Available parking places recognition system

Vitalii L. Levkivskyi[1], Dmytro K. Marchuk[1], Nadiia M. Lobanchykova[1], Ihor A. Pilkevych[2] and Dmytro I. Salamatov[1]

[1]Zhytomyr Polytechnic State University, 103 Chudnivska Str., Zhytomyr, 10003, Ukraine
[2]S. P. Korolyov Zhytomyr Military Institute, 22 Myru Ave., Zhytomyr, 10023, Ukraine

Abstract

The paper describes algorithms and methods of working of the available parking places recognition system by analyzing the flow of video data, coming from the city cameras. Search an available parking place is usually a significant problem in modern cities, which requires finding new information technologies, approaches to solve it. Therefore, the purpose of this work is to analyze existing solutions and finding new information technologies for the task of optimizing the parking process. In addition, the software implementation of the proposed algorithms and methods has been implemented. The system was developed with Python and PHP programming languages. With Python an algorithm for recognizing free spaces for car parking and a chatbot for Telegram were implemented. PHP was used for the developing of the content management system with MVC (Model-View-Controller) architectural pattern. MySQL was chosen as the database management system. Communication between the chatbot and the script for recognition is carried out using memory-resident NoSQL DBMS Redis. The implemented system allows drivers to search available parking places in the selected parking lot with a minimum of time, providing information via a Telegram chatbot. The system can be deployed on Apache2 Webserver and Linux OS.

Keywords

parking place, parking lot, Mask R-CNN, neural network, edge computing, computer vision

1. Introduction

Analyzing auto-flow in Ukrainian cities, we can mention that it always increases. This situation causes some problems for citizens. The biggest problem is searching for an available parking place near big supermarkets, multi-story buildings or other areas with crowds of people. Implementation of available parking places recognition system will allow users to determine whether the specified parking lot has available parking places and choose which part of the parking lot these places are located. Modern systems are based on artificial intelligence and machine learning, which, combined with edge computing and intelligent technologies, create digital society systems capable of solving several complex problems and automating management

CS&SE@SW 2021: 4th Workshop for Young Scientists in Computer Science & Software Engineering, December 18, 2021, Kryvyi Rih, Ukraine

levkivskij.vl@gmail.com (V. L. Levkivskyi); dimonoso96@gmail.com (D. K. Marchuk); lobanchikovanadia@gmail.com (N. M. Lobanchykova); igor.pilkevich@meta.ua (I. A. Pilkevych); sal.dima27@gmail.com (D. I. Salamatov)
https://cs.ztu.edu.ua/our-team (V. L. Levkivskyi); https://ztu.edu.ua/ua/structure/faculties/fikt (N. M. Lobanchykova)
0000-0002-1643-0895 (V. L. Levkivskyi); 0000-0001-8675-8047 (D. K. Marchuk); 0000-0003-4010-0308 (N. M. Lobanchykova); 0000-0001-5064-3272 (I. A. Pilkevych); 0000-0002-0018-9814 (D. I. Salamatov)

decisions that improve human life. Therefore, the available parking places recognition system is one of the components of an intelligent city system and a necessary component of modern digital society.

2. The purpose and objectives of the study

The primary purpose of the program system implementation is a minimization of time for searching for an available parking place to park a car. The system must provide quick information about the existence and location of public parking places on a selected parking lot.

The following system components should be implemented:

- chatbot for messenger Telegram;
- website for system management;
- sub-module for searching available parking places in real-time using video cameras installed on this parking lot and connected to the Internet.

Implementation of the objectives for Telegram chatbot consists of the following stages:

1. Creating a possibility for showing a list of available parking lots to search for available parking places.
2. Handling and processing a request for an available parking place searching for a chosen parking lot.
3. Implementation of data processing received from the recognition sub-system and sending these data to the user as text and images.
4. Implementation of the possibility of performing an automatic search of available parking places for a specified parking lot at a limited time.

Implementation of the objectives for the sub-system consists of the following stages:

1. Connection establishment with IP security camera, using required data from a database.
2. Retrieving an image from a video-flow, for the recognition.
3. Car recognition on the image and getting coordinates of cars found by the system.
4. Implementation of an algorithm that determines the availability of a parking place, using the coordinates.
5. Control of input requests from the Telegram bot.
6. Generating a response for the chatbot.
7. Saving of an image with marked available parking places.

3. Analysis of published data

Girshick et al. [1], Hariharan et al. [2, 3], Dai et al. [4] show the efficiency of the Region-Based Convolutional Neural Network (R-CNN). Most approaches for semantic segmentation use information about shape by extraction of CNN functions from masked image regions. In papers

[5, 6] utilized classification of the Fast R-CNN. However, segmentation that precedes recognition, going slowly and is less accurate. The authors of the article [7] have gotten good results. They claim that Mask R-CNN provides a practical foundation for a self-learning strategy. The work's main contribution [8] comprehensively describes region-based convolutional neural networks and their recent improvement. In paper [9], the application of the cutting-edge instance segmentation framework, Mask R-CNN, was demonstrated in cattle counting in different situations. In article [10], a detailed survey of loss combinations was provided and showed improvements on the MS COCO Mask. In a default Mask R-CNN setup, achieved a training speed-up. Today, there are many different companies involved in non-regular passenger transport. Yefimenko et al. [11] consider the possibility of reducing the speed of order processing to improve service delivery but ignore the necessity of quickly finding parking spaces. Cheng et al. [12] consider the conditions for improving the accuracy of mask localization in the process of image segmentation. Huang et al. [13] evaluated the problem of mask quality. They proposed R-CNN with mask scoring, which contains a network block for studying the quality of predicted masks of instances.

4. Materials and methods

We will analyze information technology to find an available parking place. The Region-Based Convolutional Neural Networks architecture has been developed to apply Convolution Neural Networks to object detection tasks.

Mask of R-CNN represents a two-stage structure: at the first stage, an image is scanned, and regions are generated, which can contain an object. At the second stage, suggestions are classified, and bounding boxes and masks are created. Figure 1 shows the Mask R-CNN architecture.

The Mask R-CNN, a deep neural network designed to separate machine learning and computer vision, is used to solve the problem of recognizing cars in a frame obtained from a video stream. An image or video is input, an object frame, an object contour ("mask") and an object class is output.

The Mask R-CNN neural network has proven itself very well in recognizing and distinguishing vehicles in the image. A notable increase in accuracy and performance can be achieved using the Feature Pyramid Network (FPN) as a backbone.

The COCO dataset (Common Objects in Context) is used for data analysis and machine learning [14]. It contains about 330 thousand images, including images of cars, trucks, motorcycles, buses – all types of objects that may be needed to implement this system.

Intersection Over Union (IoU) computes the intersection by joining two bounding rectangles, bounding rectangle for primary truth and bounding rectangle for predicted rectangle by algorithm[15].

When the IoU is 1, it will mean that the predicted box and the confidence bounding box overlap entirely. To detect the object in the image, in case of non-maximum suppression, all bounding boxes with IoU>0,5 takes into account.

Thus, after analysing the capabilities of these technologies, a general algorithm of the system was created:

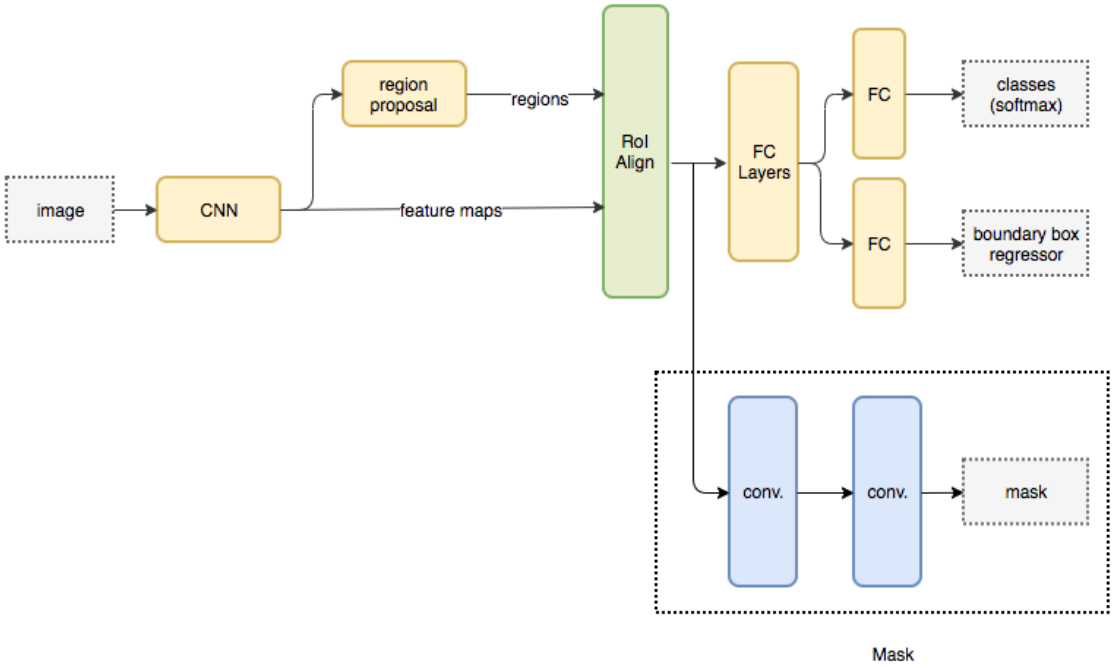

Figure 1: Architecture of the Mask R-CNN.

1. Connection to an IP security camera;
2. Obtaining the initial frame of the parking lot, where all parking places are filled with cars. This frame is used as a basis for determining the availability of the parking space;
3. Recognition of all cars in the initial frame by using the model Mask R-CNN and obtaining their coordinates;
4. Getting the current frame from the video stream;
5. Recognition of all cars in the current frame by using the model Mask R-CNN and obtaining their coordinates;
6. Finding the IoU value (figure 2) between the car at the initial frame and the vehicle at the current frame. This determines the availability of parking space. If this value is less than some constant value (for example, 0.15), then this parking space is considered available;
7. Based on the calculated data, the information is sent to the end-user.

To get more performance on a computer with a 4 GB GPU, the Mask R-CNN settings have been changed. As a result, the best system performance was obtained with the following settings:

```
class MaskRCNNConfig(Config):
 NAME = "coco_pretrained_model_config"
  IMAGES_PER_GPU = 1
  GPU_COUNT = 1
```

```
DETECTION_MIN_CONFIDENCE = 0.7
VALIDATION_STEPS = 10
STEPS_PER_EPOCH = 100
```

The system's main components are a chatbot, an algorithm for recognizing an available parking place, and a control panel. After launching the Telegram chatbot, the user has specific actions available: view information about the bot, show a list of commands on the screen, show available actions in the form of a button menu. When a user selects "Find Free Place" or "`Set Up Auto Search`", a button menu appears with a list of all available locations. After selecting one of the locations, the user gets the result, and in the second case, he needs to choose the time and days of the week for an automatic search. If the user wants to finish working with the chatbot, he can leave the conversation. Otherwise, he can select any action again.

An equally important component of the system is the script for recognizing available space for car parking. The recognition script is run once and executed access to the video stream from the IP camera. The current frame is taken from the received video, and by using the algorithm, it is checked whether free parking places are available in the given location. At the same time, it is checked whether there are requests from the chatbot. Depending on the combination of these factors, the script either generates a response, then sends it to the chatbot, or takes the next frame and starts the process again. The following is the activity diagram of the recognition script (figure 2).

Figure 3 shows the class diagram of the control panel.

The following classes can be distinguished in this diagram:

- Core – core of the MVC;
- Template – class for the view logic;
- DatabaseSignleton – the class for database queries execution and general working with the database;
- "`Admin_Controller`" – the main controller for control panel working. This class uses "`Admin_Model`" – for data processing and "`Admin_View`" – for showing results;
- "`Admin_Model`" – a model which defines working with database in the control panel;
- "`Admin_View`" – an MVC component that takes care of displaying logic. The controller uses class depends on a controller function, shows content to a user.

The following classes and methods have been implemented to create a chatbot and a recognition algorithm using Python (figure 4).

The following classes can be distinguished at this diagram:

- TelegramBot – a category that controls all the logic of the Telegram chatbot;
- FreeParkingDetector – a class that implements an algorithm for recognizing available parking places on video from a video camera and creates a response object to send to the chatbot;
- MaskRCNNConfig – a configuration class for the Mask R-CNN model that sets object recognition parameters on the image. This class is used while creating a model object.
- MysqlClient – a class for implementation of work with DBMS MySQL on the side of Python programming language. Used to insert, edit, delete and retrieve information. It is a "link" between Python scripts and the control panel implemented using PHP.

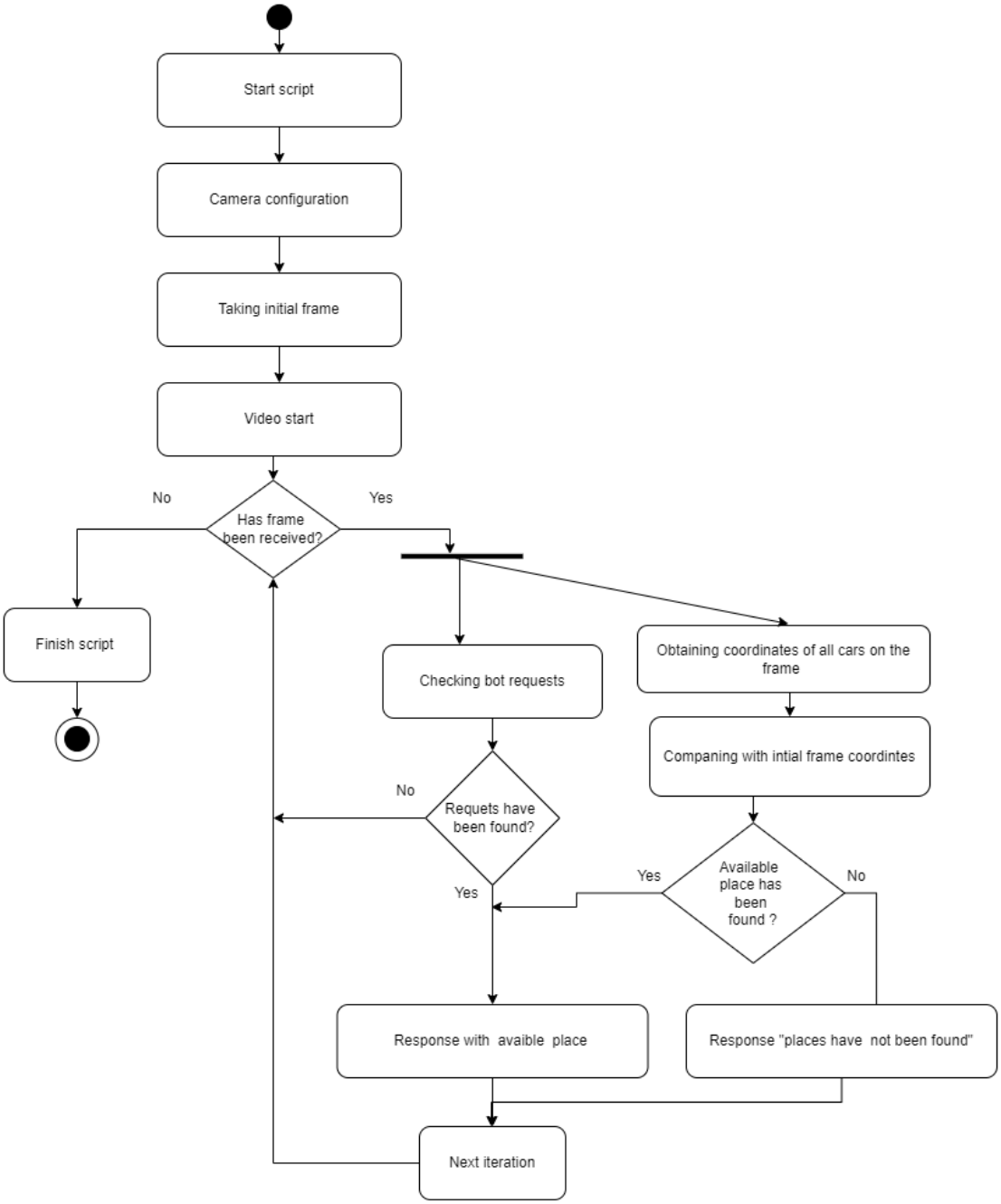

Figure 2: The activity diagram of the recognition script.

- RedisClient – a class for working with NoSQL Redis data warehouse in Python programming language. With Redis, there is constant real-time communication between the Python scripts.

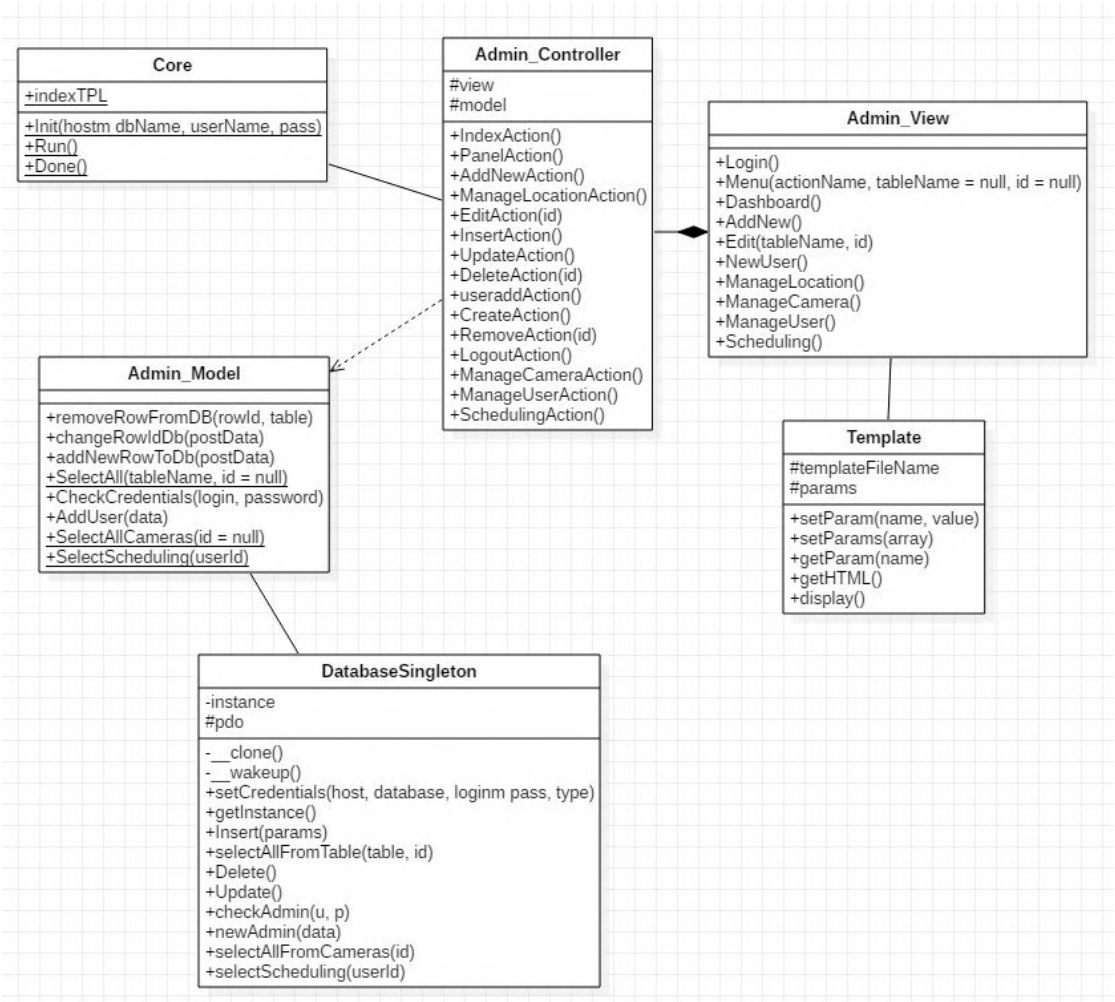

Figure 3: The activity diagram of the recognition script.

The system database “free_parking_detection_database” includes five tables, storing all required information for working the program. The database structure is shown in figure 5.

The “users” table is created to store info about administrators. The “telegram_users” table is used to store data of Telegram chatbot users. The “locations” table is designed to store parking lots and their locations. The “location_camera” table is created to store security camera access settings. This table is related to the “locations” by location identifier. The “detection_scheduling” table is used to store configurations for auto-recognition of a parking place. The table contains time and weekdays, in which the system should search available sites for specified used at a limited time.

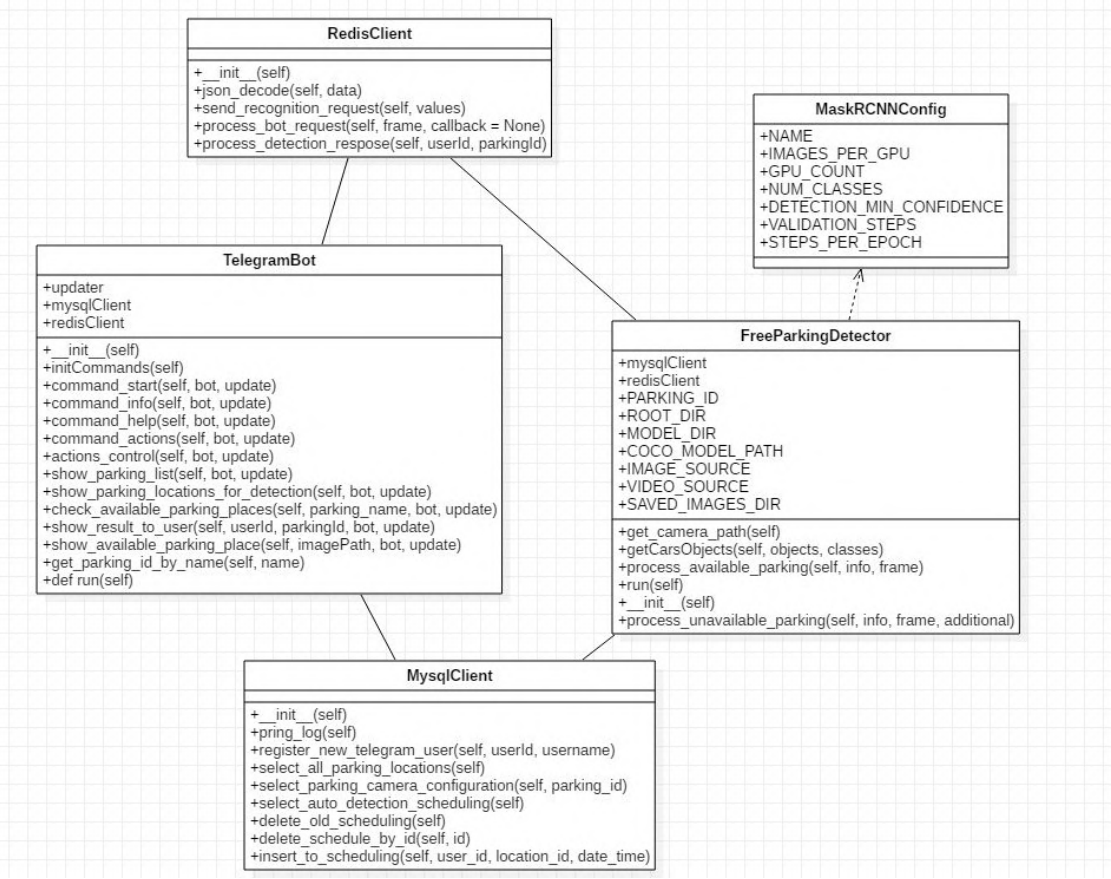

Figure 4: The class diagram of a chatbot and a recognition algorithm.

5. Results

The interface structure of the available parking places recognition system. After logging in, the administrator user sees the main page of the control panel (figure 6). This page has a menu for navigating the admin panel, a welcome message, and an exit button.

Using the menu on the left, the user can choose what he wants to change. If the user selects the “Administrators” menu item, he goes to the admin management page. Here is a form for creating a control panel user that contains login and password fields. There is also a table with current administrators. The table contains all information about administrators and a column with the “Delete” button, with which you can delete an administrator.

To find a bot, you need to search the bot name “Free Parking Recognition” in the Telegram search input. The most important command of the chatbot is “/ actions”. Using this command, a user can open a button menu, which will display the main functionality of the system. Three buttons appear on the screen with the main actions for the user: “Parking list”, “Check availability”, “Set up automatic recognition”.

The “Parking list” button is used to display information about available locations in the

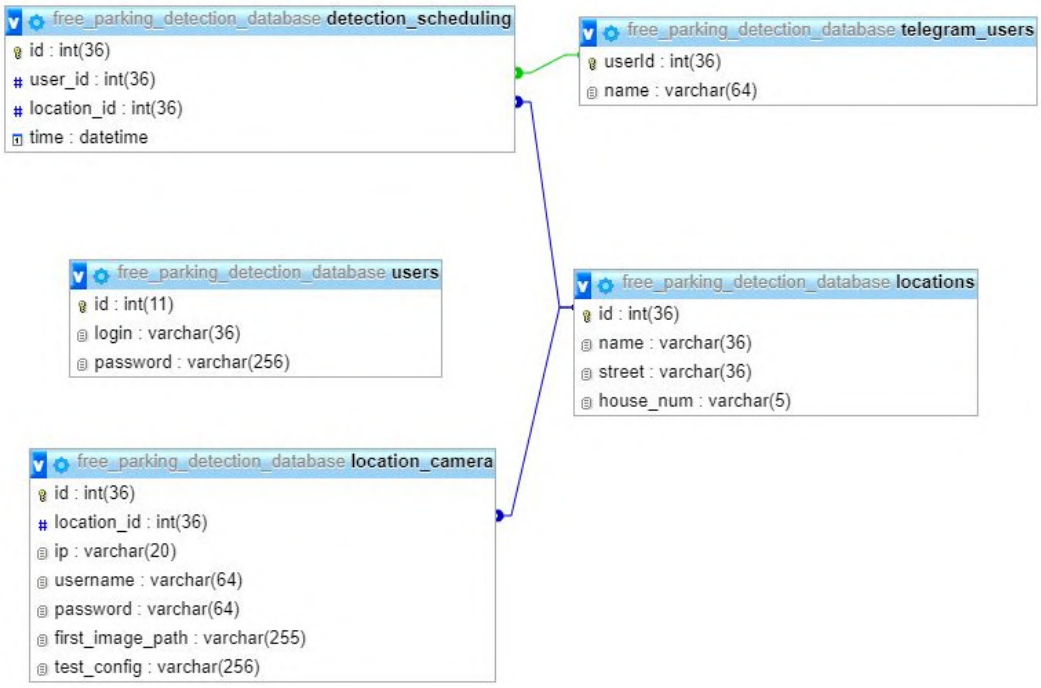

Figure 5: Database structure scheme.

system. This information can be helpful for users who know the address or name of the site but are not sure whether the procedure works in this location. The “Set up automatic recognition” button is needed to adjust the recognition at a specified time. This can be especially useful for Users who plan their trips. After clicking this button, the user must enter the date and time specified in the message. If the date and time are determined correctly, the user will be prompted to select a parking lot for automatic recognition at the specified time.

“Check availability” button allows the user to make sure that there are free parking places in the parking lot he needs at the moment. By clicking on it, the user must select a location from the list of suggested (figure 7).

After the user selects one of the locations, the bot will send a recognition request to the script responsible for recognizing available parking places in this parking lot. If no parking places are found, a message with this information will be displayed. The number of spaces in this parking lot is displayed for users.

If the algorithm at least one available parking place has been recognized, and a corresponding message will be shown (figure 8). The user can view the information about the number of available places, which will allow the user to analyze the situation in more detail and make sure that he will be able to find an open place shortly. A photo of this parking place is attached to the message about the successfully found parking place, which will allow the user to see it in large parking lots quickly.

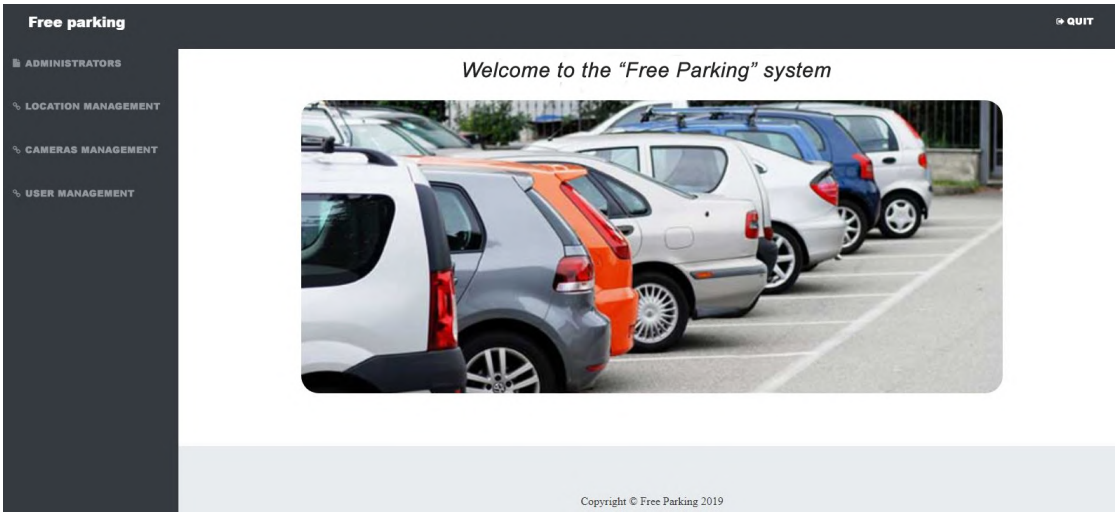

Figure 6: Main page of the control panel.

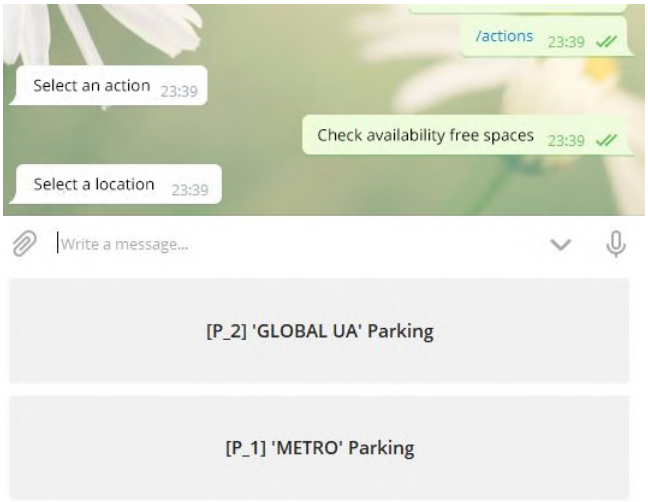

Figure 7: Checking for available places existing.

6. Conclusions

The analysis of neural networks showed that usage of the Mask R-CNN neural network is best suited to solve the problem of recognizing cars from the video flow of a security camera in real-time. This neural network has some advantages over R-CNN and Fast R-CNN. Based on various studies of this neural network, we can say that the main benefits are performance and greater accuracy. In addition, we were able to get the optimal balance between performance and accuracy through the correct selection of network settings.

The general algorithm of working of the system has been offered, and software realization

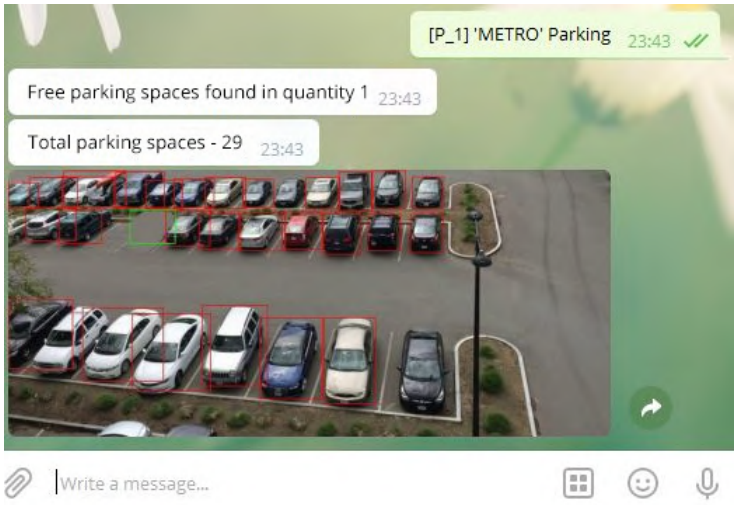

Figure 8: Available place has been found.

of availability for each parking place has been carried out. The maximum balance between productivity and accuracy has been experimentally defined as 0.7.

The activity diagram of the recognition script, the chart of the control panel classes in PHP, the graph of the chatbot classes, and the recognition algorithm in Python, the structure scheme of the database have been offered. Software implementation of the chatbot for Telegram messenger and a script for recognizing an available place for car parking has been developed.

References

[1] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587. doi:10.1109/CVPR.2014.81.

[2] B. Hariharan, P. Arbeláez, R. Girshick, J. Malik, Simultaneous detection and segmentation, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 297–312. doi:10.1007/978-3-319-10584-0_20.

[3] B. Hariharan, P. Arbeláez, R. Girshick, J. Malik, Hypercolumns for object segmentation and fine-grained localization, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 447–456. doi:10.1109/CVPR.2015.7298642.

[4] J. Dai, K. He, J. Sun, Convolutional feature masking for joint object and stuff segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, 2015, pp. 3992–4000. doi:10.1109/CVPR.2015.7299025.

[5] P. O. Pinheiro, T. Y. Lin, R. Collobert, P. Dollár, SharpMask : Learning to refine object segments, in: European Conference on Computer Vision, Springer International Publishing, 2016, pp. 75–91. URL: https://vitalab.github.io/article/2017/04/11/sharpmask.html.

[6] J. Dai, K. He, Y. Li, S. Ren, J. Sun, Instance-sensitive fully convolutional networks, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 534–549. doi:10.1007/978-3-319-46466-4_32.

[7] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, in: IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2980–2988. doi:10.1109/ICCV.2017.322.

[8] P. Bharati, A. Pramanik, Deep Learning Techniques—R-CNN to Mask R-CNN: A Survey, in: A. K. Das, J. Nayak, B. Naik, S. K. Pati, D. Pelusi (Eds.), Computational Intelligence in Pattern Recognition, Springer Singapore, Singapore, 2020, pp. 657–668. doi:10.1007/978-981-13-9042-5_56.

[9] B. Xu, W. Wang, G. Falzon, P. Kwan, L. Guo, G. Chen, A. Tait, D. Schneider, Automated cattle counting using Mask R-CNN in quadcopter vision system, Computers and Electronics in Agriculture 171 (2020) 105300. doi:10.1016/j.compag.2020.105300.

[10] R. S. Zimmermann, J. N. Siems, Faster training of Mask R-CNN by focusing on instance boundaries, Computer Vision and Image Understanding 188 (2019) 102795. doi:10.1016/j.cviu.2019.102795.

[11] A. Yefimenko, A. Kuzmenko, H. Marchuck, R. Petriv, I. Suhoniak, Geoinformation system for managing non-regular passenger transportation, E3S Web of Conferences 166 (2020) 05002. doi:10.1051/e3sconf/202016605002.

[12] T. Cheng, X. Wang, L. Huang, W. Liu, Boundary-preserving Mask R-CNN, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Computer Vision – ECCV 2020, Springer International Publishing, Cham, 2020, pp. 660–676. doi:10.1007/978-3-030-58568-6_39.

[13] Z. Huang, L. Huang, Y. Gong, C. Huang, X. Wang, Mask scoring r-cnn, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 6409–6418. URL: https://openaccess.thecvf.com/content_CVPR_2019/papers/Huang_Mask_Scoring_R-CNN_CVPR_2019_paper.pdf.

[14] COCO - Common Objects in Context, 2020. URL: https://cocodataset.org.

[15] R. Khandelwal, Computer Vision: Instance Segmentation with Mask R-CNN, 2019. URL: https://towardsdatascience.com/computer-vision-instance-segmentation-with-mask-r-cnn-7983502fcad1.