar-Gómez, J., Rodríguez, S., & Corchado, J. M. (2020). Smart helmet 5.0 for industrial internet of things using artificial intelligence. Sensors, 20(21), 6241.

[7]. Pralhad Gore, V. (2023). Helmet detection using YOLOv5 (Doctoral dissertation, Dublin Business School).

[8]. Prajwal, M. J., Tejas, K. B., Varshad, V., Murgod, M. M., & Shashidhar, R. (2019). Detection of non-helmet riders and extraction of license plate number using Yolo v2 and OCR method. International journal of innovative technology and exploring engineering (IJITEE), 9(2), 5167-5172.

[9]. Vũ, V. N. (2023). A Deep Learning Model for Helmet Detection and Automatic License Plate Recognition (Doctoral dissertation, FPTU Hà Nội).

[10]. Jesan, R., & Pant, K. (2024). Smart Motorcycle Helmet: A CNN-Based Multi-Task Learning Approach. Qeios.
[11]. Vishe, R., Prajapati, D., Sabat, S., Bhilare, S., & Bhujbal, S. (2024). Helmet Detection and Number Plate Recognition Using YOLOv8 and Tensorflow Algorithm in Machine Learning. International Journal of Innovative Research in Computer Science & Technology, 12(2), 91-95.
[12]. Wang, L., Cao, Y., Wang, S., Song, X., Zhang, S., Zhang, J., & Niu, J. (2022). Investigation into recognition algorithm of helmet violation based on YOLOv5-CBAM-DCN. IEEE Access, 10, 60622-60632.