автоматики. Врахування специфічних вимог до роботи з двигунами змінного струму дозволить забезпечити надійну та безпечну роботу стрілочних переводів в умовах інтенсивної експлуатації, особливо при впровадженні високошвидкісного руху.

Список використаних джерел

4. Kuchar K., Holasova E., Pospisil O., Ruotsalainen H., Fujdiak R. and Wagner A. Hunting Network Anomalies in a Railway Axle Counter System. 2023. Vol. 23, Is. 6. 23(6). 3122. URL: https://doi.org/10.3390/s23063122
5. Axle Counter Overlay System. URL: https://www.frauscher.com/en/ fields-of-application/references/Axle-Counter-Overlay-System-UK_re_1939 (дата звернення 04.01.24).
6. Gerhátová Z., Zitrický V., Klapita V. Industry 4.0 Implementation Options in Railway Transport. Transp. Res. Procedia. 2021. 53. 23–30.
7. Heinrich M., Gölz A., Arul T., Katzenbeisser S. Rule-based Anomaly Detection for Railway Signalling Networks. arXiv. 2020. arXiv:2008.05241.

УДК 621.396.967

***Sadovnykov B.I.**, PhD student*
Ukrainian State University of Railway Transport, Kharkiv
***Zhuchenko O.S.**, PhD. Associate Professor*
Ukrainian State University of Railway Transport, Kharkiv

DEVELOPMENT OF AN ALGORITHM FOR DETECTING MOVING OBJECTS IN IMAGES FROM A REAL-TIME VIDEO STREAM

One of the tasks of recognizing objects in video is to find their location in the current frame. In algorithms that use neural networks, such as SSD[1], YOLO[2], search and classification are performed by a single model and are inseparable steps. In the case where there is a computing node in the system that has enough power for complex image processing and can do it before the server that recognizes objects, separating the object search and recognition operations can reduce the latency, load on the network and the recognition node.

The peculiarity of video data is that it consists of frames that have a strict sequence. Provided that the video is recorded on a static camera, such as those used in video surveillance systems, objects can be searched for by calculating the difference between adjacent frames. This technique is known as background removal, and there are a number of algorithms for solving this problem. The main problem is their computational complexity and speed. The results of such a method may not be suitable for further classification and a separate algorithm must be developed for any corrections. These algorithms are based on the calculation of a background model, which is then used to highlight new objects in the image. These algorithms have a high processing time, so we will develop an alternative algorithm with lower computational requirements.

The first step of the algorithm is to convert the colors of the current frame to grayscale, since the previous frame has already been converted in the previous iteration of the algorithm. Another thing to consider is the number of frames of the video stream per second. In the case of a high number of frames and relatively slow-moving objects in the video, the difference between two adjacent frames may be insignificant, and the number of difference calculations is similar to frames with significant differences. To increase the speed of the algorithm, you can skip frames and find the difference only between every n-frames, where the value of n depends on the FPS of the video, the speed of objects in it. This leads to an artificial decrease in the number of frames per second, which reduces the use of computing resources and prevents the problem of a queue of frames for processing.

Next, you need to calculate the absolute difference in the values of the corresponding pixels between the current and the previous frame. In order to cut off changes in the frame due to lighting or noise, only pixel differences greater than a threshold value are taken into account. After that, the resulting mask must be transformed using morphological operations to fill in the gaps resulting from the comparison with the threshold value and to eliminate noise and artifacts using the absolute difference operation.

Since not all of the object can be captured in this way because some parts don't change position between frames, or because the color difference is close to zero, a circle around each non-zero pixel is included in the final mask. The larger the envelope size, the more likely it is to capture the entire object, but it can also lead to the inclusion of background objects.

The most effective method for cutting out only the differences from an image is to overlay the mask obtained in the previous steps on the current frame using the bitwise AND operator. Images are matrices in memory, and matrix operations use optimizations such as AVX commands, so they are faster than iterative approaches. As a result, only those parts of the image that are different between the two frames remain as a result of the mask.

One of the disadvantages of this method is the high sensitivity to noise due to the use of absolute pixel values, which leads to a large number of regions with differences of only a few pixels. Further processing of such regions will lead to significant time costs, so the resulting regions should be filtered by size and only large ones that may contain a potential object should be used.

One of the fast and efficient methods of background removal is the MOG2 algorithm[3]. In a number of experiments, the average speed of this algorithm is 10 milliseconds per frame. The algorithm developed in this work demonstrates an average time of 8 milliseconds.

Another disadvantage is that if an object moves little or only one of its parts moves, only that part will be found in the image. This disadvantage is partially compensated by capturing the object's neighborhood and can be completely eliminated by periodically using more computationally intensive algorithms to find objects.

References

16. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C., Berg A., SSD: Single Shot MultiBox Detector, arXiv 1512.02325, 2016.
17. Redmon J., Divvala S., Girshick R., Farhadi A., You Only Look Once: Unified, Real-Time Object Detection, IEEE Conference on Computer Vision and Pattern Recognition, 2016.
18. Kutuk S, Abri R., Abri S., Cetin S., A Light Weight Approach for Real-time Background Subtraction in Camera Surveillance Systems, IEEE 5th International Conference on Image Processing Applications and Systems, 2022

УДК 621.396.2

***Syvolovskyi I.M.**, PhD student*
Ukrainian State University of Railway Transport, Kharkiv
***Zhuchenko O.S.**, PhD. Associate Professor*
Ukrainian State University of Railway Transport, Kharkiv
***Sarapin R.O.**, Servicemen*
Military unit A7223

METHODS TO IMPROVE THE PERFORMANCE OF DISTRIBUTED TELECOMMUNICATION SYSTEMS BY CHANGING THEIR ARCHITECTURE

Currently, Internet of Things systems are among the most complex to design, due to the large number of client devices and the even greater amount of data they generate. The data generated by the devices have no value on their own - the main task of any system is to process them by structuring, cleaning, analysis, etc.

As long as the system processes numerical or textual data, the traditional approach using the cloud is suitable for any load, albeit with high latency. But when the system needs to process multimedia data (audio and video), the resource requirements increase significantly. Nowadays, with the development of artificial intelligence algorithms, media processing has begun to include their active use, for example, pattern recognition. However, the use of these algorithms imposes additional resource requirements - some algorithms can get a significant performance boost when running on hardware-accelerated processors or video cards. Also, such systems may have increased requirements for data processing delays, for example, in video surveillance, which makes the cloud-based approach inefficient.

To solve such problems, the Fog Computing paradigm was previously developed, which introduces additional layers of computing nodes between the cloud and the client device. Using this paradigm, it becomes possible to transfer part of the computation to intermediate layers, which reduces the latency relative to client devices and, accordingly, to obtain data processing results faster at each stage.

Given the growing popularity of this paradigm, researchers have begun to develop specific cases of its application in various fields, creating additional or specialized layers and forming clusters of nodes. In the context of video stream processing, this paradigm can be easily applied - a separate layer of computing nodes is allocated for each processing stage, with hardware characteristics that can effectively perform the designated type of task [1].

The stages of video stream processing in video surveillance include: preprocessing, segmentation, feature extraction, and classification. These stages show that the further the processing is carried out, the more the hardware requirements of the nodes increase, but at the same time, the cardinality of the data decreases - at each stage, the node transmits only the results of its processing and a small part of the original data (for example, key frames). In terms of network capacity, nodes in later stages can receive processing results from more nodes than nodes in the previous stage. Also, it can be noted that nodes from later stages can perform tasks from earlier stages, although this is a less efficient use of resources, as simple tasks are more efficiently distributed to weaker nodes.

These statements lead to the conclusion that the exclusive use of nodes for a specific type of task is inefficient, because in the event of load surges or failures, other nodes may not be able to compensate for the lack of resources due to the conceptual limitations of the system.

New research addresses this situation in the context of the ‘service placement problem’, where a service is a container or application that can perform one type of task. Several such services can be placed on a node, and processing optimizations include moving services to other nodes to reduce latency, which is reduced to performing tasks on graphs [2].

Given that this approach does not clearly divide nodes into layers, and large systems can have tens of