Data-Flow Processing of Data from Surveillance Cameras for Objects Detection Using Distributed Data Processing System

Egor Ivanov, Igor Tishchenko
Program Systems Institute of Russian Academy of Sciences
Pereslavl-Zalessky, Russia
{egor.s.ivanov, igor.p.tishchenko}@gmail.com

***Abstract*—The paper describes image processing methods that was realized in the distributed data processing system [1]. Basic processing method is image segmentation. Convolutional neural network is used for detection of color and texture information. Results of image segmentation are applied in background regions detection task.**

I. INTRODUCTION

Various image processing techniques can speed up the solution of a problem, and in some cases - even automate this process. For example, the solution of the inventory can be reduced to the problem of search and recognition of objects in the image. The task of changing the area occupied by one or another object can be reduced compared to areas in the images of the same area at different times. Calculation of the area of the specified object in space can be reduced to the calculation of the size of objects in the image. A selection of objects in a picture, a sub-task of segmentation. Thus, we can say that the task of monitoring the territory is also reduced to the problem of segmentation.

Image segmentation is an image processing, in which the partition is obtained a plurality of segments (called superpixels), homogeneous in some respects. Typically, such signs are visual characteristics, most of all is using color. It is assumed that in the final segments of the image correspond to the objects, and the segment boundaries are boundaries of objects. As a result of the segmentation, the subsequent analysis of the image is not on the pixel level and at the level of the selected superpixels, e.g. at the object level

II. METHODS OF MAGE SEGMENTATION

Most of common RGB-image segmentation algorithms are variety of one of the three basic methods: threshold segmentation, building areas or edge detection.

Conception of first method is that pixels age grouping by their color features. If the brightness of the pixel is within the prescribed limits, the pixel belongs to a given class. These boundaries can be specified as a, and calculated based on the characteristics of the processed image. Binarization Image is a special case of the threshold segmentation. Fig.2 shows the result of image processing threshold segmentation [2], where the amount allocated to the segments is two. We can assume that the captured image shows the white walls, and black - the floor.

Fig.1. Input image

Fig.2. Binarized image

Fig.3 shows process of image segmentation in the data processing scheme.

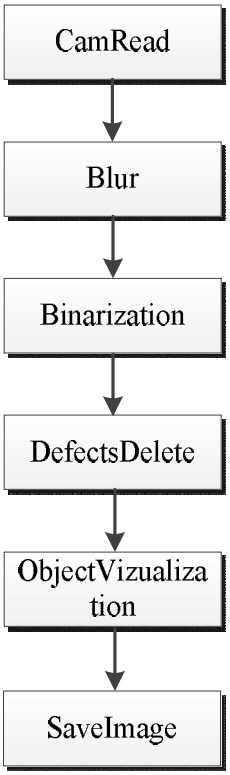


Fig.3. Scheme of process of image segmentation by threshold segmentation

The image processing is performed in the distributed data processing system. The system has a modular architecture. The main objective of the system is to provide data using the modules in the pipeline mode. This speeds up data processing by using a supercomputer. The input system is supplied description of the task, which consists of data processing modules. The system automatically determines which and how many modules are to run, and also provides data transfer between them.

The image processing consists of sequential execution processing modules. First there is a reader of the original image from the camera (CamRead module). Input channel module does not have an output channel - image. Then comes the module Blur - smoothing the image. This will eliminate excessive segmentation. As the input channel module receives the image picked up by the previous module. Also, this module receives the parameters: the method of smoothing and blur radius produced. The output channel of the module is also an image. Next, the smoothed image is transferred to the input module binarization. Input channel - image output channel - image. Binarization image is in the process of elimination of defects - removal of objects (segments), the size of which is not more than N pixels (N - parameter acquiring unit). This procedure performs module DefectsDelete. Input and output channels are also images. The next stage of processing - visualization of objects found (module ObjectVizualization). Input and output channels are also an image. Ends processing module SaveImage, which retains an image obtained in the entrance channel.

Fig.4 shows an example of an image obtained as a result of post-processing (performance of the modules and DefectsDelete OblectVizualization).

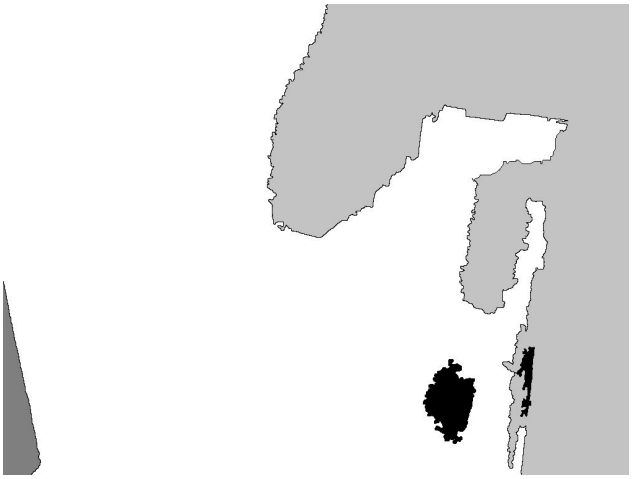


Fig.4. Results of image processing using full scheme (Fig.3).

In segmentation by building regions originally selected starting pixel. We consider it the neighboring pixels and their homogeneity is analyzed on the basis of which a decision on the inclusion of a given pixel in a particular segment or on the establishment of a new one. Thus, the final result of segmentation obtained by merging pixels in individual segments. Often after combining received segments are checked for homogeneity and nonhomogeneous field, it is divided into smaller, as long as each of the segments would not meet the criterion of homogeneity. One of the most popular algorithm used in this approach is the k-means algorithm [3]. He concludes that first selects a center pixel of each object (cluster), then occurs buildup areas on the pixels. The action of the algorithm is that it tends to minimize the total deviation of the points from the centers of clusters of clusters. This is an iterative algorithm that divides a given set of pixels into k clusters, in terms which are as close to their centers, and she clustering is due to the displacement of the center. The result of the algorithm is highly dependent on the presence of noise in the image, so you should use the pre-processing (eg, Gaussian smoothing). Fig.5 shows the result of this algorithm without pretreatment, while Fig.6 - using Gaussian pre-smoothing. On the source image have been allocated plots of 5 types, which occurred in the latter capacity areas.

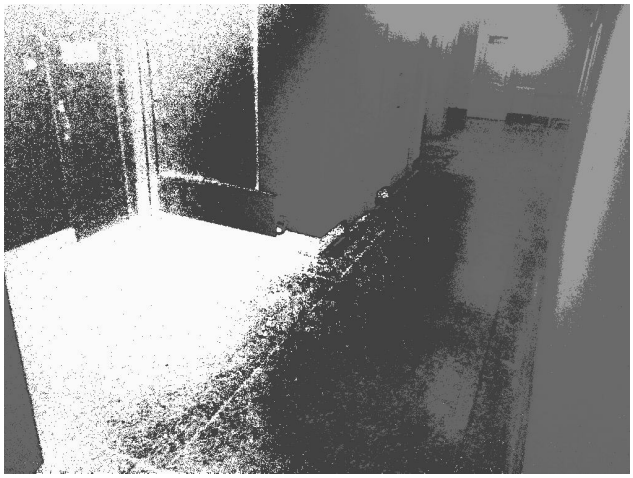


Fig.5. K-means image segmentation without pre-processing

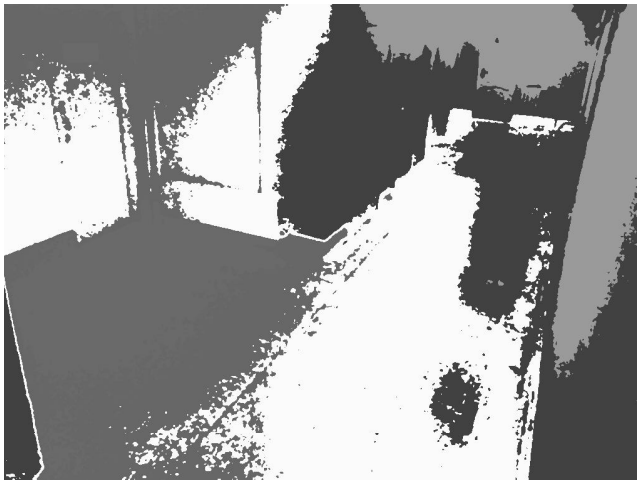


Fig.6. K-means image segmentation using pre-processing (Gaussian blur)

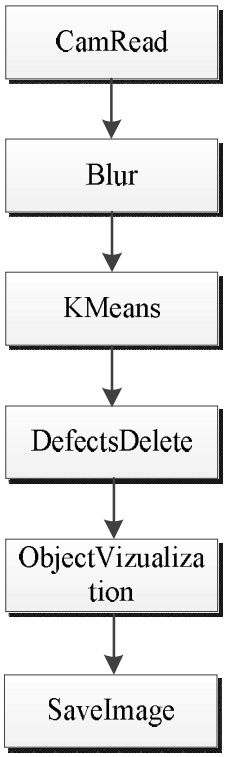


Fig.7. Scheme of process of image segmentation by k-means segmentation

The process shown on Fig.7 and consists of sequential execution modules: reading (CamRead), antialiasing (Blur), pixel clustering method k-means (KMeans). The input channels of the module k-means is an image, as the module receives the number of k- number of clusters that stand out in the picture. The output channel of the module is the image. Next comes the removal of defects (DefectsDelete), visualization of objects (ObjectVizualization) and saving results (SaveImage).

This method of segmentation is using the selection boundaries gradient operator. Then the separation threshold. The boundary pixels are obtained in the previous steps are connected to the closed curves, which can be considered the boundaries of the segments (regions). Fig.8 shows the result of allocation of borders by Sobel. Then, the resulting image has passed post-processing: it has been normalized, and to eliminate excessive segmentation Binarization. Fig.9 shows the white border segments. Even given the fact that the question of edge detection in the image well enough to lighten in modern technical literature, it still remains quite a laborious task, since the qualitative selection boundaries always depends on a number of factors affecting the result. As can be seen in the image, the border is often found torn and get a separate region becomes impossible. Thus, this approach does not always produce the desired result, and requires further revision by the user: it is necessary to eliminate gaps, which in itself is a separate task.

Fig.8. Sobel edge detection

Fig.9. Result of post-processing

Scheme of image processing shown on Fig.10.

III. NEURAL NETWORKS

Using neural networks is motivated by their similarity to successfully working biological systems, which - in comparison to the overall system - consist of very simple but numerous nerve cells simple but many processing units that

work massively in parallel and (which is probably one of the most significant aspects) have the capability to learn [4].

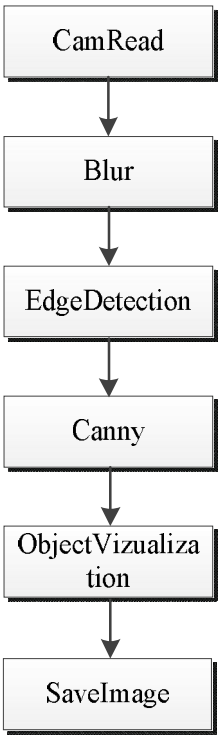


Fig.10. Scheme of process of image segmentation by edges detection

Neural networks are a set of models inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning [5].

A convolutional neural network (CNN) consists of several layers. These layers can be of three types: convolutional, pooling and fully-connected.

Convolutional layers consist of a rectangular grid of neurons. It requires that the previous layer also be a rectangular grid of neurons. Each neuron takes inputs from a rectangular section of the previous layer. The weights for this rectangular section are the same for each neuron in the convolutional layer.

Next layer is the *pooling layer*. The pooling layer takes small rectangular blocks from the convolutional layer and subsamples it to produce a single output from that block. The pooling layer can take the average, or the maximum, or learn a linear combination of the neurons in the block.

After convolutional and pooling layers, should be fully connected layers. A *fully connected layer* takes all neurons in the previous layer (be it fully connected, pooling, or convolutional) and connects it to every single neuron it has. Fully connected layers are not spatially located anymore, so there can be no convolutional layers after a fully connected layer [6].

Architecture of convolutional neural network is shown on Fig.11.

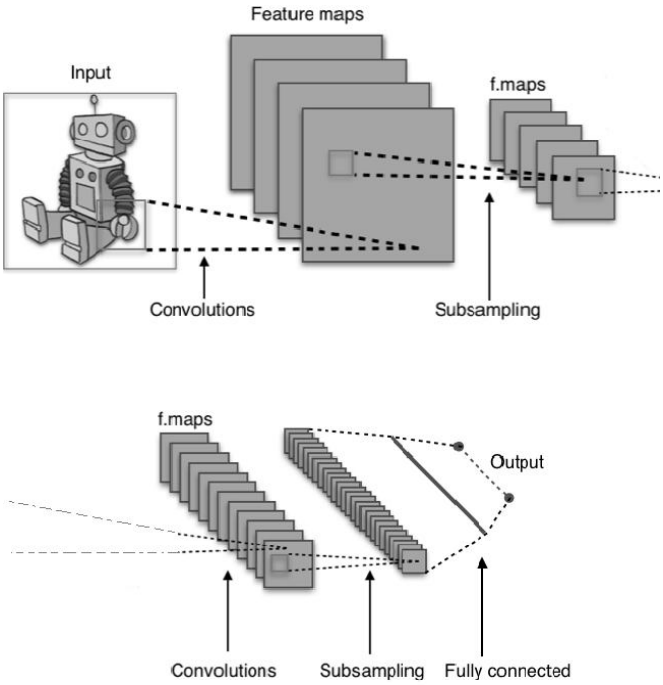


Fig.11. Typical CNN architecture

IV. Image segmentation using neural networks

A. Neural network learning

The first stage of processing - preparation of input data for neural network training. The user selects the image areas which contain the classes (module MakeClasses) (Fig.13). The sizes of these portions can be of arbitrary size. Output channel of this module is a set of images.

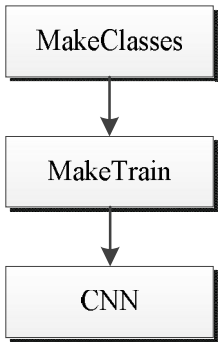


Fig.12. Scheme of learning Neural Network

The next step is a creating train dataset (MakeTrain). Input channel for this module is the set of images. The module cuts images into pictures 32x32 and writes data for learning neural network.

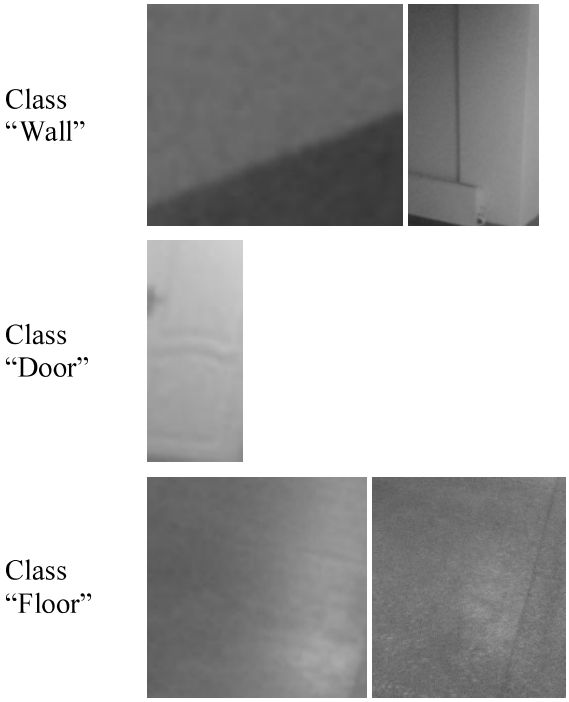


Fig.13. Samples of images marked by the user as a result of module MakeClasses

The next stage is the work of a neural network. CNN module reads the plurality of images, and, depending on the input parameter value, produces training or recognition. In this scheme is used training sample - samples given classes of the desired size (image used 32 * 32). It should be noted that the set of objects for learning should be sufficiently large to contain a variety of options objects. Making the training sample is an important step in dealing with a neural network. Learning outcomes depend on the quality of the training sample. A segmentation result depends on the quality of training the neural network. Further, in the examples, the following classes: "floor", "wall" and "door". The total number of copies for training amounted to about 40000. Table I shows some of the items available for training.

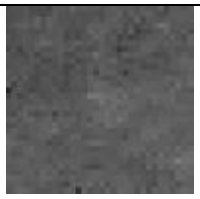
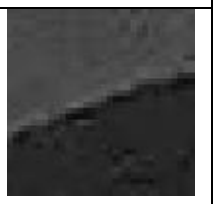
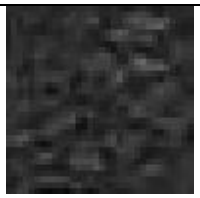
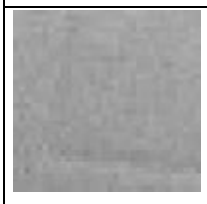
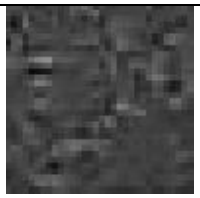
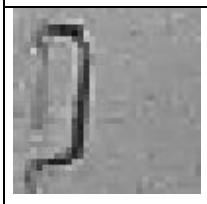
As a result, training is a set of weights recorded in the file. Thereafter, use of the instrument for subsequent processing.

Neural network training error was 0.02%.

B. Image segmentation

The idea of segmentation using neural networks is as follows. Processed image is divided into fragments using a moving mask. The mask can move with random step from 1 pixel to 32, whereby the resulting set may comprise a different number of fragments. For example, the image 64x64 step 32 pixels in the mask is divided into 4 fragments, and when 1 pixel step - to 1089 fragments. Each of the resulting fragments is the image that the neural network can be attributed to one or another class. The resulting image can be repainted in accordance with the recognition results. The quality of the segmentation depends on the quality of training networks.

Table I. SAMPLES OF TRAIN DATASET

Door	Floor	Wall
		
		
		
		
		

The schema of image processing is shown below on Fig.14.

The original image from the camera using the module CamRead, broken into multiple images (module MakeTest) and supplied the neural network to recognize the (module CNN). The result of the module is a set of numbers that indicate the recognized classes. After that, the module produces ImageColor drawing recognition results. Post-processing - removal of defects, rendering of found objects and saving results (modules DefectDelete, ObjectsVizualization, SaveImage).

Benefit of this method of image segmentation is that user knows in advance which objects are present in the image, he can control these classes. For example, if the walls in the room in different colors, then the training images should apply to samples of all colors. This will cause that wall and green, orange, and on the resulting image will refer to the same class. Furthermore, in the case of complex texture contrast (e.g., floor marble chips) obtained by segmentation of excess (such as when using k-means method).

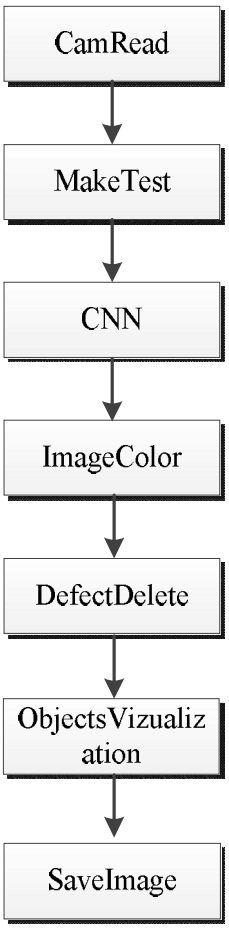


Fig.14. Scheme of image segmentation using neural network

Moreover minor disturbances do not affect the result. This is achieved by the fact that the assignment of each pixel to any occurs, given its environment - the neighboring pixels. Therefore image virtually needs no pretreatment.

Fig.15. Segmented image. White color is walls, black is doors and gray is floor.

C. Results

The original picture was split into a plurality of images, each of which was filed with the neural network to recognize. Thus, each image pixel was assigned to one of three classes, i.e. the image has been marked. The result of the segmentation is shown in Fig.15.

A feature of this approach is that the resultant image segments can be discontinuous.

V. Conclusion

Task of image segmentation is an important task in the field of image processing. Image segmentation can be seen not only as a separate task, but as a problem of data preprocessing. With the help of the image, you can select the objects that are in the interest of the future, which in turn speeds up data analysis.

The developed system of stream processing modules were implemented for image segmentation, as well as pre- and post-processing modules images.

Image segmentation using neural network - an approach that gives better results than other approaches. Due to the fact that the operation of the neural network is carried out not with the individual pixels, but with the entire image fragments, the appearance of noise in the form of one or two pixels of an image area gives a negative feedback. In addition, working with the parts of the picture, in addition to the color information is used and even texture information.

Acknowledgment

This work was supported by the Ministry of Education and Science of the Russian Federation: №14.607.21.0088 agreement for a grant on ``Development of methods and means of processing and intelligent analysis of images and dataflow obtained from a variety of stationary and mobile sensors, using high-performance distributed computing for the tasks of monitoring the premises and the surrounding area". Unique identifier: RFMEFI60714X0088.

References

[1] Aleksey Kondratyev, Igor Tishchenko, “Distributed Processing System of Images Flow”, *in Proc. RiTA2015*, in press.
[2] J. Sauvola, M. Pietikanen. Adaptive document image binarization, *in Pattern Recognition*, vol. 33, 2000, pp. 225-236.
[3] Simplest image segmentation using k-means, Web: http://habrahabr.ru/post/165087/.
[4] David Kriesel. A Brief introduction to Neural Networks // ZETA2-EN.
[5] Convolutional Neural Networks, Web: http://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/.
[6] N. Sarangi, C.C. Sekhar. Tensor Deep Stacking Networks and Kernel Deep Convex Networks. Pattern Recognition: Applications and Methods: 4th International Conference // ICPRAM 2015 pp. 267-281