Real-time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8

*Armstrong Aboah & Bin Wang
Department of Radiology
Northwestern University

`armstrong.aboah, bin.wang@northwestern.edu`

Ulas Bagci
Department of Radiology
Northwestern University

`ulas.bagci@northwestern.edu`

Yaw Adu-Gyamfi
Department of Civil Engineering
University of Missouri-Columbia

`adugyamfiy@missouri.edu`

Abstract

Traffic safety is a major global concern. Helmet usage is a key factor in preventing head injuries and fatalities caused by motorcycle accidents. However, helmet usage violations continue to be a significant problem. To identify such violations, automatic helmet detection systems have been proposed and implemented using computer vision techniques. Real-time implementation of such systems is crucial for traffic surveillance and enforcement, however, most of these systems are not real-time. This study proposes a robust real-time helmet violation detection system. The proposed system utilizes a unique data processing strategy, referred to as **few-shot data sampling**, *to develop a robust model with fewer annotations, and a single-stage object detection model, YOLOv8 (You Only Look Once Version 8), for detecting helmet violations in real-time from video frames. Our proposed method won* **7th place** *in the 2023 AI City Challenge, Track 5, with an mAP score of* **0.5861** *on experimental validation data. The experimental results demonstrate the effectiveness, efficiency, and robustness of the proposed system. The code for the few-shot data sampling technique is available at* `https://github.com/aboah1994/few-shot-Video-Data-Sampling.git`.

1. Introduction

Traffic safety is a major concern worldwide, with helmet usage being a key factor in preventing head injuries and fatalities caused by motorcycle accidents. However,

*Corresponding author

Figure 1. Illustration of predictions of motorists, passengers, and motorcycles in real-time. We present a method that is able to predict helmet rule violation in real-time. The different bounding box colors represent different predictive class.

helmet usage violations continue to be a significant problem in many countries. In order to detect such violation, various automatic helmet detection systems have been proposed and implemented. These systems use computer vision and machine learning techniques such as object detection, tracking, and recognition to detect and enforce helmet usage violations [7, 17]. Even though several techniques for helmet detection have been proposed in the literature, most of them are not able to perform in real-time. Real-time helmet detection is crucial for traffic surveillance and enforcement, as it allows authorities to quickly identify and take action against individuals who are not wearing helmets (see Fig. 1). This is especially important in developing countries, where helmet usage is not frequent, and the number of motorcycle-related accidents is high.

The need for real-time helmet detection under varying weather conditions and time of day is crucial for ensuring

traffic safety. For instance, different weather conditions and lighting can greatly affect the visibility of motorcycle riders and their helmets, making it challenging for traditional systems to accurately detect helmet usage violations. Additionally, the number of helmet usage violations may also vary depending on the time of day. For another example, violations may be more likely to occur during nighttime or in inclement weather when visibility is poor.

Traditional helmet detection systems often rely on manual inspection which is time-consuming and prone to human error. Real-time helmet detection systems, on the other hand, can automatically and continuously monitor helmet usage in real-time, providing a more efficient and accurate method of enforcing helmet laws. As a result, the current study seeks to develop a real-time helmet violation detection system as shown in Fig. 1.

The overarching goal of this study is to develop a real-time helmet violation detection system that is robust to varying weather conditions and time of day. To achieve this goal, we proposed a unique data processing strategy referred to as the ***"few-shot data sampling technique"***. This technique was implemented to develop a more robust model with fewer annotations(see Algorithm 1 and 2). In our proposed system, we have addressed the challenges of varying lighting and weather conditions by using a newly developed algorithm to classify videos under different conditions, which is significant to address the challenges of sampling evenly from all video types. With the help of our newly developed few-shot data sampling techniques and data augmentation, our proposed system has been able to achieve a high level of accuracy and robustness in detecting helmet usage violations under varying weather conditions and time of day. For developing the detection model, we utilized a single-stage object detection model, YOLOv8, for detecting helmets in real-time. YOLOv8 is a state-of-the-art object detection model that has been shown to achieve high accuracy and speed in real-world applications. One of the main advantages of YOLOv8 is its ability to detect objects in real-time, making it well-suited for our task of detecting helmet usage violations in traffic videos. The major contributions of our study are summarized below;

1. Proposed and implemented a novel data processing strategy, called the ***"few-shot data sampling technique"***, for developing a robust helmet detection model with fewer annotations. The technique involves selecting a small but representative number of images from a large dataset using our developed algorithms and then applying data augmentation techniques to generate additional images for training. By using this technique, we able to develop a robust helmet detection model with fewer annotations, which is an important contribution as it reduces the time and effort required for annotation

2. Developed a real-time helmet detection system that is robust to varying weather conditions and time of day by utilizing YOLOv8, a state-of-the-art object detection model, and data augmentation techniques. YOLOv8 is designed to be fast and accurate, making it ideal for use in real time. In addition, several data augmentation strategies were utilized in this study to overcome the issues of occlusion and viewpoint problems. To further improve prediction accuracy and confidence during inference time, the study employed test time augmentation (TTA) during its inference stage.

3. We performed a comparative analysis of three single-stage object detection models from the YOLO series, namely YOLOv5, YOLOv7, and YOLOv8. By evaluating their performance in detecting helmet violations, we aimed to identify the most effective model for detecting helmet violations. This analysis is useful in improving the accuracy and effectiveness of our helmet violation detection system.

Our experimental results demonstrated the effectiveness of our proposed helmet detection system in detecting helmets in real-world scenarios. The high precision rate achieved in our tests indicates that the system is able to accurately detect helmets in a wide range of visual conditions, including varying weather conditions and different times of day. This is crucial for the practical application of the system, as it ensures that the system can be used to detect helmets in a variety of real-world environments. Secondly, our experimental results showcase the robustness of our proposed data processing strategies. By developing a model that can generalize its predictions to account for many of the visual complexities and situational ambiguities that are often encountered in real-world scenarios, we have created a system that is more robust and reliable than existing methods. Finally, the use of a single-stage object detection model and our proposed data processing strategies have resulted in a model that is both fast and efficient, which is crucial for real-time applications. As such, our experimental results demonstrate that our proposed helmet detection system is not only accurate but also well-suited for real-world use.

The remainder of the paper is structured as follows: In section 2, we conduct a literature review on helmet detection techniques. In section 3, we describe the data set and our unique data processing strategies that were employed to develop a more robust model with fewer annotations. In section 4, we discuss the helmet detection models employed in this study. In section 5, we present our experimental findings, which demonstrate the efficacy of our proposed method in detecting helmets in real-world scenarios. In section 6, we discuss the implications of our findings and make suggestions for future research in this field.

2. Related works

Helmet violation detection, also known as helmet enforcement, has become an important area of research in recent years due to the importance of helmet usage in reducing injury and fatalities in road accidents. In a study conducted by the World Health Organization (WHO), it was estimated that helmets can reduce the risk of death by almost 40% and the risk of severe injury by 70% [12].

One of the earliest studies in the field of helmet violation detection was conducted by [15], who proposed a method for detecting helmet violations using computer vision techniques. The authors used color and texture-based features to detect helmets in real-time and reported an accuracy rate of 89.5% [2, 13, 14].

In a more recent study, [16] proposed a deep learning-based approach for helmet enforcement in real-time. The authors used a Convolutional Neural Network (CNN) trained on a large dataset of helmet and non-helmet images and reported a high accuracy rate of 97.5%. The authors also reported that their method was faster than previous methods, with a processing time of less than 50 milliseconds per frame [3, 4, 9, 13].

Another study by [6] used a deep neural network, specifically YOLOv3, for helmet enforcement in real-time. The authors used a dataset of helmet and non-helmet images to train the network and reported an accuracy rate of 96.2%. The authors also reported that their method was faster than previous methods, with a processing time of less than 30 milliseconds per frame.

In a recent study, [5] proposed a real-time helmet enforcement system using a combination of computer vision and deep learning techniques. The authors used a combination of color and texture-based features and a deep neural network to detect helmets in real-time and reported an accuracy rate of 95.6%. The authors also reported that their method was faster than previous methods, with a processing time of less than 100 milliseconds per frame.

In conclusion, helmet violation detection has been an active area of research in recent years, and various studies have proposed different approaches using computer vision and deep learning techniques. The studies mentioned above demonstrate that deep learning-based approaches have achieved high accuracy rates and processing times, making them suitable for real-time helmet enforcement systems. However, more research is needed to improve the accuracy and processing times of these systems and to make them more robust to different conditions and environments.

3. Data

3.1. Data Overview

The data used for the proposed violation detection model was sourced from the 2023 NVIDIA AI CITY CHAL-

Figure 2. Images from different time of day and weather. a) Daylight, b) Nighttime, and c)Foggy weather.

LENGE, specifically Track 5 which focuses on detecting helmet rule violations among motorcyclists and passengers. The dataset includes 100 videos for training and 100 videos for testing, each with an average length of 20 seconds, a frame rate of 10 fps, and a resolution of 1920x1080. The videos present a range of difficulties, as they feature various camera angles, lighting, and weather conditions as presented in Fig. 2. The primary objective of the model is to identify motorcycles and their riders and determine whether or not they are wearing helmets.

3.2. Data Processing

One of the major steps taken in this study to build a robust helmet violation detection model was to perform certain key data pre-processing steps.

The pre-processing steps used in this study are summarized in Fig. 3. The aim of this step was to improve the performance of the object detection model by automatically selecting frames that profoundly represent the training data and also provide enough variability in the training dataset through data augmentation to help the developed model generalize well with real-world test cases. The study carried out two main data pre-processing steps: (1) a few-shot data sampling framework to select the best representative set of data for training and (2) data augmentation to increase the variety of the training data. An illustration of the data pre-processing method, which incorporates six data augmentation techniques, is shown in Fig. 3.

Few-shot Data Sampling Framework: The initial ground truth annotations were provided as part of the challenge, however, there were some missing annotations as illustrated in Fig. 4. These missing annotations have a significant impact when training a model. To address this without manually reviewing all 20,000 frames and correcting annotations, a few-shot data sampling framework was developed. This framework was designed to help select the most representative frames of the entire dataset and minimize the
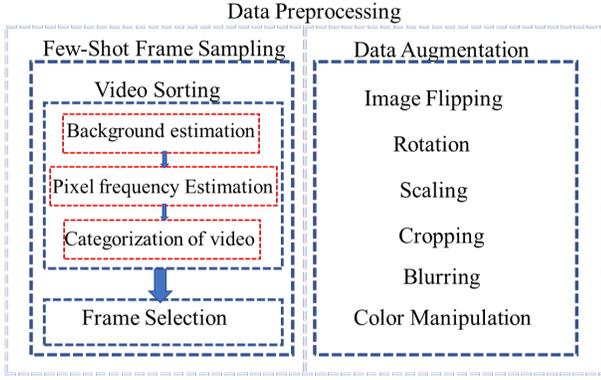
Figure 3. Data Preprocessing: Frame Sampling and Data Augmentation

need for re-annotation of all 20,000 frames. The framework consists of three primary steps as follows:

First, we determine the background in each video. To estimate the background of a video, we first randomly select frames within a 10-second period. Next, we compute the median of 60 percent of all frames in the sample. By using random sampling and determining the median of a subset of images, we are able to negate the impact of short-term video resolution changes such as zooms, and pixelation.

Second, our developed algorithm (see Algorithm 1) is used to categorize the videos according to the time of day and weather conditions such as day, night, and fog. It is important to ensure a balanced representation of all video types since there are more daylight videos in the dataset compared to other scenes causing imbalance problems. The proposed algorithm takes the estimated video background and calculates the frequency of each pixel. If the maximum frequency corresponds to a pixel value less than 150, the algorithm classifies the image as night; otherwise, the algorithm classifies the image as day or foggy. To distinguish between daytime and foggy videos, the skewness of the image frequencies is computed. The algorithm classifies the video as foggy if the absolute skewness is close to zero. The frequency distribution of the day, night, and fog images are shown in Fig. 5.

Lastly, we developed a frame sampling algorithm (see Algorithm 2) that aims to select more frames from video types that were underrepresented, as identified. The algorithm receives as input the anticipated number of frames. With the information regarding the total number of videos in each category and the fps of each, a sample rate is calculated for each video category.

Data Augmentation: Data augmentation is a common technique used in computer vision to increase the sample size and diversity of the training dataset. This is particularly critical for object detection tasks such as our current work, which can be complicated by problems of occlusion

---

Algorithm 1 Categorize Videos

Input: Video background image, I_{bg}
Output: Categorized video: "day", "night", or "foggy"

1: **for** $p \in pixels$ **do**
2:     $f_p \leftarrow calculateFrequency(p)$
3: **end for**
4: $maxFrequency \leftarrow max(f_p)$
5: **if** $maxFrequency < 150$ **then**
6:     $class \leftarrow$ "night"
7: **else**
8:     $skewness \leftarrow calculateSkewness(f_p)$
9:     **if** $|skewness| \approx 0$ **then**
10:         $class \leftarrow$ "foggy"
11:     **else**
12:         $class \leftarrow$ "day"
13:     **end if**
14: **end if**
15: **return** $class$

Algorithm 2 Frame Sampling Algorithm

Input: n_{frames}, $n_{videos}^{category}$, $fps^{category}$ **Output:** Selected frames from each video category

1: **for** $category \in 1, 2, 3, ..., n$ **do**
2:     Calculate sample rate for each category: $sample_rate^{category} = \frac{n_{frames}}{\sum_{i=1}^{n} n_{videos}^i \times fps^i}$
3:     Select frames from each video in the category at the calculated sample rate
4: **end for**
5: Return the selected frames

---

and changes in viewpoint. To solve the challenges of occlusion and viewpoint variation, several data augmentation techniques were applied in this study.

1. **Image flipping:** Flipping the image horizontally was done to aid the model to learn to detect helmets from both sides of the motorcycle.

2. **Rotation:** Rotation was applied to augment the data by changing the viewpoint angle of the helmet.

3. **Scaling:** Scaling was used to change the size of the helmet in the image, which can help the model learn to detect helmets of different sizes.

4. **Cropping:** Cropping of images was done to simulate the effect of occlusion, so that the model can learn to detect helmets even when they are partially obscured.

5. **Blurring:** Blurring of images were carried out to help the model learn to detect helmets under poor lighting conditions.

Figure 4. Examples of missed annotations from the ground truth annotations.
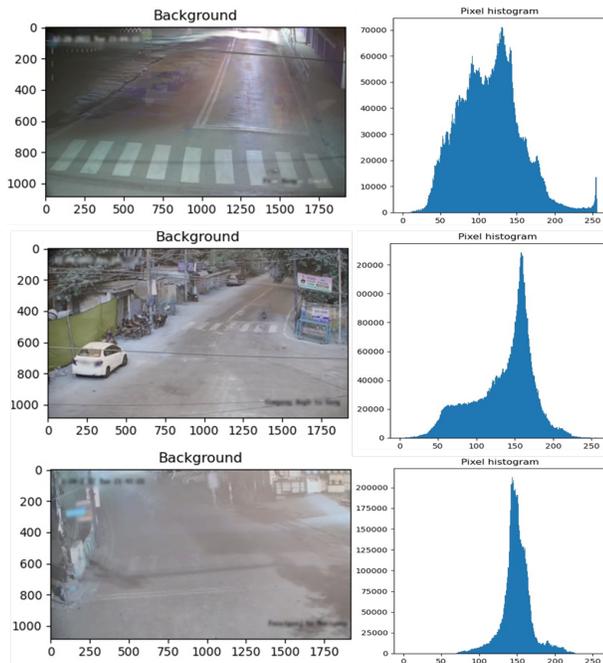

Figure 5. Sorting videos by time of the day and weather condition using frequency thresholding.

6. **Color manipulation:** We adjusted the brightness, contrast, and saturation of the image to help the model learn to detect helmets in different lighting conditions.

3.3. Training and Validation Dataset for Developing the Detection Model

All models were trained on an NVIDIA GeForce RTX 3090 GPU using 4,500 training examples. The dataset was divided in a ratio of 0.7:0.3 for training and validation re-

spectively. The test dataset was provided separately by the organizers of the competition. To prevent the model from overfitting frames with high similarity, we employed the Semantic Clustering by Adopting Nearest Neighbors (SCAN) algorithm [10] to eliminate frames with high similarity as shown in Fig 6. In order to eliminate any potential bias in evaluating our validation dataset, we took great care in selecting our training images by ensuring that sequential image sequences did not appear in both the training and validation datasets. Additionally, we added the estimated backgrounds of each video, along with their augmentations, into the training data as negative samples to minimize false positive detections.

Figure 6. Clustering of frames with high similarity. Each row represent frames of high similarity.

4. Helmet Detection Models

The study experimented three object detection models namely YOLOv5, YOLOv7, and YOLOv8. All three models are designed to detect objects in a single stage.

YOLOv5. The YOLOv5 [1] architecture is made up of three main parts which include the Backbone, Neck, and Head. The Backbone has a convolutional neural network that gathers and builds image features at different levels of detail. The Neck combines the features extracted from the Backbone and generates feature maps for the prediction task. The Head uses the feature maps from the Neck to predict object classes and their corresponding bounding boxes. The **CSPDarknet53** is the backbone network of YOLOv5 which has 29 convolutional layers with a 3x3 kernel size. This results in a receptive field of size 725x725 and a total of 27.6 million parameters. A SPP block is added to the CSPDarknet53 to increase the receptive field without affecting the speed of the operation. Additionally, PANet is used to fuse low-level features with high-level features to enhance the richness of the features.

YOLOv7. The YOLOv7 [11] model had numerous changes to its architecture that distinguished it from YOLOv5. These included the implementation of compound scaling, the extended efficient layer aggregation network (EELAN), and the use of techniques like planned and reparameterized convolution, coarseness for auxiliary loss, and fineness for lead loss. The EELAN was a crucial element of YOLOv7, and it allowed the model to learn more effectively while still maintaining the original gradient route. This was achieved through a technique called "expand, shuffle, and merge cardinality." The compound scaling approach was introduced to YOLOv7 to preserve the model's properties at the initial level and maintain its optimal design. In addition, the paper presented the use of model re-parameterization and dynamic label assignment as network optimization strategies, addressing existing problems and improving the model's overall performance.

YOLOv8. YOLOv8 is the latest iteration in the YOLO family of detection models, which are known for their capabilities for joint detection and segmentation. Similar to YOLOv5 [1, 8], the architecture consists of a backbone, head, and neck. It boasts a new architecture, improved convolutional layers (backbone), and a more advanced detection head, making it a top choice for real-time object detection. YOLOv8 also offers support for the latest computer vision algorithms such as instance segmentation, enabling multiple object detection in an image or videos. The model uses the **Darknet-53** backbone network, which is faster and more precise than the previous YOLOv7 [11] network. YOLOv8 predicts bounding boxes through an anchor-free detection head. The model is more effective than previous versions due to its larger feature map and improved convolutional network, which enhances its precision and speed. YOLOv8 also incorporates feature pyramid networks for recognizing objects of varying sizes. The model also comes with a user-friendly API, making it easy to implement in various applications.

4.1. Model Training

All three models were trained with optimal hyperparameters generated by genetic algorithm. Table 1 provides a summary of the training hyperparameters. Additionally, all models were trained for 400 epochs with a batch size of 16 and an image size of 832x832.

Table 1. Training Hyperparaters

Hyperparameter	YOLOv5	YOLOv7	YOLOv8
Initial learning rate	0.0101	0.0101	0.0106
Optimizer	Adam	Adam	Adam
Momentum	0.937	0.965	0.971
Weight decay	0.00042	0.00051	0.00048
Warmup epochs	3.862	5.541	2.689
IoU	0.881	0.867	0.912

4.2. Test Time Augmentation (TTA)

Test Time Augmentation (TTA) is a technique used in deep learning for improving the accuracy of a model's predictions on test data. TTA involves applying data augmentation techniques, such as rotation, flipping, or cropping, to the test data and then making predictions on each augmented version of the test data. The final prediction is then made by averaging the predictions made on the augmented versions of the test data. TTA can be computationally expensive, as it involves making multiple predictions on augmented versions of the test data. However, it can be implemented efficiently by using parallel processing or by batching the augmented data.

5. Results and Discussion

The 2023 NVIDIA AI City Challenge Task 5 includes 100 unannotated videos for testing. Each testing video has a 20-second duration and a resolution 1920 × 1080 pixel. Trained models must be able to detect motorcycles and their riders, whether or not they are wearing helmets, from the test videos. Our trained model's detections on the test videos are shown in Fig 7.

A submission to the challenge is a text file with the format: Video ID, frame ID, bb_left, bb_top, bb_width, bb_height, Class ID and Confidence. The Video ID is a numeric identifier for the video, starting from 1 and indicating

Figure 7. Model detection from different time of day and weather condition.

its position in the alphabetically sorted list of all Task 5, Test Set videos. The frame ID refers to the number of the frame with the detected objects. The bb_left, bb_top, bb_width, and bb_height are the bounding box coordinates for the detected object and the Class ID is the numeric identifier for the object, starting from 1. The confidence is a probability value between 0 and 1.

The evaluation of our model is based on the mean Average Precision (mAP) across all frames in the test videos, and the leaderboard ranking of submissions is determined by the mAP score, as calculated by Equation 1.

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \qquad (1)$$

where N is the number of queries.

5.1. Comparative Analysis

We conducted a comparative analysis of various single-stage object detection models belonging to the YOLO series with and without test time augmentation (TTA) on both our validation dataset and test dataset(provided by organizers of the competition).

5.1.1 Validataion Dataset

Our analysis presented in Table 2 suggests that recent iterations of YOLO, namely yolov7 and yolov8, display su-

Table 2. Experimental results on validation dataset

Model	mAP.05	mAP.05-.95	Precision	Recall
yolov5	0.823	0.465	0.892	0.811
yolov7	0.846	0.526	0.912	0.854
yolov8	0.858	0.601	0.923	0.898
yolov5+TTA	0.911	0.613	0.931	0.907
yolov8+TTA	**0.923**	**0.647**	**0.953**	**0.918**

perior performance in terms of both mAP.05 and mAP.05-.95 when compared to the older version, yolov5. Moreover, our findings propound that utilizing test time augmentation (TTA) further enhanced the performance of the models. Notably, yolov8+TTA has demonstrated the highest mAP.05-.95 score of 0.647, indicating its ability to detect objects with high accuracy across a range of IoU thresholds. Thus, our results underscore the significance of employing the latest object detection models and incorporating test time augmentation to improve their efficacy.

5.1.2 Test Dataset

On the experimental test dataset, the yolov8+TTA outperformed all other models. It achieved an overall mAP score of **0.5861** with a comparative inference speed of 95 fps as shown in Table 3. Its inference speed makes it suitable for

Table 3. Experimental results on test dataset

Model	mAP	fps
yolov5	0.3988	160
yolov7	0.4061	163
yolov8	0.5136	162
yolov5+TTA	0.5346	100
yolov8+TTA	**0.5861**	**95**

real-time predictions. The results from on experimental test data was **ranked 7th** on the public leader board illustrated in Table 4.

Table 4. Top 10 Leader Board Ranking

Rank	Team ID	Team Name	Score
1	58	CTCAI	0.8340
2	33	SKKU Automation Lab	0.7754
3	37	SmartVision	0.6997
4	18	UT_He	0.6422
5	16	UT_NYCU_SUNYAlbany	0.6389
6	45	UT1	0.6112
7	192	Legends (Ours)	0.5861
8	55	NYCUROAD BEAST	0.5569
9	145	WITAI53	0.5474
10	11	AIMIZ	0.5377

6. Conclusion

We developed a real-time multi-class helmet violation detection system that is robust to varying weather conditions and times of the day. To achieve this goal, we proposed a unique data processing strategy referred to as the ***"few-shot data sampling technique"*** and utilized YOLOv8, a state-of-the-art single-stage object detection model. Our experimental results demonstrated the effectiveness and robustness of the proposed system in detecting helmets in real-world scenarios with high precision and efficiency. The results also showed that the proposed system is well-suited for practical use in real-time applications.

References

[1] Armstrong Aboah. A vision-based system for traffic anomaly detection using deep learning and decision trees. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4207–4212, 2021. 6

[2] M Ashvini, G Revathi, B Yogameena, and S Saravanaperumaal. View invariant motorcycle detection for helmet wear analysis in intelligent traffic surveillance. In *Proceedings of International Conference on Computer Vision and Image Processing: CVIP 2016, Volume 2*, pages 175–185. Springer, 2017. 3

[3] Rao Cheng, Xiaowei He, Zhonglong Zheng, and Zhentao Wang. Multi-scale safety helmet detection based on sas-yolov3-tiny. *Applied Sciences*, 11(8):3652, 2021. 3

[4] Lixia Deng, Hongquan Li, Haiying Liu, and Jason Gu. A lightweight yolov3 algorithm used for safety helmet detection. *Scientific reports*, 12(1):10981, 2022. 3

[5] Rupjyoti Gogoi and Arindam Bora. Real-time helmet enforcement system using computer vision and deep learning. *Journal of Ambient Intelligence and Humanized Computing*, 12(1):65, 2021. 3

[6] J Huang, X Liu, and Ming-Hsuan Yang. Real-time helmet enforcement using deep learning. In *Proceedings of the 2018 ACM on Multimedia Conference*, pages 1793–1801. ACM, 2018. 3

[7] Wei Jia, Shiquan Xu, Zhen Liang, Yang Zhao, Hai Min, Shujie Li, and Ye Yu. Real-time automatic helmet detection of motorcyclists in urban traffic using improved yolov5 detector. *IET Image Processing*, 15(14):3623–3637, 2021. 1

[8] Maged Shoman, Armstrong Aboah, Alex Morehead, Ye Duan, Abdulateef Daud, and Yaw Adu-Gyamfi. A region-based deep learning approach to automated retail checkout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3210–3215, 2022. 6

[9] Hongru Song. Multi-scale safety helmet detection based on rsse-yolov3. *Sensors*, 22(16):6061, 2022. 3

[10] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X*, pages 268–285. Springer, 2020. 5

[11] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022. 6

[12] World Health Organization (WHO). Helmet promotion in low- and middle-income countries, 2013. 3

[13] Fan Wu, Guoqing Jin, Mingyu Gao, HE Zhiwei, and Yuxiang Yang. Helmet detection based on improved yolo v3 deep model. In *2019 IEEE 16th International conference on networking, sensing and control (ICNSC)*, pages 363–368. IEEE, 2019. 3

[14] Hao Wu and Jinsong Zhao. An intelligent vision-based approach for helmet identification for work safety. *Computers in Industry*, 100:267–277, 2018. 3

[15] Z Zeng, Z Zhang, and Y Liu. Helmet enforcement using computer vision techniques. In *Proceedings of the 2006 International Conference on Image Processing*, pages 453–456. IEEE, 2006. 3

[16] Y Zhang, T Liu, and L Shao. Deep learning based real-time helmet enforcement system. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1057–1065. ACM, 2017. 3

[17] Fangbo Zhou, Huailin Zhao, and Zhen Nie. Safety helmet detection based on yolov5. In *2021 IEEE Interna-*

tional conference on power electronics, computer applica-tions (ICPECA), pages 6–11. IEEE, 2021. 1