UDC 004.78

O.Vorona , O.Polshakova

SUBSYSTEM FOR RECOGNITION OF EMERGENCY SITUATIONS ON THE ROAD

*Abstract:*The article considers an approach to solving the problem of increasing the speed of response to emergency events on the road. The process of incident recognition and automatic notification of the relevant operator to improve the interaction of intelligent smart city systems with emergency services of the city is the object of research in this work.

As a result of the study of this subject area, road incidents were classified and systematized. Also, the database includes information about the services that can be involved in responding to the event, and communication channels with them have been configured.

The proposed solution involves the use of visual surveillance tools capable of recognizing an emergency situation on the road and sending a notification about it to emergency services, as well as recording information about the incident in a single data store as a component of Smart city.

*Keywords:*means of visual surveillance, emergency recognition, Yolov8 computer vision model.

Introduction

With the advent of the era of information technology, humanity is witnessing the rapid development of cities and the increase in the number of vehicle owners, which leads to a large number of road traffic accidents (traffic accidents). Road accidents are a serious threat to the safety of citizens, economic development and environmental conditions. In addition, taking into account the statistics, in the conditions of a modern city, it is important not only to implement means to reduce the number of road accidents, but also to respond quickly and effectively to the situations that have occurred. Therefore, the ability to identify and respond to road emergencies in a timely manner is vital for community engagement, thereby minimizing emergency response times, as well as the number of deaths and property that can be lost in such cases.

Problem Statement

In the context of this study, the object is the process of automated recognition of emergency situations using video streams from surveillance cameras and webcams. The subject of the research is a neural network that, using computer vision, is able to recognize incidents on the road, classify them, and notify the operator of the information about them, recording the relevant incident in the database for keeping statistics.

The main goal is to increase the efficiency of Smart City functioning in the tasks of reducing the response time to emergency situations, which will allow to reduce the number of victims and reduce material losses due to the development of appropriate software. This result can be achieved by integrating advanced technologies of computer vision, neural networks and real-time video processing.

To achieve the set goal, there is a need to solve such tasks as:

– develop a video stream processing module;

– train a neural model to recognize road accidents;

– implement the integration of the message module for informing emergency services;

– connect the subsystem to the database for storing and analyzing the collected data;

– test the implemented subsystem.

The implementation of such a decision will contribute to the improvement of overall road safety, the efficiency of emergency services, increasing the level of public trust in these services and authorities, and also saving time and resources, since fewer accidents reduce the costs of repairing infrastructure, vehicles and treating victims.

The results of the analogues analysis

As of today, there is a limited number of software tools that can recognize emergency situations on the roads in real time with high accuracy. There are no direct analogues of this system, there are solutions that partially offer a solution to the problem under study. Fig. 1 shows the three analogues considered, which formed the basis of the conducted analysis.

Figure 1. Examples of analogues

In Traffix.ai and a2-VCA, there is no extended classification of incidents, which in turn does not allow for a more differentiated analysis of traffic. In addition, the Traffix.ai system has limitations for certain camera manufacturers and models, making it less accessible to users. The disadvantage of intuVision Traffic is its narrow specialization, since this solution focuses on the tasks of classifying objects on roads. These solutions require a paid license and expensive equipment for effective work.The results of the study are shown in Tab. 1.

ISSN 1560-8956

Table 1. **Functions that are implemented in the considered systems**

The name of the subsystem	Classification of incidents	Sending event notifications	View statistics	Work in real time	Processing of local video files
Traffix.ai	Absent	Missing	Available	Available	Absent
a2-VCA	Absent	Available	Missing	Available	Absent
intuVision Traffic	Absent	Missing	Available	Available	Absent

Description of architectural components

During the analysis of the subject area, the main components of the subsystem were highlighted, which include a module for collecting and processing video data, a model for recognizing emergency situations, a user interface, as well as a module for communicating with the database and sending messages to emergency services. That is, this solution consists of four main parts.

The incident recognition module uses the YOLOv8m pre-trained model for incident recognition and the OpenCV library for video processing tasks [1].

The module for sending an incident to the mail is implemented using the requests and email libraries, forms and sends requests to the API of the Mailgun service, which is used to send an incident to the operator's mail, using a special domain created for mailings, which allows sending an unlimited number of messages to email users, without blocking the account record [2]. Also, the auxiliary library in this module is the threading library. It is used to create an auxiliary thread in which messages will be sent to avoid application slowdowns and improve the user experience.

The module for working with the database is implemented using pyodbc, which is a Python library for creating a connection and conducting manipulations with the database, which allows you to form queries to it. Thus, obtaining information about incidents, carrying out authorization, managing users and keeping statistics is implemented.

The authorization module, using the database module, is responsible for user authorization and, accordingly, will help the subsystem administrator to manage users.

The fifth and additional part is the UI module, where authorization and registration forms are developed, an administrative panel form for controlling authorized users, a form for working with video streams, where the user can choose the location of the video stream on which recognition is carried out, or it is possible to use prepared videos and form of statistical output [3].

Due to such an organization of components, the user has the opportunity to interact with the authorization module, the statistics output module, and the recognition module. The authorization module will be used to log in to the system, where when interacting with the recognition module, a streaming video will be displayed on the screen, on which the frames of recognized incidents will be applied in the event of their occurrence on the screen.

When switching to the statistics display mode, the user can interact with the database module, the structure of which is shown in Fig. 2, where the function of receiving statistics data on the screen is available through communication with it.

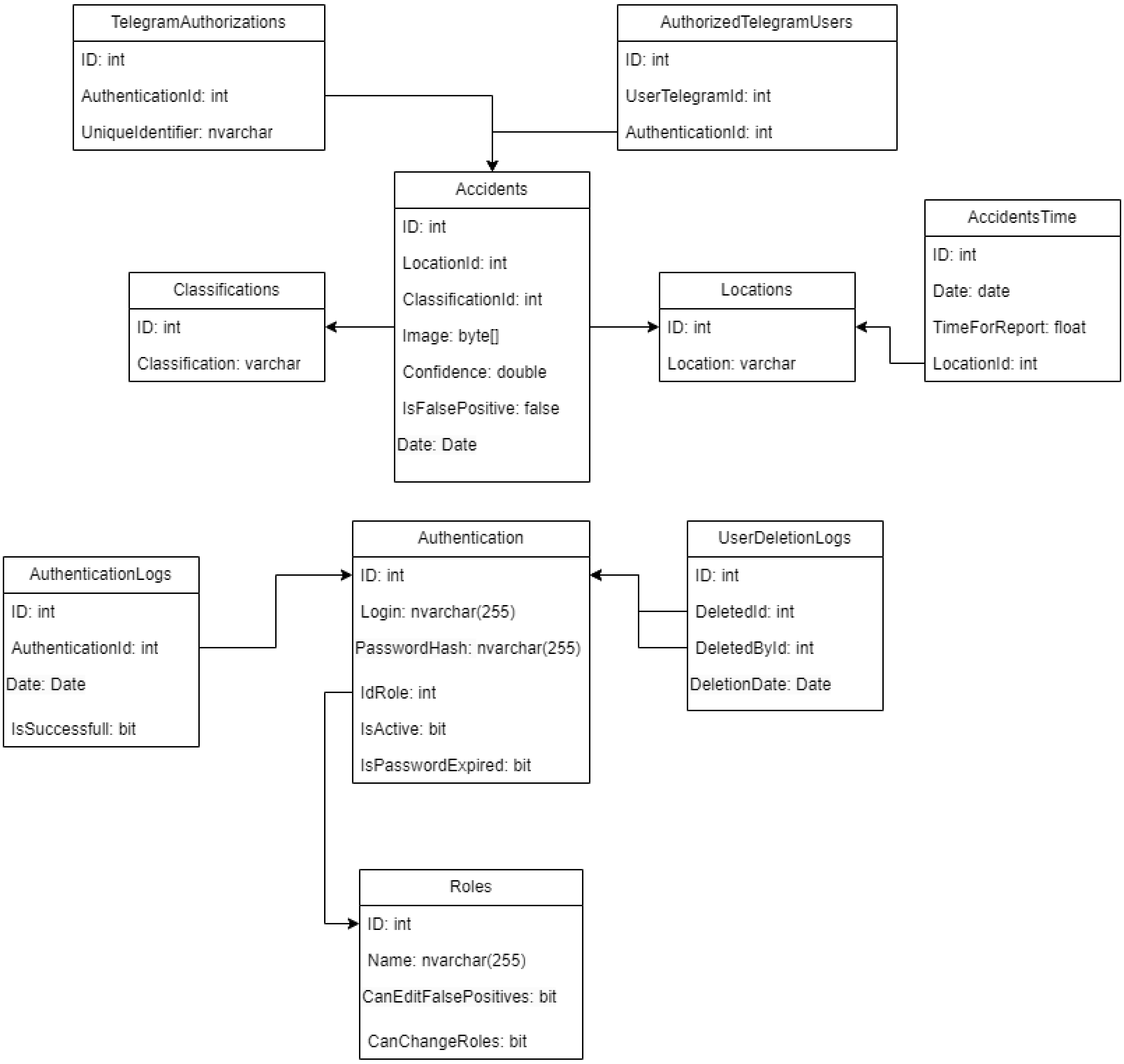


Figure 2. Database diagram

Description of the solution

In the context of the developed solution, it is proposed to expand the functions of the component systems of the smart city in the tasks of monitoring the road situation in order to reduce the delay time in responding to emergency situations by automating the process of automatic identification of road accidents and providing this information to the relevant services.

The process of incident recognition and automatic notification of the relevant operator to improve the interaction of intelligent smart city systems with the emergency services of the city is the object of research in this work. The sequence of incident recognition is shown in Fig. 3.

As a result of the study of this subject area, road incidents were classified and systematized. Also, the database includes information about the services that can be involved in responding to the event, and communication channels with them have been configured.

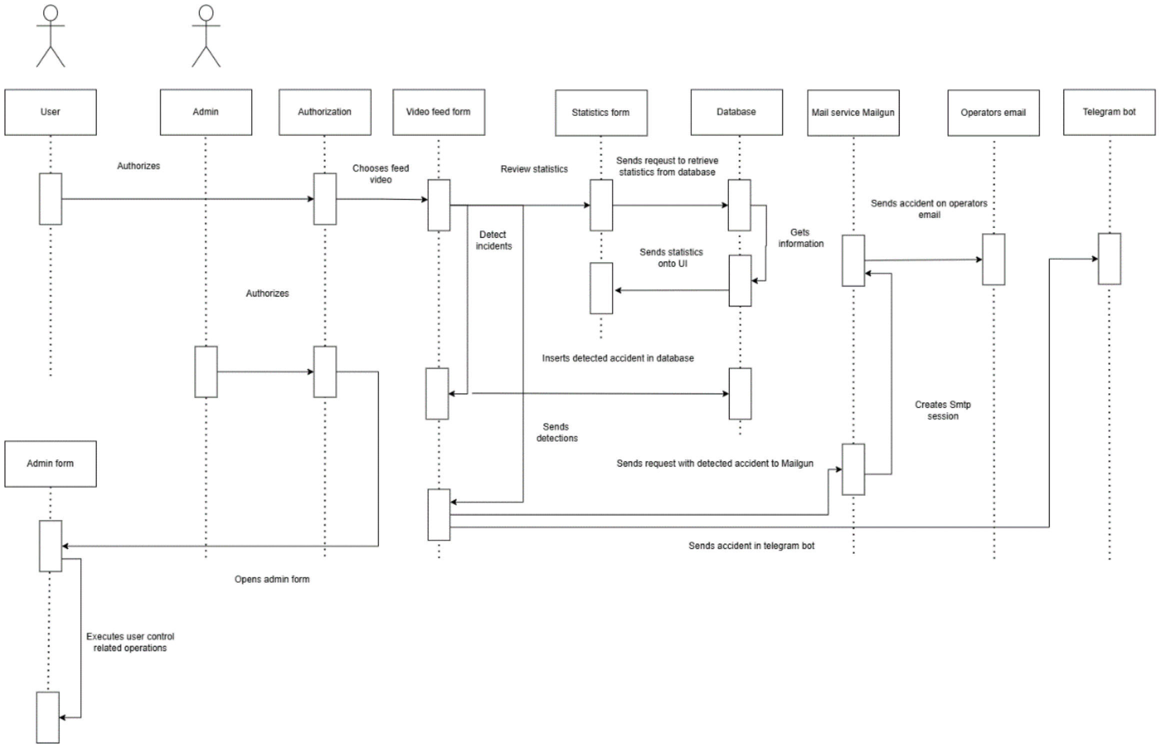


Figure 3. Incident recognition sequence

In the proposed subsystem software, the YOLOv8m model is pre-trained on a data set that has classes of road incidents, an input video is specified, on which the model searches for incidents, when an incident is detected, information about it is sent to the specified mail using the mailgun service.

In turn, after additional training of the model, the results were obtained, which can be seen on the Confusion matrix normalized graph, which is responsible for the distribution of the percentage of successful recognition by class (Fig. 4) [4]. As you can see, the model best recognizes the Moderate class with a percentage of 96%, followed by the Severe class with a recognition percentage of 93%, in the NoAccident class, the percentage of successful defoaming is 72%, 28% is recognized on the background.

The algorithm of deep machine learning of neural networks - deep learning was chosen and used as the recognition algorithm. In contrast to classical machine learning, in which new knowledge is extracted from a large array of data uploaded to the system, which is later corrected by users and machine learning rules are formed on the basis of this knowledge. In deep learning, new knowledge is formed independently without user involvement. The learning phase during the application of this algorithm is the labeling of significant volumes of data and establishing their correspondence as a result of comparison with the reference values of the knowledge base [5].

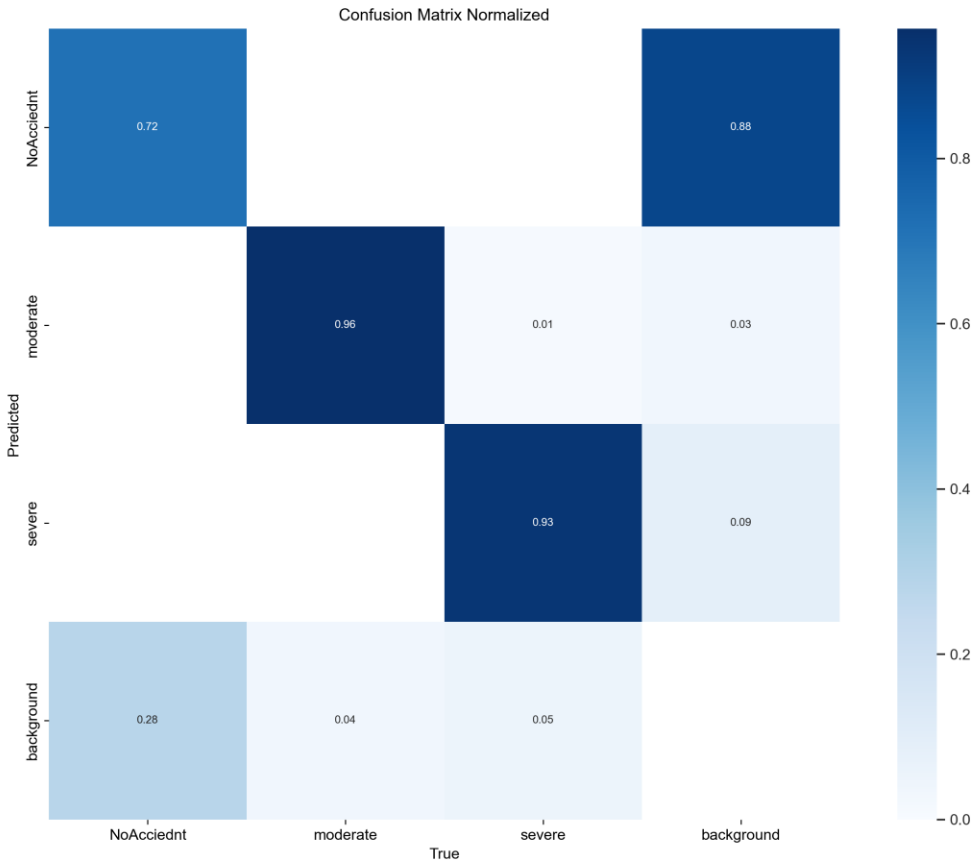


Figure 4. Confusion Matrix Normalized

The main stages of the recognition algorithm are as follows [6]:

1. The input image is fed to the input of the selected model and conditionally divided into three parts [7]:

− Backbone: The input image passes through the backbone, which consists of convolution layers. In YOLOv8, an improved version of the main block is used, which allows to increase the accuracy of object detection.

− Subsampling: after passing through the main block, the image is reduced in size (subsampled) using subsampling, which helps to save memory and speed up calculations.

− Detect Head: After subsampling, the image passes through the main detection unit, which aims to find objects and determine their locations and classes. YOLOv8 uses a custom mainframe architecture that takes into account previous versions of YOLO and additional optimizations.

 ISSN 1560-8956

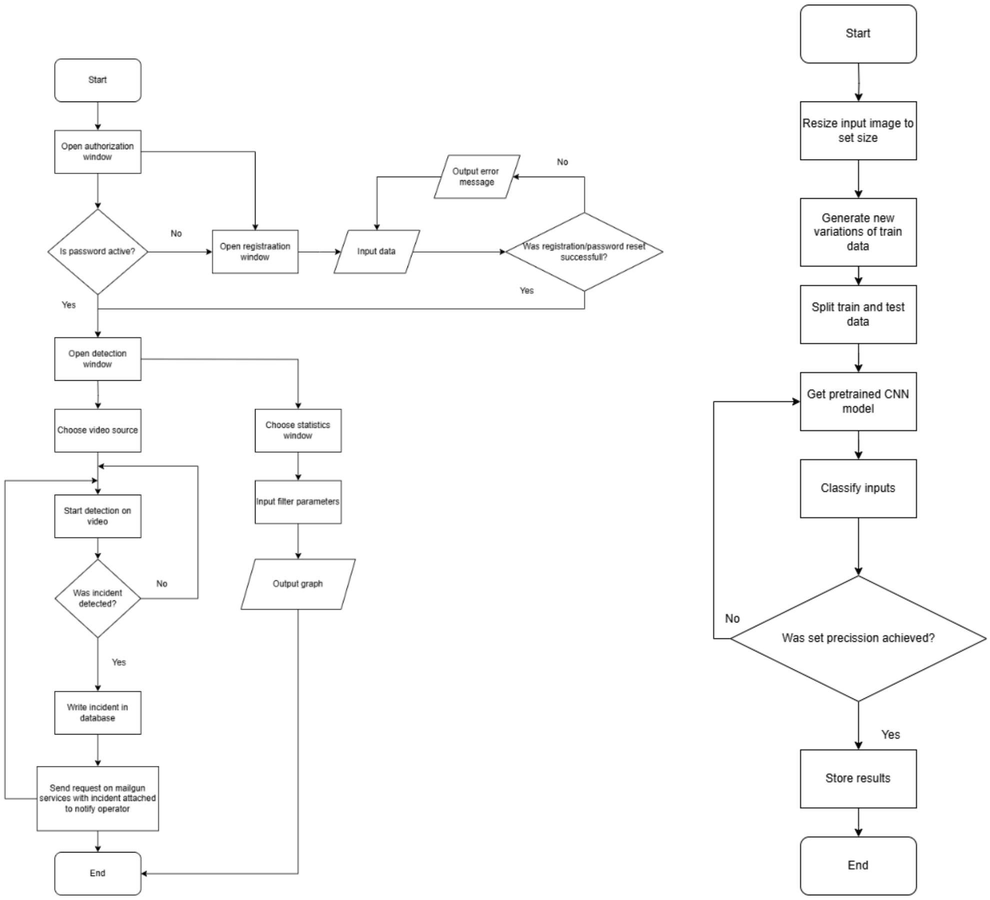
2. Output of results: after passing the image through the main unit, predictions about the location and classification of objects in the image are obtained. These predictions may contain the coordinates of areas where objects are located and the probability of their classification.

3. Thresholding: In order to take into account only reliable predictions, threshold filtering is used, where predictions with low probability or incorrect classifications are discarded.

4. Improving accuracy (Post-processing): Some additional operations can be performed to improve recognition accuracy, such as combining overlapping regions or using information from multiple frames for confidence in recognition.

Looking at the stages of the YOLO algorithm in more detail, the first step of the model is to use CNN on the received image to find objects. As for training the model, the first 20 convolutional layers are trained with ImageNet using temporal averaging pooling and the associated layer. The last fully connected layer of YOLO predicts class probabilities, and accordingly bounding box coordinates.

The general algorithm of the application is shown in Fig. 5.



(a) (b)

Figure 5. Block diagram of the proposed subsystem (a) and the CNN algorithm (b)

The input image is divided into a grid. The YOLO model determines whether a cell is responsible for detecting an object if its center falls within it. Each grid cell provides a prediction of the number of bounding boxes and a confidence estimate for those boxes. Confidence score data indicate how confident the model is that the resulting region contains an object, and how accurately it thinks it was predicted. YOLO assigns a single predictor the task of predicting the object whose prediction IOU is the highest. It specializes predictors of the boundary region. Each predictor improves the prediction of certain dimensions, ratios of object parameters, improving the overall recall score.

The results of the experiment

First, the operation of the pre-trained model in recognition tasks was checked (Fig. 6).To evaluate the effectiveness of its work for real-time traffic emergency recognition, experiments were conducted using YouTube videos containing traffic accidents. The duration of the videos varied from one minute to fifteen minutes. The YOLOv8m model, retrained on the prepared data set, analyzed the frames in the streaming video to detect possible emergency situations.

Figure 6. Recognition result

As a result, with the help of the developed solution, an improvement in the efficiency of response to incidents was obtained by reducing the burden on the operator of the emergency service. As can be seen from the comparison graph in Fig. 7, the number of incidents sent to the emergency service operator without classification after 06/11/2024 is about two thousand, while when applying the classification and dividing the incidents by severity, only about 1750 were sent to the emergency service operator incidents classified as serious requiring emergency care.

Thus, it is clear that without classification, the operator received about 250 minor incidents, which could have taken up valuable time and diverted human resources from situations that really needed help.

 ISSN 1560-8956

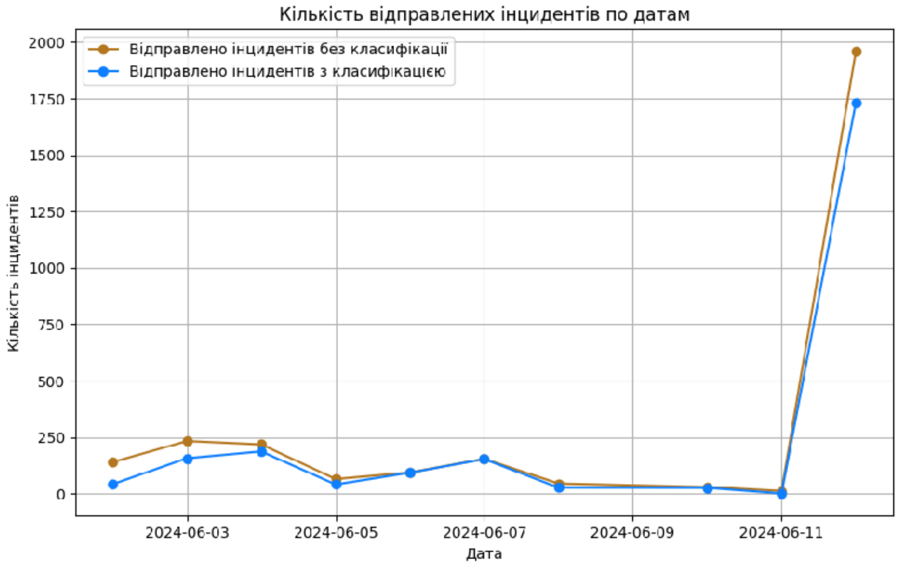


Figure 7. Comparison of the number of incidents sent without and with classification

Also, two sending options are implemented: by sending a letter to the operator's mail, and a notification in the Telegram bot [8]. As a result, in Fig. 8, it can be seen that the option of sending a message in Telegram has small time advantages over the option of sending a letter by mail, equal to a few tenths of seconds. But, taking into account that in the case of sending a large number of incidents, these tens of seconds can add up to a large amount of lost time, and in critical situations every second can cost a life, it is recommended to use the method of informing using the Telegram bot to receive information about the incident.

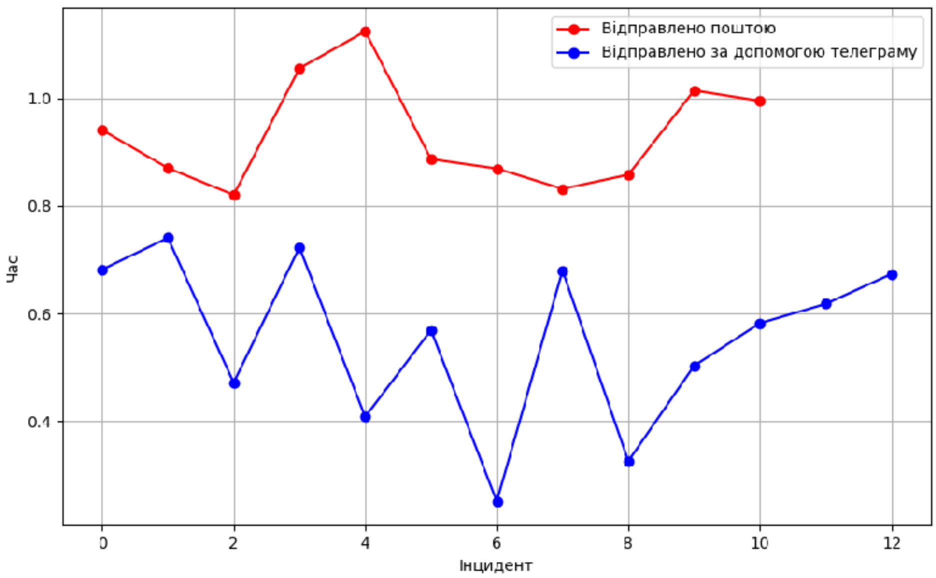


Figure 8. Comparison of sending incident information using developed solutions

It is also worth noting that when sending a message to the post office (Fig. 9), the location is also indicated in the text of the letter, in the given example the location is the text local, in real conditions it will be replaced by the inscription of the address of the camera on which the model recorded the accident, the attached photo incident.

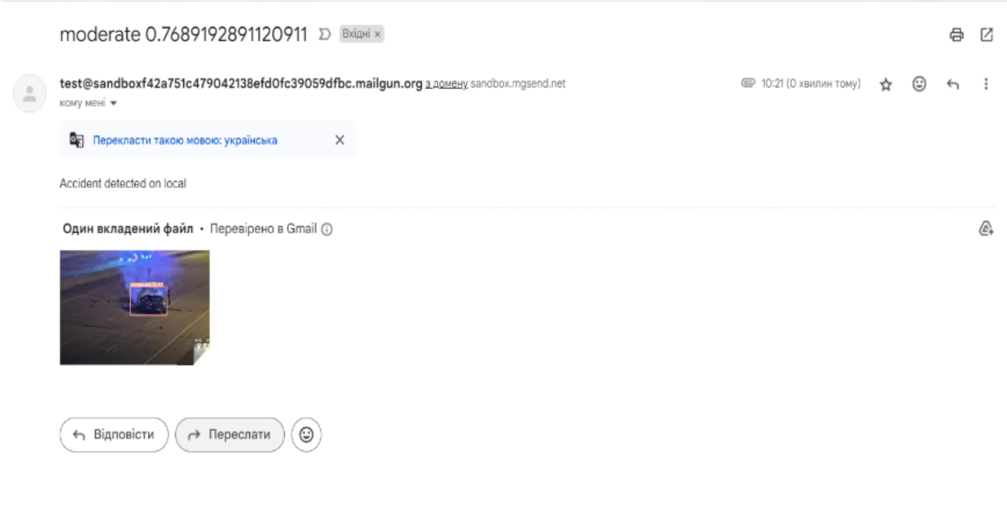


Figure 9. An example of a letter with a recognized incident

By analogy, in the process of sending a letter via the Telegram bot (Fig. 10), the operator receives a message about the classification of the incident, the confidence of the model and the image.

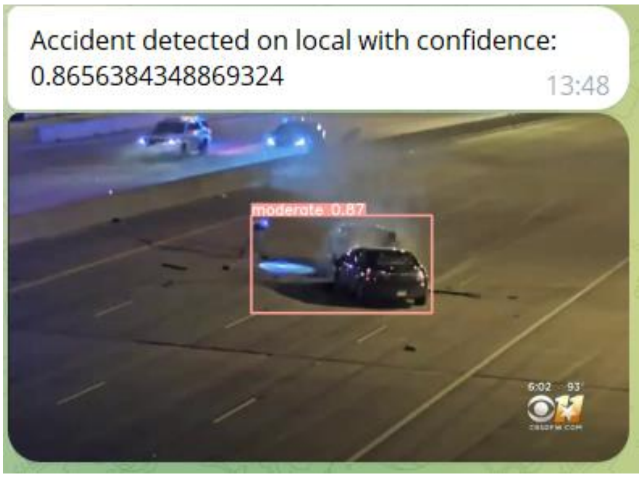


Figure 10. An example of incident recognition in Telegram

These approaches allow the operator to quickly analyze the received results, and in case the operator finds that the model correctly recognized the incident, to quickly send appropriate help to the address specified in the notification.

This is also an advantage over the traditional method of informing about the incident, because there are often cases when a person does not know the exact location where he is, or being in a state of shock, cannot explain to the operator the reason for the call and his location, which can take precious seconds, necessary to save life.

Conclusions

The article considers an approach to solving the problem of increasing the speed of response to emergency events on the road. The proposed solution involves the use of visual surveillance tools capable of recognizing an emergency situation on the road and sending a notification about it to emergency services, and recording information about the incident in a single data store as a component of Smart city.

To create this system, YOLOv8m was used as a neural model, streaming video processing was implemented using the OpenCV library. Accordingly, an architecture consisting of four main parts is created, namely an incident recognition module, in which a neural network searches for incidents in streaming video, a database module that is responsible for all database connections, an authorization module that deals with authorization, user registration and the module of sending an incident, the task of which is to create a letter, indicating the location of the incident, the operator's mail, a photo of the incident, and connecting to the Api Mailgun service to send a message.

From the obtained results, it can be seen that the introduction of classification in the recognition of incidents helps to reduce the burden on emergency service operators, by informing only about serious incidents that require immediate assistance. It was also found that the best way to inform about an incident is the Telegram bot, because the response speed with its help is faster by a few tenths of a second, which in the case of a large number of incidents can save many lives.

REFERENCES

1. *Amy Lam, Darci Heikkinen, Yoyo Chan,* Machine Learning in OpenCV, 2024, pp. 128-138
2. *Jack Chan,* Python API Development Fundamentals, 2029, pp. 163-180
3. *Saurabh Chandrakar*, Python GUI with PyQt, 2023
4. *Khandelwal S.*, Deep Learning for Data Architects: Unleash the Power of Python's Deep Learning Algorithms (English Edition), 2023, pp. 122-125)
5. *Agostino Cortesi, Nabendu Chaki, Khalid Saeed, Rituparna Chaki*, Advanced Computing and Systems for Security, 2020, pp. 125-128
6. *Mohit Sewak*, Practical Convolutional Neural Networks, 2018, pp. 30-45
7. *Munesh Chandra Trivedi, Pascal Lorenz, Ruben Gonzalez Crespo, Suyel Namasudra,* Data Science and Network Engineering, 2023, pp. 129-130
8. *Nicolas Modrzyk*, Building Telegram Bots, 2018, pp 1-15