Software complex for identifying objects and managing during sorting and loading operations

Aleksandr Tyutyunnik[a], Alexey Lazarev[a] and Pavel Kharlamov[a]

[a] *Branch of the National Research University Moscow Power Engineering Institute in Smolensk, 1 Energeticheskiy proyezd, Smolensk, 214013, Russian Federation*

Abstract

The paper presents a developed software complex for identifying objects and managing during sorting and loading operations at the preparatory stage of construction work. The software complex is based on the use of artificial neural networks of deep learning and fuzzy models.

Keywords 1

Software complex, object identification, neural networks with deep learning, fuzzy model, sorting and loading operations

1. Introduction

Nowadays, the issue of digitalization and automation of various processes, actions and operations is of vital importance. Considering its priority and vector of its formalization in the national policy of Russia, it is becoming more and more promising and in demand in various areas of industry, services, and science. More and more processes are being technologically implemented in the form of hardware and software complexes or application software. The main idea of the software complex for identifying objects and managing during sorting and loading operations (SC) described and implemented by the authors solves one of the components of the above-mentioned issue and touches upon the construction sphere, namely, preparatory work for the construction of certain structures.

In conditions of the increased competitive situation in the market of building services the following list of problems becomes actual: reduction of time limits of performance of stages and increase in quality of civil work by acceleration and change of technologies both for a stage of erection of a construction, performance of finishing works and other subsequent processes, and for a preparatory stage which key task is preparation of resources and materials for carrying out of works. Moreover, each subsequent phase requires the supply of the necessary consumables. The SC offered by the authors is aimed exactly at solving such a problem within the framework of sorting and loading works.

At the initial development stage, the project was based on the task of accounting for construction resources, such as gravel and crushed stone of various fractions, sorted by nominal grain sizes in accordance with GOST 8267-93 (according to Russian law), located on large sites in the open sky with the help of specialized equipment, carrying out the loading of consumables and subsequent transportation to the point of loading, while each machine (vehicle) has a number that binds it to one or another fraction of gravel.

2. Technical statement of the problem

Based on the task described above, certain requirements to SC are highlighted. Firstly, the complex should allow to count the number of fields from the incoming video stream at a given area of the image

Russian Advances in Fuzzy Systems and Soft Computing: selected contributions to the 8-th International Conference on Fuzzy Systems, Soft Computing and Intelligent Technologies (FSSCIT-2020), June 29 – July 1, 2020, Smolensk, Russia
EMAIL: tyutyunnik.aa@yandex.ru; anonymous.prodject@gmail.com; pavel_kharlamov.mp67@mail.ru
ORCID: 0000-0002-6394-4514; 0000-0003-3252-0409; 0000-0003-2413-9827

and determine their location in real time. As the location of the fields is dynamic, it is necessary to provide a mechanism of building a training sample for training cascade classifiers (similar cascade classifiers Haar, LBP). The system uses an advanced image processing algorithm based on a neural network with deep training, which analyzes the changes occurring on each frame of the image, with an input stream of at least 30 frames per second, which allows to process and analyze the image in real time. Collected data for a given interval of time from all video streams should arrive in a single SQL database for further statistical processing. Secondly, the program should read the vehicle number according to the above principle and recognize the fraction of gravel or rubble to which this number is attached. Thirdly, SC should process the received data and determine the field from which the crushed stone will be transported to the issuing or unloading points by sending the appropriate command.

The software product is based on the Python programming language and is multiplatform, allowing for adaptation to a specific customer operating system (OS). SC should consider the following factors:

- process a black and white image (night mode);
- consider a variety of weather conditions (rain, snow, hail, etc.) – with a strong distortion of the image is acceptable to reduce accuracy, up to the inability to count;
- consider changes in the brightness of the image in the night mode (light from headlights, instability of lighting devices, etc.);
- consider large quantities of gravel or crushed stone grains in a dynamic field;
- provide for multi-channel command transmission to vehicles.

3. Proposed solution

The above-mentioned issue has various solutions – one of them is the creation of software complex, which main functions are: implementation of a photo or video, identifying objects in the received images, processing of the obtained results and implementation of expert support solutions when performing the sorting and loading operations. Thus, the authors modeled and later developed a complex that meets the requirements. At the stage of technical statement of the problem and modeling, four modules responsible for a certain number of functions in the system were formed:

- module “Collection and primary analysis of data” – identification of the objects on the field;
- module “Systematization and fuzzification” – main part of analysis of data;
- module “Expert decision support / Fuzzy Module” – making of the decision;
- module “Output of the results of the decision support system”.

Modules of the developed SC are presented on the Figure 1.

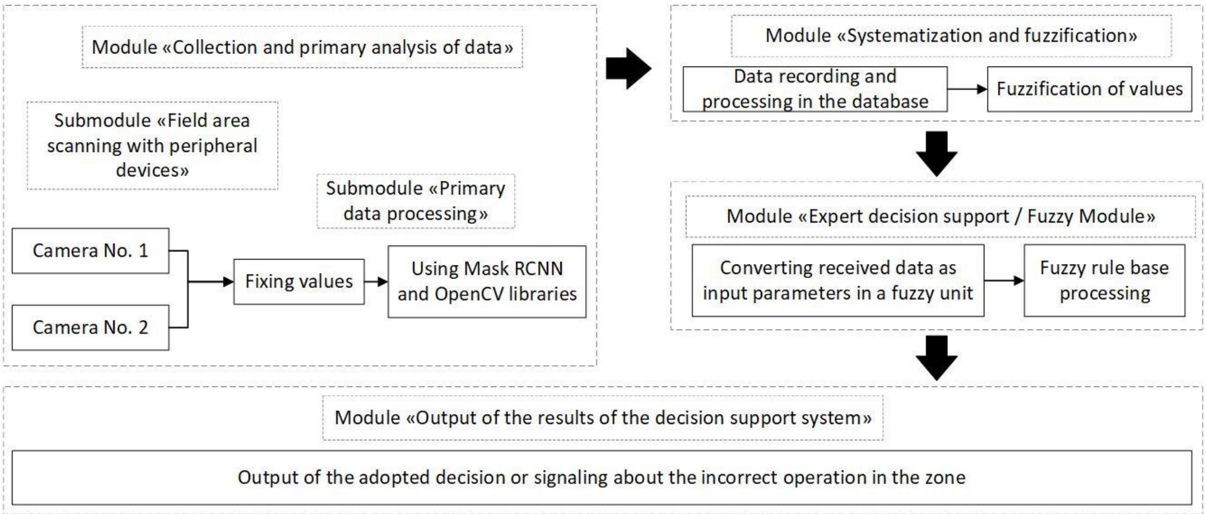

Figure 1: Scheme of modules of the developed SC

3.1. Image processing with deep-learning neural networks and fuzzy models

Open Source Computer Vision (OpenCV) libraries in Python programming language [1-4] were used to provide the possibility of image processing from direct video stream. This approach allows to implement a universal product, which is multiplatform, and if necessary, the release of critical updates will reconstruct them for a specific OS – Microsoft Windows, Linux, or OS X [5]. Besides these factors it is necessary to note the special approach applied in realization of SC – binding to the certain object is carried out on its identifier (the state number of the vehicle) as possibility of replacement of the vehicle on another is supposed, therefore installation of gauges on the vehicles (for example, geolocation) is impossible.

Definition of key objects in the image is implemented in the proposed SC with the help of Mask RCNN library, as it already contains a set of pre-trained neural network models, has an open source code and allows you to use the multi-computing technology Compute Unified Device Architecture (CUDA) based on a graphics processor, which seems to be a significant factor in the use of the end product, based on the fact that at the time of project planning the following parameters are unknown: the size of the field area and the number of working machines [6]. The specified library provides an opportunity to teach his own model, which allows you to take into account when teaching the neural network of various additional designations on the used site, for instance, signal pointers with container designations. Figure 2a shows the result of testing the capabilities of the specified library in urban traffic conditions, which shows that Mask RCNN has successfully coped with the task of detecting such objects as a car, pedestrian, and traffic lights [7]. During the development of the program structure of SC and the implementation of the model training process, plots were compiled based on the use of TensorFlow (Figure 2b) [8]. One of them shows the result of vehicle recognition, the second shows the result of material recognition and the third – other various objects [9]. As you can see, training a material recognition model requires a larger amount of data, since there are no trained models for material models [10].

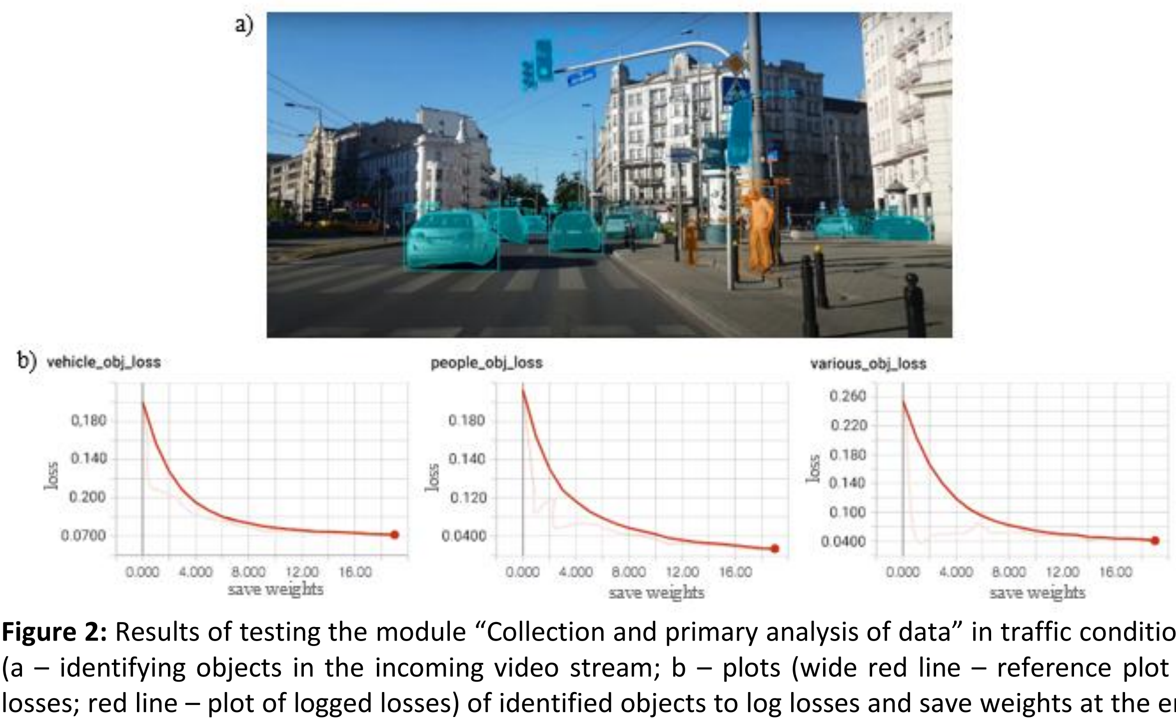

Figure 2: Results of testing the module "Collection and primary analysis of data" in traffic conditions (a – identifying objects in the incoming video stream; b – plots (wide red line – reference plot of losses; red line – plot of logged losses) of identified objects to log losses and save weights at the end of every epoch)

In the next stage of recognition, it is necessary to separate the unique identifier (number) of the vehicle from the selected object, for this purpose the Tesseract library was used, which uses a neural network based on linear character recognition technology. Optical character recognition (OCR) allows in Figure 3 to recognize a set of identification characters – the machine number [11]. From Figure 3 you can see that the camera with the configured OpenCV library allows you to recognize the machine number and, accordingly, the trajectory of the specialized vehicle to the target. Also, modification of this library allows you to teach the neural network to identify additional objects, in the case shown in Figure 3 – a pile of sandy soil. As a result, it becomes possible to implement the identification of material during its transportation and loading, which will allow to compare materials to belong to one fraction for bulk materials and establish similarities between objects belonging to other types of building materials. The use of this modification significantly reduces the risks of loading erroneous material. To organize a centralized control, it is proposed to use several cameras around the perimeter of the site. Thanks to this, the system will be able to identify the observed object as it moves towards the target, and in case of violation of a given trajectory to notify a specialist.

The implementation of the software product scheme is shown on Figure 4. Submodule No.1 is an algorithm for calculating the slope angle, the second – the lower boundary of the image and the corresponding desired angle. On the 3 submodule the algorithm of the upper boundary and its corner search is presented. After all corresponding angles are found, block 4 performs the calculation of side borders of the vehicle number. After calculation of all necessary parameters, an array of selected symbols of the searched number is submitted to the Tesseract software input. As mentioned earlier, this library uses a neural network, in this case the neural network is trained using the image of the symbol of the vehicle license plate [12]. This process is presented in block 5 of Figure 4. Semantic segmentation technology is used to recognize the object of materials – the image from the camera is broken into quantitative objects. The process in this case is the application of a segmenting neural network, at the input of which the image with the object, such as sand, is fed [13]. Application of several cameras will allow to shoot the object from several sides – so you can achieve accuracy in determining the size of the object [14, 15].

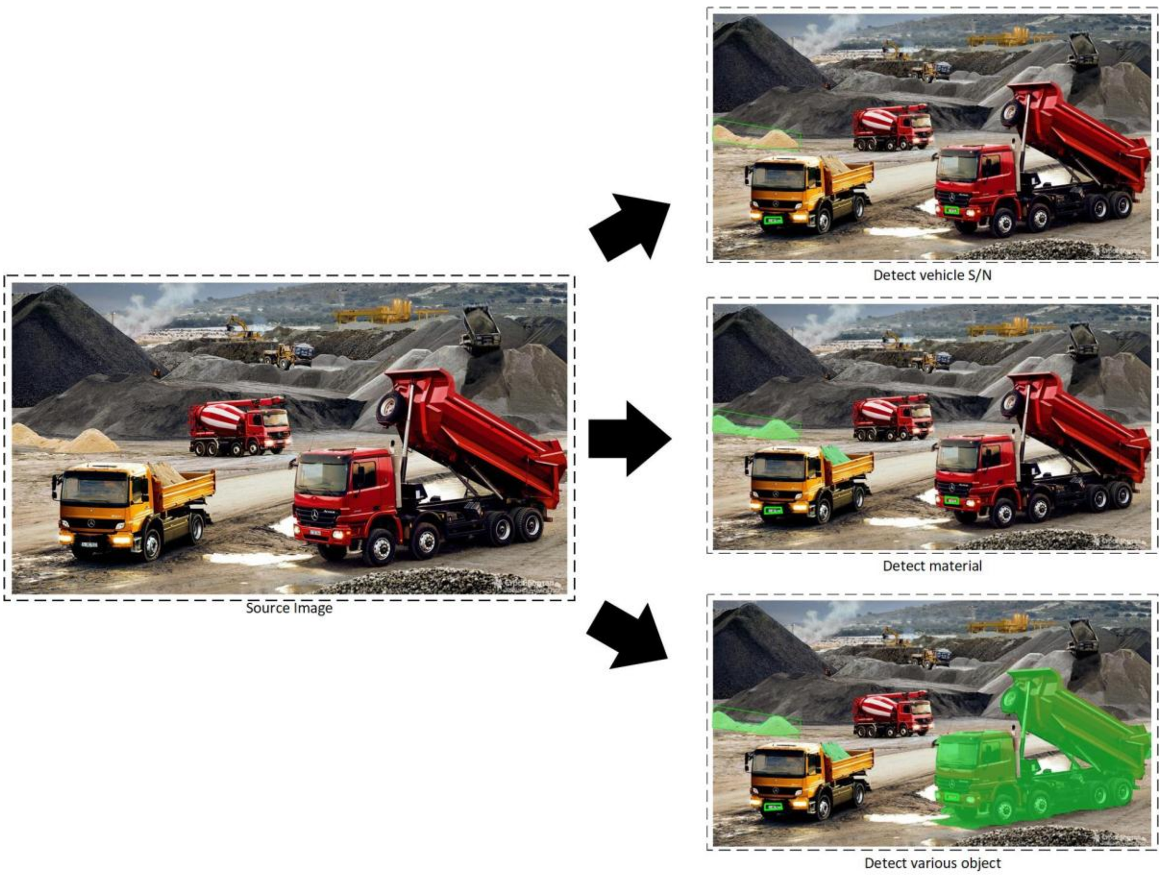

Figure 3: Recognition of target objects by means of OpenCV and semantic segmentation of the image

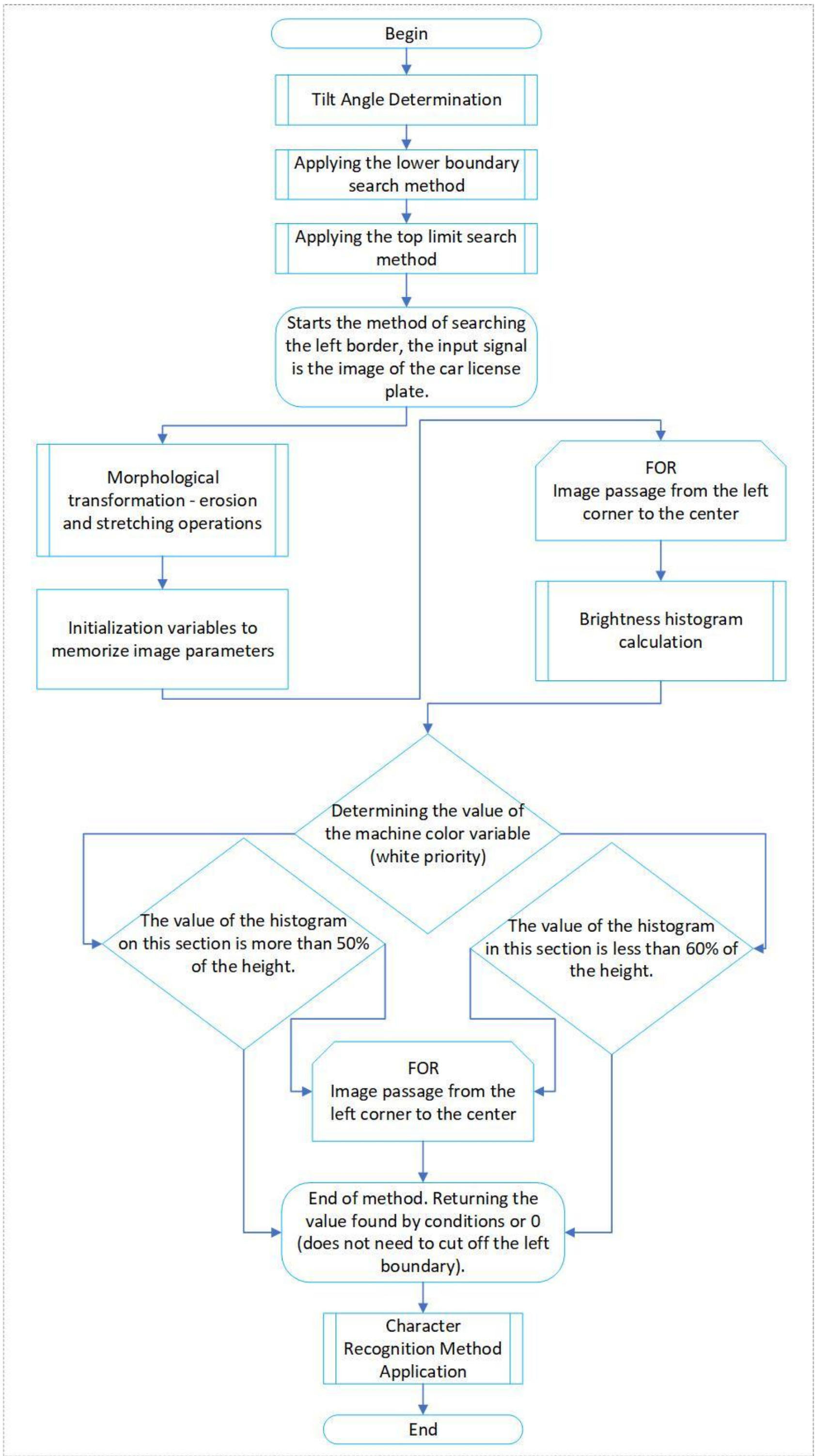

Figure 4: Block diagram of recognition of objects using OpenCV

3.2. SQL database design

The organization of a centralized control system of sorting and loading operations also requires a database (DB), which will record data on the movement of the vehicle, the transported material and the time of loading and unloading operations. Interaction of software is implemented with relational DB – MySQL or PostgreSQL [16, 17]. Such DB will allow you to connect to any OS. DB consisting of four tables has been created for SC being developed: table of vehicles ("Vehicle"), table of materials ("Resources"), table of work zones ("Section"), and table of orders ("Order").

The table “Vehicle” has the following fields:

- id (vehicle identifier, key field, unique);
- name (name of the vehicle);
- serial (vehicle serial number);
- status (current vehicle status: free/busy).

The table “Resources” assumes the following fields:

- id (material identifier, key field, unique);
- name (material name);
- amount (quantity of material on site).

The table “Section” contains the following key fields:

- id (zone identifier, key field, unique);
- material (material located within the specified zone).

The table “Order” contains the following fields:

- id (order number, key field, unique);
- vehicle (identifier of the machine servicing the order);
- resource (material identifier used in the execution process);
- date (date and time of the order);
- section (material shipment section number);
- status (status of order execution).

The table “Order” is linked to the table “Vehicle” by primary key id with the field vehicle; also, the field resource is linked to the key field id in the table “Resources”. The table “Order” is linked to the table “Section” for the specific zone assignment. To implement the design scheme of DB used software MySQL Workbench (Figure 5) [18, 19].

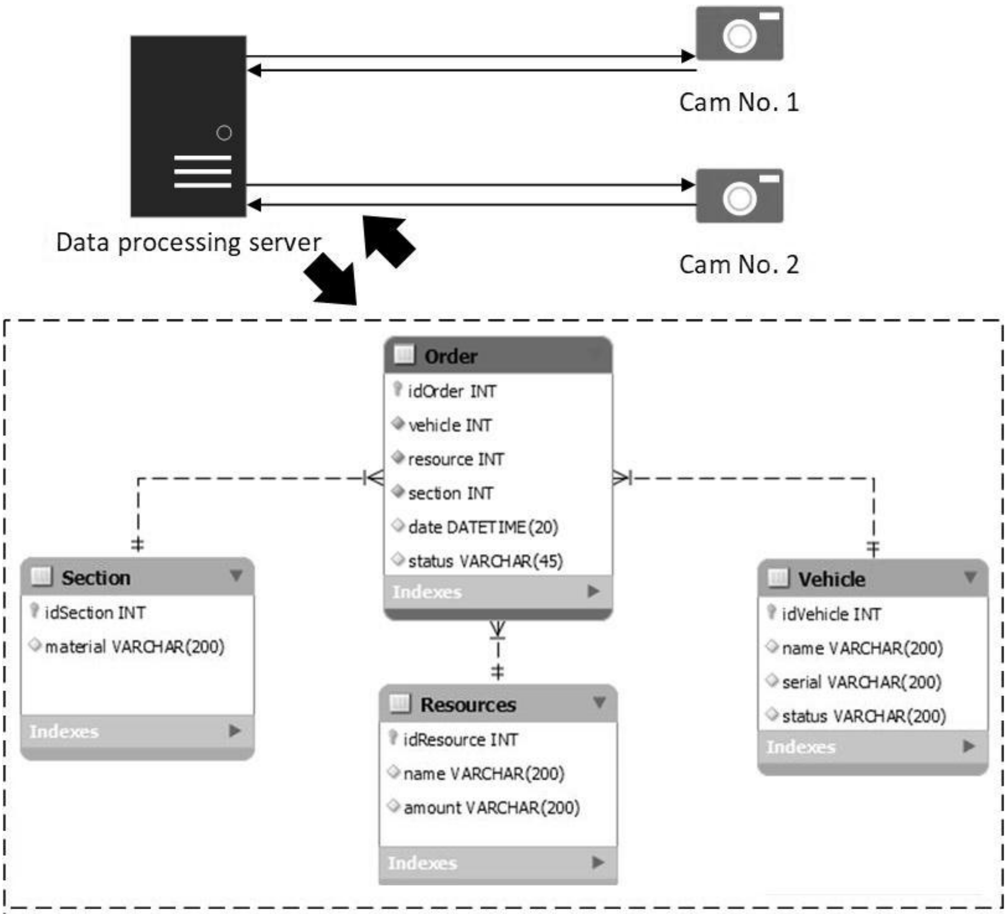

Figure 5: Database table model diagram

3.3. Fuzzy output module – decision support module

The decision-making module of the developed SC uses the data loaded into DB, converting it with the help of a fuzzifier [20, 21]. It uses the field id of the table “Vehicle” which is grouped by binding to the field section, material of the table “Resources”, the field id of the table “Section”. Based on the initial parameters, the base of fuzzy rules is used to determine the correspondence between the machine number and the field segment where the loading and unloading operations are executed and the work execution status. At the design stage, the authors created a model of the fuzzy output module in Fuzzy Logic Designer in MatLAB (Figure 6), the base of fuzzy rules which was later adapted for the programming language (Python).

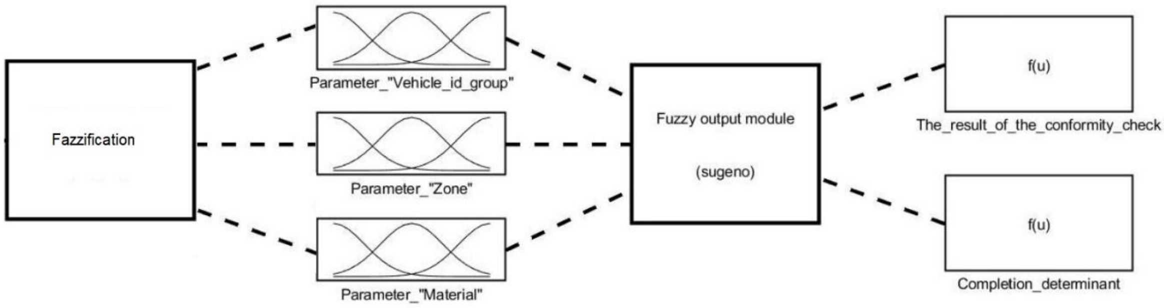

Figure 6: The structure of the fuzzy output module

To make a decision, the following rules (R) were included in the base of fuzzy rules:

R1: **IF** (Parameter_”Vehicle_id_group” IS Zone_n) AND (Parameter_”Zone” IS Zone_n) AND (Parameter_”Material” IS NOT None) **THEN** (The_result_of_the_conformity_check IS defined);

R2: **IF** (Parameter_”Vehicle_id_group” IS Zone_n) AND (Parameter_”Zone” IS NOT Zone_n) AND (Parameter_”Material” IS NOT None) **THEN** (The_result_of_the_conformity_check IS Mismatch_determined);

R3: **IF** (Parameter_”Material” IS None) **THEN** (Completion_determinant IS Work_is_done);

R4: **IF** (Parameter_”Material” IS NOT None) **THEN** (Completion_determinant IS Work_continues).

4. Conclusion

The developed SC, based on application of neural networks with deep training and fuzzy logic, allows to solve the basic problems on increase of efficiency of performance of civil work, and in particular a preparatory stage which key element is sorting and loading works [22, 23]. The above described methods of realization of stages of work of the complex unite perspective, modern and actual approaches and software of image processing and the subsequent decision making.

5. References

[1] A. Sharma, V. R. Shrimali, M. Beyeler, Machine Learning for OpenCV 4: Intelligent algorithms for building image processing apps using OpenCV 4, Python, and scikit-learn, 2nd ed., Birmingham, Packt, 2019.
[2] A. F. Villan, Mastering OpenCV 4 with Python: A practical guide covering topics from image processing, augmented reality to deep learning with OpenCV 4 and Python 3.7, Birmingham, Packt, 2019.
[3] Z. Xiangyu, Ch. Wai Ting, OpenCV (open source computer vision library) – based video target tracking algorithm, 2013. Patent No. CN102915545A, Filed Sep, 20 (2012), Issued Feb. 6th., 2013.

[4] OpenCV Documentation, 2020. URL: https://docs.opencv.org/master/db/d27/tutorial_py_table_of_contents_feature2d.html.
[5] R. Vyas, Face recognition using feature extraction and neuro-fuzzy techniques, 2012. URL: https://www.researchgate.net/publication/267425870_FACE_RECOGNITION_USING_FEATURE_EXTRACTION_AND_NEURO-FUZZY_TECHNIQUES.
[6] J. Sanders, E. Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming. 1st ed., United States of America: Addison-Wesley Professional, 2010, 238-248.
[7] S. Bose, C. Colaco, M. Bagwe, Traffic Detection Using OpenCV, 2020. URL: https://www.researchgate.net/publication/341335122_Traffic_Detection_Using_OpenCV. doi: 10.22541/au.158931021.15827522.
[8] A. Gulli, A. Kapoor, S. Pal, Deep Learning with TensorFlow 2 and Keras: Regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API, 2nd Edition, Birmingham, Packt, 2019.
[9] M. Rubel, S. Waheed, K. Ahmed, A. Hanifa, Real Time Traffic Sign Detection and Recognition using Adaptive Neuro Fuzzy Inference System. Communications on Applied Electronics,3 (2015)1-5. URL: https://www.researchgate.net/publication/283553431_Real_Time_Traffic_Sign_Detection_and_Recognition_using_Adaptive_Neuro_Fuzzy_Inference_System. doi:10.5120/cae2015651877.
[10] B. Planche, E. Andres, Hands-On Computer Vision with TensorFlow 2: Leverage deep learning to create powerful image processing apps with TensorFlow 2.0 and Keras, Birmingham, Packt, 2019.
[11] S. Reddy, S. Ahmed, P. Akshay, Number Plate Recognition using OCR. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2019, 466-471. URL: https://www.researchgate.net/publication/331848293_Number_ Plate_Recognition_using_OCR. doi:10.32628/CSEIT1952123.
[12] B. Cyganek, Neuro-fuzzy System for Road Signs Recognition, 2008, 503-512. URL: https://www.researchgate.net/publication/225624984_Neuro-fuzzy_System_for_Road_Signs_Recognition. doi:10.1007/978-3-540-87536-9_52.
[13] S. Gorbachev, V. Syryamkin, Adaptive Neuro-Fuzzy Recognition Technology Intersecting Objects. Applied Mechanics and Materials, 2015. 683-688. URL: https://www.researchgate.net/publication/277662762_Adaptive_Neuro-Fuzzy_Recognition_Technology_Intersecting_Objects. doi:10.4028/www.scientific.net/AMM.756.683.
[14] B. Somasundaram, S. Subramanian, Semantic Video Segmentation and retrieval system, Latvia, LAP LAMBERT Academic Publishing, 2019.
[15] H. Singh, Practical Machine Learning and Image Processing: For Facial Recognition, Object Detection, and Pattern Recognition Using Python, New York, Apress, 2019.
[16] P. DuBois, MySQL (5th Edition) (Developer's Library), United States of America, Addison-Wesley Professional, 2013.
[17] A. Beaulieu, Learning SQL: Generate, Manipulate, and Retrieve Data. 3rd ed., United States of America: Addison-Wesley Professional, 2010.
[18] M. McLaughlin, MySQL Workbench: Data Modeling & Development (Oracle Press) 1st ed., Kindle Edition, New York, McGraw-Hill Osborne Media.
[19] H. -J. Schönig, Mastering PostgreSQL 12: Advanced techniques to build and administer scalable and reliable PostgreSQL database applications, 3rd ed., Birmingham, Packt, 2019.
[20] A. Dorzhigulov, A. James, Deep Neuro-Fuzzy Architectures, 2020. URL: https://www.researchgate.net/publication/332301666_Deep_Neuro-Fuzzy_Architectures. doi:10.1007/978-3-030-14524-8_15.
[21] J. M. Keller, D. Liu, D. B. Fogel, Fundamentals of Computational Intelligence: Neural Networks, Fuzzy Systems, and Evolutionary Computation (IEEE Press Series on Computational Intelligence). 1se ed. Hoboken, New Jersey: Wiley-IEEE Press, 2016.
[22] J. M. Wahlen, S. P. Baginski, M. Bradshaw, Financial Reporting, Financial Statement Analysis and Valuation. 9th ed. Boston, Massachusetts, United States: Cengage Learning, 2017.
[23] J. Alexander, Financial Planning & Analysis and Performance Management. 1st ed. Hoboken, New Jersey, United States: Wiley, 2018.