Information Technology of Hot-metal Ladle Car Handwritten Numbers Recognition from Photo Image

Yevhen Chychkarov[1[0000-0002-4362-5129]], Anastasiia Serhiienko[2[0000-0003-1328-2572]], Iryna Syrmamiikh[3[0000-0003-0524-6731]]

1,3 Donetsk State University of Management, Mariupol, Ukraine
influence@meta.ua, siv.2017@ukr.net

2 Pryazovskyi State Technical University, Mariupol, Ukraine
invitery@i.ua

Abstract. The paper presents the results of the construction and testing of an information system for recognizing ladle car numbers drawn up of handwritten digits. The latest publications on solving problems of recognition and analysis of texts by their images were analyzed, a selected range of application libraries for creating a program for experiments are Keras, TensorFlow, Theano, Pandas, Matplotlib and others.

Two variants of recognition technology were analyzed and tested: using fuzzy logic and using deep learning neural networks. To construct and train neural networks, a well-known and rather complete base of handwritten digits MNIST was chosen. Two types of neural networks were considered: sequential and convolutional. The training of neural networks was carried out using a variable number of steps (epochs). Recognition images were scaled to a size of 28x28 (784 cells in a one-dimensional representation). Preliminary processing of images (filtering, scaling, etc.) was carried out using the OpenCV library. For recognition, each image of the digit was converted to a 28x28 size and fed to the input of a pre-trained neural network.

A technique to select the area of interest in photographs containing hand-written digits for further recognition has been worked out. For handwritten digit recognition, the best recognition accuracy is provided by a convolutional neural network, with which 97.6% of car ladle digits were correctly recognized.

Keywords: Keras; TensorFlow; MNIST; Python; Deep learning; Neural networks; Digit recognition.

1 Introduction

Currently, automatic recognition systems for numbers of moving objects (the license plate of the vehicles, wagon numbers, etc.) are demanded in a wide range of fields.

In this paper, a method of recognizing the digit of a moving hot-metal ladle car by a photo using various mechanisms (fuzzy output and deep learning neural networks) is proposed. Hot-metal ladle cars are used to transport cast iron from blast furnaces to the mixer department of the oxygen-converter plant (OCP). The number of the hot-metal ladle car is its unique identifier. Therefore, the ladle car number is quite important information about it, and it is actively used in information systems operating at integrated iron-and-steel works. For inventory and control of their condition, an automated monitoring system is used to track the movement of platforms carrying hot-metal ladle cars.

The digits of car ladle number are applied manually (without stencil) with lime. During transportation and while conducting technological operations, the ladle cars are exposed to high temperatures, and the digits may be distorted due to dust, cast iron drips, etc.

The image of the hot-metal ladle car is obtained by shooting with the camera. The initial data for solving the problem of recognizing the ladle car number is a fragment of the photo image of the ladle car containing its number.

Recognizing handwritten digits or letters is obvious enough for a person, but for computers, the identification of such characters is a difficult task. Various methods are often used to fulfill this task: neural networks, classification algorithms, etc.

Recently, deep learning neural networks have been widely used to identify range of interest and recognize image elements.

2 Analysis of literature data and formulation of the problem

Currently, there are many areas of science and technology, which are largely focused on the development of systems that analyze the information presented as images. However, most of the tasks associated with image processing and recognition are classified as difficult to formalize.

The task of segmentation and recognition of image areas containing handwritten or printed characters is relevant due to the presence of a large number of technical applications.

A number of works on the recognition of license plates of cars [1-4] using neural network technologies have been published. A similar approach is used in the systems of registration and recognition of wagons and tanks numbers [5-6]. The functionality of all these systems is approximately the same – they automate the process of reading digits and save the received information.

Differences in the digit recognition systems associated with the format of car and wagon numbers, as well as with the observation conditions, are discussed in works [8–9]. In particular, according to [8], weather conditions and poor illumination are obstacles for the qualitative identification of digits. To eliminate these obstacles, additional lighting and image filtering should be used.

Various methods and algorithms to recognize digits are used: a modified algorithm for comparing an object with a standard in Hausdorff distance [10], a perceptron neural network with sigmoid activation functions [8], and a scale-invariant transformation method [5].

According to [11], the support vector method (SVM) is the most effective for recognizing license plate numbers. The advantage of the SVM method is that a relatively small training sample is sufficient for constructing a classifier for character recognition. In addition, it has a low probability of error.

In work [12], the effect of preliminary processing and segmentation of license plate numbers images on the results of their recognition was noted. For any recognition methods, incorrect character segmentation does not allow to achieve accurate results. It was also noted there that license plate recognition is easily implemented in relation to standard plates. But in different countries the sizes and contents of the license plate differ, so there is a need to solve recognition problems without sticking to strict rules.

In recent years, much attention has been paid to the recognition of digits on various visual objects using deep learning neural networks [13-16]. For example, in work [16], a method for detecting and segmenting car license plates in real time on the basis of image analysis and processing methods is presented. According to the authors, the computation effort and accuracy degree, taking into account the proposed approach, are acceptable for real-time applications with a run time of less than 1 second.

A deep learning neural network is an artificial neural network with several hidden layers [17]. Additional layers make it possible to build abstractions of ever higher levels, making it possible to form models for recognizing complex real-life objects. Deep direct distribution networks are commonly used, however, recent studies have shown the successful use of deep architectures in recurrent networks [18]. In tasks related to image processing, convolutional neural networks (CNN) are predominantly used, taking into account their high efficiency for this range of tasks.

Deep learning technologies are used to solve a range of practically important tasks: tasks related to the natural language (grammar check, recognition of printed or handwritten texts, etc.), search for information on the Internet, image processing and recognition, etc. [20, 21]. Machine learning methods are used to build algorithms that provide self-learning on certain datasets, and then reliable forecasting and classification of test data [22-26].

To solve the problems of recognition and analysis of texts by their images, various types of neural networks are used: sequential neural networks [24], convolutional neural networks [21, 22, 24, 25], impulse neural networks [26].

To solve the problems of machine learning and building neural networks, the Python programming language and a number of applied libraries (Keras, TensorFlow, Theano, Pandas, Matpoltlib) [27] are widely used.

TensorFlow is a specialized library for calculations using graphs of data streams (graph nodes – mathematical operations, edges – multidimensional data arrays). In TensorFlow, parallel computing on the GPU is available after installing necessary drivers and configuring special parameters.

Keras is a library for building Deep Learning neural networks, which provides a high-level API that uses TensorFlow (or Theano) as a backend.

Optimization of the structure of deep learning neural networks and optimization of their parameters are of considerable interest for the researchers.

Advantages related to recognition accuracy close to the human level when processing large data arrays have led to an increase in the use of CNN in recent years [22-23]. But the picture of the effect of CNN's hidden layers on overall network performance remains unclear.

According to [24], in practice, impulse neural networks with multilayer learning turned out to be difficult to learn.

3 Goal and objectives of the research

Goal of the research: to develop an information technology for recognizing the handwritten digit of a hot-metal ladle car number by its photo image, capable of fast localizing the area of interest, identifying the digit, and testing this technology.

To solve this practical problem, it is necessary to solve several subproblems:

- development and testing of an algorithm of preliminary image processing and selection of areas containing a digit for recognition, as applied to the images of hot-metal ladle cars;
- setup and optimization of an artificial neural network to recognize handwritten or printed digits;
- development and adjustment of the car number recognition method by means of a fuzzy comparison with the sample.

The image of the ladle car with the number is obtained using the camera in grayscale.

The images under consideration (see the example in Fig. 1) have a number of features that significantly distinguish their processing from the well-known solutions on recognizing wagon or car numbers:

1. hot-metal ladle cars work in difficult conditions around the clock, therefore, processed images may be distorted by the presence of dust, drips and spray of cast iron, changing the lighting conditions;
2. the processed image contains non-informative areas (sky, wall, elements of nearby metal structures) with high or low brightness;
3. digits to be recognized are applied manually on an uneven porous surface, therefore, even on freshly applied numbers there are heterogeneities;
4. there may be drips when applying numbers, as well as areas with a light background at the edges of the image.

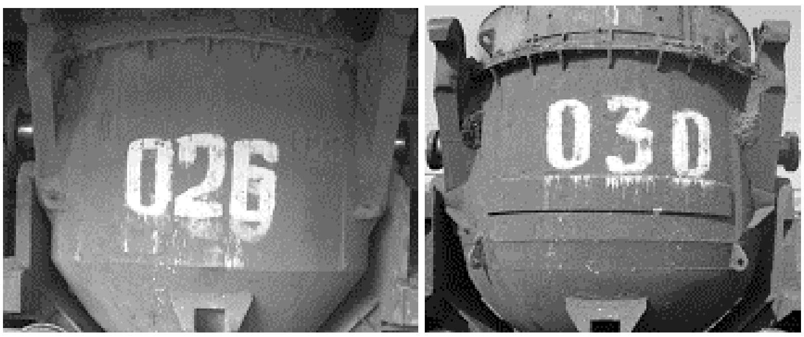


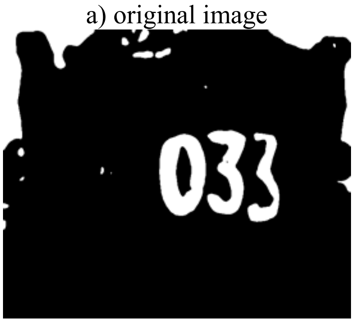
Fig. 1. Example of original photo images with distortions

4 Description of the proposed algorithm of selecting recognition area

All operations of selection a region of interest (see list below) were performed in a program in Python using functions of the OpenCV library. An illustration of the process of selecting recognition areas is presented in Fig. 2.

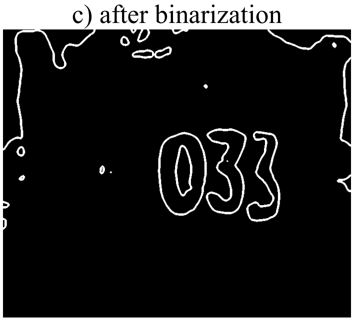
a) original image

b) after filtration

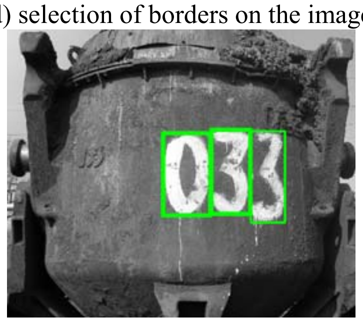


c) after binarization

d) selection of borders on the image



e) after binarization and dilatation



f) original image with selected areas of recognition

Fig. 2. Operating sequence of selection recognition areas algorithm

The basis of automatic recognition of car numbers is based on the following procedures [27-28]:

1. image filtering to reduce the noise level (a Gaussian filter was used - cv2.GaussianBlur function);
2. binarization of the image to cut off noise (the cv2.threshold function was used, its

parameters were adapted to reliably select the outlines of the digits);

3. highlighting the contrasting borders on the image (the Canny edge detector was used - cv2.Canny function, for which the values of the maximum and minimum values of the gradient were preliminarily selected);
4. filtering to reduce the effect of image heterogeneity (the median filter was used - cv2.medianBlur function);
5. image binarization (cv2.threshold function was also used);
6. morphological transformation (dilatation - function cv2.dilate);
7. selection of contours and their sorting (selection of contours was carried out using the cv2.findContours function);
8. image segmentation, i.e. selection of recognition areas in the form of rectangle set containing previously selected contours of digits (cv2.boundingRect functions were used).

Computer experiments showed that the proposed method of localization of informative is effective and has a reliability of localization of 97.6 %.

5 Description of digit recognition algorithms

The process of recognizing text areas in images can be divided into three stages: the stage of preliminary processing of the image, the stage of selecting an area containing characters, and the stage of their recognition.

5.1 Digit identification using fuzzy logic

Localization of the area of interest described in [29] is performed using clustering by nearest neighbors method. This method is effective and has a localization reliability of 96%. However, the algorithm for identifying the recognition area described above can also be applied in this information technology for recognizing a handwritten digit of a hot-metal car by its image, since it has a higher localization reliability of 97.6%.

The number of the hot-metal car is identified using the original method, based on a fuzzy representation of the images and comparing them with the standard segments [30]. The idea of the identification method is as follows. First it is necessary to find the common area of two comparable segments of different sizes, called the core. Associate a coordinate system with this core and recalculate the pixel representation of both images into one coordinate system.

Place the origin of the core coordinate system at the characteristic point of the digits image representation. A point has been adopted, which is calculated as the center of gravity of the image. The cores of the analyzed segment and prototype $\{A''\},\{B''\}$ are divided into blocks $\Omega_{i,j}$ of size $n \times n$. The number of blocks for sets $\{A''\},\{B''\}$ is calculated: $C = \mathrm{round}(W/n);\ D = \mathrm{round}(H/n)$.

The set of units of image and prototype cores are denoted as:

$$\Omega^A = \left\{\Omega^A_{r,s} \middle| r = \overline{1,C}; s = \overline{1,D}\right\},\quad \Omega^B = \left\{\Omega^B_{r,s} \middle| r = \overline{1,C}; s = \overline{1,D}\right\}.$$

Separate units $\Omega^{A}_{r,s}$ and $\Omega^{B}_{r,s}$ of sets Ω^{A} and Ω^{B}, respectively, are set by two characteristics: the set of pixels coordinates included in the unit:

$$\Omega^{A}_{r,s} = \Omega^{B}_{r,s} = \left\{(i, j) \middle| i = \overline{(C-1) \cdot n + 1, C \cdot n}, j = \overline{(D-1) \cdot n + 1, D \cdot n}\right\},$$

and set of binarized brightness functions: $g_A : \Omega^{A}_{r,s} \to [0,1]$, $g_B : \Omega^{B}_{r,s} \to [0,1]$.

Two fuzzy sets $\underset{\sim}{\Omega}^{A}$, $\underset{\sim}{\Omega}^{B}$, which are defined on universal sets Ω^{A}, Ω^{B} are introduced. Membership functions are given in formula 1:

$$\mu_{\underset{\sim}{\Omega}^{A}}\left(\underset{\sim}{\Omega}^{A}_{r,s}\right) = \frac{\sum\limits_{i,j \in \Omega_{r,s}} g_A(i, j)}{\left|\Omega^{A}_{r,s}\right|}, \quad \mu_{\underset{\sim}{\Omega}^{B}}\left(\underset{\sim}{\Omega}^{B}_{r,s}\right) = \frac{\sum\limits_{i,j \in \Omega_{r,s}} g_B(i, j)}{\left|\Omega^{B}_{r,s}\right|}. \tag{1}$$

As a result fuzzy sets are received (formula 2):

$$\underset{\sim}{\Omega}^{A} = \left\{\Omega^{A}_{r,s} \middle| \mu_{\underset{\sim}{\Omega}^{A}}\left(\Omega^{A}_{r,s}\right)\right\}, \quad \underset{\sim}{\Omega}^{B} = \left\{\Omega^{B}_{r,s} \middle| \mu_{\underset{\sim}{\Omega}^{B}}\left(\Omega^{B}_{r,s}\right)\right\}. \tag{2}$$

Then Hamming distance is determined (formula 3):

$$\rho\left(\underset{\sim}{\Omega}^{A}, \underset{\sim}{\Omega}^{B}\right) = \frac{\sum\limits_{r=1}^{C}\sum\limits_{s=1}^{D}\left|\mu_{\underset{\sim}{\Omega}^{A}}\left(\underset{\sim}{\Omega}^{A}_{r,s}\right) - \mu_{\underset{\sim}{\Omega}^{B}}\left(\underset{\sim}{\Omega}^{B}_{r,s}\right)\right|}{\left|\Omega^{A}\right|} \tag{3}$$

For the analyzed segment and all prototypes available in the database, the values of the relative Hamming distance are calculated $\rho\left(\underset{\sim}{\Omega}^{A}, \underset{\sim}{\Omega}^{B}\right)$ and sorted in descending order: $\rho\left(\underset{\sim}{\Omega}^{A_1}, \underset{\sim}{\Omega}^{B_1}\right) \le \rho\left(\underset{\sim}{\Omega}^{A_2}, \underset{\sim}{\Omega}^{B_2}\right) < \ldots < \rho\left(\underset{\sim}{\Omega}^{A_N}, \underset{\sim}{\Omega}^{B_N}\right)$.

5.2 Digit identification a using a neural network deep learning

An important point for deep learning neural networks training is building a dataset to train the model.

To construct and train the model, a well-known and rather complete database of handwritten digits MNIST was chosen [18-24, 30]. This database (MNIST - Modified

NIST) is a subset of the larger NIST database [31], which contains handwritten images segmented by images of specially prepared templates filled by respondents from the Census Bureau and students of US educational institutions. MNIST consists of a training (60,000 images) and test (10,000) parts, in addition, images obtained from different authors were placed in different parts to increase uniqueness.

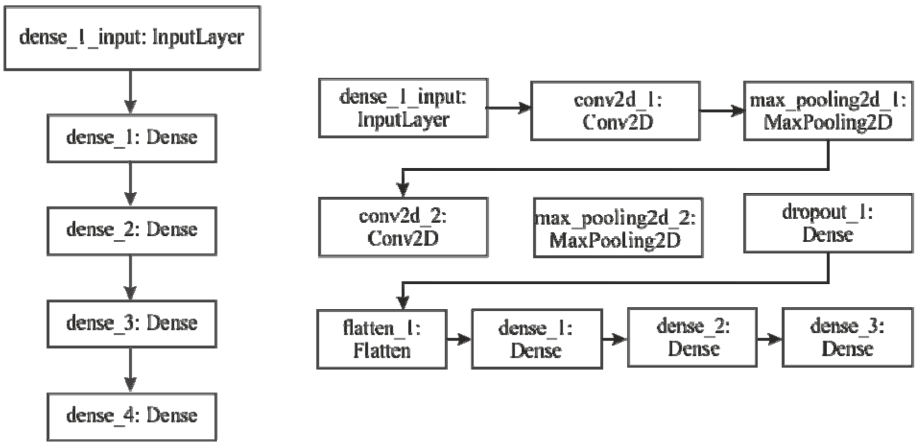
The Keras library contains numerous implementations of widely used building units of neural networks, such as layers, target and transfer functions, optimizers, and many tools to simplify the work with images and text.

The model for recognizing digits included the input and output layers, as well as one or more hidden layers. The model was trained using a variable number of steps (epochs). Recognition images to be scaled to a size of 28x28 (784 cells in a one-dimensional representation), so the number of neurons in the input and hidden layers was assumed to be 784.

The output level is a layer with 10 nodes tf.nn.softmax, which returns an array of ten probability estimates, the sum of which is 1. Each node contains an estimate that indicates the likelihood that the current image belongs to one of 10 classes.

For recognition, each image of the digit was converted to a size of 28x28, and then fed to the input of a pre-trained neural network.

Variants of structures of the deep learning neural network based on Keras framework, which were used to recognize car number elements, are shown in Fig. 3. The structure of simpler version of a fully connected neural network is shown in Fig.3a. The structure of a more complex convolutional neural network (CNN) is shown in Fig. 3b.



a) sequential neural network (feed-forward network – FNN) with three dense inner layers

b) convolutional neural network (CNN)

Fig. 3. Variants of structures of the deep learning neural network based on Keras framework, which were used to recognize car number elements

The stages of constructing and training a model and recognition of practical samples are easily separated due to the possibility of exporting models. To do this, the model.save method (filepath) was used to save the Keras model in one HDF5 file, which contains:

- architecture of the model, allowing to restore the model;
- model weight;
- training configuration (loss calculation function, optimizer)
- the state of the optimizer, which allows to resume training exactly where it was stopped.

To restore the model, the function keras.models.load_model (filepath) was further used. This function also allows to build a model using the saved training configuration (if the model has never been compiled).

6 Discussion

The stepwise processing of the input image with an illustration of the achieved results is performed. The processing of ladle car images with numbers 026 and 030 is presented. During identification with different standards, the ladle car image segment with #026 was compared. Examples of original images with a highlighted recognition area are shown in Fig. 4.

Fig. 4. Examples of original images with a selected recognition area

A comparison of the obtained car segments with all the prototype segments stored in the database was made. When highlighting the elements of the digit, it is clear that due to the distortion of the number by the traces of previous numbers in the image of ladle car #26, the recognition area is slightly increased. When using this recognition method, the resulting area (full image of the digit) was obtained by drawing a rectangle covering all areas with digits. The smallest value of the Hamming distance was used to determine the standard and the corresponding digit in the car image. From Fig. 5, it is seen that the analyzed segment corresponds to the reference #026, i.e.

identification is successful.

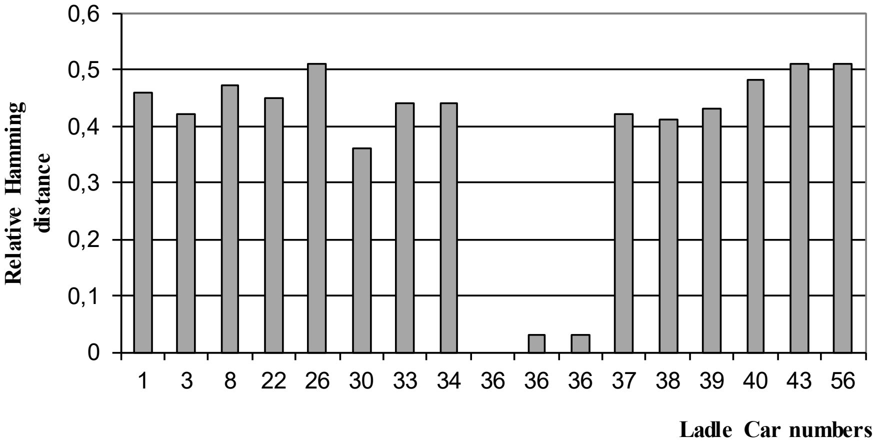
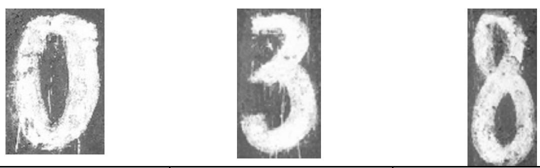


Fig. 5. Relative Hamming distance when comparing ladle car segment #026 and ladle car standards with numbers: № 01, 03, 08, 022, 026, 026, 030, 033, 034, 036, 037, 038, 039, 040, 043, 056

A simpler version of a fully connected neural network in the test set provided a recognition error level of 1.3-1.5%, but in real examples, the recognition accuracy did not exceed 70% (see Fig. 6). For elements of the ladle car number (see the example in Fig. 6), binarization and morphological "closing" operations (MORPH_CLOSE in terms of the OpenCV library) were performed before recognition.

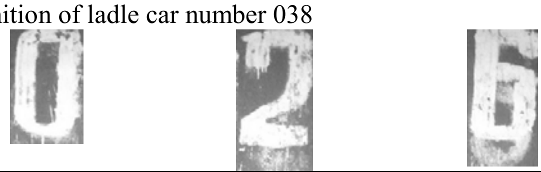
Highlighted igits of number



Recognized by FNN	0	3	5
Recognized by CNN	0	3	8

a) example of recognition of ladle car number 038

Highlighted digits of number



Recognized by FNN	0	2	5
Recognized by CNN	0	2	8

b) example of recognition of ladle car number 026

Fig. 6. Examples of recognition of car number digits by various neural networks

The disadvantage of this type of networks is a significantly longer duration of training. On all versions of test photographs of cars with a qualitatively applied numbers, all digits were accurately recognized. The exception was ladle car number

026 (see Fig. 6b), where one of the digits was recognized incorrectly due to a defect in number application.

In the sample of 15 ladle car numbers, which included a total of 42 digits (12 three-digit and 3 two-digit numbers), 41 digits were correctly recognized using a convolutional neural network, so that the recognition accuracy was 97.6%. Recognition errors are connected with poorly written numbers.

When setting up the model according to the MNIST test data or a set of images for further recognition, it was found that the recognition accuracy increases slightly with the increase in the number of hidden layers. The model setup is much more sensitive to the choice of the type and parameters of the optimizer and the number of training epochs.

7 Conclusions

1. Two versions of information technology for recognizing the manually written numbers of hot-metal ladle cars from their photographs were built.
2. A technique to select the area of interest in photographs containing handwritten digits for further recognition, based on the use of the OpenCV library has been worked out.
3. A computer experiment to identify the numbers of car ladles using fuzzy comparisons according to the technology proposed in the article confirmed a rather high processing efficiency. Thus, the relative Hamming distance for comparing images of photo segments of the same car for different conditions was 0.01-0.16, for different cars – 0.26-0.49, which is sufficient to decide whether the proximity estimate is in the range of external variations of conditions for shooting the car number corresponding to the standard or does not belong to this standard. Errors of the first kind are in the range 0-0.07, and errors of the second kind are in the range 0-0.01. The average identification error is 4%.
4. The possibility of using deep learning neural networks constructed with the help of Keras library to solve the problem of recognizing the numbers of hot-metal cars, including handwritten digits, is shown. For handwritten digit recognition, the best recognition accuracy is provided by a convolutional neural network, with which 97.6% of car ladle digits were correctly recognized.

References

1. González, A., Bergasa, L.M.: A text reading algorithm for natural images. Image and Vision Computing. **31(3)**, 255–274 (2013)
2. Ubozhenko, N.V.: Analysis of the effectiveness of character recognition methods in the framework of the license plate recognition task of vehicles. Prospects for the development of information technologies. **12**, 41–45 (2013)
3. Kachanovsky, Yu.P., Yavtukhovich, A.G.: Development of a license plate localization algorithm for use in a distributed hardware-software complex for license plate recognition. Information Technologies for Modeling and Control. **39**, 508–516 (2007)
4. Petrov, S.P.: Convolutional neural network for recognition of license plate symbols of a car. System analysis in science and education. **21**, 66–73 (2013)

5. Sanver, U.: Identification of train wagon numbers. In: Proceedings of the 2014 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference, pp. 63–68. St. Petersburg (2014)
6. Lisanti, G.A, Karaman, S.,·Pezzatini, D.,· Del Bimbo, A.: Multi-Camera Image Processing and Visualization System for Train Safety Assessment. Multimedia Tools and Applications. **77(2)**, 1583–1604 (2018)
7. Shin, Shi: Emgu CV Essentials. Packt Publishing (2013)
8. Chernopyatov, A.V.: Neural network recognition system for railway car numbers. Perm University Herald. Mathematics. Mechanics. Computer science. **4 (23)**, 96–102 (2013)
9. Kazansky, N.L., Popov, S.B.: Distributed system of technical vision for registration of trains. Computer Optics. **36(3)**, 419–428 (2012)
10. Volotovsky, S.G., Kazansky, N.L., Popov, S.B., Khmelev, R.V.: Vision system for recognition of railway tank numbers using a modified correlator in the Hausdorff metric. // Computer Optics. **27**, 177–184 (2005)
11. Obukhov, A.V., Lyasheva, S.A., Shleimovich, M.P.: Methods of automatic recognition of license plates. Bulletin of the Chuvash University. **3**, 201–208 (2016)
12. Surekha, P., Pavan Gurudath, P., Prithvi, R., Ritesh Ananth, V.G.: Automatic license plate recognition using image processing and neural network. ICTACT journal on image and video processing. **8(4)**, 1786–1792 (2018)
13. Hashmi, S., Kumar, K., Khandelwal, S., Lochan, D., Mittal, S.: Real Time License Plate Recognition from Video Streams using Deep Learning. International Journal of Information Retrieval Research. **9(1)**, 65–87 (2019)
14. Björklund, T., Fiandrotti, A., Annarumma, M., Francini, G., Magli, E.: Automatic license plate recognition with convolutional neural networks trained on synthetic data. In: Proceedings of the 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), Luton, UK, 16–18 October 2017, pp. 1–6 (2017)
15. Khan, J.A., Shah, M.A., Wahid, A., Khan, M.H., Shahid, M.B.: Enhanced car number plate recognition (ECNPR) system by improving efficiency in preprocessing steps. In: 2017 International Conference on Communication Technologies (ComTech), pp. 156–161. Rawalpindi (2017)
16. Lofrano Corneto, G.A., Assis da Silva, F., Pereira, D.R., et al.: New Method for Automatic Vehicle License Plate Detection. IEEE Latin America Transactions. **15(1)**, 75–80 (2017)
17. Schmidhuber, J.: Deep Learning in Neural Networks: An Overview. Neural Networks. **61**, 85–117 (2015)
18. Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., Khudanpur S.: Recurrent neural network based language model. In: 11th Annual Conference of the International Speech Communication Association. Japan, pp. 1045–1048 (2010)
19. Nikolenko, S., Kadurin, A., Arkhangelskaya, E.: Deep Learning, Peter, St. Petersburg (2018)
20. Vyugin, V.V.: Mathematical foundations of the theory of machine learning and forecasting. Moscow (2013)
21. Arif, R.B., Siddique, M.A., Khan, M.M., Oishe, M.R.: Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Convolutional Neural Network. In: 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), pp. 112–117. Dhaka, Bangladesh (2018)
22. Siddique, F., Sakib, S., Abu Bakr Siddique, Md.: Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers. In: 2019 5th International Conference on

Advances in Electrical Engineering (ICAEE) (2019). https://arxiv.org/pdf/1909.08490.pdf. Accessed 16 Dec 2019

23. Brownlee, J.: Handwritten Digit Recognition using Convolutional Neural Networks in Python with Keras. https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/. Accessed 16 Dec 2019
24. Arif, R.B., Siddique, A.B., Khan, M.R., Oishe, M.R.: Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Convolutional Neural Network. In: 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), pp. 112–117. Dhaka, Bangladesh (2018)
25. Isogawa, K., Ida, T., Shiodera, T., Takeguchi, T. : Deep Shrinkage Convolutional Neural Network for Adaptive Noise Reduction. IEEE Signal Processing Letters. **25(2)**, 224–228 (2018)
26. Tavanaei, A., Maida, A.S.: Multi-layer unsupervised learning in a spiking convolutional neural network. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2023–2030. Anchorage, AK (2017)
27. Sable, A.: Building Custom Deep Learning Based OCR models. https://nanonets.com/blog/attention-ocr-for-text-recogntion. Accessed 16 Dec 2019
28. Gonzalez, R., Woods, R.: Digital image processing. 3rd ed., Rev. and add. Technosphere. Moscow (2012)
29. Serhiienko, A.V.: Method for the localization of informative areas using threshold image processing and clustering. Railway information and control systems: Collection of scientific papers, Issue. **4 (101)**, 8–14 (2013)
30. Kargin, A.A., Serhiienko, A.V.: Identification of the number of iron bucket by image segment by comparing it with the reference segment. Problems of Information Technologies, Collection of articles, scientific proceedings. **7**, 114–121 (2010)
31. LeCun Y. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist. Accessed 16 Dec 2019
32. Grother P.J. Nist special database 19 – handprinted forms and characters database // National Institute of Standards and Technology (NIST), Tech. Rep. 1995. https://www.nist.gov/srd/nist-special-database-19. Accessed 16 Dec 2019