



Article A Cloud-Based Ambulance Detection System Using YOLOv8 for Minimizing Ambulance Response Time

Ayman Noor ¹, Ziad Algrafi ¹, Basil Alharbi ¹, Talal H. Noor ^{1,*}, Abdullah Alsaeedi ¹, Reyadh Alluhaibi ¹

- Department of Computer Science, College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia; anoor@taibahu.edu.sa (A.N.); ziadalgrafi@gmail.com (Z.A.); basil.khalid.alharbi@gmail.com (B.A.); aasaeedi@taibahu.edu.sa (A.A.); rluhaibi@taibahu.edu.sa (R.A.)
- ² Department of Computer Science, College of Computer Science and Engineering, Taibah University, Yanbu 966144, Saudi Arabia; mwateer@taibahu.edu.sa
- * Correspondence: tnoor@taibahu.edu.sa

Abstract: Ambulance vehicles face a challenging issue in minimizing the response time for an emergency call due to the high volume of traffic and traffic signal delays. Several research works have proposed ambulance vehicle detection approaches and techniques to prioritize ambulance vehicles by turning the traffic light to green for saving patients' lives. However, the detection of ambulance vehicles is a challenging issue due to the similarities between ambulance vehicles and other commercial trucks. In this paper, we chose a machine learning (ML) technique, namely, YOLOv8 (You Only Look Once), for ambulance vehicle detection by synchronizing it with the traffic camera and sending an open signal to the traffic system for clearing the way on the road. This will reduce the amount of time it takes the ambulance to arrive at the traffic light. In particular, we managed to gather our own dataset from 10 different countries. Each country has 300 images of its own ambulance vehicles (i.e., 3000 images in total). Then, we trained our YOLOv8 model on these datasets with various techniques, including pre-trained vs. non-pre-trained, and compared them. Moreover, we introduced a layered system consisting of a data acquisition layer, an ambulance detection layer, a monitoring layer, and a cloud layer to support our cloud-based ambulance detection system. Last but not least, we conducted several experiments to validate our proposed system. Furthermore, we compared the performance of our YOLOv8 model with other models presented in the literature including YOLOv5 and YOLOv7. The results of the experiments are quite promising where the universal model of YOLOv8 scored an average of 0.982, 0.976, 0.958, and 0.967 for the accuracy, precision, recall, and F1-score, respectively.

Keywords: ambulance detection; machine learning; YOlOv8; cloud computing

1. Introduction

In today's fast-paced environment, we require devices that can respond swiftly to real-time emergency calls. Ambulance vehicles, for example, face a serious, systemic challenge in determining the best route to the desired destination. No matter how hard the optimal path algorithms try, traffic light jamming will always be one of the ambulance's obstacles. By creating a model that can detect an ambulance before it arrives at the traffic light and provide a green signal to the traffic control system to clear the road, it can pass without having to wait and the ambulance's travel time can be cut down to the absolute minimum [1–4]. This approach requires two sides of implementation: one side is for building the model that detects the ambulances, and the other side is the system that uses this model to detect the ambulances and takes the results, and sends them to the traffic controller.

Delay in hospitalization is the most frequent and substantial cause of fatality in traffic accidents [1,5–7]. The injured may pass away before they reach the hospital if



Citation: Noor, A.; Algrafi, Z.; Alharbi, B.; Noor, T.H.; Alsaeedi, A.; Alluhaibi, R.; Alwateer, M. A Cloud-Based Ambulance Detection System Using YOLOv8 for Minimizing Ambulance Response Time. *Appl. Sci.* **2024**, *14*, 2555. https://doi.org/10.3390/app14062555

Academic Editor: Andrea Prati

Received: 8 February 2024 Revised: 7 March 2024 Accepted: 12 March 2024 Published: 19 March 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/). there is even a one-minute delay. Research in [7–9] found that if immediate aid is delayed, the victim may pass away before reaching the hospital. Since a 6% gap in reaction time necessitates immediate treatment for the sufferers, it is critical to arrive at the hospitals as soon as possible. To address this issue and identify ambulance vehicles as quickly as feasible, numerous advancements in this field have been investigated and tested in several scenarios. Ambulance vehicles can be found using security cameras by employing machine learning models. A number of techniques, including a Motion History Image (MHI), K-Nearest Neighbor (KNN), Hidden Markov Model (HMM), Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), Artificial Neural Network (ANN), Recurrent Neural Network (RNN), Support Vector Machine (SVM), Long Short-Term Memory network (LSTM), and others, can be used to automatically detect ambulance vehicles [1,10–15].

The proper detection of ambulance vehicles will help emergency services to keep up with evolving emergency calls [16,17]. To develop a proper ambulance vehicle detection system that can detect ambulances and distinguish them from other objects, the work is heavily dependent on data collection and training the model to detect the required objects through the camera [1,18-20]. The YOLOv8 model is fed with a large number of object (i.e., in this case, ambulance vehicle) values. After training the model, we will create a system that can run the model, pass values (i.e., a video live feed or images), and return values as a detection signal as well as information about the detection signal. Therefore, we gathered data using crawling techniques from 10 different countries. In the early stages, we managed to collect 40,000 images, and later we filtered them to finally have 3000 images from 10 countries. These countries include Saudi Arabia, Japan, Italy, Russia, the United Kingdom, Sweden, Turkey, Germany, Spain, and Norway. Then, the 3000 images were cropped, and each image was renamed for the country that it belongs to, including a name and a number. These images were trained with two types of models; one is a pre-trained model, which is generally known as a faster way to train because the first layers of the network were trained before, and the model can now know what object to look for instead of trying to identify the shape and color of that object, which in our paper is the ambulance. The second model is training from scratch; this approach generally will have less accuracy than the pre-trained one. On some occasions, all of them may have the same accuracy only if the non-pre-trained model is given the maximum epochs that it needs, but in our case, we chose 100 epochs, so there is room left to compare both sides.

In this paper, we propose a layered architecture system to develop an ambulance vehicle detection system that consists of four layers including (i) the data acquisition layer (DAL), which is the camera on the traffic light. (ii) The ambulance detection layer (ADL), which has the YOLOV8 model, will inherit the data from the DAL and process each frame. It will then export the data for the application programming interface (API), and from there, the API sends the data to (iii) the monitoring layer (ML). Both the ADL and ML are connected through (iv) the cloud layer (CL) for process and storage purposes. The main contributions of this work are summarized as follows:

- We collected and labeled 3000 images of ambulances from 10 different countries that cover various domains and languages. The dataset can be used to train and evaluate computer vision processing models for cross-lingual tasks.
- We proposed a novel layered architecture system that can handle multiple types of signal detection reports from different sources and formats. The system can efficiently process, store, and query reports using a unified interface.
- To the best of our knowledge, we are the first to use YOLOv8 to detect the ambulance to allow the traffic signals to turn green to speed up the ambulance's estimated arrival time to the hospital.
- We conducted a comprehensive evaluation of the pre-trained model and the non-pre-trained model on our dataset and compared its performance among 10 different countries. We showed that the pre-trained model outperforms the non-pre-trained model on most metrics and achieves state-of-the-art results.

The remainder of the paper is structured as follows: Section 2 discusses the related work. We describe the architecture of the ambulance detection system in Section 3. The YOLOv8 model we used is presented in Section 4, and the experiment's results are described in Section 5. Section 6 discusses the conclusion and future work.

2. Related Work

The issue of the detection of ambulance vehicles has recently attracted the attention of many researchers. For instance [21], presents a system that uses two models. YOLOV3 is used as a filtering input for distinguishing whether it is a truck or a car, and then the output goes to the CNN model to detect whether the truck is an ambulance or not. This conventional method has more than several major problems that we can overcome. Firstly, the two-step detection methodology suffers from the YOLOv3's tendency to skip actual inputs of ambulance vehicles, and in that respect, the CNN is left helpless to do anything. Second, the system has significant input lag because it filters the data in more than one step. Third, the system is inefficient in processing the trucks because the ratio of ambulance vehicles to trucks is very low, causing the system to waste time filtering and processing the data, resulting in a large waste of data and time. In solving the first matter, we will not use a hybrid model because the YOLOv8 model has achieved outstanding speed and accuracy. For the second matter, we would not need to filter or have input lag because we are not using a hybrid model. Eventually, our system will be very efficient due to the optimization of the YOLOv8 model. In [22], the authors used ResNet-50, whereas we used YOLOv8, which is superior; they trained on 100 images, whereas we trained on over 2000 images with around 1000 background images to reduce the error rate. We used a vision model from there and were able to identify where the ambulance came from the cameras at the traffic light using a First In First Out (FIFO) algorithm. In [23], the authors used an audio-model to detect the ambulance from the siren, but this approach does not work at the traffic light because there are roads in more than one direction at the traffic light. In [24], the authors have chosen to employ the monolithic approach in their system design, whereas we chose the microservice development strategy because it is more adaptable and flexible while theirs is more expensive, demanding, and incompatible with other systems. In [25], the authors used color and blob detection on OpenCV, an extremely outdated conventional method that lacks precision, while we employed the most cutting-edge model. In [26], a different approach was employed, using a keypad mounted in the ambulance that the driver used to activate the traffic light. However, this method lacks the automated system for controlling the traffic light that we implemented by utilizing YOLOv8 to automatically detect the ambulance vehicles and send the signal to the traffic light. In [27], it is crucial to the proposed system to detect objects from different angles; however, our system can detect objects from any angle even if the camera was flipped. Furthermore, to the best of our knowledge, we are the first to use YOLOv8 to detect the ambulance to allow the traffic signals to turn green to speed up the ambulance's estimated arrival time to the hospital.

3. System Architecture

The system is designed to detect ambulances on the road using a camera and a YOLOv8 model and to send a signal to open the road for ambulance vehicles. The system aims to improve the efficiency and safety of emergency services by reducing traffic congestion and response time. The system consists of four layers: the data acquisition layer, ambulance detection layer, monitoring layer, and cloud layer (see Figure 1).



Figure 1. Ambulance detection system architecture.

3.1. Architecture Layers

Each layer performs a specific function and communicates with the other layers through APIs. The system can handle multiple cameras and traffic lights simultaneously and can adapt to different scenarios and environments. The system architecture is as follows:

1. Data Acquisition Layer: This layer gathers the data from the camera and breaks it into frames. The frames are then sent to the next layer for processing. This layer is responsible for interfacing the external data sources, such as the camera, to the system. It also performs some preprocessing tasks such as filtering, scaling, or sampling the data.

2. Ambulance Detection Layer: This layer runs the YOLOv8 model on every frame that is received from the data acquisition layer (i.e., for more details about the YOLOv8 model, see Section 3.2). It determines if an ambulance is present or not in each frame. If an ambulance is detected, it posts a detection signal to its API. The ADL is connected through the cloud layer where the processing takes place.

3. Monitoring Layer: This layer fetches the detection signal from the ambulance detection layer and processes it. It displays the signal for the admin or the traffic light system so that they can open the road for the ambulance. This layer also stores the signal in the cloud for a future analysis and reporting. This layer provides a user interface for the admin to monitor and control the system. It also communicates with the traffic light system to send commands to change the traffic signals according to the detection signal. This layer uses cloud services to store and access data securely and efficiently.

4. Cloud Layer: This layer provides a platform as a service (PaaS) and infrastructure as a service (IaaS) for the other layers. It enables the communication and storage of data among the layers through the internet. It also provides scalability, security, and reliability for the system. PaaS is a cloud computing model that provides tools and services for developing and deploying applications without managing the underlying infrastructure. Meanwhile, IaaS is a cloud computing model that provides computing resources such as servers, storage, or networks on demand without managing them.

3.2. YOLOv8 Model

The You Only Look Once (YOLO) model was originally derived from the human visual system [28]. As the human system can glance at an image and instantly know what objects are in it, the model performs the same in stages. The YOLO model had a vast amount of support and was made by Ultralytics, although YOLO models are well known to the community of computer vision. The previous versions of YOLO started with the founders of the YOLO model [28]. From there, Joseph and Ali improved the YOLO

to YOLOv2 or YOLO9000 [29]. The reason for calling it YOLO9000 is that it can detect more than 9000 object categories. They also incorporate batch normalization, anchor boxes to improve detection accuracy, and dimension clusters for visualizing the 9000 objects. Then, further improvements were made to the YOLOv3 performance by implementing a new, improved network architecture from Darknet-19 that was used in YOLOv2 and was called Darknet-53 [30]. In YOLOv4 [31], the focus was on incorporating a mosaic data augmentation, which further boosts the model's performance by making it robust to where the objects may appear. And in YOLOv6 [32], a different approach was taken, where they took YOLOv5 for further optimization at the industry levels, which was made by Meituan [33]. YOLOv7 introduced a new strategy for training that further boosts the performance of the model [34]. To support a variety of vision tasks, including object identification, segmentation, pose estimation, tracking, and classification, YOLOv8 uses a backbone similar to that of YOLOv5 with a few changes to the CSP Layer, now known as the C2f module. In this work, we choose to use the YOLOv8n (nano) based on the scale of our study.

In order to detect ambulance vehicles, the YOLO model divides the image into an $S \times S$ grid, and for each cell, there is a probability. These probabilities are calculated for each cell, and if the ambulance vehicle falls in one of these cells, every cell of that ambulance vehicle is responsible for a small part of the predicted value. Figure 2 illustrates how the $S \times S$ grid is configured and how the ambulance vehicle is detected.



Figure 2. Ambulance vehicle detection in an $S \times S$ grid.

The YOLO model consists of several functions including the localization loss function, the confidence loss function, and the classification loss function. Multiple sum-squared errors make up the localization loss function. The bounding box predictions are given more weight in the localization loss function by setting λ to 5 and the boxes devoid of ambulances are given less weight by setting λ_{Δ} to 0.5 in the confidence loss function. Equation (1) offers further information about the YOLO model functions.

$$\lambda \sum_{n=0}^{S^2} \sum_{k=0}^{B} 1^{\alpha}_{nk} \Big[(x_n - \hat{x}_n)^2 + (y_n - \hat{y}_n)^2 \Big] + \lambda \sum_{n=0}^{S^2} \sum_{k=0}^{B} 1^{\alpha}_{nk} \Big[\left(\sqrt{w_n} - \sqrt{\hat{w}_n} \right)^2 + \left(\sqrt{h_n} - \sqrt{\hat{h}_n} \right)^2 \Big] + \sum_{n=0}^{S^2} \sum_{k=0}^{B} 1^{\alpha}_{nk} \Big[\left(\Theta_n - \widehat{\Theta}_n \right)^2 \Big] + \lambda_{\Delta} \sum_{n=0}^{S^2} \sum_{k=0}^{B} 1^{\Delta}_{nk} \Big[\left(\Theta_n - \widehat{\Theta}_n \right)^2 \Big] + \sum_{n=0}^{S^2} 1^{\alpha}_n \sum_{c \in \gamma} \Big[(\rho_n(c) - \widehat{\rho}_n(c))^2 \Big]$$
(1)

As we can see in Equation (1), the first two lines represent the localization loss function. λ denotes the coefficient that multiplies the loss for the coordinates in each grid cell $\sum_{n=0}^{S^2}$ (i.e., in the $S \times S(S^2)$ grid) and in each grid box $\sum_{k=0}^{B}$ where an ambulance can be found, denoted as α . 1^{α}_{nk} represents the identity function, set to 1 if there is an ambulance in the n^{th} cell, and the k^{th} bounding box is responsible for the prediction of an ambulance (i.e., object). The sum-squared error is aggregated for both the x and y coordinates in the n^{th} cells. The square root for the width (denoted as w) and the height (denoted as h) in the n^{th} cells are aggregated to consider the equality of small and large bounding boxes to ensure that the error is handled uniformly across both large and small boxes. The second two lines represent the confidence loss function, where Θ denotes the confidence score and Θ the predicted confidence score (i.e., the first line in the confidence loss function represents the confidence error when an ambulance is detected in the cell). λ_{Δ} denotes the coefficient that multiplies the ambulance losses for each grid cell in the $S \times S$ grid (i.e., the second line in the confidence loss function represents the confidence error when an ambulance is not detected in the cell). The last line represents the classification loss function where the conditional probability denoted as ρ is aggregated for the ambulance class c, which is \in the set classes denoted as γ . We defined two different classes including an ambulance and background.

4. Implementation

For the implementation, we used Amazon Web Services (AWS) as our cloud service provider. AWS helped us to set up and manage our system in a reliable, fast, and scalable manner. AWS provided us with various services and features that suited our system needs and use cases. For example, we used AWS Elastic Compute Cloud (EC2) (https://aws.amazon.com/ec2/instance-types/t3/ (accessed 15 December 2023)) to launch virtual servers with different resources that we could adjust as needed. We also used AWS security and governance services to protect our data and applications from unauthorized access and threats. To set up and manage our system on AWS, we chose an EC2 instance, and the configuration parameters and values are detailed in Table 1.

Table 1. List of the configuration parameters and values.

Parameter	Value
Cloud service provider	AWS
Instance type	t3.xlarge
Operating system	Windows Server 2022
CPU	Intel Xeon Platinum 8259CL
RAM	16 GB
Disk size	30 GB

After launching the EC2 instance, we installed Docker version 4.16.2 (https://docs. docker.com/desktop/release-notes/#4160 (accessed 17 December 2023)) on the virtual machine. We created two containers based on Arch Linux using Docker. The first container ran the code for the data acquisition layer and ambulance detection layer of our system. The data acquisition layer collected data from various sources, such as sensors, cameras, GPS devices, etc. The ambulance detection layer analyzed the data and detected any ambulances in the vicinity. This container also included a Python interpreter version 3.10.11 (https://www.python.org/downloads/release/python-31011/ (accessed 18 December 2023)) and NodeJS version 18.16.0 (https://nodejs.org/en/blog/release/v18.16.0 (accessed 18 December 2023)). Python and NodeJS helped us write and run our code for the data acquisition layer and ambulance detection layer of our system. Python enabled us to write concise and readable code that could handle complex tasks and data structures. NodeJS allowed us to use JavaScript for both front-end and back-end development, simplifying our codebase and improving performance. The second container ran the code for the monitoring system for our system. The monitoring system enabled us to monitor the system's performance and status in real-time using a web-based dashboard. This container only included NodeJS version 18.16.0. This container was used for the monitoring system for our system.

4.1. Data Collection

We gathered data using crawling techniques from 10 different countries. Over 40,000 images were gathered, but only 3000 images were chosen because some were irrelevant to the study purposes. The countries chosen were Saudi Arabia, Russia, Japan, Germany, Sweden, Norway, Turkey, the United Kingdom, Italy, and Spain. Each country received 300 images of its ambulance vehicles, 200 of which were trained on YOLOv8 and the rest were for validation. In addition, we trained the model with 948 background images, as our data on ambulance vehicles were small (i.e., the model would not be able to identify the ambulance vehicles from the background). Two types of models were implemented: one a pre-trained model and the other a non-pre-trained one. Eventually, we collected the data for each country out of need rather than interest because there is no such thing as a universal ambulance car; each country has its own vehicle with its own shape and color, so we had to construct our own data. Nonetheless, we currently have 22 models. Ten are pre-trained models, and ten are non-pre-trained models; each country has each of them, and 2 are the crucial ones, which we term the universal model because they can work in all of the nations that we have listed; one is pre-trained and the other is non-pre-trained. More details regarding the dataset such as the image size, size range of the ambulance objects (in pixels), possible number of ambulance vehicles in each image, etc., can be found in Table 2.

Data Labeling

For the training, 2000 of the obtained images were manually labeled and categorized as ambulance vehicles. A total of 1000 images, however, were utilized for testing. One text file per image is used to store the annotations and labeling.

Country	Data Type	Total Images	Total Labels	Min. Object Size (px)	Max. Object Size (px)	Avg. Object Size (px)	Min. Ambulance Count	Max. Ambulance Count	Avg. Ambulance Count
All data	Training	2000	2306	247.86	15,538,778.40	109,254.97	1	8	1.15
All data	Validation	1000	1132	470.20	10,116,762.21	191,851.62	1	6	1.13
Ambulance—Germany	Training	200	220	1007.49	1,186,804.65	51,727.80	1	8	1.10
Ambulance—Germany	Validation	100	112	1491.36	3,201,209.98	275,850.71	1	4	1.12
Ambulance—Italy	Training	200	203	1376.61	3,937,977.63	151,716.60	1	2	1.01
Ambulance—Italy	Validation	100	111	483.85	887,085.47	122,364.63	1	4	1.11
Ambulance—Japan	Training	200	211	1187.14	15,538,778.40	165,862.70	1	2	1.06
Ambulance—Japan	Validation	100	113	923.05	4,434,590.25	214,963.03	1	5	1.13
Ambulance—Norway	Training	200	219	812.92	8,467,322.79	153,564.89	1	2	1.10
Ambulance—Norway	Validation	100	108	601.87	1,791,667.76	37,890.46	1	5	1.08
Ambulance—Russia	Training	200	253	666.66	4,447,982.09	91,370.16	1	4	1.26
Ambulance—Russia	Validation	100	116	1033.73	1,401,825.30	92,035.72	1	3	1.16
Ambulance—Saudi Arabia	Training	200	215	247.86	313,236.73	24,322.19	1	4	1.08
Ambulance—Saudi Arabia	Validation	100	104	470.20	926,459.14	138,453.29	1	2	1.04
Ambulance—Spain	Training	200	233	1165.99	9,633,192.61	324,977.95	1	5	1.17
Ambulance—Spain	Validation	100	114	2008.01	9,597,678.94	385,183.35	1	3	1.14
Ambulance—Sweden	Training	200	216	993.95	50,081.73	25,045.00	1	4	1.08
Ambulance—Sweden	Validation	100	112	1908.89	2,348,605.69	138,267.40	1	5	1.12
Ambulance—Turkey	Training	200	250	398.81	5,857,812.51	71,457.92	1	6	1.25
Ambulance—Turkey	Validation	100	111	1540.82	941,968.69	44,960.89	1	3	1.11
Ambulance—United Kingdom	Training	200	286	481.81	5,063,227.92	46,318.38	1	6	1.43
Ambulance—United Kingdom	Validation	100	131	781.50	10,116,762.21	419,075.79	1	6	1.31

Table 2. Dataset Specification	
--------------------------------	--

5. Experimental Results

5.1. YOLOv8 Model Performance

For YOLOv8 model performance, we have 22 different experiments, 10 of which are pre-trained while the other 10 are non-pre-trained (i.e., two different experiments for each country). Moreover, we conducted another two experiments, which we call the universal model (i.e., pre-trained vs. non-pre-trained) for all of the countries that we have listed. Figure 3 shows how the YOLOv8 model uses the bounding box in red around the ambulance vehicles. Meanwhile, Figure 4 shows the validation of the detection of the ambulance by drawing boxes around the ambulance vehicles with an accuracy measure that states how confident the YOLOv8 model is regarding that detected ambulance. The training and running time of the proposed model are detailed in Table 3.



Figure 3. The bounding box in red around the labeled ambulance vehicles.



Figure 4. The validation of the detection of the ambulance vehicles.

The results of our experiments show promising accuracy scoring. Nonetheless, we found that some of the results took a leap down when we experimented using a non-pre-trained scenario. For example, Figure 5 illustrates the accuracy bar charts of the non-pre-

trained scenario for the countries Saudi Arabia (SA), Russia (RU), Japan (JP), Germany (DE), Sweden (SE), Norway (NOR), Turkey (TR), the United Kingdom (UK), Italy (IT), and Spain (ES). In the results for the non-pre-trained scenario, we observe that SA received the highest score in accuracy, reaching 0.98. This is because of the high resolution and good angle of the captured images, as well as the large logo of the ambulance vehicles. In addition, we can see that the SE scored second (i.e., 0.976 in accuracy) because most of the types of ambulance vehicles in the SE validation set have a higher object size than others, which contributes to a higher accuracy. In other words, larger object sizes in pixels and large-size ambulances with more distinguishable features tend to yield higher accuracy, whereas smaller object sizes in pixels and medium-size ambulances with fewer distinguishable features result in lower accuracy (e.g., Sport Utility Vehicles (SUVs)); especially the YOLOv8 model works on dividing the image into an $S \times S$ grid.

Table 3. The Training and Running Time of the Proposed Model

Stage	Time
Pre-process	10 ms
Inference	100 ms
Data Transfer Time	100 ms
Server Processing Time	200 ms
Response Time	310 ms
Data Acquisition Layer	70 ms
Monitoring Layer	30 ms
Ambulance Detection Layer	110 ms
Cloud Layer ¹	650 ms

¹ Response time is calculated as an average for all figures where it takes less than 1 s on the cloud.





Figure 5. The accuracy of the non-pre-trained model among 10 different countries.

On the other hand, Figure 6 illustrates the accuracy bar charts of the pre-trained scenario for the aforementioned countries. We can see that JP received the best accuracy rating of 0.995. The reasons behind this are that JP data have high resolution for the high object size in pixels in the validating set and not more than a four-view angle of the captured images, as well as the logo size and one vehicle type for the ambulance vehicles. While SE data also have a large logo of the ambulance vehicles, which will make it easier for the YOLOv8 model to distinguish this particular ambulance vehicle from other ones, the SE ambulance vehicles have only two colors as well, including yellow and green.

Figure 7 depicts the results of our YOLOV8 accuracy performance for pre-trained vs. non-pre-trained scenarios among 10 different countries. Overall, we can see that the results in the pre-trained scenario overcome the non-pre-trained scenario in all countries' experiments. That is because the YOLOV8 model's first layers already knew what to look for (i.e., ambulance vehicles' shapes and colors). In the results for the pre-trained scenario,

we observe that SE and RU received the best accuracy rating of 0.99 for the aforementioned reasons. However, IT and NOR received the worst accuracy rating of 0.94; that is because the IT dataset has different ambulance vehicle types and colors, and the NOR dataset has low-resolution images, images taken from different angles, and medium-size ambulance vehicles (i.e., SUVs). In other words, the medium-size ambulance will have fewer boxes than the large-size ambulance (e.g., truck vehicles), which will affect the accuracy's result in a negative correlation.





Figure 6. The accuracy of the pre-trained model among 10 different countries.

Figure 7. Cont.



Figure 7. Pre-trained vs. non-pre-trained accuracy performance among 10 different countries. (a): Accuracy performance in Germany dataset. (b): Accuracy performance in Italy dataset. (c): Accuracy performance in Japan dataset. (d): Accuracy performance in Norway dataset. (e): Accuracy performance in Russia dataset. (f): Accuracy performance in Saudi Arabia dataset. (g): Accuracy performance in Spain dataset. (h): Accuracy performance in Sweden dataset. (i): Accuracy performance in Turkey dataset. (j): Accuracy performance in UK dataset.

Figure 8 illustrates the results of our universal YOLOV8 model performance for pretrained vs. non-pre-trained scenarios among all of the countries that we have listed. For this experiment, we trained our YOLOv8 model using 2000 images (i.e., for all ambulance vehicles in 10 different countries) and validated the universal model using 1000 images. As we can see, the universal model has scored 0.98 in accuracy (see Figure 8a) for both pre-trained and non-pre-trained scenarios, which is the highest result among all countries. The reason for that is the universal model has the largest amount of data for training where the model could understand the ambulance vehicles better. The reason for going with 100 epochs is to leave room to compare both the pre-trained and the non-pre-trained models. We can also observe that the universal model reached a precision of 0.97 and 0.96 in the pre-trained and non-pre-trained scenarios, respectively (see Figure 8b). For the recall, the universal model has scored 0.95 in both scenarios as shown in Figure 8c. Meanwhile, the universal model regarding the F1-score recorded 0.96 in the pre-trained scenario and 0.95 in the non-pre-trained scenario (see Figure 8d). More details about the accuracy, precision, recall, and F1-score results after reaching 100 epochs can be found in Table 4.

Universal YOLOv8

40

(b)

Universal YOLOv8

epoch

60

80

100

Pre-trained
 Non-pre-trained

1.0

0.8

0.2

0.0

1.0

0.8

0.6 Ш 0.4 0

20

Pre-trained Non-pre-trained

Precision 0.6 0.4

Universal YOLOv8

40

(a)

Universal YOLOv8

60

epoch

80

100

Pre-trained • Non-pre-trained

1.0

8.0

Accuracy 0.4

0.2

0.0

1.0

8.0

Becall 0.6 0.6

0

20

Pre-trained •Non-pre-trained



cision universal performance of YOLOv8. (c): Recall universal performance of YOLOv8. (d): F1-score universal performance of YOLOv8.

Country	JP	SE	RU	SA	TR	UK	DE	ES	IT	NOR	Universal	
Accuracy of pre-trained	95.5	99.4	99.3	99.1	98.8	98.7	95.8	96.9	94.5	94.2	98.2	
Accuracy of non-pre-trained	93.8	97.6	95.7	98.0	94.4	97.2	95.6	95.8	90.3	89.5	98.0	
Precision of pre-trained	99.1	100	97.1	97.0	97.3	93.2	97.8	93.3	99.0	98.8	97.6	
Precision of non-pre-trained	95.8	97.3	94.8	94.8	95.6	93.5	94.8	93.8	89.8	97.7	96.3	
Recall of pre-trained	99.0	97.9	98.3	98.1	99.0	98.5	91.1	98.2	91.0	87.0	95.8	
Recall of non-pre-trained	89.4	95.2	87.9	92.3	89.2	90.1	87.5	93.4	85.6	78.7	95.1	
F1-score of pre-trained	99.0	98.9	97.7	97.5	98.1	95.8	94.3	95.7	94.8	92.5	96.7	
F1-score of non-pre-trained	92.5	96.2	91.2	93.5	92.3	91.8	91.0	93.6.6	87.6	87.2	95.7	

Table 4. Performance Results Among Countries and the Universal Model

Figure 9 shows the results of the universal model in the confusion matrix where the actual values of true and false concerning predicted values are either positive or negative. As we can see, the true positives reached 1093 when the model predicts that it is an ambulance vehicle, while the true negatives are when the model predicts that it is not an ambulance vehicle (i.e., background), which is empty in our case (i.e., because the model is not supposed to identify non-ambulances). This means that when evaluating the performance of an object detection model, the number of true negatives can easily overwhelm the number of false positives and false negatives, making it difficult to compare the performance of the models. The false positive scored 57, and it occurs when the model predicts an ambulance but it is not present. The false negative scored 39, and it occurs

when the model predicts there is no ambulance but there is. Based on the recorded true positives, true negatives, false positives, and false negatives, we can easily calculate the accuracy, precision, recall, and F1-score (i.e., more details on how to calculate the accuracy, precision, recall, and F1-score can be found in [35]).





5.2. Comparison between YOLOv5, YOLOv7, and YOLOv8

To compare our proposed YOLOv8 model with other models presented in the literature [33,34], we conducted another two experiments for YOLOv5 and YOLOv7 that represent all of the countries that we have listed (i.e., the universal model) in two different scenarios including pre-trained and non-pre-trained.

Figure 10 depicts the results of accuracy for YOLOv5 and YOLOv7. We can observe that both models show slightly lower accuracy scores comparatively with YOLOv8 (see Figure 8a). YOLOv7 has achieved 0.953 for the pre-trained scenario and 0.978 for the non-pre-trained scenario, while YOLOv5 has achieved 0.979 for the pre-trained scenario and 0.975 for the non-pre-trained scenario. Meanwhile, YOLOv8 has scored 0.982 and 0.980 for the pre-trained and non-pre-trained scenarios, respectively.



Figure 10. Universal YOLOv5 vs. YOLOv7 accuracy. (a): Accuracy of YOLOv5. (b): Accuracy of YOLOv7.

Figure 11 illustrates the results of precision for YOLOv5 and YOLOv7. YOLOv5 surpasses YOLOv7, with both pre-trained (0.960) and non-pre-trained (0.953) scenarios exhibiting higher precision. However, YOLOv8 (see Figure 8b) has recorded an even higher precision rate than both YOLOv5 and YOLOv7, scoring 0.976 for the pre-trained scenario and 0.963 for the non-pre-trained scenario.



Figure 11. Universal YOLOv5 vs. YOLOv7 precision. (a): Precision of YOLOv5. (b): Precision of YOLOv7.

Figure 12 shows the results of recall for YOLOv5 and YOLOv7. YOLOv5 has achieved 0.951 for the pre-trained scenario and 0.938 for the non-pre-trained scenario. In addition, YOLOv7 has achieved 0.914 for the pre-trained scenario and 0.940 for the non-pre-trained scenario. On the other hand, YOLOv8 (see Figure 8c) has overcome both YOLOv5 and YOLOv7 models in recall, scoring 0.958 for the pre-trained scenario and 0.951 for the non-pre-trained scenario.



Figure 12. Universal YOLOv5 vs. YOLOv7 recall.(a): Recall of YOLOv5. (b): Recall of YOLOv7.

Figure 13 depicts the results of the F1-score for YOLOv5 and YOLOv7. YOLOv5 has achieved a slightly higher F1-score for pre-trained models (0.955) compared to YOLOv7 (0.919). Meanwhile, YOLOv8 (see Figure 8d) has scored 0.967 and 0.957 for pre-trained and non-pre-trained scenarios, respectively.

In summary, YOLOv8 emerges as the top performer among the three versions, offering the highest accuracy, precision, recall, and F1-score for both pre-trained and non-pretrained models. Having carefully evaluated the performance metrics, we elected to proceed with YOLOv8 as our preferred choice. This decision stems from a thorough examination of various factors, including computational resources and implementation constraints, ensuring alignment with our specific application requirements.

Table 5 shows the analysis of YOLOv5, YOLOv7, and YOLOv8, based on the empirical data, shedding light on their performance differences in object detection tasks. YOLOv8 stands out with robust performance, achieving high accuracy scores of 0.982 for pre-trained models and 0.980 for non-pre-trained scenarios. YOLOv8 demonstrates high precision scores for both pre-trained (0.976) and non-pre-trained (0.963) models. Additionally, YOLOv8 recall values are commendable, with pre-trained models achieving 0.958 and non-pre-trained models achieving 0.951, resulting in an impressive F1-score of 0.967 and 0.957 for pre-trained and non-pre-trained models, respectively.



Figure 13. Universal YOLOv5 vs. YOLOv7 F1-score. (a): F1-score of YOLOv5. (b): F1-score of YOLOv7.

Table 5. The Universal Model renormance	Table 5.	The L	Jniversal	Model	Performanc
---	----------	-------	-----------	-------	------------

Model	Universal YOLOv5	Universal YOLOv7	Universal YOLOv8
Accuracy of pre-trained	97.9	95.3	98.2
Accuracy of non-pre-trained	97.5	97.8	98.0
Precision of pre-trained	96.0	92.5	97.6
Precision of non-pre-trained	95.3	95.8	96.3
Recall of pre-trained	95.1	91.4	95.8
Recall of non-pre-trained	93.8	94.0	95.1
F1-score of pre-trained	95.5	91.9	96.7
F1-score of non-pre-trained	94.5	94.9	95.7

In light of our observation, the YOLOv7 pre-trained model did not outperform the non-pre-trained one, and the reason behind that could be attributed to its ability to adapt more effectively to the specific characteristics of the dataset over time. While the pre-trained model initially demonstrates better performance due to its transfer learning from a general dataset, it might struggle to fully adapt to the nuances of the target domain as training progresses. On the other hand, the non-pre-trained model starts from scratch but gradually learns to capture the intricacies of the dataset more effectively, leading to its superior performance in the long run. This hypothesis suggests that the non-pre-trained model's superiority is a result of its ability to learn domain-specific features more deeply over the course of training, ultimately surpassing the initial advantage provided by the pre-trained model's transfer learning.

6. Conclusions and Future Work

This work presents the design and implementation of a novel ambulance detection system architecture to prioritize ambulance vehicles by turning the traffic light to green for saving patients' lives. The system is capable of handling multiple types of signal detection reports from different sources and formats by processing and storing them in the cloud. In particular, we exploit a YOLOv8 model to detect the ambulance and allow the traffic signals to turn green to speed up the ambulance's estimated arrival time at the hospital. We are the first to exploit a YOLOv8 model for the detection of ambulances, to the best of our knowledge. To demonstrate the performance of our YOLOv8 model, we collected and labeled 3000 images of ambulances from 10 different countries that cover

various domains and languages. The dataset can benefit the research community in training and evaluating computer vision models for cross-lingual tasks. Moreover, we conducted 22 different experiments, 10 of which are pre-trained while the other 10 are non-pre-trained (i.e., 2 different experiments for each country). We conducted another two experiments, which we call the universal model (i.e., pre-trained vs. non-pre-trained) for all of the countries that we have listed. Furthermore, we compared the performance of our YOLOv8 model with other models presented in the literature including YOLOv5 and YOLOv7. The results of the experiments are quite promising where the universal model of YOLOv8 scored an average of 0.982, 0.976, 0.958, and 0.967 for the accuracy, precision, recall, and F1score, respectively. The universal model scored the highest result among all countries because of the large amount of data for training where the model could understand the ambulance vehicles better. For future work, we intend to alter the network layers to suit our datasets, resulting in a high-efficiency model. In addition, we are planning to develop a new segmentation technique; semantic segmentation is the counterpart of object detection, where objects are identified with a square box, but in segmentation, objects are drawn at the pixel level.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data is available on the following link (https://github.com/basilalharbi/Ambulance-Detection-System) (accessed on 11 March 2024).

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

References

- 1. Almukhalfi, H.; Noor, A.; Noor, T.H. Traffic management approaches using machine learning and deep learning techniques: A survey. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108147. [CrossRef]
- Lupa, M.; Chuchro, M.; Sarlej, W.; Adamek, K. Emergency ambulance speed characteristics: A case study of Lesser Poland voivodeship, southern Poland. *GeoInformatica* 2021, 25, 775–798. [CrossRef]
- 3. Cerna, S.; Arcolezi, H.H.; Guyeux, C.; Royer-Fey, G.; Chevallier, C. Machine learning-based forecasting of firemen ambulances' turnaround time in hospitals, considering the COVID-19 impact. *Appl. Soft Comput.* **2021**, *109*, 107561. [CrossRef] [PubMed]
- Nguyen, V.L.; Hwang, R.H.; Lin, P.C. Controllable Path Planning and Traffic Scheduling for Emergency Services in the Internet of Vehicles. *IEEE Trans. Intell. Transp. Syst.* 2021, 23, 12399–12413. [CrossRef]
- Colla, M.; Santos, G.D.; Oliveira, G.A.; de Vasconcelos, R.B.B. Ambulance response time in a Brazilian emergency medical service. Socio-Econ. Plan. Sci. 2023, 85, 101434. [CrossRef]
- Chang, F.R.; Huang, H.L.; Schwebel, D.C.; Chan, A.H.; Hu, G.Q. Global road traffic injury statistics: Challenges, mechanisms and solutions. *Chin. J. Traumatol.* 2020, 23, 216–218. [CrossRef] [PubMed]
- Foggia, P.; Petkov, N.; Saggese, A.; Strisciuglio, N.; Vento, M. Audio surveillance of roads: A system for detecting anomalous sounds. *IEEE Trans. Intell. Transp. Syst.* 2015, 17, 279–288. [CrossRef]
- 8. Kumar, N.; Acharya, D.; Lohani, D. An IoT-based vehicle accident detection and classification system using sensor fusion. *IEEE Internet Things J.* **2020**, *8*, 869–880. [CrossRef]
- 9. World Health Organization. *World Health Statistics 2023: Monitoring Health for the SDGs, Sustainable Development Goals;* World Health Organization: Geneva, Switzerland, 2023.
- 10. Kong, Y.; Fu, Y. Human action recognition and prediction: A survey. Int. J. Comput. Vis. 2022, 130, 1366–1401. [CrossRef]
- 11. Noor, A.; Pattanaik, P.; Khan, M.Z.; Alromema, W.; Noor, T.H. Deep Feature Detection Approach for COVID-19 Classification based on X-ray Images. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 141–146. [CrossRef]
- 12. Gutiérrez, J.; Rodríguez, V.; Martin, S. Comprehensive review of vision-based fall detection systems. *Sensors* **2021**, *21*, 947. [CrossRef]

- 13. Saif, S.; Tehseen, S.; Kausar, S. A survey of the techniques for the identification and classification of human actions from visual data. *Sensors* **2018**, *18*, 3979. [CrossRef]
- 14. Noor, T.H. Human Action Recognition-Based IoT Services for Emergency Response Management. *Mach. Learn. Knowl. Extr.* 2023, *5*, 330–345. [CrossRef]
- 15. Sun, Z.; Ke, Q.; Rahmani, H.; Bennamoun, M.; Wang, G.; Liu, J. Human action recognition from various data modalities: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 3200–3225. [CrossRef]
- 16. Neira-Rodado, D.; Escobar-Velasquez, J.W.; McClean, S. Ambulances deployment problems: Categorization, evolution and dynamic problems review. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 109. [CrossRef]
- Li, M.; Vanberkel, P.; Zhong, X. Predicting ambulance offload delay using a hybrid decision tree model. *Socio-Econ. Plan. Sci.* 2022, *80*, 101146. [CrossRef]
- Shamrat, F.J.M.; Mahmud, I.; Rahman, A.S.; Majumder, A.; Tasnim, Z.; Nobel, N.I. A smart automated system model for vehicles detection to maintain traffic by image processing. *Int. J. Sci. Technol. Res.* 2020, 9, 2921–2928.
- 19. Yu, X.; Marinov, M. A study on recent developments and issues with obstacle detection systems for automated vehicles. *Sustainability* **2020**, *12*, 3281. [CrossRef]
- Arul, S.J.; Mithilesh, B.; Shreyas, L.; Kaliyaperumal, G.; K. A., J.K. Modelling and Simulation of Smart Traffic Light System for Emergency Vehicle using Image Processing Techniques. In Proceedings of the 2023 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM), Uttar Pradesh, India, 22–24 February 2023; IEEE: Toulouse, France, 2023; pp. 1–4.
- 21. Agrawal, K.; Nigam, M.; Bhattacharya, S.; Sumathi, G. Ambulance detection using image processing and neural networks. *J. Physics: Conf. Ser.* **2021**, 2115, 012036. [CrossRef]
- 22. Jiménez-Moreno, R.; Martínez Baquero, J.E.; Rodriguez Umaña, L.A. Ambulance detection for smart traffic light applications with fuzzy controller. *Int. J. Electr. Comput. Eng.* **2022**, *12*, 2088–8708. [CrossRef]
- Usaid, M.; Asif, M.; Rajab, T.; Rashid, M.; Hassan, S.I. Ambulance Siren Detection using Artificial Intelligence in Urban Scenarios. Sir Syed Univ. Res. J. Eng. Technol. 2022, 12, 92–97. [CrossRef]
- Rubini, K.; Vidya, M.; Yeshaswini, S.; Gowthami, A. Automatic Ambulance Detection and Intimation Using RSSI. Int. J. Emerg. Technol. Eng. Res. (IJETER) 2019, 7, 40–43.
- Yang, O.W.; Suriani, N.S. Vision based traffic control for intelligence ambulance detection system. *Evol. Electr. Electron. Eng.* 2020, 1, 333–341.
- 26. Ahir, D.; Bharade, S.; Botre, P.; Nagane, S.; Shah, M. Intelligent traffic control system for smart ambulance. *IRJET* 2018, *5*, 355–358.
- 27. Srinivasan, V.; Rajesh, Y.P.; Yuvaraj, S.; Manigandan, M. Smart traffic control with ambulance detection. *Iop Conf. Ser. Mater. Sci. Eng.* **2018**, 402, 012015. [CrossRef]
- Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
- Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 2017; pp. 7263–7271.
- 30. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv:1804.02767.
- 31. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. arXiv 2020, arXiv:2004.10934.
- 32. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* 2022, arXiv:2209.02976.
- 33. Terven, J.; Cordova-Esparza, D. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv* 2023, arXiv:2304.00501.
- Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
- Noor, T.H.; Noor, A.; Elmezain, M. Poisonous Plants Species Prediction Using a Convolutional Neural Network and Support Vector Machine Hybrid Model. *Electronics* 2022, 11, 3690. [CrossRef]

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.