Article

Enforcing Traffic Safety: A Deep Learning Approach for Detecting Motorcyclists' Helmet Violations Using YOLOv8 and Deep Convolutional Generative Adversarial Network-Generated Images

Maged Shoman [ID], Tarek Ghoul [ID], Gabriel Lanzaro, Tala Alsharif, Suliman Gargoum [ID] and Tarek Sayed *[ID]

Department of Civil Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada; magedsho@mail.ubc.ca (M.S.); tarek.ghoul@ubc.ca (T.G.); lanzaro@mail.ubc.ca (G.L.); tala172@mail.ubc.ca (T.A.); suliman.gargoum@ubc.ca (S.G.)
* Correspondence: tsayed@civil.ubc.ca

Abstract: In this study, we introduce an innovative methodology for the detection of helmet usage violations among motorcyclists, integrating the YOLOv8 object detection algorithm with deep convolutional generative adversarial networks (DCGANs). The objective of this research is to enhance the precision of existing helmet violation detection techniques, which are typically reliant on manual inspection and susceptible to inaccuracies. The proposed methodology involves model training on an extensive dataset comprising both authentic and synthetic images, and demonstrates high accuracy in identifying helmet violations, including scenarios with multiple riders. Data augmentation, in conjunction with synthetic images produced by DCGANs, is utilized to expand the training data volume, particularly focusing on imbalanced classes, thereby facilitating superior model generalization to real-world circumstances. The stand-alone YOLOv8 model exhibited an F1 score of 0.91 for all classes at a confidence level of 0.617, whereas the DCGANs + YOLOv8 model demonstrated an F1 score of 0.96 for all classes at a reduced confidence level of 0.334. These findings highlight the potential of DCGANs in enhancing the accuracy of helmet rule violation detection, thus fostering safer motorcycling practices.

Keywords: deep learning; DCGANs; YOLOv8; helmet detection; imbalanced classes

1. Introduction

Motorcycles have long been favored for their flexibility and economic advantages as a mode of transportation. For example, motorcycles tend to occupy smaller spaces, which helps riders avoid traffic congestion and offers advantages in limited parking spaces. This makes motorcycles more appealing for mixed and less organized traffic environments in densely populated urban areas, where motorcyclists can take evasive actions more freely [1]. In addition, motorcycles have a lower fuel consumption when compared to passenger vehicles, which leads to their widespread use in many developing countries, particularly for commercial applications and delivery services [2]. However, motorcycle ridership is generally associated with a higher crash fatality risk that may be attributed to several factors [3,4]. First, motorcyclists tend to be less risk aware, which leads to committing more traffic violations and engaging in risky maneuvers (e.g., lane filtering, sudden overtaking) more frequently [5,6]. Second, motorcycle ridership might be associated with inexperienced driving and lack of training [7]. Finally, motorcycle riders are more susceptible to injuries due to the lack of protection [2,8]. To improve motorcycle safety, it is essential to address the factors contributing to the high number of fatalities and enforce stricter traffic laws to deter motorcyclists from engaging in hazardous ridership practices. Laws mandating helmet use have been implemented in various countries as helmets have been an effective measure to

reduce injury severity in motorcycle crashes [9]. Helmet regulations have proven to be a crucial step in improving motorcycle safety and reducing the number of fatalities resulting from crashes [10]. Consequently, there is a growing interest in the use of automated helmet detection algorithms that may be used in less organized road environments. Challenges associated with these algorithms are the diversity in the dataset when there are multiple riders and low-quality video data.

Most of these detection models have been developed using advanced machine learning algorithms that require large datasets. However, in locations where noncompliance is considerable, such as in busy environments with a lack of enforcement, the number of motorcyclists with helmets may be limited. This leads to an unbalanced dataset where the number of motorcyclists with helmets is much greater than the number of non-compliant motorcyclists. Numerous frameworks have been proposed to deal with these datasets. For instance, generative adversarial networks (GANs) can enhance the quality of training datasets by addressing class imbalance. This technique considers a discriminator and a generator, where the generator is responsible for creating artificial observations and the discriminator should verify if the observations belong to the original distribution or to the distribution sampled by the generator. These neural networks are trained until the generator samples observations so well that the discriminator is not able to correctly classify them as either fake or real. Therefore, GANs can be used to generate more data and increase the number of observations in an unbalanced dataset. Results from our experiments show that the framework adopted can be used to detect motorcyclist helmet usage with high accuracy, which demonstrates the potential of implementing this system for automated enforcement.

The objective of this study is to develop a framework for helmet detection of motorcycle riders that can help overcome issues related to class imbalance experienced in previous research. This framework can then be used to enforce helmet-use laws and improve road safety overall. The method proposed in this study includes several steps, namely, (1) data cleaning, (2) augmenting the cleaned data, (3) generating synthetic images using deep convolutional generative adversarial networks (DCGANs) to handle unbalanced classes, and (4) applying YOLOv8 to detect helmet violations. The main contributions of this paper are as follows: (1) Developing a real-time helmet violation detection system that utilizes YOLOv8, data augmentation methods, and DCGANs for image generation that is able to perform accurate detections despite varying weather and light conditions. Data augmentation and generation techniques were used in this work to address the occlusion and perspective concerns, including test time augmentation (TTA) throughout its inference step to further increase prediction accuracy and confidence. (2) Analyzing the performance of the developed system using three object detection models from the YOLO series—YOLOv5, YOLOv7, YOLOv8 (with and without DCGANs)—to determine the most efficient model for identifying helmet violations. This study helps in increasing the precision for detecting helmet violations.

This paper is organized as follows: Section 2 contains a summary of the relevant literature. The data used in this paper are presented in Section 3. Section 4 shows the methodology used, and Section 5 presents the results. Finally, Section 6 concludes the study, presents its implications, and highlights opportunities for future work.

2. Materials and Methods

2.1. Helmet Use and Motorcyclist Safety

Motorcyclists are generally over-represented in crash fatalities worldwide [11,12]. Among other factors, this is due to their vulnerability and to their proneness to engage in risky behavior [2,8,13]. For example, motorcyclists are generally faced with visibility issues that make it more difficult to observe them [14]. The usage of cell phones has also proven to be related to reckless behavior [15], which leads to increasing crash occurrence. Also, the motorcycling experience might produce a thrill for risk seeking riders [16], and this behavior leads to excessive speeding [17]. To account for these factors that greatly increase the severity of motorcyclist crashes, countermeasures must be put in place. Numerous studies

have shown the significant role of helmet use in reducing motorcyclist fatal injuries [18,19]. For example, Ref. [3] conducted a study in the US analyzing approximately 4000 motorcycle crashes, of which 77% resulted in fatalities. In addition, out of those fatal incidents, 37% of the victims did not wear helmets. The study further revealed that nearly $2.2 billion in losses occurred each year due to riders not wearing helmets. Other studies in different environments also underpinned the importance of helmets to prevent serious injuries in motorcycle crashes. Using data from Taiwan, Ref. [20] found that using a helmet reduces the death probability by 40%. Additionally, Ref. [21] used ordered logit models to show that the motorcycle crash severity levels are significantly associated with helmet usage. To address the risks associated with motorcycle crashes, various countries worldwide have enacted legislation requirements to mandate helmet use [4].

2.2. Detection Algorithms

Despite the importance of enforcing helmet usage in reducing the severity of motorcycle collisions, this process may be perceived as costly. This is especially true for jurisdictions with limited police resources. To overcome these challenges, automated helmet detection frameworks have been considered. Object detection involves identifying and locating objects (e.g., motorcyclists without helmets) within videos or images using image processing techniques or deep neural networks. Image processing for object detection consists of several steps, such as pre-processing, feature extraction, thresholding, edge detection, and contour analysis. Various methods have been proposed, including the Haar cascade classifier [22], histogram of oriented gradients (HOG) [23], and scale-invariant feature transform (SIFT) [24] to detect objects in images. Furthermore, deep learning methods, which are able to learn complex patterns, have emerged as the leading approach due to their adaptability and ability to handle real-world scenarios [25]. These techniques employ convolutional neural networks (CNNs) for feature extraction and include models such as Region-based CNN (R-CNN) [26], Fast and Faster R-CNN [27], Single Shot Multi-Box Detector (SSD), and You Only Look Once (YOLO) [28–30]. Among these many techniques for object detection, YOLO has been a popular choice for real-time object detection due to its speed and efficiency. This algorithm was utilized in various studies for detecting objects, such as helmets [31,32], license plates [33], and road users in conflict interactions [34]. Many improvements have been recently made, with numerous model architecture modifications increasing accuracy and reducing processing time [35–38]. These models, which continue to be improved over time, are very useful for image detection due to their easy implementation, pre-trained existing weights, and open-source availability. However, the accuracy of YOLO models is associated with the quality of the training dataset, which may be limited. Additional deep learning techniques can be implemented to overcome this drawback.

Several different studies have proposed helmet detection algorithms. For example, Ref. [39] used a YOLOv5 detector in two steps to detect helmets in China, which first detects a motorcyclist and then helmet usage. Also, [40] proposed a deep learning framework to verify motorcyclist helmet usage in Myanmar. The algorithm provided an accuracy that was approximately 5% lower than a human observer. Furthermore, Ref. [41] employed feature extraction techniques and a multi-layer perceptron classifier to verify if motorcyclists were using helmets. Ref. [42] used a similar approach for motorcyclists in Thailand, and the method found a low rate of false positives. In addition, Ref. [43] used support vector machines to identify helmet wearing in busy environments. Most of the existing research focused on helmet detection algorithms rely on standard detection frameworks (e.g., YOLO) or classifiers (e.g., support vector machines) using real data only [44–46]. However, these methods become less effective when employed in very dense locations that have multiple violations occurring in one scene with poor visibility.

Generative adversarial neural networks [47], have gained popularity in recent years due to their ability to generate fake images that are convincing replicas of real images. Other approaches include diffusion networks which add random noise to the models in order to obtain the process to construct samples, and variational autoencoders, which apply

regularization to ensure that adequate data can be generated. To overcome the limitations associated with poor visibility and class imbalance, this paper proposes a framework that considers synthetic data. Therefore, generative adversarial networks have the potential of improving the model accuracy by enhancing the training process overall. This research fills this gap by combining the YOLOv8 model with DCGANs to improve helmet detection and identify motorcycle riders that do not comply with traffic regulations.

3. Dataset

The dataset used in this study, collected as part of the 2023 AICity (https://www.aicitychallenge.org/) (accessed on 20 March 2023), comprises of 100 videos from India, each with a resolution of 1920 × 1080 at a length of 20 seconds with 10 frames per second. Bounding box labels were included in the dataset for each of the classes. The dataset poses a number of difficulties due to the numerous visual complications brought on by the weather, glare, and time of day, as observed in Figure 1. Additionally, the photos' objects provide additional challenges including pixelation and occlusion, which are very common issues experienced by practitioners when analyzing images from CCTV cameras. Eight classes of interest are included in the dataset and their respective frequencies are listed in Table 1. Another major challenge associated with this dataset is that the majority of scenes suffer from moderate to extreme mislabeling. This mislabeling was predominately omission, where a large number of objects in a frame were ignored. In many cases, timestamps were labelled as motorcycles, among other erroneous labels, which would have severely affected any model's quality. This establishes the first objective of developing a methodology that considers the potential for incorrect ground truth and establishes a remedial measure. In addition to the challenges associated with the lack of accurate labels, the raw dataset consisted of images that were low resolution, coupled with additional challenges such as fog and low lighting. Moreover, the configuration of the cameras resulted in a need to track motorcycles at a distance where the image size is extremely small. To address this issue, substantial augmentation and variation in environmental conditions should be observed in the training dataset. Lastly, as is evident from Table 1, the dataset suffers from unbalanced classes which requires novel minority oversampling techniques to be used to allow for these under-represented classes to be identified accurately.
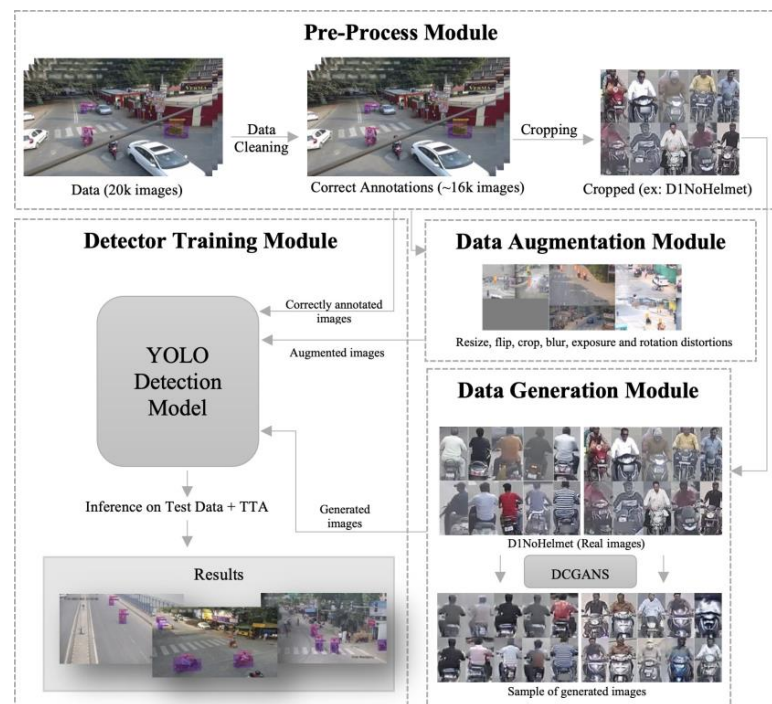

Figure 1. Sample of the collected data with visual complexities such as dark, blurry, and occlusion conditions.

Table 1. Training data instances per class.

Class	Description	Instances
Motorcycle	A motorcycle being driven	31,135
D1Helmet	A motorcycle driver wearing a helmet	23,260
D1NoHelmet	A motorcycle driver not wearing a helmet	6856
P1Helmet	The first passenger wearing a helmet	94
P1NoHelmet	The first passenger not wearing a helmet	4280
P2Helmet	The second passenger wearing a helmet	0
P2NoHelmet	The second passenger not wearing a helmet	40

4. Methodology

In order to develop a solution that correctly detects all seven classes and overcomes the issues listed in the previous section, we propose a system that contains a pre-processing module, a data augmentation module, a data generation module, and a detector training module as illustrated in Figure 2. Firstly, the frames (20,000 images) are fed into the pre-processing module, classifying correct and missed detections. Secondly, the correct detections (16,000 images) are cropped and grouped into their respective classes. The correct images are used in the data generation module to generate augmented images and the cropped images are then utilized as input into the data generation module to generate new images of the same class. Inside the detector training module, the correctly annotated, augmented, and generated images are used to train the detector. Finally, inference on the test data is performed using the trained detector and test time augmentation (TTA).

Figure 2. Overview of the proposed system consisting of the pre-processing, data augmentation, data generation, and detector training modules.

4.1. Pre-Processing Module

Given the amount of mislabeling in the ground truth dataset, each frame was manually examined to ensure that all frames were properly labelled and that there were no omissions or mislabeling. This resulted in less data being available for training (16k images), but greatly improved the performance of the model. The main challenges associated with the available data are as follows:

- Addressing missing and mislabeled ground truth data.

- Ensuring that there is a sufficient variety in environmental conditions to allow for adequate detections.
- Addressing issues surrounding class imbalance.

As such, the missed/incorrect images (∼4k images) were set aside for the trained detector later and the correct detections were cropped with their respective class and used for the data generation module. We obtained the bounding box coordinates for every object from the ground truth text file, and then extracted the corresponding images by cropping them. To ensure consistency, we normalized the crops by resizing them to match an image size of 64×64.

4.2. Data Augmentation Module

Various data augmentation strategies were used to develop a more adaptable model with increased detection accuracy. A common issue associated with the use of computer vision in traffic safety applications is the need for a long recording period due to sample-size concerns and data quality. Data augmentation can be used to address this issue, with methods such as: blur, mosaic, flipping, rotation, and flipping. One method of performing augmentation is to rotate the original picture at various angles, while flipping produces a mirror image that may be either horizontally or vertically oriented. The blur method uses a filter to lessen the sharpness of the picture. A random section is then chosen from the mosaic picture and utilized as the final enhanced image. In contrast, the mosaic process resizes four separate photos and merges them. The advantage of this method is that it improves the visual complexity of the photos, giving the model a more demanding and realistic environment to recognize. These methods allow the model to process a wider variety of pictures, which improves the accuracy of identifying the classes of interest in the dataset. The videos used contain a wide range of variations, such as different camera angles, lighting conditions, and rider styles. By applying data augmentation techniques such as random cropping, horizontal flipping, and color jittering, we created a larger and more diverse dataset as shown in Figure 3, which can help the model to learn more robust and discriminative features.

Figure 3. Sample of the augmented frames using resize, hue, flip, crop, saturation, blur, exposure, and rotate distortions.

4.3. Data Generation Module

After relabeling the training dataset and validating the results, it became evident that certain classes were severely under-represented and that there was a need to use a generative network to accommodate the low instance classes. GANs have a unique approach compared to other popular neural network architectures, as they aim to solve two distinct problems simultaneously. These problems are discrimination, which involves effectively distinguishing between real and fake images, and generating "realistic" fake data, which involves creating samples that are perceived as real. Although these objectives are essentially opposites, GANS combine them into a single model. Alternatively, if we were to separate these tasks into different models, we would have a generator (*G*) and discriminator (*D*) model. The generator model takes a random noise vector of N dimensions as input and uses a learned target distribution to transform it. Its output also are N-dimensional. On the other hand, the discriminator model models a probability distribution function, similar to a classifier, and outputs a probability between 0 and 1 that the input image is real or fake. In this way, the two main objectives of the generation task can be defined:

1. The objective of training *G* is to increase *D*'s classification error to the maximum extent possible. This will ensure that the generated images appear authentic and realistic.
2. The objective of training *D* is to reduce the final classification error as much as possible. This will enable *D* to correctly differentiate between real and fake data.

To accomplish this, during the backpropagation process, the weights of *G* are adjusted using gradient ascent in order to maximize the error, whereas *D* employs gradient descent to minimize it. It is important to note that during training, the two networks do not directly use the actual distribution of images. Instead, they use each other's outputs to evaluate their performance (1). We use the absolute error to estimate the error of *D*, and then use the same function for *G*, but with the aim of maximizing it (2).

$$E(G, D) = \frac{1}{2}(E_{x \subset p_t}[1 - D(x)] + E_{x \subset p_g}[D(x)]) \tag{1}$$

$$E = max_G(min_D E(G, D)) \tag{2}$$

In this case, p_t represents the true distribution of images, while p_g is the distribution created from *G*. Deep convolutional generative adversarial networks (DCGANs) incorporate key principles of convolutional neural networks (CNNs) and have become a popular architecture due to their quick convergence and ease of adaptation to more complex variations (such as incorporating labels as conditions or using residual blocks). They address several significant challenges, including:

- *D* is structured to perform a supervised image classification task (for example, identifying if an image contains a driver helmet or not).
- The filters learned by the GAN can be utilized to generate specific objects in the resulting image.
- *G* has vectorized properties that can learn highly intricate semantic representations of objects.

Figure 4 (left) presents the structure of a DCGAN generator. The starting input of the Generator is a (1, 100) noise vector. This vector then passes through four convolutional layers with up-sampling and a stride of 2 to generate an RGB image result of size (64, 64, 3). To achieve this, the input vector is projected onto a 1024-dimensional output to match the input of the initial convolutional layer. Figure 4 (right) presents the structure of a DCGAN discriminator. In contrast to the generator, the discriminator takes an input image of size (64, 64, 3), which is the same size as the output generated by the generator. The input image then passes through four standard down-sampling convolutional layers, with a stride of 2. In the final output layer, the image is flattened into a vector, which is usually fed to a sigmoid function that outputs the discriminator's prediction for that image—a single value representing the probability of the image having the class of interest within the range of [0, 1].
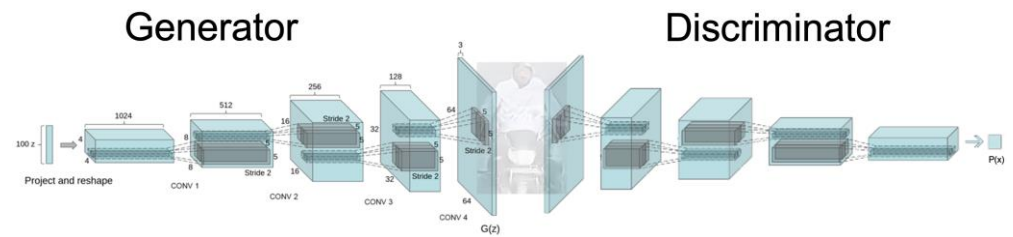
Figure 4. DCGAN generator and discriminator model architecture.

Table 2 presents the hyperparameters used to train the DCGAN model. To speed up the DCGAN's convergence, we use spectral normalization, a novel method of initializing weights, that has been developed for GANs using (3) and (4) to enhance the stability of model training.

$$\sigma = \|W_v\| = u^T W_v \tag{3}$$

$$W_{SN}(W) = \frac{W}{\sigma(W)} \tag{4}$$

Table 2. Hyperparameters for the trained DCGAN model.

Hyperparameter	Value
batch_size	128
weight_init_std	0.02
weight_init_mean	0.0
leaky_relu_slope	0.2
downsize_factor	2
dropout_rate	0.5
scale_factor	4 (downsize_factor)
optimizer	Adam
lr_initial_d	tfe.Variable (0.0002)
lr_initial_g	tfe.Variable (0.0002)
lr_decay_steps	1000
noise_dim	128

The vectors u and v, which have the same size, are randomly generated and used in a power iteration process for a specific weight during each learning step. This approach is more computationally efficient compared to simply penalizing the gradients. During the backpropagation step, we update the weights using $W_{SN}(W)$ instead of W. The convolutional and dense layers are initialized with the truncated normal distribution. Additionally, we removed the bias term from the convolutional layers, which helps to further stabilize the model. During training, label smoothing was used as a regularization technique to avoid the discriminator from becoming either too overconfident or underconfident in its predictions. If the discriminator becomes too certain that a particular image contains a driver with a helmet, the generator may exploit this fact and continuously generate only such images, thereby ceasing to improve its performance. To counteract this, we can set the class labels for the negative classes to be within the range of [0, 0.3] and [0.7, 1] for the positive ones. This strategy prevents the overall probabilities from approaching the two thresholds too closely. Additionally, we introduced some noise to the labels (5%), so the actual and predicted distributions become more dispersed and begin to intersect with each other.

Consequently, creating a customized distribution of generated images during the learning phase becomes simpler. The Adam optimizer, with a standard learning rate of 0.0002 and a beta of 0.5, was the most effective optimization algorithm for this task. This learning rate is applied to both models. To assess the probability that the actual data is more

realistic than the generated data, we use the relativistic average least squares (RaLSGAN) defined by (5) and (6).

$$L_D^{RSGAN} = -E_{(x_r,\ x_f)}[\log\left(sigmoid\left(C(x_r) - C\left(x_f\right)\right)\right)] \tag{5}$$

$$L_G^{RSGAN} = -E_{(x_r,\ x_f)}[log\left(sigmoid\left(C\left(x_f\right) - C(x_r)\right)\right)] \tag{6}$$

The objective is to evaluate the likeness between the real (*r*) and synthetic (*f*) data distributions. *RSGAN* is considered to have reached the optimal point when $D(x)$ equals 0.5, indicating that $C(x_r)$ and $C\left(x_f\right)$ are equivalent. In addition to issues such as non-convergence and vanishing/exploding gradients, GANs may also encounter a significant problem known as mode collapse. This occurs when the generator begins generating a restricted range of samples. To address this issue, we applied a few techniques such as: label smoothing, instance noise, and weight initialization. Another technique that we employed during training is Experience Replay, which helped in retaining some of the recently produced images in memory. After every (replay_step) iteration, we train *D* on those previous images to remind the network of past generations, reducing the likelihood of overfitting to a specific instance of data batches during training.

Utilizing DCGANs in our study significantly contributes to performance enhancement by addressing the challenge of class imbalance in the dataset. DCGANs are particularly effective in generating high-quality synthetic images that are indistinguishable from real images. This capability allowed us to augment the training dataset, especially for under-represented classes, thereby providing a more balanced dataset. Enhanced dataset balance improves the learning process, enabling the neural network to generalize better across different scenarios and not overfit to the over-represented classes. Moreover, the synthetic images generated by DCGANs add variability to the training data, which helps in improving the robustness of our model against variations in real-world inputs, such as different lighting conditions, angles, and helmet types. This directly translates to higher accuracy and reliability in detecting helmet violations across diverse environments.

4.4. Detector Training Module

In selecting a model for detecting motorcyclists at varying distances, weather, and lighting conditions, several options were initially considered, including Faster R-CNN and Mask R-CNN. YOLO was chosen for its versatility and ability to generalize to objects of varying sizes and backgrounds, making it well-suited to the task. Unlike the other models, YOLO is a single-shot detector that allows for multiple objects to be detected in a single pass as opposed to multiple stages. This makes it more efficient than other models. Moreover, YOLO models are more generalizable as objects of varying sizes and backgrounds may be detected. Lastly, YOLOv8 offers data augmentation tools such as mosaic that significantly aid in training the model to handle low-quality images. Thus, the YOLOv8 model architecture was used. Compared to previous iterations of YOLO such as YOLOv5, YOLOv8 includes several additional features that help address the issues identified in the previous section. Notably, YOLOv8 introduces focal loss as shown in Ref. [48], an improved version of cross-entropy loss, to address the class imbalance issue. This works by assigning more weights to difficult-to-detect objects and less weight to easy to detect objects. Thus, the balanced cross-entropy loss function shown in (7) is modified as shown in (8):

$$CE(p_t) = -\alpha\log(p_t) \tag{7}$$

$$FL(p_t) = -(1 - p_t)^{\gamma}\log(p_t) \tag{8}$$

where p_t is the predicted probability of the ground truth class, γ is a class imbalance tuning parameter, and α is a static class imbalance weighting factor. In addition to better-addressing class imbalance, YOLOv8 is also better suited to work in busy urban environments where objects may be occluded or differ in size. Unlike its previous iterations, YOLOv8 is an

anchorless model which does not rely on predefined anchor boxes. Instead, it uses a series of learned key points which represent each object in the image. This anchorless architecture allows more flexibility in detecting objects of different sizes and aspect ratios, as well as better handling of object occlusion. An 80:10:10 split for the train/validation/test components was performed on the dataset, and Table 3 presents the hyperparameters used to train the YOLO models used (YOLOv5, YOLOv7, and YOLOv8) in our experiments.

Table 3. Hyperparameters for the trained YOLO Models.

Hyperparameter	Value
batch_size	8
imgsz	1088
optimizer	Adam
learning_rate	0.001
dropout	0.1
iou	0.7
momentum	0.937
weight_decay	0.0005
warmup_bias_lr	0.1
fl_gamma	2.0
label_smoothing	0.1
mosaic	0.1
close_mosaic	10

4.5. Test Time Augmentation

Test time augmentation (TTA) has been widely used to improve the accuracy of computer vision models in various domains, including object detection. By applying transformations such as rotations, flips, and changes in brightness and contrast to test images, TTA can effectively increase the diversity of the test data, leading to more robust and accurate models. This is particularly important in the context of detecting helmets, where variations in lighting conditions, camera angles, and helmet types can greatly affect the accuracy of the model. TTA can help mitigate these issues by providing the model more diverse test data, allowing it to better generalize to unseen data. Several studies have demonstrated the effectiveness of TTA in improving the accuracy of helmet detection models, suggesting that it is a valuable technique for this task. However, it should be noted that TTA can increase the computational cost of testing, and careful consideration should be given to the trade-off between accuracy and computational efficiency.

5. Results

The objective of this study was to develop a framework that improves the accuracy of existing methods for detecting helmet use violations, which often rely on manual inspection and can be prone to error. Three different object detection models were considered, namely YOLOv5, YOLOv7, and YOLOv8. Additionally, augmented and synthetic images were generated using DCGANs to further improve performance on under-represented classes. Each of the techniques was then applied and their performance was evaluated to identify the best-performing detector. TTA was applied during the inference phase to further improve the model performance. The models were trained and tested on a dataset of 100 traffic videos collected from the AI City Challenge Dataset (data from India). This dataset presents a wide variety of different challenges common to the transportation domain, including varying image quality, lighting, and occlusion. The DCGAN model was trained on a device equipped with an NVIDIA RTX3080 Ti graphics card and 12 GB GDDR6X RAM. Its run-time was approximately twelve hours per class, for a total of 1000 epochs using an early stopping window of 30 epochs. The YOLO detector models, trained on the same device, had a run-time of approximately three days for a total of 300 epochs, using an early stopping window of 20 epochs.

5.1. Performance Metrics

The results for the performed experiments were evaluated using the mAP50, mAP50-95, precision, and recall on the validation dataset, and mAP, F1, and frames per second (fps) on the test dataset. mAP50 (mean average precision at 50% intersection over union) is a common metric used to evaluate the performance of object detection models. It measures the average precision of the model at different levels of confidence when the overlap between the predicted bounding box and the ground truth bounding box is at least 50%. In other words, it measures the accuracy of the model in localizing objects in the image, and it takes into account both the precision and the recall of the model. A mAP50 score of 1.0 means that the model has achieved perfect precision and recall, while a score of 0 means that the model has failed to detect any objects. The mAP50-95 (mean average precision at the 50–95% intersection over union range) is another common metric used to evaluate the performance of object detection models. It measures the average precision of the model at different levels of confidence when the overlap between the predicted bounding box and the ground truth bounding box is between 50% and 95%. The mAP50-95 is often considered a more stringent evaluation metric compared to the mAP50, as it requires the model to detect objects with higher levels of precision and recall. A higher mAP50-95 score indicates that the model is more accurate in localizing objects in the image, and it takes into account both the precision and the recall of the model over a wider range of intersection over union thresholds.

Precision measures the proportion of true positive predictions (i.e., the number of correctly identified positive instances) out of all positive predictions made by the model (i.e., the sum of true positives and false positives). A high precision score indicates that the model is making accurate positive predictions and has a low false positive rate. Precision can be calculated using:

$$\text{Precision} = \frac{TP}{(TP + FP)} \tag{9}$$

Recall measures the proportion of true positive predictions (i.e., the number of correctly identified positive instances) out of all actual positive instances in the data. Recall can be calculated using:

$$\text{Recall} = \frac{TP}{(TP + FN)} \tag{10}$$

The F1 score was calculated using (11), which takes into account both the precision and recall of the algorithm in order to provide a single measure of its effectiveness. The F1 score ranges from 0 to 1, with a score of 1 indicating perfect precision and recall, and a score of 0 indicating that the model did not correctly identify any objects.

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{11}$$

5.2. Detector Model Results

A summary of the experimental results is displayed in Table 4, showing the DCGANs + YOLOv8 + TTA model performing the best, achieving a mAP on the test data of 0.810. YOLOv5 and YOLOv7 achieved mAPs of 0.621 and 0.643, respectively, while YOLOv8 trained on the original data only achieved a mAP of 0.680. As shown in Table 4, the precision and recall of YOLOv8 were 0.928 and 0.891, respectively, which were superior to that of YOLOv7 (0.911 and 0.83) and YOLOv5 (0.890 and 0.804), indicating that the newer YOLO architectures tend to have more true positive detections and fewer false positive detections. When comparing the inference speed of the different YOLO architectures, YOLOv8 and YOLOv7 had approximately the same speed and were 3% faster than YOLOv5 but yielded a 13% improvement in the F1 score when comparing YOLOv5 to YOLOv8.

Table 4. Experimental results on the validation and test datasets.

Model	mAP50	mAP50-95	Precision	Recall	mAP	F1	fps
YOLOv5	0.842	0.459	0.890	0.804	0.621	0.80	150
DCGANs + YOLOv5	0.823	0.465	0.754	0.799	0.601	0.78	150
YOLOv7	0.851	0.521	0.911	0.830	0.643	0.82	155
DCGANs + YOLOv7	0.862	0.526	0.915	0.836	0.654	0.82	155
YOLOv8	0.866	0.603	0.928	0.891	0.680	0.91	155
DCGANs + YOLOv8	0.913	0.721	0.945	0.893	0.782	0.93	143
DCGANs + YOLOv8 + TTA	0.949	0.749	0.972	0.919	0.810	0.96	92

When combined with DCGANs, the precision and recall continue to improve. The YOLOv8 model achieved an F1 score of 0.91 at a confidence level of 0.617 for all classes, while the DCGANs + YOLOv8 + TTA model resulted in an improved F1 score of 0.96 at a lower confidence level of 0.334. The precision and recall of the DCGANs + YOLOv8 + TTA model were also greater than that of YOLOv8 alone (0.928 and 0.891) and the DCGANs + YOLOv8 model (0.945 and 0.893). When combining DCGANs and TTA with YOLOv8, the synthetic data and augmentation data helped the model generalize better over diverse scenarios, making it more accurate at lower confidence levels. By lowering the confidence threshold, the model can detect more nuanced or difficult-to-detect objects, thus increasing the overall recall. DCGANs and TTA provide more robust and diverse training data where the model can afford to lower its confidence threshold while still improving accuracy. In contrast, traditional models like YOLOv5 and YOLOv7, though improved upon by their iterations, still fall short compared to YOLOv8. When extended to incorporate DCGANs, these models see improvements, particularly in recall and precision, but do not achieve the same high standards set by the latest enhancements in YOLOv8. While the DCGANs + YOLOv8 + TTA model was more accurate than only using DCGANs or only running a YOLOv8 model, the processing speed was greatly reduced (155 fps to 92 fps). However, given the sensitive nature of this application and the high cost of false positives associated with traffic enforcement, the approach remains viable despite the greater computational power required.

5.3. Synthetic Augmentation Results

Synthetic data were created to improve the dataset and address class imbalance. Figure 5 presents a sample of the synthetic images generated by DCGANs with size 64×64, which are very close to authentic images in terms of crispness, naturalness, and realism. A total of 2000 augmented and synthetic images per class were combined with real images to improve the performance of the helmet recognition system. The results in Table 4 demonstrated that the DCGANs + YOLOv8 + TTA model was more robust to variations in the input data, resulting in more accurate results.

Figure 5. Sample output from DCGANs for "DriverNoHelmet" class.

These variations include changes in lighting conditions, camera angles, image quality, and occlusion. Additionally, the DCGANs + YOLOv8 + TTA model is more effective at detecting minority classes. Since small datasets can limit the ability of neural networks to locate objects and can result in overfitting, greater improvements can be expected

in these minority classes as a result of synthetic minority sampling. As expected, the model's performance on the classes P1Helmet, P1NoHelmet, and P2NoHelmet improved as shown in Figure 6. Figure 6 compares the F1-confidence and precision-recall curves for the YOLOv8 vs. DCGANs + YOLOv8 + TTA models. Therefore, by saturating the original dataset with augmented and synthetic images generated by DCGANs, the model's detection and recognition performance can be greatly enhanced, as presented in Figure 7, which shows inference on a sampled test data.
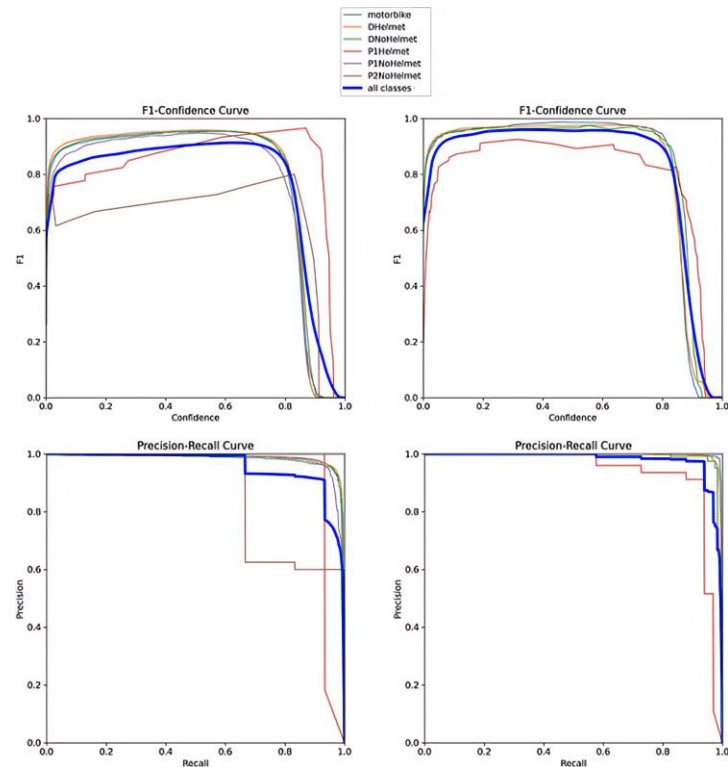

Figure 6. F1 and precision-recall curves for the trained YOLOv8 model (**left**: YOLOv8, **right**: YOLOv8 + DCGANs + TTA).

Figure 7. Inference of trained DCGANs + YOLOv8 + TTA model on test data.

6. Conclusions

The objective of this work is to propose a novel methodology to detect motorcycle riders and identify helmet violations. As helmet usage is highly associated with lower

crash risk, the methodology proposed in this research aims to improve safety overall by enforcing traffic laws. This study uses a CNN-based solution for helmet detection and expands the CNN training set using synthetic data, which were generated using DCGANs. This procedure was conducted to improve both identification and classification outcomes. The latest YOLO algorithm to date (i.e., YOLOv8) and test time augmentation were also used to improve the model performance. Findings reveal that combining original and synthetic images improves the ability to detect helmets. More specifically, by generating synthetic images for the most common classes, the model achieved a higher F1-score with 2000 additional images of a 64×64 image size. Additionally, incorporating noise during training reduced errors and improved the training phase. The proposed technique was demonstrated to greatly improve the effectiveness of the model in detecting if motorcyclists were wearing helmets. Also, test time augmentation greatly contributed to helmet violation detection. While mosaic data augmentation works in the training stage, adding an extra layer of augmentation at the test stage substantially improved the performance in conditions of poor lighting and bad weather conditions. This could thereby make detection systems more adaptable and effective in a wider range of scenarios.

As the field of computer vision continues to evolve and new YOLO architectures improve upon previous versions, there remains a substantial equity concern. The majority of the datasets used in YOLO come from very organized traffic environments, with higher image qualities and more readily available data. This makes adopting the existing frameworks more challenging in less organized environments, where image quality may be lower, the road environment may be more complex, and additional data may not be available. In these cases, the DCGAN approach allows generating data classes which are missing. This helps addressing this equity concern by considering diverse needs and obstacles faced by various communities worldwide.

Several opportunities for future work can be derived from this study. For example, the proposed framework might be improved by incorporating a tracking algorithm, which can track helmet detections over several images to improve the model accuracy. This can further be achieved by using ensembling. Also, future research can explore the impact of synthetic image size and quality by training synthetic images of different sizes and comparing with other real data benchmarks. This enables obtaining a wider collection of traffic scenarios and thus allows for developing models for various environments. Furthermore, exploring the use of other GAN models, such as BigGAN, StyleGAN, and MGAN can provide potential improvements in model performance. Finally, there is great potential to incorporate diffusion models, allowing for further styles to be applied to the generated images.

Author Contributions: Conceptualization, M.S. and T.G.; methodology, M.S. and T.G.; software, M.S. and G.L.; validation, T.A., G.L., S.G. and T.S.; formal analysis, S.G. and T.S.; investigation, T.A. and M.S.; resources, T.G.; data curation, T.G. and M.S.; writing—original draft preparation, M.S., T.G., G.L., T.A., S.G. and T.S.; writing—review and editing, G.L., T.A. and S.G.; visualization, M.S., T.G. and G.L.; supervision, S.G. and T.S. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: The data presented in this study are available for download and use at https://www.aicitychallenge.org/ (accessed on 20 March 2023).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lanzaro, G.; Sayed, T.; Alsaleh, R. Modeling Motorcyclist–Pedestrian Near Misses: A Multiagent Adversarial Inverse Reinforcement Learning Approach. *J. Comput. Civ. Eng.* **2022**, *36*, 04022038. [CrossRef]
2. Haworth, N. Powered two wheelers in a changing world—Challenges and opportunities. *Accid. Anal. Prev.* **2012**, *44*, 12–18. [CrossRef]

3. Dua, A.; Wei, S.; Safarik, J.; Furlough, C.; Desai, S.S. National mandatory motorcycle helmet laws may save $2.2 billion annually: An inpatient and value of statistical life analysis. *J. Trauma Acute Care Surg.* **2015**, *78*, 1182–1186. [CrossRef]

4. World Health Organization. *Global Status Report on Road Safety 2018*; World Health Organization: Geneva, Switzerland, 2018; Available online: https://apps.who.int/iris/handle/10665/276462 (accessed on 3 January 2023).

5. Rutter, D.R.; Quine, L. Age and experience in motorcycling safety. *Accid. Anal. Prev.* **1996**, *28*, 15–21. [CrossRef]

6. Clabaux, N.; Fournier, J.-Y.; Michel, J.-E. Powered two-wheeler riders' risk of crashes associated with filtering on urban roads. *Traffic Inj. Prev.* **2017**, *18*, 182–187. [CrossRef] [PubMed]

7. Mullin, B.; Jackson, R.; Langley, J.; Norton, R. Increasing age and experience: Are both protective against motorcycle injury? A case-control study. *Inj. Prev.* **2000**, *6*, 32–35. [CrossRef]

8. Vlahogianni, E.I. Powered-Two-Wheelers kinematic characteristics and interactions during filtering and overtaking in urban arterials. *Transp. Res. Part F Traffic Psychol. Behav.* **2014**, *24*, 133–145. [CrossRef]

9. Gabella, B.; Reiner, K.L.; Hoffman, R.E.; Cook, M.; Stallones, L. Relationship of helmet use and head injuries among motorcycle crash victims in El Paso County, Colorado, 1989–1990. *Accid. Anal. Prev.* **1995**, *27*, 363–369. [CrossRef]

10. Ichikawa, M.; Chadbunchachai, W.; Marui, E. Effect of the helmet act for motorcyclists in Thailand. *Accid. Anal. Prev.* **2003**, *35*, 183–189. [CrossRef]

11. Vanlaar, W.; Hing, M.M.; Brown, S.; McAteer, H.; Crain, J.; Faull, S.M. Fatal and serious injuries related to vulnerable road users in Canada. *J. Safety Res.* **2016**, *58*, 67–77. [CrossRef]

12. Chang, H.-L.; Yeh, T.-H. Risk Factors to Driver Fatalities in Single-Vehicle Crashes: Comparisons between Non-Motorcycle Drivers and Motorcyclists. *J. Transp. Eng.* **2006**, *132*, 227–236. [CrossRef]

13. Yousif, M.T.; Sadullah, A.F.M.; Kassim, K.A.A. A review of behavioural issues contribution to motorcycle safety. *IATSS Res.* **2020**, *44*, 142–154. [CrossRef]

14. RogÅLe, J.; El Zufari, V.; Vienne, F.; Ndiaye, D. Safety messages and visibility of vulnerable road users for drivers. *Saf. Sci.* **2015**, *79*, 29–38. [CrossRef]

15. Truong, L.T.; Nguyen, H.T.T.; De Gruyter, C. Correlations between mobile phone use and other risky behaviours while riding a motorcycle. *Accid. Anal. Prev.* **2018**, *118*, 125–130. [CrossRef] [PubMed]

16. Rowden, P.; Watson, B.; Haworth, N.; Lennon, A.; Shaw, L.; Blackman, R. Motorcycle riders' self-reported aggression when riding compared with car driving. *Transp. Res. Part F Traffic Psychol. Behav.* **2016**, *36*, 92–103. [CrossRef]

17. Manan, M.M.A.; Ho, J.S.; Arif, S.T.M.S.T.; Ghani, M.R.A.; VÅLarhelyi, A. Factors associated with motorcyclists' speed behaviour on Malaysian roads. *Transp. Res. Part F Traffic Psychol. Behav.* **2017**, *50*, 109–127. [CrossRef]

18. Drysdale, W.F.; Kraus, J.F.; Franti, C.E.; Riggins, R.S. Injury patterns in motorcycle collisions. *J. Trauma Inj. Infect. Crit. Care* **1975**, *15*, 99–115. [CrossRef] [PubMed]

19. MacLeod, J.B.A.; DiGiacomo, J.C.; Tinkoff, G. An Evidence-Based Review: Helmet Efficacy to Reduce Head Injury and Mortality in Motorcycle Crashes: EAST Practice Management Guidelines. *J. Trauma Inj. Infect. Crit. Care* **2010**, *69*, 1101–1111. [CrossRef] [PubMed]

20. Keng, S.-H. Helmet use and motorcycle fatalities in Taiwan. *Accid. Anal. Prev.* **2005**, *37*, 349–355. [CrossRef]

21. Cunto, F.J.C.; Ferreira, S. An analysis of the injury severity of motorcycle crashes in Brazil using mixed ordered response models. *J. Transp. Saf. Secur.* **2017**, *9* (Suppl. 1), 33–46. [CrossRef]

22. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001; pp. I-511–I-518. [CrossRef]

23. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; pp. 886–893. [CrossRef]

24. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]

25. Shoman, M.; Aboah, A.; Adu-Gyamfi, Y. Deep learning framework for predicting bus delays on multiple routes using heterogenous datasets. *J. Big Data Anal. Transp.* **2020**, *2*, 275–290. [CrossRef]

26. He, K.; Gkioxari, G.; DollÅLar, P.; Girshick, R. Mask R-CNN. *arXiv* **2018**. Available online: http://arxiv.org/abs/1703.06870 (accessed on 29 April 2023).

27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**. [CrossRef]

28. Aboah, A.; Shoman, M.; Mandal, V.; Davami, S.; Adu-Gyamfi, Y.; Sharma, A. A vision-based system for traffic anomaly detection using deep learning and decision trees. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Nashville, TN, USA, 19–25 June 2021; pp. 4202–4207.

29. Shoman, M.; Aboah, A.; Morehead, A.; Duan, Y.; Daud, A.; Adu-Gyamfi, Y. A region based deep learning approach to automated retail checkout. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, New Orleans, LA, USA, 18–24 June 2022; pp. 3209–3214.

30. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]

31. Dahiya, K.; Singh, D.; Mohan, C.K. Automatic detection of bike-riders without helmet using surveillance videos in real-time. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 3046–3051. [CrossRef]

32. Desai, M.; Khandelwal, S.; Singh, L.; Gite, S. Automatic helmet detection on public roads. *Int. J. Eng. Trends Technol.* **2016**, *35*, 185–188. [CrossRef]

33. Medimi, V.S.; Mavilla Vari Palli, A.J. *A Comparative Study of YOLO and Haar Cascade Algorithm for Helmet and License Plate Detection of Motorcycles*; Blekinge Institute of Technology: Belkinge, Sweden, 2022.

34. Liu, Y.; Alsaleh, R.; Sayed, T. Modeling Pedestrian Temporal Violations at Signalized Crosswalks: A Random Intercept Parametric Survival Model. *Transp. Res. Rec. J. Transp. Res. Board* **2022**, *2676*, 707–720. [CrossRef]

35. Liu, Y.; Alsaleh, R.; Sayed, T. Modeling lateral interactions between motorized vehicles and non-motorized vehicles in mixed traffic using accelerated failure duration model. *Transp. A Transp. Sci.* **2022**, *18*, 910–933. [CrossRef]

36. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020. Available online: http://arxiv.org/abs/2004.10934 (accessed on 29 April 2023).

37. Jocher, G. ultralytics/yolov5: v6.0—YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support. *Zenodo* **2021**. [CrossRef]

38. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo Algorithm Developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]

39. Jia, W.; Xu, S.; Liang, Z.; Zhao, Y.; Min, H.; Li, S.; Yu, Y. Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector. *IET Image Process.* **2021**, *15*, 3623–3637. [CrossRef]

40. Siebert, F.W.; Lin, H. Detecting motorcycle helmet use with deep learning. *Accid. Anal. Prev.* **2020**, *134*, 105319. [CrossRef]

41. Silva, R.R.V.E.; Aires, K.R.T.; De Melo Souza Veras, R. Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers. In Proceedings of the 2014 27th SIBGRAPI Conference on Graphics, Patterns and Images, Rio de Janeiro, Brazil, 26–30 August 2014; pp. 141–148. [CrossRef]

42. Doungmala, P.; Klubsuwan, K. Helmet Wearing Detection in Thailand Using Haar Like Feature and Circle Hough Transform on Image Processing. In Proceedings of the 2016 IEEE International Conference on Computer and Information Technology (CIT), Nadi, Fiji, 8–10 December 2016; pp. 611–614. [CrossRef]

43. Jie, L.; Huanming, L.; Tianzheng, W.; Min, J.; Shuai, W.; Kang, L.; Xiaoguang, Z. "Safety helmet wearing detection based on image processing and machine learning. In Proceedings of the 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), Doha, Qatar, 4–6 February 2017; pp. 201–205. [CrossRef]

44. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**. [CrossRef]

45. Shoman, M.; Amo-Boateng, M.; Adu-Gyamfi, Y. Multi-purpose, multi-step deep learning framework for network-level traffic flow prediction. *Adv. Data Sci. Adapt. Anal.* **2022**, *14*, 2250010. [CrossRef]

46. Shoman, M.; Aboah, A.; Daud, A.; Adu-Gyamfi, Y. Graph Convolutional Gated Recurrent Unit Network for Traffic Prediction using Loop Detector. *Data Adv. Data Sci. Adapt. Anal.* **2024**. [CrossRef]

47. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]

48. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007. [CrossRef]