vehicles

MDPI

Article

An Enhanced Model for Detecting and Classifying Emergency Vehicles Using a Generative Adversarial Network (GAN)

Mo'ath Shatnawi [1] and Maram Bani Younes [2,*]

[1] Software Engineering, Philadelphia University, Amman 19392, Jordan
[2] Information Security and Cybersecurity, Philadelphia University, Amman 19392, Jordan
* Correspondence: mbani047@uottawa.ca

Abstract: The rise in autonomous vehicles further impacts road networks and driving conditions over the road networks. Cameras and sensors allow these vehicles to gather the characteristics of their surrounding traffic. One crucial factor in this environment is the appearance of emergency vehicles, which require special rules and priorities. Machine learning and deep learning techniques are used to develop intelligent models for detecting emergency vehicles from images. Vehicles use this model to analyze regularly captured road environment photos, requiring swift actions for safety on road networks. In this work, we mainly developed a Generative Adversarial Network (GAN) model that generates new emergency vehicles. This is to introduce a comprehensive expanded dataset that assists emergency vehicles detection and classification processes. Then, using Convolutional Neural Networks (CNNs), we constructed a vehicle detection model demonstrating satisfactory performance in identifying emergency vehicles. The detection model yielded an accuracy of 90.9% using the newly generated dataset. To ensure the reliability of the dataset, we employed 10-fold cross-validation, achieving accuracy exceeding 87%. Our work highlights the significance of accurate datasets in developing intelligent models for emergency vehicle detection. Finally, we validated the accuracy of our model using an external dataset. We compared our proposed model's performance against four other online models, all evaluated using the same external dataset. Our proposed model achieved an accuracy of 85% on the external dataset.

Keywords: autonomous vehicle; machine learning; emergency vehicle; GANs; CNN; object detection

check for updates

Citation: Shatnawi, M.; Bani Younes, M. An Enhanced Model for Detecting and Classifying Emergency Vehicles Using a Generative Adversarial Network (GAN). *Vehicles* **2024**, *6*, 1114–1139. https://doi.org/10.3390/vehicles6030053

Academic Editor: Yiqun Liu

Received: 4 April 2024
Revised: 22 June 2024
Accepted: 25 June 2024
Published: 29 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Smart cities are widely developed and have been investigated differently recently. Technological development helps to build and monitor these cities. They aim to improve the quality of life by improving the quality of services such as education, healthcare, and transportation. These services have been linked to technological innovation [1–3]. Traffic crises are the most critical challenges in traditional cities, especially crowded ones. Modern technologies emerge and provide solutions, especially in the matter of enhancing the safety conditions on road networks. Emergency vehicle management is one of the most critical problems that require sensitive and real-time solutions. Special driving rules are announced to vehicles around emergency vehicles, such as opening the way for them and giving them the highest priority to traverse a signalized road intersection.

Detecting emergency vehicles on the road network is the first step in reacting according to location, speed, and other parameters. Drivers used to be able to see these emergency vehicles and respond accordingly visually. Emergency vehicles produce sirens in more crowded and fast road scenarios to alert the surrounding drivers. Several drivers panic to react to these sudden sirens. Sometimes, they fail to determine these emergency vehicles' exact location, speed, or other characteristics. Thus, they fail to respond promptly and adequately, threatening the road network's safety conditions. Benefiting from modern vehicles' advanced equipment and technologies, it is necessary to develop an intelligent

emergency vehicle detection system. Models that use artificial intelligence technology can improve and increase the efficiency of logistics transportation services, reduce response time, choose the healthiest and fastest routes, and alert surrounding vehicles to open the way for them [4–7].

Artificial intelligence (AI) offers many varied solutions in this field. It has excellent potential in Internet applications to monitor and manage traffic and predict future events through machine learning (ML). Many scientific research studies have tried to address this topic by finding a way to detect and classify vehicles on the roads. Some of this research specifically studied the issue of detecting emergency vehicles through photos and videos to inform and alert the concerned authorities [8,9]. Some problems are reported regarding these previous studies that consider either the used method, the tested dataset, or the accuracy of the obtained model.

Several studies have introduced intelligent emergency vehicle detection models in the literature to classify vehicles on road networks. The main weakness of the previous studies is the suitability of the datasets used. The unbalanced dataset rarely involves an emergency vehicle and is not the best choice for training the machine learning mechanisms and predicting the existing emergency vehicles [10]. We have noticed that some datasets used are unrealistic in several scenarios. These datasets' considered photos or videos are not taken from road scenarios (e.g., toys or vehicles on an agency) [11–13]. In addition, the method of determining the accuracy as a criterion is not always the most appropriate criterion to rely on to accept the results [14]. An additional study, exemplified by Sheng et al. [15], has put forth a learning-oriented strategy addressing video temporal coherence due to the inadequacy of recently devised techniques, specifically, filters designed for improving, restoring, editing, and analyzing static images when applied to video clips. The inherent distinction lies in recognizing that a video transcends a mere sequence of individual images. This disparity poses a challenge in the context of effectively detecting emergency vehicles in transit on roads.

The real problem we may face in applying the proposed model is the same dataset on which we will train the model. It turns out that several conditions must be achieved in the dataset to obtain the precise and accurate detection of emergency vehicles. Among these conditions are the size of the dataset and the quality of the images, in addition to the balance characteristics [16]. Consequently, this work aims to find a model that achieves these goals with the required accuracy standards. We also plan to produce a suitable new dataset that achieves acceptable accuracy using Generative Adversarial Networks (GANs) due to their proven ability to improve and enhance dataset quality and address specific challenges in vehicle detection. GANs have been successfully used to enhance datasets in many fields [17–20], while GANs have been used to improve data quality in image translation models to improve detection under poor lighting conditions to improve night images and increase the accuracy of vehicle detection in the dark [21,22].

In addition, some image restoration models using GANs have proven effective in mitigating the negative effects of occlusion on vehicle detection [23]. This confirms the benefit of GAN networks to create artificial images that enhance the dataset and simulate reality in various difficult circumstances while maintaining the quality of the dataset, which ultimately leads to more accurate detection of emergency vehicles. Finally, we aim to test and verify the generated dataset. Figure 1 illustrates the general steps of the proposed work, encompassing all the stages from obtaining the initial dataset and its refinement using GANs, to the evaluation and validation of the newly generated dataset.

The remainder of this paper is organized as follows: Section 2 studies previous work in this field of research. Then, Section 3 investigates the main characteristics of existing datasets that have been used to detect emergency vehicles on road networks. It clarifies the main weaknesses and problems in each dataset. Section 4 presents the steps of gathering, preparing, and augmenting the images for the dataset. It also clearly shows the steps of generating the new images using GANs. The details of the testing, verification, and

comparing processes are explained in Section 6. Finally, Section 7 concludes the entire paper and recommends some future studies.

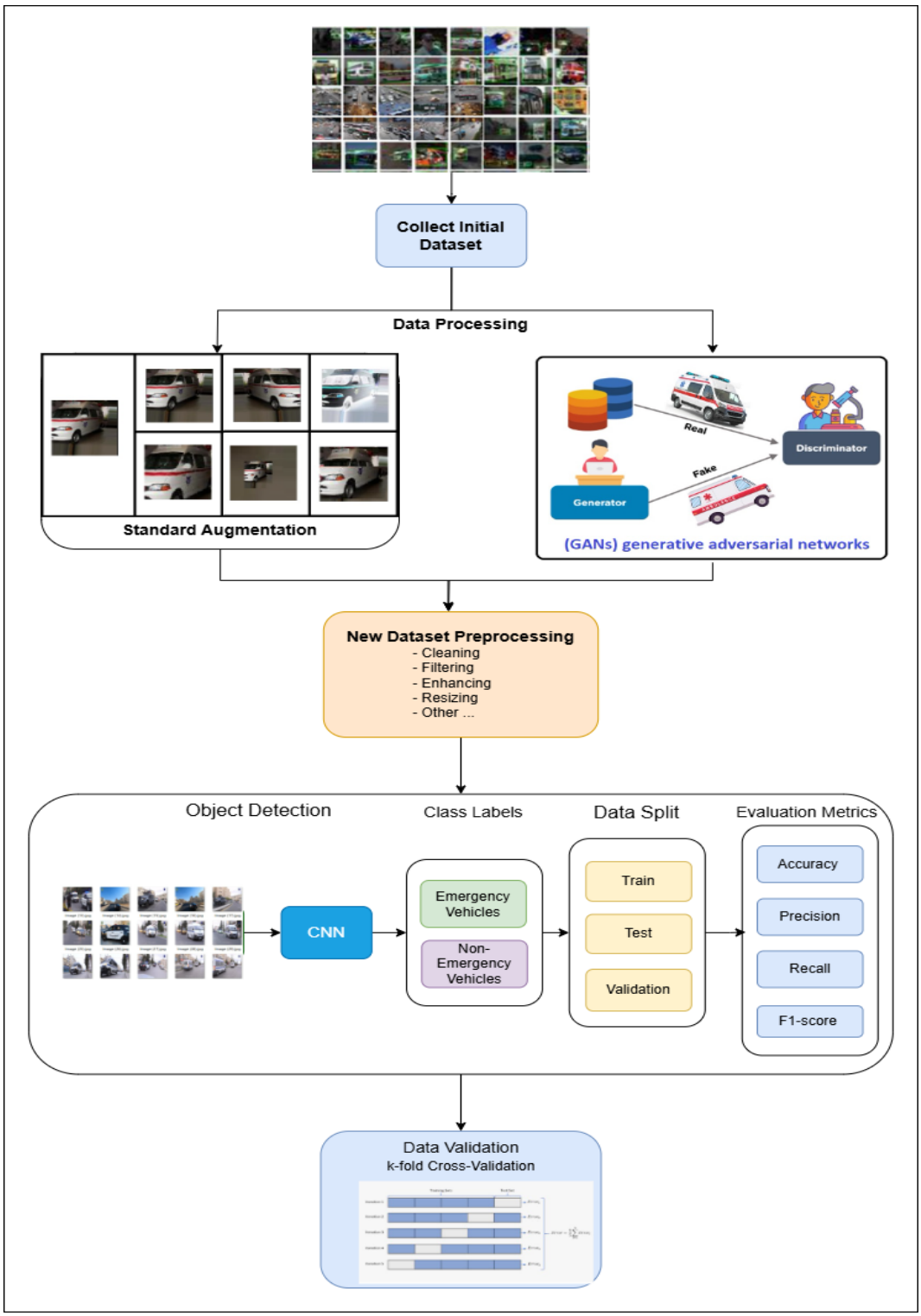


Figure 1. Methodology for developing the emergency vehicle detection model.

2. Related work

The significant increase in the number of vehicles and the traffic crises that may disrupt the mobility of emergency vehicles encourage several researchers to work in this field. Using new technology, such as the Internet of Things (IoT), to propose new tools or

models should help reduce these problems' effects and assist emergency vehicles over the road network. This section first investigates previous studies of vehicle and object detection methods over the road network. Then, we study the usage of the GANs model to improve or augment the existing datasets and its role in improving the detection of vehicles.

2.1. Vehicle and Object Detection Methods

In our exploration of studies related to object detection, numerous research endeavors were identified, particularly those focusing on feature extraction from images. For instance, the investigation undertaken by Jiang al. [24] sought to seamlessly incorporate empirical aesthetic rules with conventional machine learning algorithms and deep learning techniques for feature extraction. This integration demonstrated efficacy in object detection; however, a noteworthy challenge surfaced. The protracted duration required for feature extraction from images posed a hindrance, rendering it incongruent with the exigencies of real-time emergency vehicle detection. Subsequently, it was observed that several research studies have leveraged advanced artificial intelligence and machine learning technologies to detect and categorize vehicles within the road network. Shuvendu Roy and Sakif Rahman [9] proposed an automated detection system to detect emergency vehicles from a closed circuit television camera (CCTV) using a neural network. They used a pre-trained model to detect objects containing 80 categories (e.g., bike, car, and motorcycle). This study aimed to classify the detected vehicles into emergency vehicles and non-emergency vehicles. Then, a Convolutional Neural Network (CNN) technology was used for the emergency vehicle classification class based on analyzing the visually detected images. They augmented the dataset by changing properties, such as mirroring the images, zooming in, zooming out, rotating, etc. [14]. After that, they used the YOLOv3 algorithm to detect emergency vehicles [25]. In the obtained results, the accuracy of classifying the vehicles is more than 97%. This is of better accuracy than the traditional system that had previously been used. However, this study never determined the size of the used dataset before or after the augmentation process. Moreover, the confusion matrix has not been investigated to show if the data are balanced and to ensure that the accuracy standard used can be adopted or needs more improvement.

On the other hand, two recent works were proposed by D. Ganesh [26] and G. Punyavathi et al. [27]. The first study (i.e., D. Ganesh, [26]) presented an automated system to detect emergency vehicles using machine learning and CNN technologies. The researcher standardized the dataset with some algorithms and adjusted the images. This study developed a new model by the neural network, using the pre-trained CNN model named "ResNet152 CNN Model". The researchers explained how the model works and the required mechanism for building it. This is without exposing the actual application of the model or giving any results or accuracy measures of the detection objects. The latter study proposed by Punyavathi et al. [27] investigated various vehicle detection methods, including traditional statistical methods, a novel algorithm for vehicle classification, and deep learning using the YOLOV3 algorithm. The study demonstrated the effectiveness of the YOLOV3 algorithm for vehicle detection and tracking, achieving an average precision of 95.8% and a recall of 97.5% on a dataset of vehicles captured under varying lighting and weather conditions. The authors collected the dataset using cameras placed on vehicles and fixed locations. While the study did not explicitly state whether the dataset was balanced or imbalanced, the confusion matrix presented in their research suggests that it is unbalanced. Therefore, as previously noted, relying on accuracy and recall as measures may not be appropriate.

Moreover, Pillai and Valles [28] designed a new model to classify and detect vehicles using CNN and YOLO algorithms based on their types and colors. The researcher first designed two CNN models to classify the vehicle's types and colors. Then, they designed a faster model to detect vehicles using YOLO. In this study, the images were improved by eliminating fog, enhancing contrast, and removing noise, accomplished with various tools, including DCP [29] and CLAHE [30]. This process led to an improvement in the

accuracy of vehicle detection and classification. The object was extracted in several ways, such as a colored graph, HOG, and pre-trained CNN, using the coordinates of the four corners of the image perimeter and the proportion of the vehicle's presence relative to the square. However, the other proposed methodology uses YOLO9000 to classify the vehicle by detecting the type of vehicle as well as the color of the vehicle. Despite the precise sequence of the work method, this study has presented a proposal for the detection method without testing or discussing the results of the experiments.

Table 1 summarizes some previous studies that have mainly proposed to detect emergency vehicles on the road network. The main objectives, algorithm, datasets, obtained results, and drawbacks of each study are illustrated in this table.

Table 1. Main characteristics of previous emergency vehicles detection mechanisms.

Related Work	Main Objective	Algorithm	Dataset	Result	Drawbacks and Limitations
Shuvendu Roy and Sakif Rahman [9]	Detect emergency vehicles	CNN, YOLOv3	COCO	Accuracy > 97%	The researcher did not explain the number of data after processing, nor did he indicate the confusion matrix to show whether the data were balanced.
D. Ganesh [26]	Propose an automated system to detect emergency vehicles	CNN	UCI 75% train 25% test	The proposed system, according to the researcher, is better than the traditional	The researcher did not show any actual results showing the experiment's success and accuracy values.
Punyavathi et al. [27]	Detect vehicles	Traditional algorithms, YOLOv3	The source of the dataset is not mentioned	Accuracy > 97%	According to the number of data in the confusion matrix, the data were not balanced, and acc and Recall could not be adopted as real accuracy indicators.
Pillai and Valles [28]	Classification and detection of the type and color of vehicles	CNN, YOLO9000	CompCar, Vehicle color recognition	New model for classification and detection of the type and color of vehicles by two algorithms	The author presented only a proposal without offering any results of the experiments.

2.2. Using Generative Adversarial Networks (GANs) to Generate Augmented Datasets

On the other hand, while numerous research studies have proposed creating improved datasets to enhance results in this field, a distinct lack of focus on studies dedicated to emergency vehicles is observed. Notably, Agrawal and Choudhary [17] recently utilized Generative Adversarial Networks (GANs) in the medical domain, specifically generating chest images. This study highlights the importance of GANs in overcoming challenges related to limited or rare images.

Tanaka and Aranha [18] used GANs to generate new artificial datasets. Then, they used the generated datasets to train the machine learning algorithms. These trained algorithms were evaluated to prove the usefulness of GANs in generating new datasets. The training did not use all the synthetic datasets, and the decision tree classifier was used to examine the quality of the training datasets. The performance evaluation result of using the generated GANs' datasets was better than the original dataset. The artificially generated datasets obtain better accuracy and recall than the original datasets. The results confirm the need to oversample data using the GANs method for images specifically.

V. Kukreja et al. [19] proposed creating a new model using GANs to identify vehicle plates with high accuracy and speed. After collecting the images and changing their size, they omitted many images that were less than the size required by the experiments. They need to use GANs to over-sample the datasets of images. The proposed model is that after the camera captures the images of the vehicles, it sends them to the GANs. In turn, the GANs model generates new artificial images accordingly. Then, the CNN algorithm uses deep learning technology to detect the plates inside the images. The generator in GANs

generates new images similar to authentic images, and the discriminator (classifier) tries to differentiate between fake and real images. The dataset was used from Pascal, increased from other datasets, and the images were taken by the camera and used STARGAN for implementation. The classifier trains the dataset and creates a noisy subset sample with size m. The proposed method achieves an accuracy of up to 99% in recognizing car plates compared to other studies that did not use the GANs to increase the dataset sample.

Furthermore, Y. Gao et al., 2021 [20] proved that data augmentation using GANs is suitable for generating new images of unmanned aerial vehicles (UAVs). Infrared data collection is the primary method for detecting all-weather UAVs through CNN. To obtain better detection effects with better accuracy, the researcher increased the infrared dataset of UAVs through the hostile generation network (i.e., GANs). The work was divided into two parts; the first was to train the GANs to generate new images that are identical to reality by entering the original set of images for the GANs. Second, after testing, the generated images were combined with the authentic images to be trained to detect UAVs through CNN. After that, the GAN dataset was tested, and the detection of UAVs was examined on the trained dataset. The results obtained from the artificially generated dataset are better than those obtained from the original dataset in terms of accuracy, recall, and F-Score metrics.

On the other hand, some image retrieval models using GANs have proven effective in alleviating the negative effects of occlusion on vehicle detection. Xu et al. [23] proposed a framework aimed at accurately determining the spatial and temporal distribution of vehicle wheel loads in scenarios involving occlusion. They used object detection models and key points to identify the load of the vehicle and the wheels, respectively. A binary image classification model was used for the covered vehicles, followed by a novel image mapping approach using Generative Adversarial Networks (GANs). This method aims to effectively convert images of occluded vehicles into unoccluded images, which facilitates the accurate determination of wheel-bearing locations and has been validated through field tests.

Kandasamy and D. Rajamanickam [21] merged YOLO to detect vehicles with GANs algorithms to improve night images and increase the accuracy of detecting vehicles at dark. Thus, night images are introduced to GANs, which generate new daytime images to increase vehicle detection accuracy, that is, by translating night images into daytime images using (Cycle GANs) to improve the features of images and vehicles and improve lighting conditions and bad weather. Two GAN models are used in this work; one generator and one discriminator are assigned for each model. The first generator is responsible for translating nighttime images into daytime images. Then, the first discriminator checks the fake "generated" daytime images for the real-time image. The second generator takes real daytime images and generates nighttime images. Then, the second discriminator checks the output to see if it is a real or fake image. Cycle GANs are the first proposed model for improving the images. After that, another model is proposed to detect inside the tested images using the neural network and YOLOv5.

Finally, a similar successful study includes using GANs specifically to reduce the decrease in accuracy in vehicle detection models in low-light conditions at night. Wu and Yixun [22] proposed a model to improve detection accuracy using image translation technology Using CycleGAN. They relied on training datasets BDD and UA-DETRAC, and custom nighttime vehicle images. They converted well-established daytime vehicle datasets into nighttime equivalents to enhance the training sets for detection models based on YOLO-v5. This approach significantly improved detection accuracy, with a 10.4% increase in the PR curve area and a 9% rise in peak F1 score. The study results highlight the potential for image translation to enhance detection accuracy at night, albeit with some practical limitations and computational requirements.

As we can see, all previous research shows that using the GANs model to pre-process data either enhances or augments the images, or balances the datasets. Table 2 summarizes the main previous studies that have used GANs to generate new datasets. It

illustrates the main objective, used algorithm, original dataset, obtained results, drawbacks, and limitations.

Table 2. Main characteristics of previous studies that used GANs.

Related Work	Main Objective	Algorithm	Dataset	Result	Drawbacks and Limitations
Tanaka and Aranha [18]	Generate new data to be trained in machine learning to oversample and balance data	GANs, SMOTE, ADASYN	Pima Indians Diabetes Database Breast Cancer Wisconsin (Diagnostic) Dataset Credit Card Fraud Detection	Achieved 98% accuracy, proving that trained data using GANs is equal to or better than the original data	The researcher confirmed to us the need for augmentation and oversample of the data, which is reflected in improving the accuracy of the detection data.
V. Kukreja et al. [19]	Detection of vehicle plates	GANs, CNN	Pascal, Images through the camera, Another dataset	The researcher reached an accuracy of 99% in real-time, which is much better than without GANs	The researcher confirmed that GAN data augmentation increases object detection accuracy in real-time.
Yuan Gao et al. [20]	Generate new data for Unmanned aerial vehicles (UAV) to train and detect it	DCGANs, GANs, CNN	Data from the researcher (760 Images) Train 75% Test 25%	Higher accuracy in UAV detection using GANs, validated by F1 Score and mAP@0.5	This is another evidence of the need to use GAN to augment data and improve detection accuracy.
Xu et al. [23]	Accurately determining the spatial–temporal distribution of vehicle wheel loads, using GANs for image inpainting.	Generative Adversarial Networks (GANs)	UA-DETRAC dataset	Developed a framework integrating object and keypoint detection models for vehicle and wheel load identification, respectively. Introduced an image binary classification model for occluded vehicles, followed by an innovative image inpainting approach using GANs.	Requires high-performance GPUs for real-time monitoring; future work aims to improve algorithm efficiency for practical engineering applications.
Kandasamy and D. Rajaman-ickam [21]	Improving night images using GANs and converting them into daytime images to improve vehicle detection accuracy.	Cycle GANs, YOLO v5	Berkeley Deep Drive (BDD)	The researcher proved that using GANs was very effective in converting night images into day images and improving vehicle detection accuracy by a large difference.	The researcher confirmed another important aspect of GANs in addition to data augmentation: improving and enhancing the images, which is very important to raise vehicle detection accuracy.
Wu and Yixun [22]	Enhance accuracy and precision of vehicle object detection in low-light nighttime conditions.	Cycle GANs, YOLO v5	BDD, UA-DETRAC, custom nighttime vehicle images	Increased area under PR curve by 10.4% and peak F1 score by 9%; improved nighttime detection accuracy without additional labeling.	Model performance sensitive to training image content, issues with high contrast and mixed color distribution in certain translated images.

3. Available Traffic Datasets Contains Emergency Vehicles

This section analyzes various datasets containing images of emergency vehicles, specifically, ambulances, fire trucks, and police vehicles. We assess the datasets based on the type, quantity, and quality of images, emphasizing their realism and the fact that they were captured from real-world scenarios. Furthermore, we investigate the image augmentation techniques used in these datasets, how researchers used them, and for what purposes. We aim to identify the limitations and shortcomings in the available datasets. Moreover, we validate our findings by testing the previously proposed models in this field on real-world images to determine their effectiveness.

3.1. Emergency Vehicle Detection

The "emergency vehicle detection" dataset [13] from Roboflow [31] contains 365 training images and 158 testing and validation images with a medium quality of 640 × 640 pixels. However, all the images in the training set are of ambulance vehicles, and there are no other emergency vehicles, such as police or fire truck vehicles. It is an unbalanced dataset; only a few images contain ambulance vehicles. Additionally, no image augmentation techniques were applied. Dissanayake et al. [32] used this dataset by the Yolo3 detection algorithm. The dataset was divided into 80% training and 20% testing and validation, aiming to detect emergency vehicles upon their arrival at the traffic light and give them a higher priority to pass through the signalized intersection. This study obtained an accuracy of 82%. After testing the online model available on Roboflow [13] with several realistic images of emergency vehicles, it was observed that the model's detection performance was poor. This suggests that the model is not trained to detect many types of emergency vehicles outside its limited dataset.

3.2. "JanataHack_AV_ComputerVision" and "Emergency vs. Non-Emergency Vehicle Classification"

The second dataset, found in multiple locations on Kaggle "JanataHack_AV_ComputerVision" [11], "Emergency vs. Non-Emergency Vehicle Classification" [33], contains approximately 3300 images, including 1000 emergency vehicles such as ambulances, fire trucks, and police vehicles. It also contains around 1300 images of other vehicles, making it a nearly balanced dataset. However, the images are of poor quality, with a resolution of 224 × 224 pixels. Kherraki and Ouazzani [34] used this dataset for emergency vehicle classification, achieving over 90% accuracy. Still, the primary issue remains the quality of the images and the limited use of data augmentation.

3.3. Ambulance Regression

The "Ambulance Regression" dataset [35] contains 307 images of ambulances in the YOLOv8 format, with 294 training images and 13 testing images. This dataset applies only standard augmentation techniques such as rotation, cropping, and brightness adjustment. However, the dataset lacks real road images, and there are very few test images.

3.4. Ambulans

The "Ambulans" dataset [36] contains 2134 images of ambulances in YOLOv8 format and uses only standard augmentation techniques, including rotation, cropping, brightness adjustment, exposure adjustment, and Gaussian blur. However, there is a problem with the low quality of some images due to augmentation. In addition, many of the images are not from real roads but from exhibitions or the Internet.

3.5. Ambulance_detect

The "ambulance_detect" dataset [37] contains 1400 images of ambulances in the YOLOv8 format without any augmentation techniques applied. The dataset contains 1400 images divided into 980 training images, 140 test images, and 280 validation images. However, the main challenge is the lack of real images of roads in the dataset and the fact that some images in the dataset are from car exhibitions. In contrast, other images were taken from real-world scenarios. In addition, the number of test images is relatively small compared to the number of training images, which may affect the model's ability to generalize.

3.6. Emergency Vehicle Detection

The "Emergency Vehicle Detection" dataset [38] contains 1680 vehicle images in the YOLOv8 format, as it relies only on applying standard augmentation techniques such as horizontal flip, random cropping, and salt noise. The dataset is divided into 1470 training images, 71 test images, and 139 validation images. The challenge here is that the images do not focus solely on emergency vehicles for training purposes. Most of the images in

the “Emergency Vehicle Detection” dataset were captured using video cameras placed in specific locations. This may limit the ability of the dataset to train the model to detect vehicle emergencies in different locations. In addition, some of the captured images may not show the detailed features of emergency vehicles due to the cameras being located at a far distance, which creates challenges for the model in classifying and detecting emergency vehicles accurately.

3.7. FALCK

The "FALCK" dataset [12] contains 176 images of ambulances and firefighting vehicles in the YOLOv8 format, with no augmentation techniques applied. The dataset has 140 training images, one testing image, and 35 validation images. The images are of good quality, but there is a lack of real road images, and the number of test images is too small for training images.

3.8. Sirens

Similarly, the “Sirens” dataset [39] comprises 213 medium-quality images of ambulances, firefighting, and police vehicles in the YOLOv8 format, including 145 training images, 22 testing images, and 44 validation images. However, this dataset is very small, which limits its ability to build high-accuracy and realistic detection models. There is also a lack of real-world road images in this dataset.

3.9. Smart Car

The “Smart car” dataset [40] was designed for detecting emergency vehicles, including ambulances, firefighting, and police vehicles. It includes 1152 images pre-processed with auto-orientation and resizing to 640 $\times$ 640 (stretch), with no augmentation techniques. The dataset is split into 921 training and 231 testing images, with medium-quality images. However, the dataset lacks real road images, and some images are unrealistic. Moreover, most of the images were not taken on the road. Furthermore, the dataset contains no images of vehicles from other classes.

The datasets we reviewed exhibit various limitations and inadequacies.

1. Many available datasets lack realism. They comprise images not obtained from real-world scenarios. This may affect the models’ ability to generalize to practical situations.
2. Available datasets often suffer from a class imbalance. Some types of emergency vehicles have a disproportionate number of images compared to others. Consequently, the models’ performance may be biased toward certain classes and suboptimal for others.
3. Most datasets have limited test data, making it challenging to assess the models’ performance accurately. Poor test data quality also makes developing models that work effectively in real-world scenarios difficult.
4. Some datasets have few images and lack comprehensive data augmentation techniques, hindering the model’s generalization of different scenarios. Excessive augmentation techniques may also reduce image quality.
5. The limited usage of these datasets in published research suggests that they are not widely recognized or effective for ambulance detection.

Table 3 summarizes the main findings from the datasets examined. After defining the main limitations in the available datasets, to address them, we explored the potential of Generative Adversarial Networks (GANs) for augmentation as discussed in the next section.

Table 3. Summary of the most important findings from the available datasets.

Dataset Name	Number of Images	Quality	Vehicle Types	Augmentation	Limitations	Sample
Emergency vehicle detection [13]	523 Images 365 (train) 158 (test) and (validate)	640 × 640 Pixels	Only ambulances	None	Unbalanced dataset, cannot detect emergency vehicles outside of the training dataset.	
JanataHack AV ComputerVi-sion [11] Emergency vs. Non-Emergency Vehicle Classification [33]	Approx. 3300 Images	Poor (224 × 224 Pixels)	Ambulances, fire, police, and other vehicles	Limited augmentation	Low-quality images, limited augmentation	
Ambulance Regression [35]	333 Images 307 (train) 26 (test) and (validate)	Medium (416 × 416 Pixels)	Only ambulances	rotation, shear, brightness	Lack of real road images, too few test images for training images	
Ambulance [36]	2134 Images	Medium (416 × 416 Pixels)	Only ambulances	Rotation, Shear, Brightness, Exposure, Gaussian	Some images may be low quality due to augmentation, lack of test data	
Ambulance detection [37]	1400 Images 980 (train) 420 (test) and (validate)	Medium (640 × 640)	Only ambulances	None	Lack of real road images, too few test images for training images	
Emergency Vehicle Detection [38]	1680 Images 1470 (train) 210 (test) and (validate)	Medium (416 × 416)	Cars (includes emergency vehicles)	flip, shear, Salt and pepper noise	Does not contain images of only emergency vehicles for training	
FALCK [12]	176 Images 140 (train) 36 (test) and (validate)	High (1280 × 720)	Ambulances and firetruck Vehicles	None	Lack of real road images, too few test images for training images	
Sirens [39]	213 Images 145 (train) 66 (test) and (validate)	Medium (416 × 416 Pixels)	Ambulances, firetrucks, and police Vehicles	None	Lack of real road images, small dataset images	
Smart car [40]	1152 Images 921 (train) 231 (test)	Medium (640 × 640 Pixels)	Ambulances, firetrucks, and police Vehicles	None	unrealistic images, Lack of real road images, small dataset images	

4. Generating a Newly Balanced Dataset

In this section, we gather images from available datasets and augment them with realistic images captured from public streets, utilizing Generative Adversarial Networks (GANs) to generate new images that were integrated into the existing dataset to enhance its diversity and balance to address the limitations imposed on previous datasets and improve the model's ability to detect emergency vehicles accurately. Therefore, providing a large and balanced dataset that includes realistic images of all categories of emergency vehicles, such as ambulances, fire engines, police vehicles, and other vehicles, ensures training an effective and accurate model for emergency vehicle detection.

4.1. Gathering a Dataset for Emergency Vehicle Detection

Building a new dataset for emergency vehicle detection requires going through the following steps:

4.1.1. Initial Data Collection

In the first step, we gathered an initial dataset of emergency and non-emergency vehicle images by identifying relevant sources and collecting representative images. We obtained about 500 images of emergency vehicles from previously available datasets [35,36,39,40] and another approximately 500 images of non-emergency vehicles from the available dataset [41]. This enabled us to curate a diverse and realistic set of images for training our model.

Moreover, we conducted a more extensive search for datasets containing at least one class of emergency vehicles. We came across the Firetruck Dataset [42] and the Police Cars Dataset [43]. The latter two datasets helped us with augmentation to expand our dataset by around 700 additional images. Thus, the total number of images became approximately 1700.

4.1.2. Image Collection through Video Recording and Frame Selection

Here, we utilized our cameras to capture additional images. We conducted multiple tours across Jordan, specifically in Irbid, and recorded videos of emergency and non-emergency vehicles on public streets. These videos were converted into individual frames, and we selected appropriate frames that captured the vehicles at various stages. This added approximately 2900 images of emergency and non-emergency vehicles to our dataset, increasing the total number of images to more than 4600.

To further expand our dataset, we searched for publicly available videos on YouTube featuring emergency vehicles, such as the ones in [44]. We downloaded these videos and extracted individual frames, carefully selecting the ones that captured emergency vehicles. We obtained 6000 images of both emergency and non-emergency vehicles from this step. There were 3200 images of emergency vehicles, and the remainder are non-emergency vehicles. Figure 2 displays a sample of the images obtained through this step.



Figure 2. Samples images collected during video recording and frame selection.

4.1.3. Dataset Pre-processing

To prepare the dataset for analysis, several pre-processing steps are required. The first step involves resizing the images to a uniform size of 416 × 416 pixels. After that, the pixel values of the images need to be scaled down to the range between 0 and 1. These steps are crucial in standardizing the data and preparing them for analysis. Removing fake images from the dataset is important, as they can adversely affect the results. Removing duplicates is also necessary to avoid redundancy and ensure each data point is unique. These pre-processing steps are essential for cleaning and standardizing the dataset for further analysis.

4.1.4. Dataset Augmentation

Dataset augmentation aims to expand and increase the dataset size using a data generator from TensorFlow [45]. This tool generates new variations of data. In our model, an image data generator takes existing images as input and applies random transformations to generate diverse images. Transformations include rotation, shifting, shearing, zooming, and flipping. A callback function saves newly generated images if validation accuracy exceeds 0.7, ensuring high-quality augmented images. This enhances the detection model's performance during training. The dataset grew from around 6000 to 18,000 images by applying these techniques, boosting training data quantity and diversity.

5. Generating New Images Using GANs

Generative Adversarial Networks (GANs) comprise a generator and a discriminator, making them adept at generating images resembling real ones [46]. The generator produces synthetic data akin to real data, while the discriminator discerns between real and synthetic data [47]. Deep Convolutional Generative Adversarial Network (DCGAN) is a type of GAN that employs CNNs in both the generator and discriminator networks, allowing it to capture spatial dependencies in images and generate realistic images [48]. The DCGAN model can help generate new images similar to real ones. The generated images can be added to

the existing dataset, increasing its diversity and size and improving the performance of machine learning models trained on the dataset.

Figure 3 visually represents the image generation process using GANs. In this process, the generator initiates by taking random noise as input and progressively transforms it through multiple layers, ultimately producing the desired image. On the other hand, the discriminator plays a crucial role in assessing the authenticity of the generated image. It takes the generated image as input and passes it through its layers to determine its realism by comparing it to the original dataset.

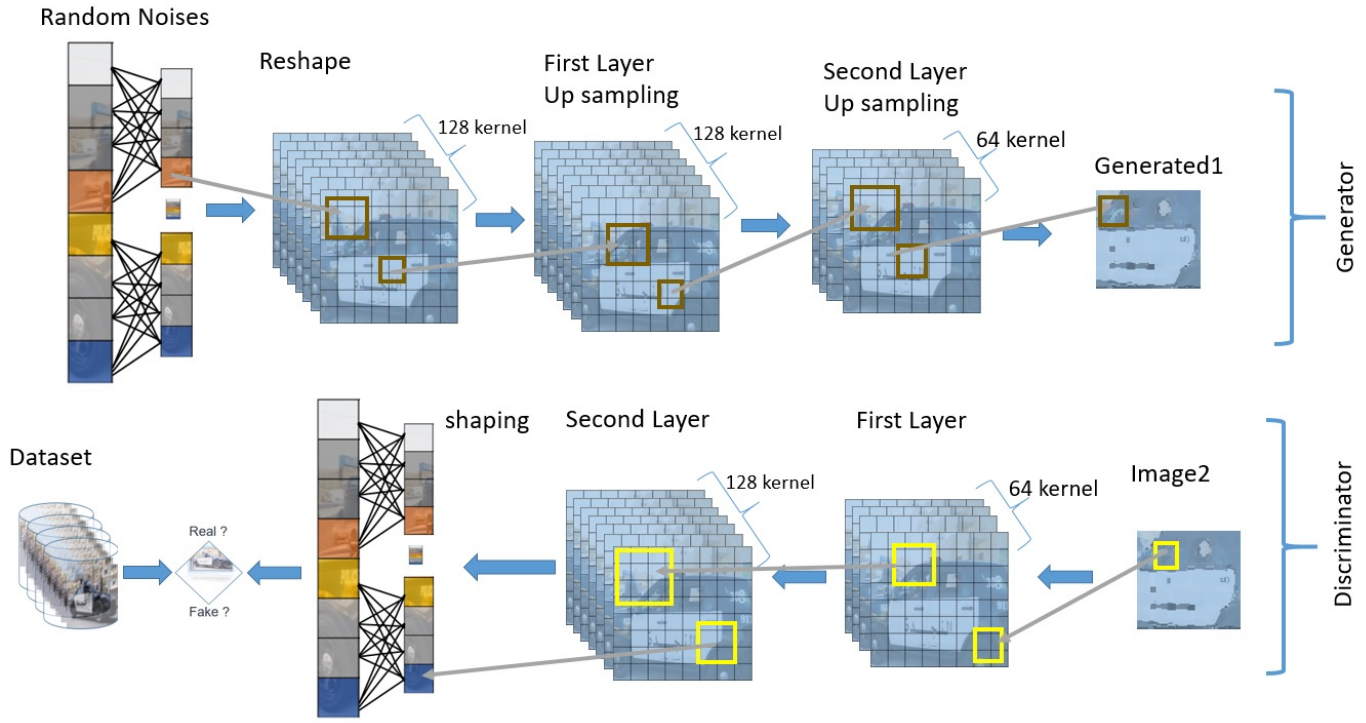


Figure 3. GANs Image Generation Process.

As we discussed earlier, the GANs model consists of two networks (i.e., generator and discriminator). Figure 4 illustrates the general architecture of the designed GANs in our work. The exact architectures of each included network are presented in this section. First, the generator network contains four hidden layers and one output layer as shown in Figure 5. These layers are explained in detail here:

- Hidden Layer 1: The input to the generator is a random noise vector of size latent_dim. This layer has n_nodes nodes, calculated as $16 \times 16 \times 128$. The reshape layer is used to reshape the output of this layer into a 4D tensor of shape (16, 16, 128).
- Hidden Layer 2: This layer uses a transposed convolutional layer (Conv2DTranspose) to upsample the input from the previous layer to a size of 52×52 pixels. It has 128 filters with a kernel size of (4, 4) and a stride of (2, 2). The ReLU activation function is used to introduce non-linearity.
- Hidden Layer 3: This layer further upsamples the input to a size of 104×104 pixels. It has 256 filters with a kernel size of (4, 4) and a stride of (4, 4). The ReLU activation function is again used.
- Hidden Layer 4: This layer upsamples the input to a size of 416×416 pixels. It has 512 filters with a kernel size of (4, 4) and a stride of (2, 2). The ReLU activation function is used.
- Output Layer: The final output layer uses a convolutional layer (Conv2D) with three filters (for the three color channels in the image) and a kernel size of (5, 5). The tanh activation function generates pixel values in the range $[-1, 1]$.

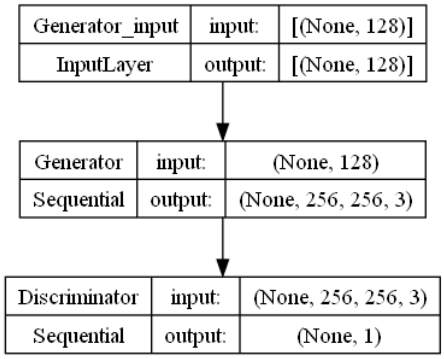


Figure 4. The architecture of the designed GANs.

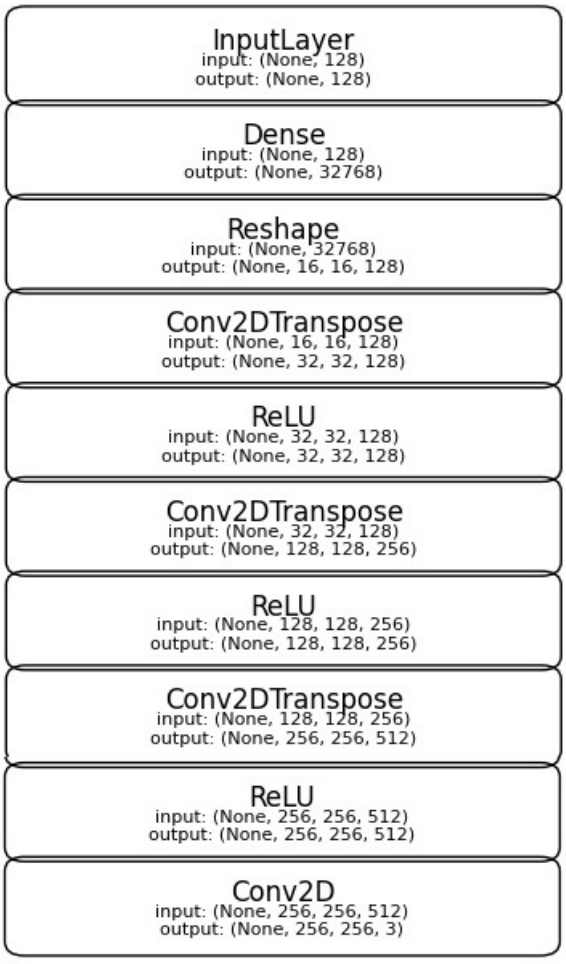


Figure 5. Generator architecture.

On the other hand, the discriminator network has three hidden layers and one output layer as shown in Figure 6. The details of these layers are explained here:

- **Hidden Layer 1**: This layer uses a convolutional layer (Conv2D) with 64 filters and a kernel size (4, 4). It has a stride of (2, 2) and uses the LeakyReLU activation function with a negative slope of 0.2.
- **Hidden Layer 2**: This layer uses a convolutional layer (Conv2D) with 128 filters and a kernel size (4, 4). It has a stride of (4, 4) and uses the LeakyReLU activation function with a negative slope of 0.2.

- **Hidden Layer 3**: This layer uses a convolutional layer (Conv2D) with 128 filters and a kernel size (4, 4). It has a stride of (2, 2) and uses the LeakyReLU activation function with a negative slope of 0.2.
- **Flatten and Output Layer**: This layer flattens the output from the previous layer into a 1D tensor and applies a dropout layer to drop some connections for better random generalization. The final output layer has a single node with a sigmoid activation function that outputs a probability between 0 and 1, indicating whether the input image is real or fake. The discriminator model is trained to classify the input images as real or fake, so the loss function used during training is binary cross-entropy.

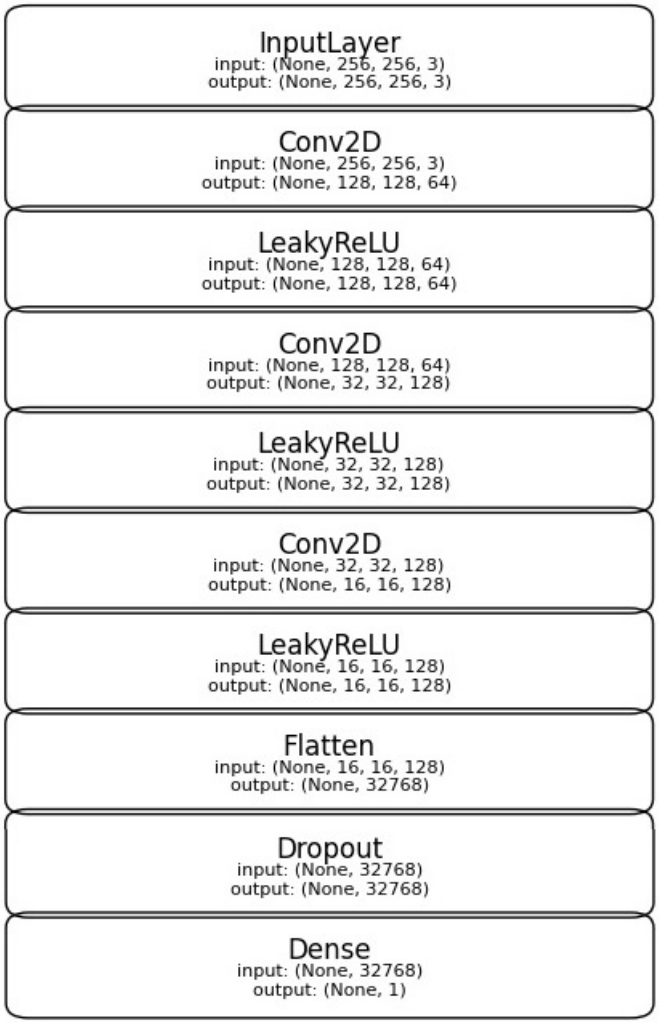


Figure 6. Discriminator architecture.

The key hyperparameters to be adjusted in the designed GANs model for training and generating the desired images are latent_dim, learning rate, and batch size. The latent_dim hyperparameter is specifically related to the generator model. It determines the size of its input and is set to 128 based on preliminary experiments and the literature, suggesting it provides a good balance between the diversity and quality of generated images [49]. Our model explicitly set the learning rate hyperparameter to 0.0002 for the Adam (Adaptive Moment Estimation) optimizer [50] used by the discriminator model. The Adam optimizer is a popular optimization algorithm that adapts the learning rate during training. By setting the learning rate to 0.0002, informed by its widespread use in stabilizing GAN training and providing effective convergence rates, we can control the step size used by the Adam optimizer to update the model's parameters. Additionally, the batch size was set to 32, which defines the number of samples processed together in each iteration during train-

ing, considering that smaller batch sizes can help reduce overfitting while maintaining manageable memory usage [51]. Moreover, the models were trained for 20,000 epochs, indicating the complete passes through the training dataset to ensure thorough learning without overfitting. The dataset used in the experiment consisted of 1000 images with a shape of (256, 256, 3), and the pixel values were scaled to be between −1 and 1. As for selecting additional parameters such as beta_1 (set to 0.5) and dropout_rate (0.3), as well as filters, kernel_sizes, and strides, they were determined based on their proven effectiveness to improve GAN performance in image generation [52]. Table 4 illustrates the parameters in the designed GANs model.

Table 4. GANs model parameters.

Parameter	Value
latent_dim	128
epochs	20,000
batch	32
learning rate	0.0002
beta_1	0.5
dropout_rate	0.3
filters	[64, 128, 128]
kernel_sizes	[(4, 4), (4, 4), (4, 4)]
strides	[(2, 2), (4, 4), (2, 2)]

The DCGAN model was trained using an adversarial approach, where the generator generates images that deceive the discriminator while the discriminator learns to differentiate between real and fake images. During training, we employed binary cross-entropy loss as the loss function. We recorded each epoch's discriminator and generator loss to monitor the training progress. The discriminator loss indicates the accuracy of the discriminator in classifying real and fake images, while the generator loss measures how well the generator can deceive the discriminator. Additionally, after each training epoch, we evaluated the performance of the discriminator on both real and generated images. The discriminator accuracy on real images indicates its ability to distinguish between real and fake images. In contrast, the discriminator accuracy on generated images shows how well the generator can deceive the discriminator. The experimental study utilized a DCGAN model with 7,579,332 parameters out of 7,149,955 that were trainable. The discriminator and generator components had 429,377 and 7,149,955 trainable parameters, respectively. The model's performance was evaluated during the training process after 300 epochs.

The results showed that as the model continued to train, the discriminator and generator losses decreased, while the discriminator's accuracy on real and fake images increased. For instance, at epoch 300, the generator loss was 1.749, and the discriminator accuracy on real and fake images was 0.4 and 0.8, respectively. However, after 12,000 epochs, the generator loss decreased to approximately 1, and the discriminator accuracy on real and fake images significantly improved to 0.94. The outcomes of the model demonstrated the effectiveness of the DCGAN architecture in generating new images of emergency vehicles. The model's performance gradually improved throughout the training process, although with some degree of fluctuation in the outcomes. A set of sample images generated during various epochs of the training is displayed in Figure 7. And Figure 8 displays a sample of images illustrating the progression of image enhancement produced by GANs as the number of epochs.

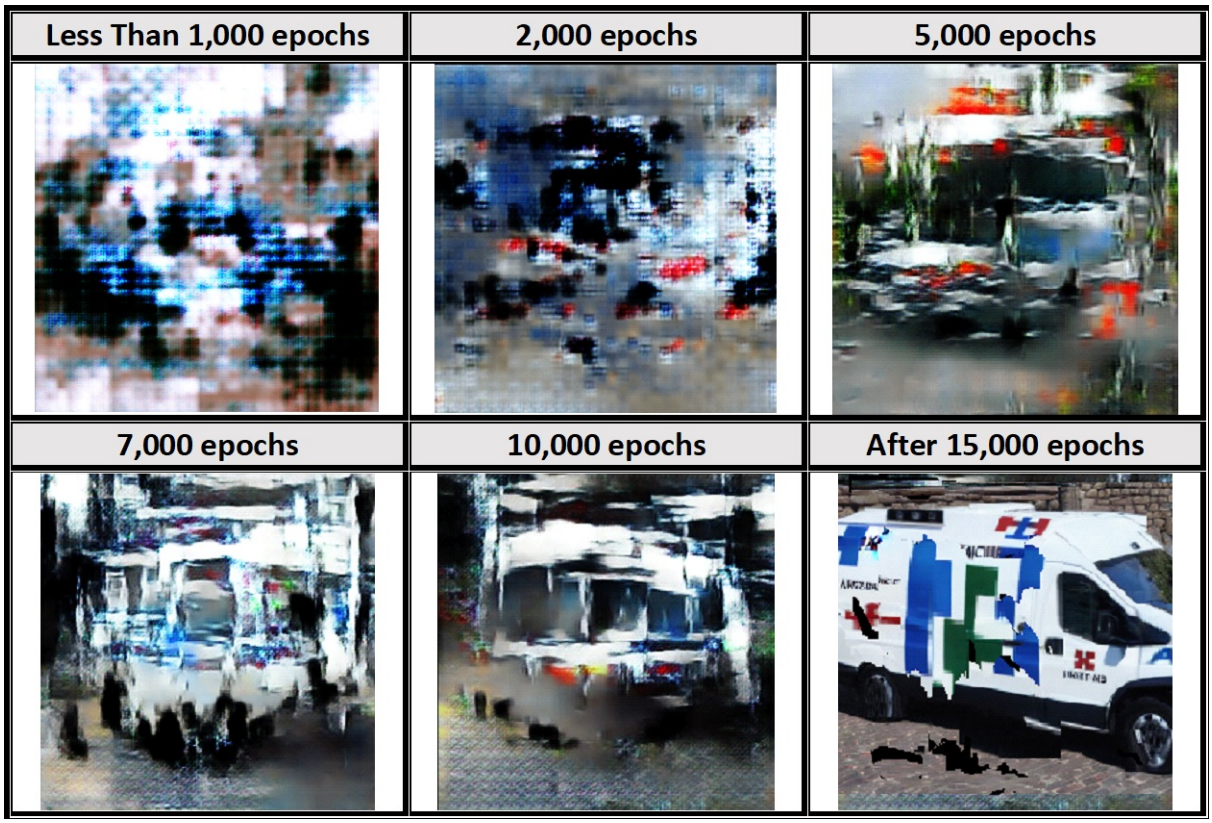


Figure 7. Samples images generated during various epochs.



Figure 8. Image enhancement progression with increasing GANs epochs.

However, despite producing many images during the training, the model still requires refinement, particularly when dealing with the existing dataset. Additionally, further improvement in the results is anticipated through continuous training and increased epochs, which could necessitate using a supercomputer or cloud computing to accelerate and enhance the training process. At the end of this section, a new dataset containing approximately 20,000 images was obtained. The new dataset is well balanced, comprising 10,000 images of emergency vehicles and 10,000 images of non-emergency vehicles. Most of the images in the dataset were taken from real-life scenarios and can be detected over real road scenarios.

6. Performance Evaluation: Test and Validate the New Dataset

As previously mentioned, the lack of a model with accurate detection capabilities for emergency vehicles can be attributed to the characteristics and realism of the dataset used. After obtaining a new dataset with approximately 20,000 images, we aimed to test and validate the dataset and develop an effective model for emergency vehicle detection. The first step involved developing an emergency vehicle detection model using the Convolutional Neural Network (CNN) algorithm. Then, the 10-fold cross-validation process was applied to validate the performance of the proposed detection model. Finally, we used a small external dataset collected using our smartphones to compare the performance of our proposed model with previously developed and online available emergency detection models. We employed several standard evaluation metrics to measure the performance of our emergency vehicle detection system. These metrics include accuracy, precision, recall, and F1-score [53]. The definitions of these metrics, their meanings, and how to compute each metric are illustrated below:

- **Accuracy** It measures the overall correctness of the system's predictions by calculating the ratio of correctly classified instances to the total instances:

$$Accuracy = (True\ Positives + True\ Negatives)/Total\ number\ of\ predictions \quad (1)$$

- **Precision** It quantifies the proportion of correctly predicted emergency vehicle instances out of all those predicted as emergency vehicles:

$$Precision = (Total\ number\ of\ True\ Positives)/(True\ Positives + False\ Positives) \quad (2)$$

- **Recall** It measures the proportion of correctly predicted emergency vehicle instances out of all the actual emergency vehicle instances:

$$Recall = (Total\ number\ of\ True\ Positives)/(True\ Positives + False\ Negatives) \quad (3)$$

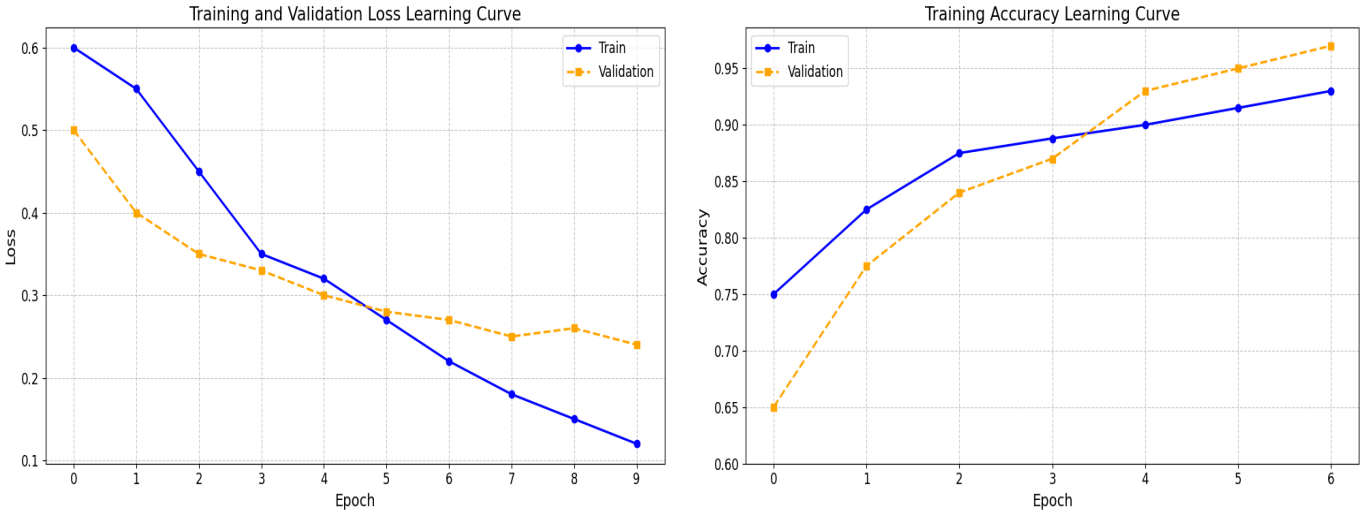
- **F1-score** It provides a balanced measure of precision and recall, considering both metrics to evaluate the system's performance:

$$F1\text{-}score = (2 * (Precision * Recall))/(Precision + Recall) \quad (4)$$

6.1. Emergency Vehicle Detection Model Using CNN

The detection model utilized in this study employed the Convolutional Neural Network (CNN) algorithm, a highly effective model for deep learning and particularly suitable for tasks involving image classification. The classification process focused on distinguishing between emergency and non-emergency vehicle images, and it was conducted in two stages. The initial stage involved 6000 images before augmentation operations, with 3200 images for emergency vehicles and 2800 non-emergency vehicles. Subsequently, the second stage incorporated 20,000 images after augmentation. The CNN model was constructed using the TensorFlow-Keras framework. The model architecture encompassed various layers, including convolutional, max-pooling, flattening, and fully connected layers. The key parameters included a kernel size of (3, 3) for the four convolutional layers, a pool size

of (2, 2) for the max-pooling layers, and a dropout rate of 0.5 in the first dense layer. The training involved 10 epochs and a batch size of 32. The initial dataset comprised 6000 images collected from the previous datasets and taken by our camera, with no augmentation applied. It achieved an accuracy of 90.3%, precision of 91.1%, recall of 90.0%, and the F1-score reached 90.5%. The evaluation results for the training, validation, and testing datasets are presented in Table 5. In contrast, the confusion matrix for the testing dataset is shown in Table 6. Figure 9a graphically illustrates the training accuracy for the initial dataset across different epochs, whereas Figure 9b depicts the learning results for the initial dataset during training and validation.



(**a**) (**b**)

Figure 9. Training and learning results on the initial dataset. (**a**) The training learning curve of the initial dataset.; (**b**) the training accuracy learning curve of the initial dataset.

Table 5. CNN results on the initial dataset.

Metrics	Training	Validations	Testing
Accuracy	95.2%	91.0%	90.3%
Precision	94.9%	89.3%	91.1%
Recall	95.4%	93.8%	90.0%
F1-score	95.1%	91.5%	90.5%

Table 6. CNN on initial dataset training confusion matrix.

		Actual	
		Emergency	**Non-Emergency**
Predicted	**Emergency**	279	27
	Non-Emergency	31	263

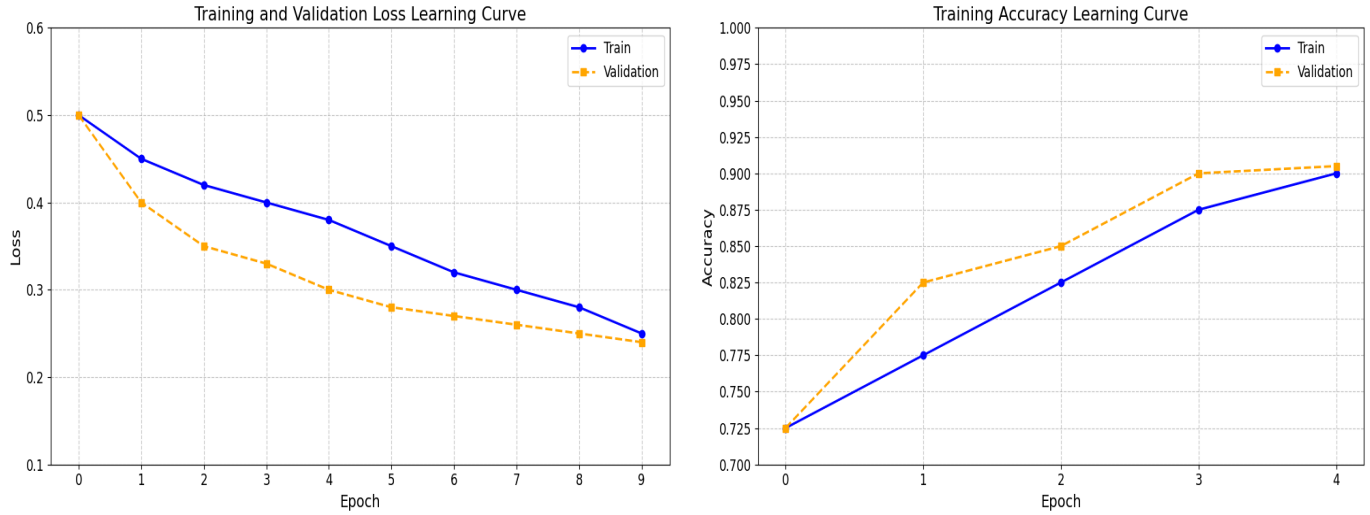
The final obtained dataset that comprises 20,000 images was employed, including adding images generated using Generative Adversarial Networks (GANs). Each category consisted of 10,000 images. The results showed notable improvement, with accuracy exceeding 90.9%, precision 93.0%, recall 88.2%, and the F1-score reaching 90.5%. The evaluation results for the training, validation, and testing datasets are presented in Table 7. In contrast, Table 8 shows the confusion matrix for the testing dataset. Figure 10a illustrates the training accuracy results for the final dataset across different epochs, whereas Figure 10b depicts the learning results for the final dataset during training and validation.

Table 7. CNN results on final dataset.

Metrics	Training	Validations	Testing
Accuracy	93.7%	91.2%	90.9%
Precision	96.5%	94.2%	93.0%
Recall	90.4%	88.0%	88.2%
F1-score	93.4%	91.0%	90.5%

Table 8. CNN on Final Dataset training Confusion matrix.

		Actual	
		Emergency	**Non-Emergency**
Predicted	**Emergency**	950	116
	Non-Emergency	65	869



(**a**) (**b**)

Figure 10. Training and Learning Results on Final Dataset. (**a**) The Training Learning Curve of Final Dataset. (**b**) The Training Accuracy Learning Curve of Final Dataset.

Figure 11 graphically illustrates comparative results of the emergency vehicle detection model using the initial and final datasets. The model significantly improves when the number of images is increased from 6000 to 20,000 as indicated by the higher accuracy, precision, and F1-score. However, the recall for detecting negative cases is slightly decreased (90.0% to 88.2%). This can be attributed to the increased diversity of negative cases in the augmented dataset, challenging the model to identify a wider range of non-emergency vehicle instances.

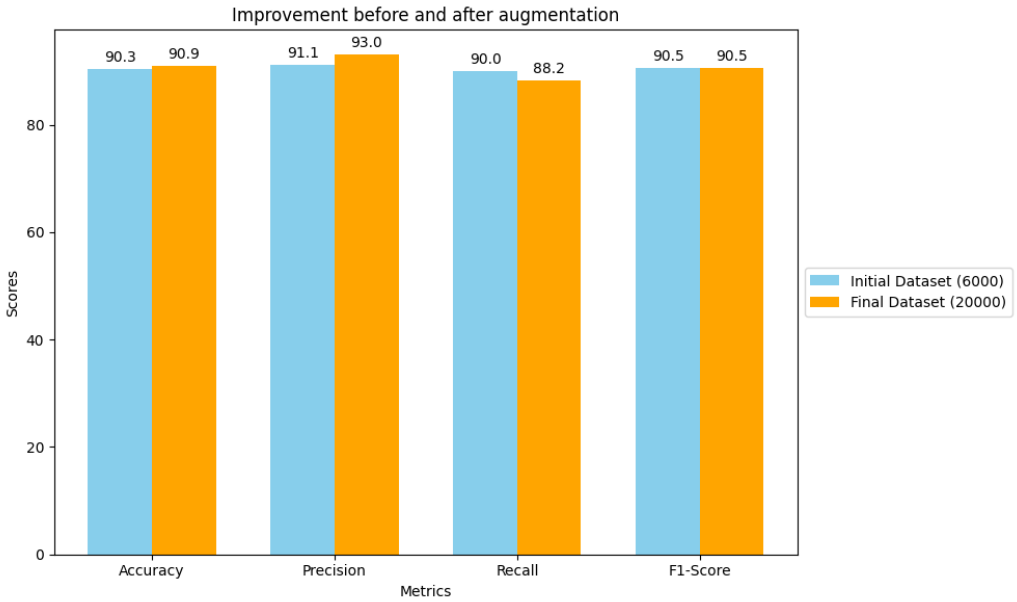


Figure 11. Performance evaluation of the detection model.

6.2. 10-Fold Cross-Validation

After classifying the model using the CNN algorithm, the dataset had to be evaluated to validate and ensure the new balanced dataset's performance. The 10-fold cross-validation technique was used on the two datasets: 6000 images in the initial dataset and 20,000 images in the final dataset. In the initial dataset of 6000 images before augmentation, the 10-fold cross-validation was applied to evaluate the model's performance. The results showed an average accuracy of 83.32%, precision of 83.44%, recall of 79.82%, and F1-score of 80.99%. Table 9 displays the results of all folds.

Table 9. The 10-fold cross-validation results on the initial dataset.

Eval. Metrics	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean
Accuracy	83.50%	85.29%	84.10%	81.91%	85.69%	83.10%	82.70%	81.51%	82.70%	82.70%	83.32%
Precision	83.36%	86.19%	83.79%	82.87%	85.56%	82.40%	82.71%	82.23%	83.96%	81.32%	83.44%
Recall	80.44%	82.00%	81.73%	79.42%	82.50%	78.65%	78.61%	76.75%	78.61%	79.46%	79.82%
F1-score	81.48%	83.37%	82.52%	80.34%	83.66%	80.00%	79.96%	78.28%	80.10%	80.23%	80.99%

The final dataset, consisting of 20,000 images (10,000 for each category), significantly improved the model's performance. Accuracy clearly improved, indicating the better identification of positive cases. While the recall is increased and more positive cases are captured, further improvements are possible. In general, the F1-score increased convergently for accuracy and recall, reflecting a well-balanced performance between accuracy and recall. Reinforcement and balancing positively affected the ability of the model to detect both positive and negative states. Table 10 presents the results of the 10-fold cross-validation in the final dataset. Figure 12 illustrates the improvement before and after augmentation.

Table 10. The 10-fold cross-validation results on the final dataset.

Eval. Metrics	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean
Accuracy	88.50%	88.90%	87.00%	89.10%	87.70%	87.29%	85.20%	87.30%	87.20%	87.90%	87.61%
Precision	89.20%	90.10%	88.15%	87.10%	89.91%	88.88%	88.05%	86.45%	88.63%	86.90%	88.34%
Recall	86.10%	87.98%	85.58%	84.21%	87.00%	85.90%	84.41%	83.31%	85.25%	84.36%	85.41%
F1-score	87.96%	88.69%	86.25%	86.68%	85.58%	88.58%	86.36%	86.84%	85.25%	86.20%	86.84%

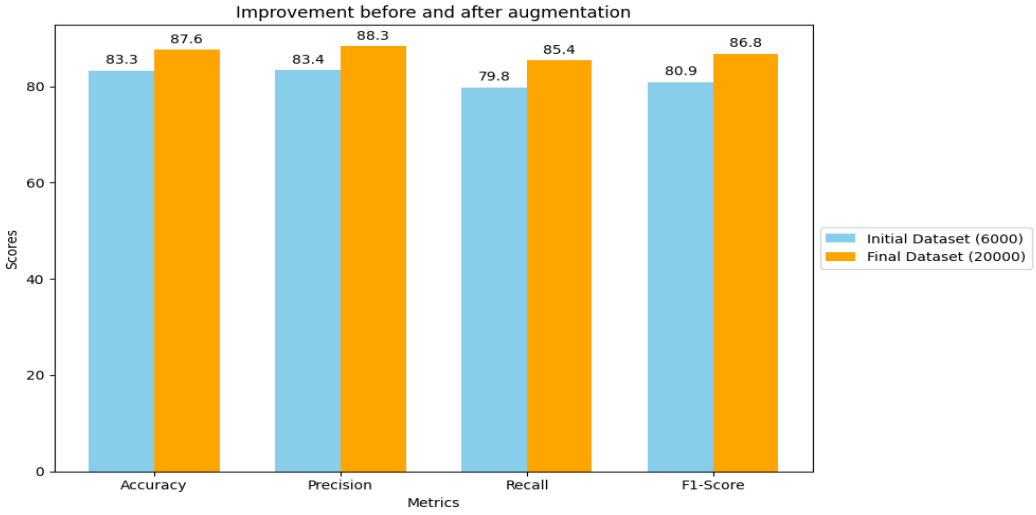


Figure 12. Improvement before and after augmentation.

6.3. Comparative Performance Study

In this section, we first test the performance of our developed model on an external dataset. Then, we aim to compare the performance of our proposed emergency vehicle detection model to previous models in this field. An external dataset of 200 images was used to test and compare the models' performance to ensure unbiased evaluation. The tested dataset was not included in the training process. Our developed model achieved an accuracy of over 85% on the external dataset, which is considered satisfactory. After evaluating the performance of our new model and confirming its effectiveness in detecting emergency vehicles, we conducted a brief comparison with some other available online models for detecting emergency vehicles [35,54–56]. This comparison was carried out using the same external dataset consisting of 200 images that were not included in the training set of our model. The objective was to assess how our model performed compared to these existing models.

The results of this comparison are presented in Table 11. The latter table indicates that our new model outperformed the other models in terms of accuracy and overall performance. It demonstrated superior abilities to detect emergency vehicles when presented with unseen images. It is essential to highlight that the primary obstacles faced by existing models stem from the quality and sufficiency of the datasets employed during their training. Our newly proposed model stands out, as it was trained on a more extensive and realistic dataset, contributing to its superior emergency vehicle detection performance.

Table 11. Emergency vehicle detection model comparison.

Model	FP	FN	TP	TN	Accuracy	Precision	Recall	F1-Score
Our Proposed Model	11	18	89	82	85.5%	89.0%	83.2%	86.0%
Emergency Vehicles Sans Domain Regression	18	27	82	73	77.5%	82.0%	75.2%	78.5%
Emergency Vehicles Computer Vision Project	14	49	86	51	68.5%	86.0%	63.7%	73.2%
Ambulance Regression	34	28	66	72	69.0%	66.0%	70.2%	68.0%
AI-Emergency Project	22	52	78	48	63.0%	78.0%	60.0%	67.8%

Figure 13 graphically compares the performance results between our new model for detecting emergency vehicles and the previously discussed models. As we infer from the figure, our proposed model performed better than previous models in this field in all the measured metrics. This means that the generated dataset has intensively improved the performance of the detection mechanisms.

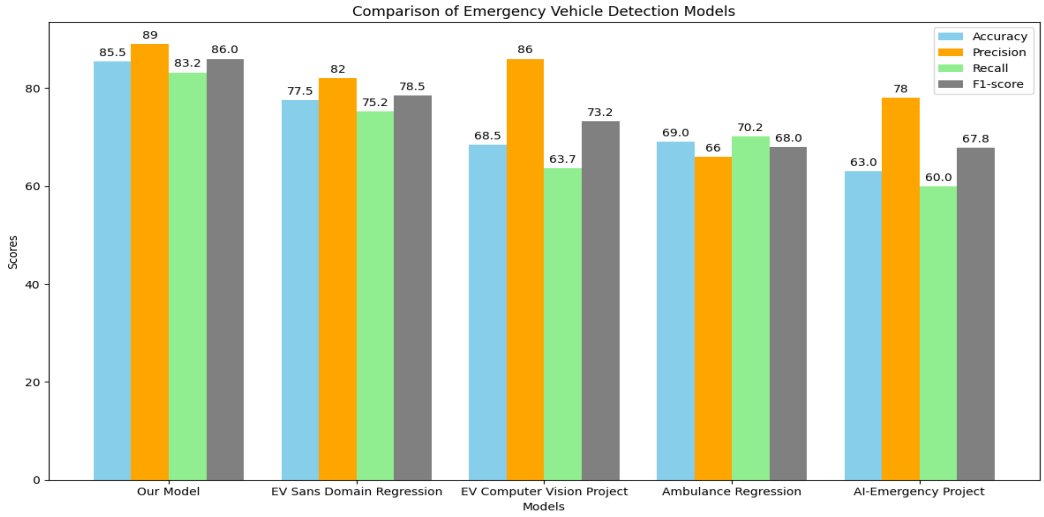


Figure 13. Comparison of emergency vehicle detection models.

7. Conclusions

In conclusion, the study provided a comprehensive overview of the necessity of having a realistic model capable of detecting emergency vehicles. We started with the main challenges in the current datasets and proposed a solution by creating a new dataset using several stages, the most important of which was the use of GANs, and we ultimately collected a new dataset consisting of 20,000 images. The new dataset was verified using 10-fold cross-validation, where a new model was built using the new dataset for emergency vehicle detection based on CNNs. The proposed model achieved a high accuracy of about 86%, outperforming several existing models with which it was compared, proving the dataset's realism and the possibility of using it for practical purposes such as traffic management. We believe that our study with the new dataset provides a valuable contribution to computer vision, confirms the effectiveness of using GANs to generate realistic images, and can inspire future research to improve object detection models' accuracy. With the new dataset generated, future research will be easier to use in other areas, such as determining the actual location of emergency vehicles or linking the model with traffic lights to open or close roads for emergency vehicles.

Author Contributions: Conceptualization, M.B.Y.; methodology, M.B.Y.; software, M.S.; validation, M.S. and M.B.Y.; formal analysis, M.S.; investigation, M.S. and M.B.Y.; resources, M.S. and M.B.Y.; data curation, M.S. and M.B.Y.; writing—original draft preparation, M.S.; writing—review and editing, M.B.Y.; visualization, M.S.; supervision, M.B.Y.; project administration, M.B.Y.; funding acquisition, M.B.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset supporting the findings of this study is openly available in the GitHub repository at https://github.com/Shatnawi-Moath/EMERGENCY-VEHICLES-ON-ROAD-NETWORKS-A-NOVEL-GENERATED-DATASET-USING-GANs.

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

References

1. Campisi, T.; Severino, A.; Al-Rashid, M.A.; Pau, G. The development of the smart cities in the connected and autonomous vehicles (CAVs) era: From mobility patterns to scaling in cities. *Infrastructures* **2021**, *6*, 100. [CrossRef]
2. Younes, M.B. Real-time traffic distribution prediction protocol (TDPP) for vehicular networks. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 8507–8518. [CrossRef]
3. Younes, B.M.; Boukerche, A.; Rango, F.D. SmartLight: A smart efficient traffic light scheduling algorithm for green road intersections. *Ad Hoc Netw.* **2023**, *140*, 103061. [CrossRef]
4. Eliiyi, U. Artificial Intelligence for Smart Cities: Locational Planning and Dynamic Routing of Emergency Vehicles. In *The Impact of Artificial Intelligence on Governance, Economics, and Finance*; Springer: Singapore, 2022; Volume 2, pp. 41–63.
5. Younes, B.M.; Boukerche, A. An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems. *Wirel. Netw.* **2018**, *24*, 2451–2463. [CrossRef]
6. Younes, B.M.; Boukerche, A. Towards a sustainable highway road-based driving protocol for connected and self-driving vehicles. *IEEE Trans. Sustain. Comput.* **2021**, *7*, 235–247. [CrossRef]
7. Younes, M.B.; Boukerche, A. Traffic efficiency applications over downtown roads: A new challenge for intelligent connected vehicles. *Acm Comput. Surv. (CSUR)* **2020**, *53*, 1–30. [CrossRef]
8. Haque, S.; Sharmin, S.; Deb, K. Emergency Vehicle Detection Using Deep Convolutional Neural Network. In Proceedings of the International Joint Conference on Advances in Computational Intelligence, Vienna, Austria, 17–19 September 2019; Springer: Singapore, 2022; pp. 535–547.
9. Roy, S.; Rahman, M.S. February. Emergency vehicle detection on heavy traffic from CCTV footage using a deep convolutional neural network. In Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), IEEE, Cox's Bazar, Bangladesh, 7–9 February 2019; pp. 1–6
10. Sunitha, P.; Srinath, S. Performance Evaluation of Feature Extraction Algorithms for Vehicle Shape Classification. *U. Porto J. Eng.* **2022**, *8*, 62–75.
11. SHRAVAN KUMAR KONINTI. JanataHack_AV_ComputerVision. Open Source Dataset. Kaggle. 2021. Available online: https://www.kaggle.com/shravankoninti/janatahack-av-computervision (accessed on 10 March 2023).
12. Folio3. FALCK Dataset. Open Source Dataset. Roboflow Universe. Roboflow. 2023. Available online: https://universe.roboflow.com/folio3-krxsh/falck (accessed on 10 March 2023).
13. Project-dbqtw. Emergency Vehicle Detection Dataset. Open Source Dataset. Roboflow Universe. Roboflow. 2023. Available online: https://tinyurl.com/2ne4j3hz (accessed on 10 March 2023).
14. Shung, K.P. "Accuracy, Precision, Recall or F1?" Towards Data Science. 15 March 2018. Available online: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9 (accessed on 27 May 2024).
15. Bin, S.; Li, P.; Ali, R.; Chen, C.L.P. Improving video temporal consistency via broad learning system. *IEEE Trans. Cybern.* **2021**, *52*, 6662–6675.
16. Mo'ath, S.; Younes, M.B. Intelligent Detecting of Emergency Vehicles on the Road Networks: Available Datasets Assessment. In Proceedings of the 2023 International Conference on Information Technology (ICIT), Amman, Jordan, 9–10 August 2023.
17. Tarun, A.; Choudhary, P. Segmentation and classification on chest radiography: A systematic survey. *Vis. Comput.* **2023**, *39*, 875–913.
18. Santos, T.F.H.K.d.; Aranha, C. Data augmentation using GANs. *arXiv* **2019**, arXiv:1904.09135.
19. Kukreja, V.; Kumar, D.; Kaur, A. GAN-based synthetic data augmentation for increased CNN performance in Vehicle Number Plate Recognition. In Proceedings of the 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 5–7 November 2020; pp. 1190–1195.
20. Gao, Y.; Luo, Z.; Yu, X.; Ren, K.; Ye, Y.; Chen, Q. November. Infrared unmanned aerial vehicle detection based on generative adversarial network data augmentation. In *AOPC 2021: Infrared Device and Infrared Technology*; SPIE: San Francisco, CA, USA, 2021; Volume 12061, pp. 209–214.
21. Sathananthavathi, V.; Kandasamy, K.; Rajamanickam, D. Nighttime Vehicle Detection using Improved CycleGAN. *Preprints* **2022**. [CrossRef]

22. Wu, Y.; Wang, T.; Gu, R.; Liu, C.; Xu, B. Nighttime vehicle detection algorithm based on image translation technology 1. *J. Intell. Fuzzy Syst.* **2024**, *46*, 5377–5389. [CrossRef]
23. Xu, B.; Liu, X.; Feng, G.; Liu, C. A monocular-based framework for accurate identification of spatial-temporal distribution of vehicle wheel loads under occlusion scenarios. *Eng. Appl. Artif. Intell.* **2024**, *133*, 107972. [CrossRef]
24. Jiang, N.; Sheng, B.; Li, P.; Lee, T.Y. PhotoHelper: Portrait photographing guidance via deep feature retrieval and fusion. *IEEE Trans. Multimed.* **2022**, *25*, 2226–2238. [CrossRef]
25. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
26. Aswathy, T.G.; Ganesh, D. Emergency and Non-Emergency Vehicle Classification using Machine Learning. *Int. J. Sci. Res. Eng. Manag. (IJSREM)* **2022**, *6*, 1–9. [CrossRef]
27. Punyavathi, G.; Neeladri, M.; Singh, M.K. Vehicle tracking and detection techniques using IoT. *Mater. Today Proc.* **2022**, *51*, 909–913. [CrossRef]
28. Pillai, U.K.; Valles, D. Vehicle Type and Color Classification and Detection for Amber and Silver Alert Emergencies Using Machine Learning. In Proceedings of the 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, USA, 9–12 September 2020; pp. 1–5.
29. Norein, I.H.K. Video Dehazing Based on Preprocessing Contrast Enhancement and Dark Channel Prior. Ph.D. Thesis, University of Gezira, Wad Madani, Sudan, 2022.
30. Seong, P.K.; Cho, S.; Kim, J.; Song, H. Effect of contrast-limited adaptive histogram equalization on deep learning models for classifying bone scans. *J. Nucl. Med.* **2022**, *63*, 3240.
31. Dwyer, B.; Nelson, J.; Solawetz, J.; Warner, J.; Smith, A.; Johnson, M.; Lee, K.; Brown, R.; Clark, L.; Harris, T.; et al. Roboflow (Version 1.0) [Software]. 2024. Available online: https://roboflow.com.computervision (accessed on 24 June 2024).
32. Dissanayake, D.M.; Aluvihare, W.B.; Rajapaksha, K.D. "SmartGo"—Intelligent Traffic Controlling System with Violators Detection. In Proceedings of the 2022 4th International Conference on Advancements in Computing (ICAC), Colombo, Sri Lanka, 9–11 December 2022; pp. 304–309.
33. Parth, C. Emergency vs. Non-Emergency Vehicle Classification", vol. 1, Kaggle. 2020. Available online: https://www.kaggle.com/datasets/parthplc/emergency-vs-nonemergency-vehicle-classification (accessed on 4 March 2023).
34. Kherraki, A.; El Ouazzani, R. Deep convolutional neural networks architecture for an efficient emergency vehicle classification in real-time traffic monitoring. *IAES Int. J. Artif. Intell.* **2022**, *11*, 110. [CrossRef]
35. SipalingAI. Ambulance Regression Dataset. Last updated 6 months ago. Object Detection. Subject: Ambulance. License: CC BY 4.0. Available online: https://universe.roboflow.com/sipalingai/ambulance-regression (accessed on 10 March 2023).
36. SipalingAI.Ambulans Dataset. Object Detection. Available online: https://universe.roboflow.com/sipalingai/ambulans (accessed on 10 March 2023).
37. Hacknjill. Ambulance Detect Dataset. Object Detection. Subject: Ambulances. Available online: https://universe.roboflow.com/hacknjill/ambulance_detect (accessed on 10 March 2023).
38. Binay, J.E. Final Exam Emergency Vehicle Detection Dataset. Last Updated a Year Ago. Object Detection. Subject: Cars. Available online: https://universe.roboflow.com/john-edward-binay/finalexam_emergencyvehicledetection (accessed on 10 March 2023).
39. Martin. Siren Dataset. Last Updated 11 Days Ago. Object Detection. Subject: Siren. Available online: https://universe.roboflow.com/martin-nc8pb/siren (accessed on 10 March 2023).
40. Ali, A. Ambluance Dataset. Last Updated 11 Days Ago. Object Detection. Subject: Smart Car. Available online: https://universe.roboflow.com/ahmed-ali-pz4fk/smart-car-zjpdw (accessed on 10 March 2023).
41. Sapra, V. Object detection Computer Vision Project. [Data Set]. Roboflow Universe. Available online: https://universe.roboflow.com/vishi-sapra/object-detection-axukj (accessed on 30 April 2023).
42. Maleki, P. Firetruck Dataset. Open Source Dataset. Roboflow Universe. Roboflow. 2022. Available online: https://universe.roboflow.com/pouria-maleki/firetruck (accessed on 4 March 2023).
43. FYP TC. Police Cars Dataset. Open Source Dataset. Roboflow Universe. Roboflow. 2022. Available online: https://universe.roboflow.com/fyp-tc-idn2o/police-cars-sumfm (accessed on 4 March 2023).
44. Rainman14. Best of Fire Trucks Responding Compilation 2021—Best of Sirens. Video, 1:25:46. Available online: https://www.youtube.com/watch?v=A1kZUDIEchY&t=2675s&ab_channel=Rainman14 (accessed on 30 April 2023).
45. Sridhar, S.; Sowmya, S. Detection and prognosis evaluation of diabetic retinopathy using ensemble deep convolutional neural networks. In Proceedings of the 2020 International Electronics Symposium (IES), Surabaya, Indonesia, 29–30 September 2020; pp. 78–85.
46. Shan, Z.; Yu, D.; Zhou, Y.; Wu, Y.; Ma, Y. Enhanced visual perception for underwater images based on multistage generative adversarial network. *Vis. Comput.* **2022**, *39*, 5375–5387.
47. Pan, Z.; Yu, W.; Yi, X.; Khan, A.; Yuan, F.; Zheng, Y. Recent progress on generative adversarial networks (GANs): A survey. *IEEE Access* **2019**, *7*, 36322–36333. [CrossRef]
48. Jia, L.; Huang, J.; Li, H. A case study of conditional deep convolutional generative adversarial networks in machine fault diagnosis. *J. Intell. Manuf.* **2021**, *32*, 407–425.
49. Youssef, S.; Jodoin, P.; Lalande, A. Gans for medical image synthesis: An empirical study. *J. Imaging* **2023**, *9*, 69. [CrossRef] [PubMed]

50. Hamza, K.A.; Cao, X.; Li, S.; Katsikis, V.N.; Liao, L. BAS-ADAM: An ADAM based approach to improve the performance of beetle antennae search optimizer. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 461–471.
51. Jiha, K.; Park, H. Limited Discriminator GAN using explainable AI model for overfitting problem. *ICT Express* **2023**, *9*, 241–246.
52. Can, U.; Çolakoğlu, M.B.; Inceoğlu, A. GAN as a generative architectural plan layout tool: A case study for training DCGAN with Palladian Plans and evaluation of DCGAN outputs. *A Z ITU J. Fac. Archit.* **2020**, *17*, 185–198.
53. Reda, Y.; Axman, D. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, Online, 20 November 2020; pp. 79–91.
54. Emergency Vehicles. Emergency Vehicles Sans Domain Object Detection. Available online: https://universe.roboflow.com/emergencyvehicles/emergency_vehicles_sans_domain (accessed on 16 May 2023).
55. Nirma University. Emergency Vehicles Computer Vision Project Object Detection. Available online: https://universe.roboflow.com/nirma-university-xrbw5/emergency-vehicles-i10gn (accessed on 16 May 2023).
56. Adamson University. AI-Mergency Computer Vision Project. Open Source Dataset. Roboflow Universe. Roboflow. 2023. Available online: https://universe.roboflow.com/adamson-university-at786/aida-3_-ai-mergency (accessed on 20 May 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.