Using Open Source Libraries in the Development of Control Systems Based on Machine Vision

Vasiliy N. Kruglov

HAL Id: hal-03647274

https://inria.hal.science/hal-03647274

Submitted on 20 Apr 2022

Using open source libraries in the development of control systems based on machine vision

Vasiliy N. Kruglov

Ural Federal University. Yekaterinburg, Russia
`v.krouglov@mail.ru`

Abstract. The possibility of the boundaries detection in the images of crushed ore particles using a convolutional neural network is analyzed. The structure of the neural network is given. The construction of training and test datasets of ore particle images is described. Various modifications of the underlying neural network have been investigated. Experimental results are presented.

Keywords: Grain-size analysis · Machine vision · Object boundaries detection · Convolutional neural network · Open source libraries · Machine learning

1 Introduction

When processing crushed ore mass at ore mining and processing enterprises, one of the main indicators of the quality of work of both equipment and personnel is the assessment of the size of the crushed material at each stage of the technological process. This is due to the need to reduce material and energy costs for the production of a product unit manufactured by the plant: concentrate, sinter or pellets.

The traditional approach to the problem of evaluating the size of crushed material is manual sampling with subsequent sieving with sieves of various sizes. The determination of the grain-size distribution of the crushed material in this way entails a number of negative factors:

– the complexity of the measurement process;
– the inability to conduct objective measurements with sufficient frequency;
– the human error factor at the stages of both data collection and processing.

These shortcomings do not allow you to quickly adjust the performance of crushing equipment. The need for obtaining data on the coarseness of crushed material in real time necessitated the creation of devices for in situ assessment of parameters such as the grain-size distribution of ore particles, weight-average ore particle and the percentage of the targeted class. The machine vision systems are able to provide such functionality. They have high reliability, performance and accuracy in determining the geometric dimensions of ore particles. At the moment, several vision systems have been developed and implemented for the

operational control of the particle size distribution of crushed or granular material. In [9], a brief description and comparative analysis of such systems as: SPLIT, WIPFRAG, FRAGSCAN, CIAS, IPACS, TUCIPS is given.

Common to the algorithmic part of these systems is the stage of dividing the entire image of the crushed ore mass into fragments corresponding to individual particles with the subsequent determination of their geometric sizes. Such a segmentation procedure can be solved by different approaches, one of which is to highlight the boundaries between fragments of images of ore particles. Classical methods for borders highlighting based on the assessment of changes in brightness of neighboring pixels, which implies the use of mathematical algorithms based on differentiation [4,8]. Fig. 1 shows typical images of crushed ore moving on a conveyor belt.
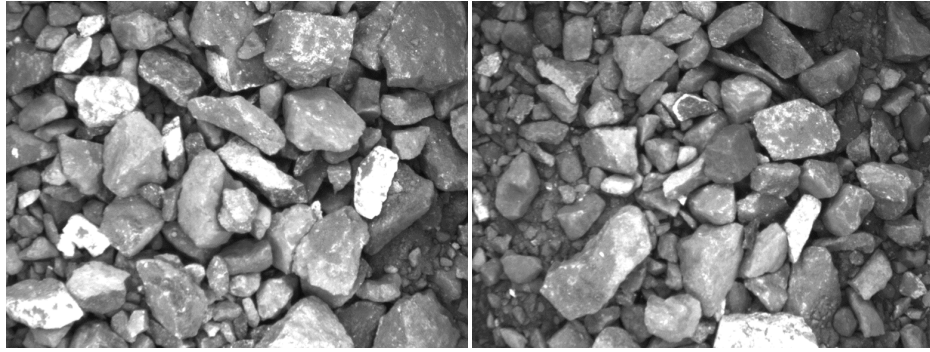

Fig. 1: Examples of crushed ore images

Algorithms from the OpenCV library, the Sobel and Canny filters in particular, used to detect borders on the presented images, have identified many false boundaries and cannot be used in practice.

This paper presents the results of recognizing the boundaries of images of stones based on a neural network. This approach has been less studied and described in the literature, however, it has recently acquired great significance in connection with its versatility and continues to actively develop with the increasing of a hardware performance [3,5].

2   Main part

To build a neural network and apply machine learning methods, a sample of images of crushed ore stones in gray scale was formed . The recognition of the boundaries of the ore particles must be performed for stones of arbitrary size and configuration on a video frame with ratio 768x576 pixels.

To solve this problem with the help of neural networks, it is necessary to determine what type of neural network to use, what will be the input information

and what result we want to get as the output of the neural network processing. Analysis of literary sources showed that convolutional neural networks are the most promising when processing images [3, 5–7].

Convolutional neural network is a special architecture of artificial neural networks aimed at efficient pattern recognition. This architecture manages to recognize objects in images much more accurately, since, unlike the multilayer perceptron, two-dimensional image topology is considered. At the same time, convolutional networks are resistant to small displacements, zooming, and rotation of objects in the input images. It is this type of neural network that will be used in constructing a model for recognizing boundary points of fragments of stone images.

Algorithms for extracting the boundaries of regions as source data use image regions having sizes of 3x3 or 5x5. If the algorithm provides for integration operations, then the window size increases. An analysis of the subject area for which this neural network is designed (a cascade of secondary and fine ore crushing) showed: for images of 768x576 pixels and visible images of ore pieces, it is preferable to analyze fragments with dimensions of 50x50 pixels.

Thus, the input data for constructing the boundaries of stones areas will be an array of images consisting of (768-50)*(576-50) = 377668 halftone fragments measuring 50x50 pixels. In each of these fragments, the central point either belongs to the boundary of the regions or not. Based on this assumption, all images can be divided into two classes.

To mark the images into classes on the source images, the borders of the stones were drawn using a red line with a width of 5 pixels . This procedure was performed manually with the Microsoft Paint program. An example of the original and marked image is shown in Fig. 2.

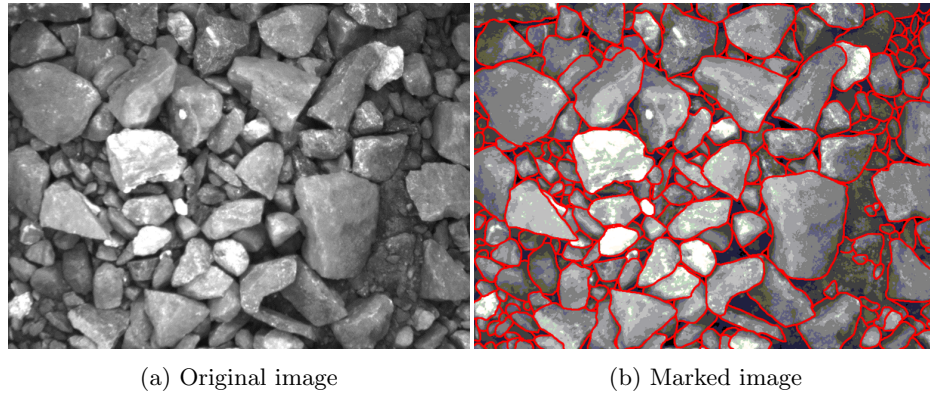(a) Original image   (b) Marked image

Fig. 2: Image processing

Then Python script was a projected, which processed the original image to highlight 50x50 pixels fragments and based on the markup image sorted frag-

ments into classes preserving them in different directories To write the scripts, we used the Python 3 programming language and the Jupyter Notebook IDE. Thus, two data samples were obtained: training dataset and test dataset for the assessment of the network accuracy.

As noted above, the architecture of the neural network was built on a convolutional principle. The structure of the basic network architecture is shown in Fig. 3 [7].
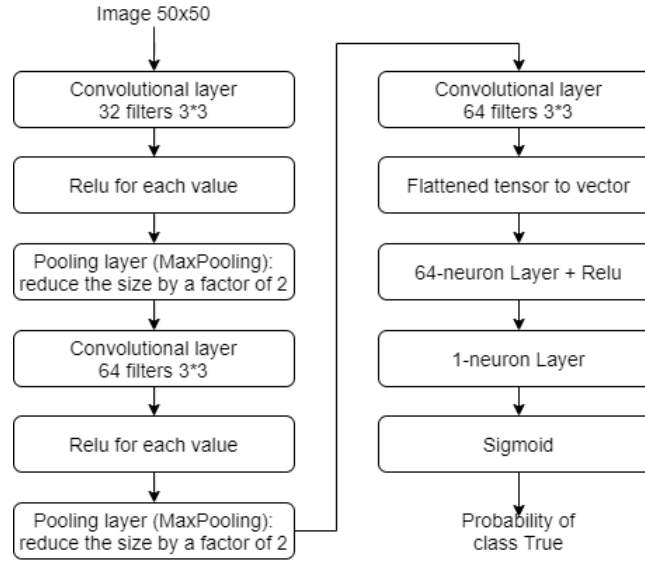

Fig. 3: Basic convolution network architecture

The network includes an input layer in the format of the tensor 50x50x1 The following are several convolutional and pooling layers. After that, the network unfolds in one fully connected layer, the outputs of which converge into one neuron, to which the activation function, the sigmoid, will be applied. At the output, we obtain the probability that the center point of the input fragment belongs to the "boundary point" class.

The Keras open source library was used to develop and train a convolutional neural network [1, 2, 6, 10].

The basic convolutional neural network was trained with the following parameters:

- 10 epoch;
- error - binary cross-entropy;
- quality metric - accuracy (percentage of correct answers);
- optimization algorithm - RMSprop.

The accuracy on the reference data set provided by the base model is 90.8%. In order to improve the accuracy of predictions, a script was written that trains models on several configurations, and also checks the quality of the model on a test dataset.

To improve the accuracy of the predictions of the convolutional neural network, the following parameters were varied with respect to the base model:

- increasing the number of layers: + 1 convolutional + 1 pooling;
- increasing of the number of filters: +32 in each layer;
- increasing the size of the filter up to 5*5;
- increasing the number of epochs up to 30;
- decreasing in the number of layers.

These modifications of the base convolutional neural network did not lead to an improvement in its performance - all models had the worst quality on the test sample (in the region of 88-90% accuracy).

The model of the convolutional neural network, which showed the best quality, was the base model. Its quality in the training sample is estimated at 90.8%, and in the test sample - at 83%. None of the other models were able to surpass this figure. Data on accuracy and epoch error are shown in Fig. 4 and 5.
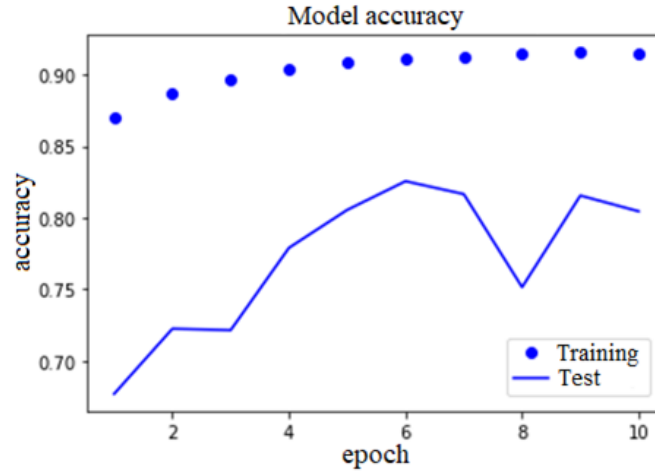

Fig. 4: The dependence of the accuracy on the training and test datasets from the training epochs

If you continue to study for more than 10 epochs, then the effect of retraining occurs: the error drops, and accuracy increases only on training samples, but not on test ones.
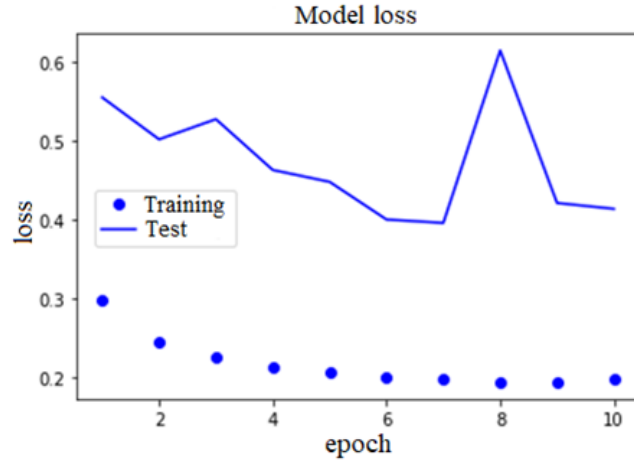
Fig. 5: The dependence of the error on the training and test datasets from the training epochs

Fig. 6 shows examples of images with neural network boundaries. As you can see from the images, not all the borders are closed. The boundary discontinuities are too large to be closed using morphological operations on binary masks; however, the use of the "watershed" algorithm [8] will reduce the identification error of the boundary points.

3   Conclusion

In this work, a convolutional neural network was developed and tested to recognize boundaries on images of crushed ore stones. For the task of constructing a convolutional neural network model, two data samples were generated: training and test dataset. When building the model, the basic version of the convolutional neural network structure was implemented. In order to improve the quality of model recognition, a configuration of various models was devised with deviations from the basic architecture. An algorithm for training and searching for the best model by enumerating configurations was implemented.

In the course of the research, it was found that the basic model has the best quality for recognizing boundary points. It shows the accuracy of the predictions for the targeted class at 83%.

Based on the drawn borders on the test images, it can be concluded that the convolutional neural network is able to correctly identify the boundary points with a high probability. It rarely makes mistakes for cases when there is no boundary (false positive), but often makes mistakes when recognizing real boundary points (false negative). The boundary breaks are too large to be closed using morphological operations on binary masks, however, the use of the "watershed" algorithm will reduce the identification error for boundary points
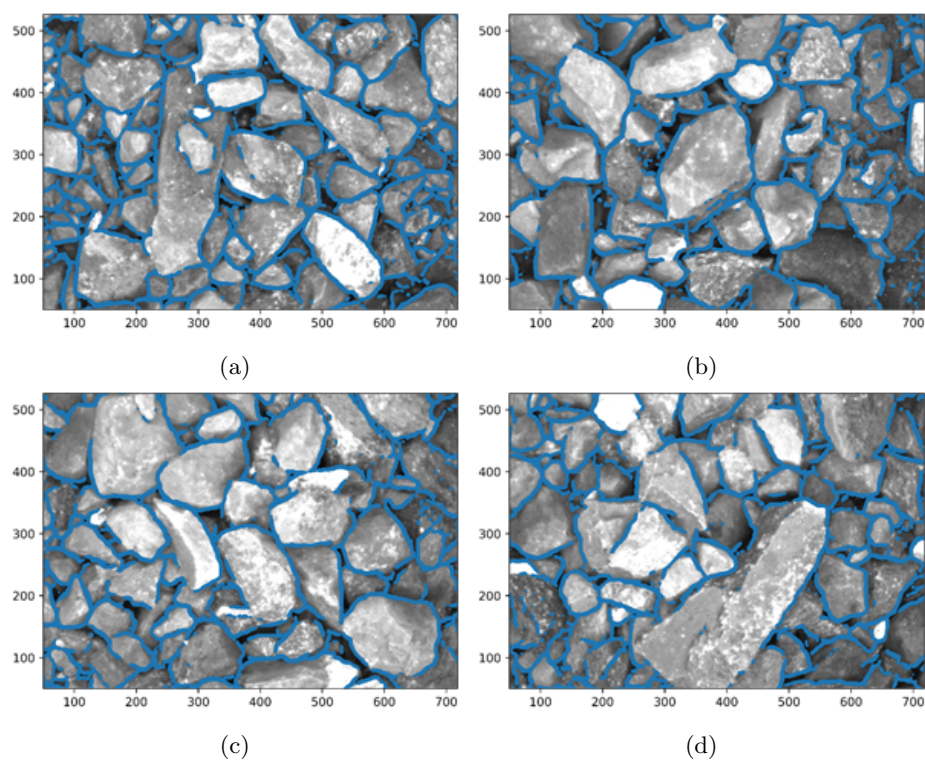
(a)

(b)

(c)

(d)

Fig. 6: Images of crushed ore particles with boundaries detected by a neural network

References

1. Keras: The python deep learning library. https://keras.io/, accessed: 2020-14-01
2. Chollet, F.: Deep Learning with Python. Manning Publications Co., USA, 1st edn. (2017)
3. Flach, P.: Machine Learning: The Art and Science of Algorithms That Make Sense of Data. Cambridge University Press, USA (2012)
4. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice-Hall, Inc., USA, 3 edn. (2006)
5. Gron, A.: Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Inc., 1st edn. (2017)
6. Gulli, A., Pal, S.: Deep learning with Keras: implement neural networks with Keras on Theano and TensorFlow. Packt Publishing, Birmingham, UK (2017)
7. Saha, S.: Comprehensive guide to convolutional neural networks - the eli5 way. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, accessed: 2020-12-01
8. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision. CL Engineering, 2 edn. (1998)
9. Thurley, M.J., Ng, K.C.: Identifying, visualizing, and comparing regions in irregularly spaced 3d surface data. Computer Vision and Image Understanding **98**(2), 239–270 (May 2005). https://doi.org/10.1016/j.cviu.2003.12.002, https://doi.org/10.1016/j.cviu.2003.12.002
10. VanderPlas, J.: Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media, Inc., 1st edn. (2016)