

Application of CNN-Based Method for Automatic Detection and Classification of the IC Packages

Kamil Marek Maliński¹ and Krzysztof Okarma¹

Abstract—Automatic detection and classification of integrated circuits' packages is one of the methods supporting the traditional production of electronic parts based on through-hole technology, typical for Printed Circuit Boards (PCBs), utilizing the advantages of modern machine vision solutions. As a result of the growing availability of cameras and 3D printers, as well as the popularity of IoT systems, prototyping of some electronic circuits with the use of simpler robotic systems may be supported by an automatic analysis of electronic components based on machine vision. Considering recent advances in the applications of Convolutional Neural Networks (CNNs) in computer vision, their applicability for this task has been analyzed and experimentally verified in this paper, also in comparison to previously proposed approach based on handcrafted features.

I. INTRODUCTION

Increasing importance of modern hardware and software solutions in everyday life is beyond doubt. Development of „intelligent” vehicles, robots, portable and mobile devices, progress in 3D printing, Internet of Things (IoT) or sensing technology are some of the most relevant indications of hardware aspects of Industry 4.0. Although nowadays, the mass production of electronic parts is mainly based on sophisticated industrial robotic systems and pick-and-place machines used for assembling small surface-mount devices (SMD), there is still a room for an independent development of some dedicated solutions based on traditional PCBs. Nevertheless, the required precision of assembling electronic parts, as well as their proper recognition, should be supported by machine vision systems to ease the development of the prototyped devices. One of the possible solutions may be related to semi-automated assembling systems with recognition of shapes and marking of electronic packages, whereas another possibility of the use of image analysis is the final verification of the appropriateness of the assembled integrated circuits (IC) packages.

Considering the possible limitations related to lighting conditions, a classification of the integrated circuits may be based on the shape analysis utilizing binary images combined with additional recognition of markings. A two-step solution based on the detection of the IC packages using the binary shape analysis with further classification with the use of a multi-descriptor approach applied for natural images, has been presented in [1], leading to 88.3% of the classification accuracy. Nevertheless, since the results obtained initially

with the use of the CNNs have not been satisfactory, mainly due to a limited number of available training images, the next step of research has been related to the improvement of the CNN architecture and parameters, preserving its relatively simple structure.

The contribution of this paper is twofold. Firstly, the innovative CNN-based solution for the automatic detection and classification of IC packages and their types has been proposed and verified. Secondly, the application of a genetic algorithm for the evolution of the network structure has been presented making it possible to avoid the time-consuming experimental verification of many possible combinations of parameters. Finally, the combination of the developed CNN-based method with the use of handcrafted features based on shape descriptors has been proposed and verified experimentally, leading to promising results.

The detailed structure of the proposed network has been obtained using a genetic algorithm, as described further, whereas some starting parameters have been selected experimentally. The choice of descriptors used in the shape descriptor analysis is based on the results presented in our previous paper [1].

The significance of the proposed approach might be related to the support of a recycling of electronic parts and products as well as an automatic detection of defects introduced by pick-and-place machines or diagnostics of electronic devices (using a proper recognition method of electronic parts as the basis for further testing of their work and I/O signals), as well as the verification of the assembled electronic elements.

II. RELATED RESEARCH

The main motivation of the conducted research is the development of a system presented in 2017 by Fujitsu company [2], where some undoubted advantages of an automatic machine vision in electronic product manufacturing have been confirmed. Application of image recognition system on a parts assembly line for inspection has been based on the use of genetic programming, since its inventors have stated that „with a lot of data, deep learning could be used to generate an original algorithm, but here we need a different approach” [2]. However, as the result the 80% reduction of the inspection processes time as well as shortening the work hours by 33% has been achieved. Nevertheless, a potential application of CNNs and its comparison with the approach based on handcrafted features [1] seems to be worth investigating even for a limited amount of data.

Most of research papers related to the problem of shape recognition applied to electronics consider the automatic

¹Kamil Marek Maliński and Krzysztof Okarma are with Department of Signal Processing and Multimedia Engineering, Faculty of Electrical Engineering, West Pomeranian University of Technology in Szczecin, 70-313 Szczecin, Poland kamil.malinski@zut.edu.pl okarma@zut.edu.pl

analysis of circuit diagrams. Some interesting results have been obtained for hand-drawn symbols of electronic elements [3], [4], [5], although the main assumption of the presented methods has been the analysis of binary images obtained from well-illuminated photos or scans. Nevertheless, the experiments have been conducted for relatively small number of images and therefore the CNN-based approach has not been used. Another example may be the method [6] utilizing Hidden Markov Models (HMMs) applied for the classification of 6 types of electronic components on hand-drawn sketches with the accuracy above 83%.

The authors of the paper [7] have presented an automatic inspection vision system for chip position and direction inspection combined with character recognition in tape-and-reel process. Another recently proposed solution for the inspection of the ball grid array (BGA) package electronic components [8] makes it possible to determine various cracking types of solder balls. Although there are some commercial systems available, which may be helpful in industrial machine vision applications, many solutions are specialized and may be applied effectively only for specific purposes. Hence, the results and methods presented in the paper, might be considered as an addition to some existing solutions, improving their universality, providing a good starting point for further development.

III. IDEA OF THE PROPOSED SOLUTION

The proposed method of automatic analysis of IC packages is based on the assumption that their classification may be based on their shape without the use of the OCR engines, which could be applied during the second stage of the analysis, not discussed in this paper. Since there are no assumptions regarding the mounting of elements of the PBC and it has been assumed that all IC packages might be randomly scattered, the proposed method should be rotation independent. Therefore, the dataset of images of the IC packages used for the development and verification of the classification method should contain a variety of images acquired for various locations and orientations of electronic elements. Hence, the dataset of 400 binary images obtained from the STL 3D models of the IC packages has been created, which contains the cropped shapes of them obtained assuming various locations of the virtual camera. Next, the images have been rotated 10 times with a step of 18 degrees, leading to 4000 images due to the assumed symmetry of the packages. Such a dataset has been used together with a database of 1000 randomly selected shapes, other than the IC packages, e.g. geometrical shapes, various objects, symbols, characters, animal and plant shapes, etc., in the first step of experiments related to the detection of the IC packages among some other objects using binary images. The dataset of 5000 binary images has been divided into 5 groups (1000 images each): DIP-16 packages, DIP-20 packages, DIP-8 packages, TO220 packages and random non-IC images. In the first two experiments, determining the optimal input image size and internal structure of the network, 700 images from each set were used as training

data, and 300 as validation data. During the next test, the ratio was adjusted gradually, to achieve best accuracy. Additionally, a set of 488 natural images has been used in the final verification, with 288 images of IC packages. Some of them are presented in Fig. 1.

The basic neural network was built in MATLAB, using the Deep Learning Toolbox. Its structure, illustrated in Fig. 2, is based on the CNN (Convolutional Neural Network) architecture with multiple convolutional stages. The network consists of the following layers:

- image input layer — takes square, binary images as inputs,
- one or more internal stages (described below),
- fully connected layer — applies all the weights and bias vectors,
- softmax layer — applies the softmax function, normalizing the outputs into a probability distribution,
- classification layer — computes the cross-entropy loss for classification.

Each internal stage is built using the same layers, but the number of neurons in each convolution layer is doubled in comparison to the previous one. The first convolution layer has 8 neurons, the second — 16, third has 32 layers and so on. The structure of each internal stage is as follows:

- convolution layer — applies the convolution filters of size 3,
- batch normalization layer — normalizes the results of the convolution,
- Rectified Linear Unit (ReLU) layer — performs the thresholding operation, where all negative values are set to 0,
- one or more convolutional stages,
- max pooling layer — downsamples the input by dividing it into regions and computing the maximum for each region.

The parameters and the structure of the network were subject to optimization during the experimental verification stage.

IV. EXPERIMENTAL VERIFICATION

A. Tuning the Network Parameters

Three experiments have been conducted to find the most effective parameters of the network built. The goal of the first test has been the comparison of results of training using images of different size — 25×25 , 50×50 and 100×100 pixels. The second test has been conducted to compare the accuracy of networks trained using various numbers of convolutional stages — from 1 to 5. During the final experiment, the ratio of training data to test data has been progressively changed, and its impact on the classification accuracy has been evaluated.

The results of the first experiment related to the choice of the appropriate image size are presented in Table I. Since the 50×50 pixels images have led to the best accuracy, this size has been chosen for the subsequent tests. However, this accuracy is only about 1.5% higher than achieved for 25×25



Fig. 1. Exemplary natural images used in the experiments

TABLE I
EXPERIMENTAL RESULTS ACHIEVED FOR VARIOUS IMAGE SIZE

Input image size	Accuracy [%]	Total training time [s]
25 × 25 pixels	88.27	7
50 × 50 pixels	89.73	18
100 × 100 pixels	89.00	80

pixels images, being much slower, hence for applications where computing resources are limited, the smaller image size might be a reasonable choice as well.

The goal of the second experiment has been to find out how many internal stages may ensure the best results. Hence, the network has been trained five times using the image size determined in the previous step (50 × 50 pixels), adding one internal stage each time. The resulting accuracy and the impact of computation time are compared in Table II. Obtained results seem to indicate, that the use of more stages leads to better results, although testing even more layers was impossible due to the number of neurons compared to the selected input size.

In the next experiment consisting of 5 training sessions, different ratio of training data to test data have been chosen. The ratio which has given the best results is 90% training

TABLE II
EXPERIMENTAL RESULTS ACHIEVED FOR VARIOUS NUMBER OF STAGES

Number of internal stages	Accuracy [%]	Total training time [s]
1	82.67	13
2	85.40	16
3	86.67	19
4	89.53	21
5	91.50	26

TABLE III
EXPERIMENTAL RESULTS ACHIEVED FOR VARIOUS RATIOS OF TRAINING TO TESTING DATA

Training/test ratio	Accuracy [%]
50 / 50	90.12
60 / 40	91.90
70 / 30	91.67
80 / 20	92.60
90 / 10	93.80

TABLE IV
EXPERIMENTAL RESULTS ACHIEVED FOR THE DATASET OF 488 NATURAL IMAGES WITH 288 REPRESENTING IC PACKAGES

Classification method	Shape analysis	Neural network
Total	488 (100%)	488 (100%)
IC packages	240 (49.18%)	240 (49.18%)
Others	248 (50.82%)	248 (50.82%)
TP + TN	481 (88.32%)	265 (54.30%)
FP + FN	57 (11.68%)	223 (45.70%)
FP	42 (8.61%)	223 (45.70%)
FN	15 (3.07%)	0 (0.00%)

data and 10% test data, as expected (results provided in Table III). Interestingly, the choice of 60% of the training data has led to slightly better results than obtained for 70% of images used for training. Finally, the final design of the network takes 50 × 50 pixels images with 5 internal stages and 90% of the dataset is used as training images, whereas 10% is used as the test data. The illustration of the training progress for these parameters is presented in Fig. 3.

B. Evaluation of the Created Network and Comparison with Multi-Descriptor Binary Shape Analysis

As the final evaluation of the proposed method of classification of binary images, some tests for a similar dataset of images, as in the previously proposed classification method [1] have been conducted. Both approaches have been evaluated considering their ability to detect IC packages and classify the package type. First, the ability of both methods to detect integrated circuits has been compared leading to the results presented in Table IV, where TN, TP, FN and FP denote true negatives, true positives, false negatives and false positives, respectively.

The result of detection has been significantly lower for the neural network since it has confused almost all non-IC images for integrated circuits. Interestingly, there have been no false negatives, which indicates that the recognition of random images is particularly troublesome for the network. This might be caused by too small dataset of random images as well as its insufficient diversity. Probably a separate network, which only uses two classes (IC and non-IC) would be better in this case, however that would cause the necessity of having two separate processes which is not an optimal solution. In such case, using the additional network as a detector would be practically pointless.

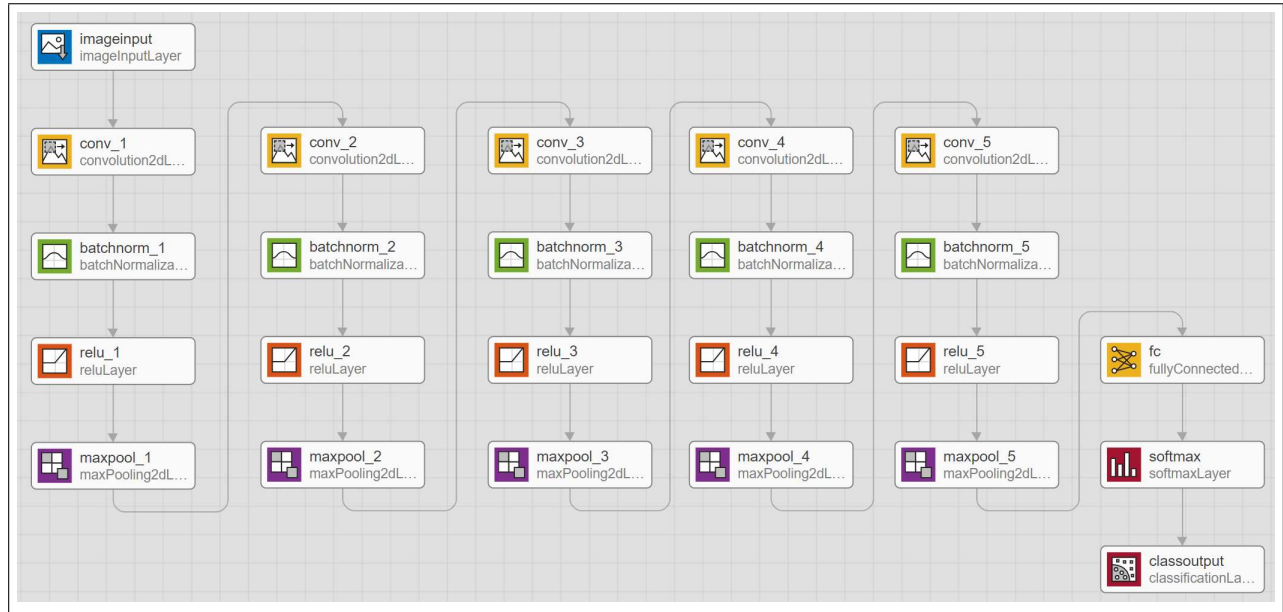


Fig. 2. The basic structure of the proposed convolutional network

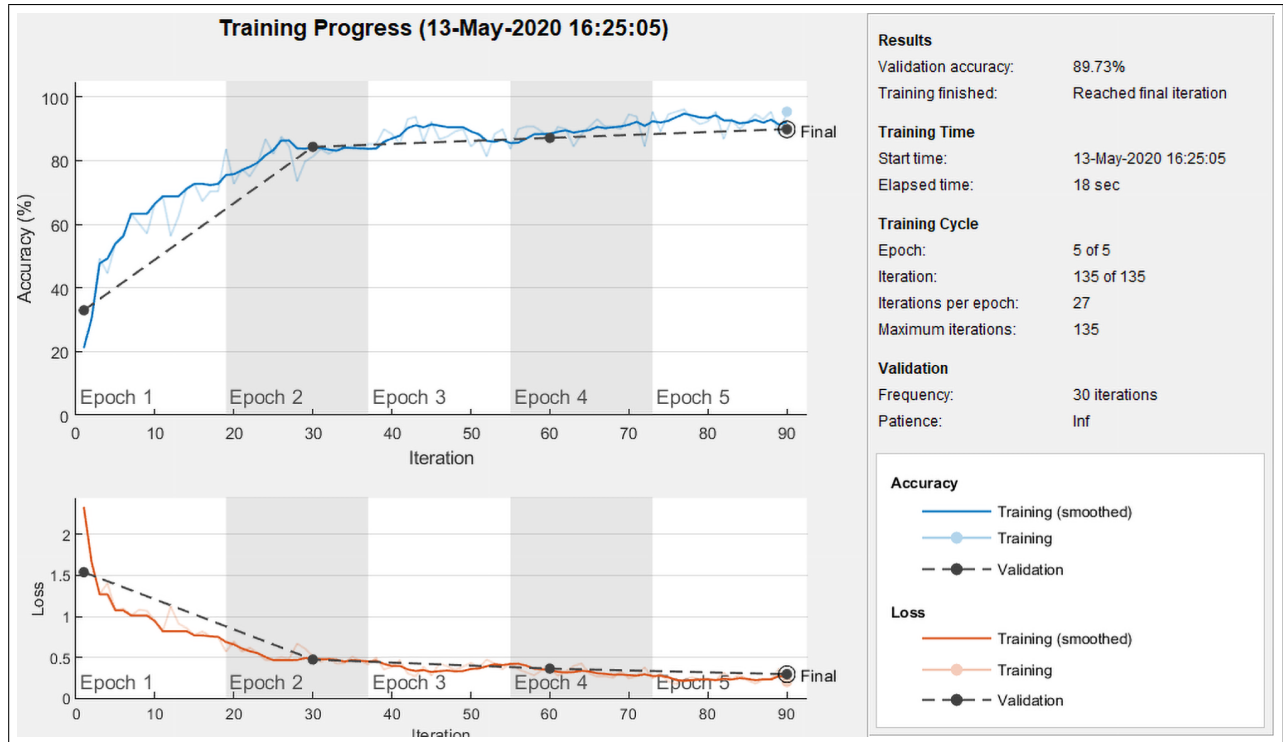


Fig. 3. Training progress of the developed network for 50×50 pixels input images (for 5 internal stages and 90% training data)

Nevertheless, the final experiment has been related to testing the network's accuracy for determining the IC package on the image and comparing it to the results achieved using the Multi-Descriptor Binary Shape Analysis method [1]. The accuracy of the neural network falls behind the accuracy of previous method based on shape analysis and handcrafted features, but only by less than 1%, as shown in Table V. This means, that this type of network can be very successful when used as an IC package classifier, especially if a larger

image dataset would be used for training.

Analysing the results presented in Table V, significant differences may be observed for the DIP-16 and DIP-20 packages, which are quite similar to each other (dual in-line packages with 8 or 10 pins on each side). Hence, a natural direction of future research may be the combination of both proposed methods towards further increase of the classification accuracy.

TABLE V
CLASSIFICATION ACCURACY FOR INDIVIDUAL IC PACKAGES

Package type	Shape analysis	Neural network
DIP-8	93.3%	91.5%
DIP-16	81.7%	72.8%
DIP-20	83.3%	90.7%
TO220	95.0%	96.0%
Average	88.3%	87.8%

Another interesting problem is related to improperly classified images, which is partially caused by the presence of some background information, such as shadows, as illustrated in Fig. 4. In fact, achieving the 100% accuracy is most likely impossible even for perfectly binarized images, since different IC packages might be really indistinguishable from each other, even for a human expert. For example, looking at DIP packages side-on, in low resolution, makes their recognition extremely difficult even for a human expert, as illustrated in Fig. 5.

C. Evolution of the Neural Network Structure Using a Genetic Algorithm

After determining the network parameters, which were easily found by iterating through all possibilities, the more detailed parameters (convolution mask size for each layer and the number of neurons in each layer) were established using a genetic algorithm. Although, theoretically, the entire set of parameters could have been found using this method, it would be very expensive from computational point of view. Hence, a genetic algorithm was used for this purpose, partially inspired by some recent works of the other researchers, where genetic algorithms were applied together

Algorithm 1 The genetic algorithm for finding the best neural network parameters

```

1: procedure EVOLVENETWORK
2:   Generate initial population of size  $N$ 
3:   Generate the gene pool template
4:   for  $it \leftarrow 1$  to  $numberOfIterations$  do
5:     for  $i \leftarrow 1$  to  $N$  do
6:       Generate a neural network from the selected specimen
7:       Train the network
8:       Evaluate the network
9:     end for
10:    Sort the generation by score in descending order
11:    Remove the worst performing networks
12:    Check if a new maximum fitness score has been achieved
13:    Generate a new generation based on the scores of the previous one
14:    Add random specimen to the new generation
15:  end for
16: end procedure

```

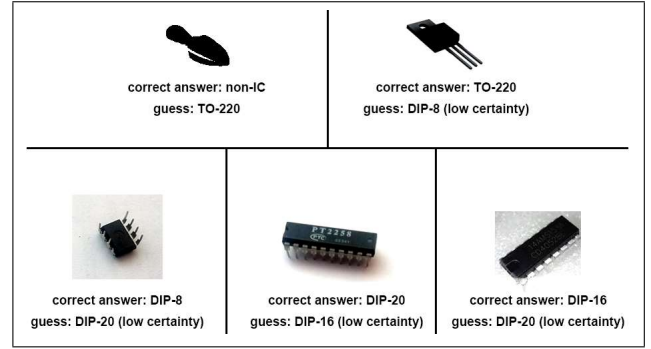


Fig. 4. Illustration of exemplary improperly classified IC packages



Fig. 5. Illustration of similarity of the side-view of different IC packages

with neural networks for similar purposes, like handwriting recognition [9] and image classification [10], [11], [12].

The genome for the genetic algorithm consisted of ten integers. The first five represented the mask size (in the range from 3 to 6 pixels), whereas the latter last five denoted the numbers of neurons in each layer, as illustrated in Fig. 6.

The maximum number of neurons for each layer was set to be 128, whereas the minimum was derived from the formula:

$$N_{min} = 1 + 10 \cdot (L_n - 1), \quad (1)$$

where L_n is the number of layer.

In each iteration, the algorithm created and trained networks for each set of variables and evaluated them. The

Algorithm 2 Generating the gene pool template

```

1: procedure GENERATE THE GENE POOL TEMPLATE
2:   Generate initial population of size  $N$ 
3:   Generate gene pool template
4:   for  $i \leftarrow 1$  to  $\text{round}(P * C_{survival}) - i + 1$  do
5:     Add  $i$  to the gene pool
6:   end for
7: end procedure

```

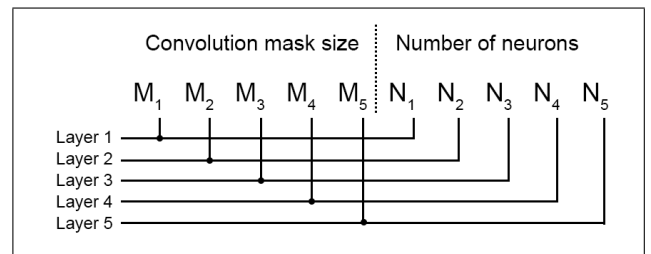


Fig. 6. Illustration of the structure of the genome used for the evolution of the neural network

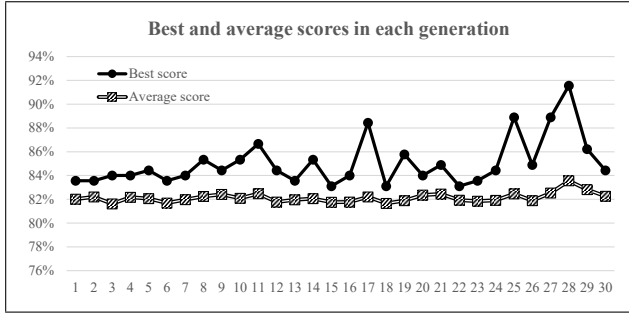


Fig. 7. Illustrations of the best and average results obtained in each generation of the neural network's evolution

best networks in each generation were used to create the gene pool for the next generation – the higher the score, the more slots in the gene pool the network occupies. After creating a new set of network templates, they were subject to random mutations, which changed the selected parameters in the genome. The occupied space was determined as:

$$GP_{spaces} = \text{round}(P \cdot C_{survival} - Rank + 1), \quad (2)$$

where GP_{spaces} is the number of gene pool spaces the specific network will occupy, P is the population size, $C_{survival}$ is the survival coefficient and $Rank$ is the ranking place of each individual network in its generation. The idea of the method is presented as the Algorithms 1 – 2. The illustration of the best results obtained in each generation is presented in Fig. 7.

Considering the computational requirements for each validated network, the number of specimen in each generation was limited to 20, whereas the number of generations was set to 30. Furthermore, each generated specimen has to be unique, so that no combination of network parameters is checked more than once. Although these assumptions somewhat limit the ability of the algorithm to traverse the search space, they significantly decrease the execution time.

Algorithm 3 Final classification using both methods in conjunction

```

1: procedure CLASSIFY
2:   Analyze the image using shape analysis
3:    $G_s$  is the IC package guess with certainty  $C_s$ 
4:   Analyze the image using the neural network
5:    $G_n$  is the IC package guess with certainty  $C_n$ 
6:   if  $G_s == G_n$  then
7:     return  $G_s$ 
8:   else
9:     if  $C_s > C_n$  then
10:      return  $G_s$ 
11:    else
12:      return  $G_n$ 
13:    end if
14:  end if
15: end procedure

```

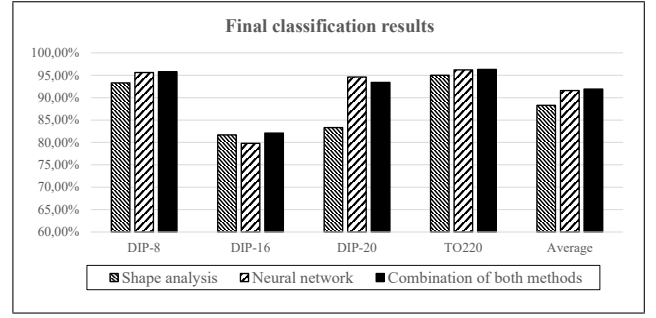


Fig. 8. Comparison of the classification accuracy obtained for various methods

D. Final Combination of Both Methods

After the best neural network was found and trained, the final concept for IC package classification was formed as the use of the neural network and shape descriptor analysis in conjunction. Both methods give „certainty” percentages, which were used as weights for their respective guesses if the methods led to different ones, as presented in Algorithm 3.

Finally, the application of both approaches together provides better results than their separate use, as shown in Fig 8 where the obtained classification accuracy is compared with the results obtained for the shape analysis (provided earlier in Table V) and the improved neural network (with the parameters optimized using the genetic algorithm).

Regardless of the overall increase of the recognition accuracy provided by the proposed combined approach in comparison to individual methods, an important advantage of the improved method is its better universality, which may be observed e.g. comparing the classification results of DIP-16 and DIP-20 packages.

V. SUMMARY

The obtained experimental results for the neural network are not completely unexpected, given the limited database of images. Since neural networks need big and diverse sets of input images to train them, this result might be much better with a much larger database of natural images used during the training stage. Another issue, which might be less troublesome for classical shape analysis methods, is the fact that the neural network has been trained using binary images, hence a proper classification of natural images has been challenging. Since such images need to be scaled and converted to binary images, some distortions influencing the object's shape may cause the decrease of the classification accuracy. Nevertheless, its improvement would require a significant enlargement of the testing dataset consisting of natural images, apart from synthetic ones.

The presented approach to the application of CNNs for an automatic recognition and classification of IC packages confirms the usefulness of the deep learning methods in machine vision, including some specific tasks with limited amount of training data, also using only synthetic training images. Nevertheless, the results obtained using the CNN-based method, although promising, do not outperform in

a significant manner those achieved using handcrafted features. Although in general both considered methods are very effective for classification of IC packages, none of them is significantly better than the other. Using them together has largely given the best results so far, however this process might be too complicated for some applications.

Therefore, the development of some even better methods combining both such approaches, due to their complementary properties, should be considered as a topical challenge also for further research.

REFERENCES

- [1] K. Maliński and K. Okarma, "Application of multi-descriptor binary shape analysis for classification of electronic parts," *Journal of Universal Computer Science*, vol. 26, no. 4, pp. 479–495, 2020.
- [2] Fujitsu. (2017) AI-enabled image recognition system to revolutionize the manufacturing line. Fujitsu Journal. [Online]. Available: <https://journal.jp.fujitsu.com/en/2017/04/19/01/>
- [3] B. Edwards and V. Chandran, "Machine recognition of hand-drawn circuit diagrams," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 6, June 2000, pp. 3618–3621.
- [4] J.-P. Valois, M. Cote, and M. Cheriet, "Online recognition of sketched electrical diagrams," in *Proceedings of Sixth International Conference on Document Analysis and Recognition (ICDAR)*, September 2001, pp. 460–464.
- [5] S. Ma, Y. Sun, P. Lyu, S. Polsley, and T. Hammond, *DCSR: A Digital Circuit Sketch Recognition System for Education*. Cham: Springer International Publishing, 2017, pp. 137–146.
- [6] Y. Zhang, C. Viard-Gaudin, and L. Wu, "An online hand-drawn electric circuit diagram recognition system using Hidden Markov Models," in *2008 International Symposium on Information Science and Engineering*, vol. 2, Dec 2008, pp. 143–148.
- [7] J. Y. Lee, C. H. Han, K. W. Ko, and S. Lee, "Development of vision system for defect inspection of electric parts in the tape and reel package," in *2016 16th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2016.
- [8] C.-Y. Huang, J.-H. Hong, and E. Huang, "Developing a machine vision inspection system for electronics failure analysis," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 9, no. 9, pp. 1912–1925, 2019.
- [9] A. Baldominos, Y. Saez, and P. Isasi, "Hybridizing evolutionary computation and deep neural networks: An approach to handwriting recognition using committees and transfer learning," *Complexity*, vol. 2019, pp. 1–16, 2019.
- [10] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, and A. Kurakin, "Large-scale evolution of image classifiers," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017, pp. 2902–2911.
- [11] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 394–407, 2020.
- [12] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, "Automatically designing CNN architectures using the genetic algorithm for image classification," *IEEE Transactions on Cybernetics*, pp. 1–15, 2020, (In Press).