Meta-heuristic Algorithm As Feature Selector For Convolutional Neural Networks

Dawid Połap[1], Marcin Woźniak[1], Jacek Mańdziuk[2]
[1]Faculty of Applied Mathematics, Silesian University of Technology,
Kaszubska 23, 44-100 Gliwice, Poland
[2]Faculty of Mathematics and Information Science, Warsaw University of Technology,
Koszykowa 75, 00-662 Warsaw, Poland
e-mail: dawid.polap@polsl.pl, marcin.wozniak@polsl.pl, j.mandziuk@mini.pw.edu.pl

2021 IEEE Congress on Evolutionary Computation (CEC) | 978-1-7281-8393-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CEC45853.2021.9504915

***Abstract*—The huge popularity of heuristics contributes not only to the improvement and modeling of new solutions but also to their adaptation to selected goals. Recent years have shown the popularity of their use also in machine learning as a training algorithm or allowing for the selection of optimal architecture or hyper-parameters. In this paper, we propose an adaptation of a nature-inspired algorithm for preprocessing images in a parallel way for obtaining higher classification results. The proposed idea is based on analyzing images by heuristic representative which is Red Fox Optimization Algorithm and returning a specific value. These values are used in deciding to classify the entire image or trim it to eliminate unnecessary objects. We modeled this solution and evaluated using the learning transfer method for VOC 2007 dataset. The obtained results were compared on selected classes to show the advantages of a proposal.**

***Keywords*—Red Fox Optimization, Heuristics, Key-points search, Feature selection.**

I. Introduction

Heuristic algorithms are one of the main methods for analyzing and finding optimal solutions in multi-model problems. This type of technique does not guarantee a proper solution, but infinite time can return an approximate one. Despite this fact, algorithms are good at searching huge solutions spaces that iteratively might be impossible to calculate in a finite time. The applications of this type of technique find a place in many areas of today's industry and research. The main place is primarily finding values that meet your specific conditions in the selected problem.

The last years brought many new algorithms, many of them are inspired by natural phenomena. In [1], the heuristic model was modeled by analyzing the behavior of slime mould. Again in [2], the authors proposed a model inspired by red foxes' hunting, where the individuals surround the victim and then attack. A similar idea can be found in [3] where the mating of black widow spiders was the basics of the creation of a new nature-inspired method. Not only the behavior of different animals is used for inspiration, but also many phenomena which occur in physics. An example is an algorithm that resembles a billiard game, where given solutions are balls and the sought are pockets [4]. Again in [5], the authors described a hyper-heuristic for improving the first population which in general is created at random. Their mechanism was tested on heuristic-based on whale hunting using the bubble-net technique. Many different heuristic algorithms are created for obtaining better results against others. This is measured by the convergence of the method or the way of analyzing solutions in a given space. All of the proposed solutions are built primarily on two stages that are repeated in each iteration, i.e. local and global searching. Some of the newest proposition offers mechanisms which add a new stage for better performance. An example of it is the manipulation of the first generation of the solution in [5], or reproduction stage in [2].

This kind of approach to obtain the approximate solution for a given problem found many applications. In [6], the authors show a new metaheuristic with a communication strategy for the prediction problem of wind power. Another issue where heuristic was applied is accounting model choices [7], or resource allocation in the Internet of Things [8]. A similar approach was shown in [9], where this technique was used for solving the load-balancing problem in a cloud environment. An important issue, where the heuristic is used is machine learning. In general, many issues can be defined as optimization problems. For instance, the training process of many classifiers is based on minimizing loss function, so the heuristic can be used as a method for finding the best configuration of weights [10], or even in hyperparameter selection, [11]. In many cases, hybrid techniques are created by combining heuristic with other solutions like support vector machine in [12]–[14] which was applied for prediction of diffuse solar radiation in selected regions and ecological environmental quality estimation.

It should be noted, that in the last few years, image analysis is made by convolutional neural networks (CNN), which perform feature extraction stage and then classified. However, the hybrid solution is created for obtaining better classification results. An example was shown in [15], where deep learning was used with a heuristic approach in image post-processing in the application for crack detection for tunnel inspection. On the other hand, CNN can be optimized by a nature-inspired algorithm that was proposed in [16], [17]. Another application is to analyze neural architecture and performing optimization of it to find the best configuration. Evolving architectures can be created by improving the classic heuristic algorithm what was shown in [18], where the idea of the ga-pso method was proposed. A similar solution was developed by the use of particle swarm optimization algorithm with binary encoding

978-1-7281-8393-0/21/$31.00 ©2021 IEEE

in [19]. Other ideas are the creation of hybrid, or modified algorithms in combination with CNN [20]. In [21], the ant lion algorithm was used in the texture classification problem, and as a result, the authors proposed a moving convolutional mask. In visual evaluation very important is good model of clustering [22], [23]. Another important research is using swarm methods for feature selection in neuro-fuzzy classifier and other classification methods [24]–[26].

Based on this observation, we propose a new model for preprocessing images using a heuristic approach before the classification by the convolutional neural network. The presented idea is based on analyzing images by the selected representant of nature-inspired heuristic (Red Fox Optimization Algorithm) using a different fitness function modeled for the image classification problem. A heuristic population returns a function's values that are used in the basic analysis of image areas. Based on these values, a simple validation mechanism decides whether the image should be cropped or not. In case of a positive decision to crop, the cropped operation is performed and the remaining area is extended to the required size of the convolutional network. This solution offers a quick decision on whether the whole image should be analyzed by the network or just a fragment. This solution was evaluated and discussed to show the potential advantages.

II. Proposed mechanism

A given image I is defined as $w \times h \times d$, which describes the main three quantities. w and h means respectively, the width and height of the image, and d means the dimension of it. The last parameter d is defined according to the selected color model like RGB (*Red-Green-Blue*), or HSL (*Hue-Saturation-Lightness*). In all cases, the d means the number of components in the used color system (for RGB and HSL, $d = 3$). The main idea is based on adding three steps before the classification by CNN.

The first step is a parallel analysis of the image by Red Fox Optimizer on the resized image (resize is important due to the constant size of the image used by the classifier). A second one is making the decision about obtained results from heuristics whether the image should be cropped. If no, there is no additional action made and it is sent to the classifier. If yes, the image is cropped by the selection of the most important function (chosen in the previous step). After cropping the image, the obtained result is resized to the dimension required by the classifier and send to it.

A. Red fox optimization algorithm as feature selector

The proposed idea is based on applying many different instances of the same heuristic algorithm in a parallel way for analyzing the same image but with other fitness functions. For this purpose, we define a set of fitness function which can be used in image analysis based on two-color models RGB and HSL. Proposed functions can be freely modified to the particular problem, but we wanted to show some popular function which operates on color models values. It must be noted, that a function gets one point $\overline{x} = (x_0, x_1)$, but analyze a grid of size 3×3, where a given pixel is in the middle (the point $\overline{x}$ is understood as a pixel on that position). A first function analyzes image by searching the darkest areas and can be defined as an average value for the RGB model as:

$$\begin{aligned} f_1(\overline{x}) =& \frac{1}{3} \cdot \frac{1}{9} \sum_{i=-1}^{1} \left[R(x_0+i, x_1+i) \right. \\ & \left. + G(x_0+i, x_1+i) + B(x_0+i, x_1+i) \right], \end{aligned} \tag{1}$$

where $R(\cdot)$, $G(\cdot)$ and $B(\cdot)$ are a specific color components in RGB model.

The second function focuses on the saturation level but is analyzed on the original image I and modified I'. The modification is based on an edge detection filter which leaves white edges on a black background. If the point is on edge, the value from the original image is obtained, in another case, the value is the opposite maximum. This function can be defined as:

$$f_2(\overline{x}) = \frac{1}{9} \sum_{i=-1}^{1} \phi_1(x_0+i, x_1+i), \tag{2}$$

and a function $\phi_1(\cdot)$ is defined as follows:

$$\phi_1(\overline{x}) = \begin{cases} S_I(x,y), & \text{if} \quad \begin{matrix} R_{I'}(x_0,x_1) > 240 \\ G_{I'}(x_0,x_1) > 240 \\ B_{I'}(x_0,x_1) > 240 \end{matrix} \\ 100, & \text{otherwise} \end{cases}, \tag{3}$$

where $S_I(\cdot)$ means a saturation level for a given pixel on image I, and $R_{I'}(\cdot)$, $G_{I'}(\cdot)$, $B_{I'}(\cdot)$ means a color values on I'.

The third function search for the opposite values of saturation. The sense of it is to analyze different areas and leave the analysis for the next module. This function can be described by the following formula:

$$f_3(\overline{x}) = \frac{1}{9} \sum_{i=-1}^{1} \phi_2(x_0+i, x_1+i), \tag{4}$$

where

$$\phi_2(\overline{x}) = \begin{cases} S_I(x,y), & \text{if} \quad \begin{matrix} R_{I'}(x_0,x_1) < 15 \\ G_{I'}(x_0,x_1) < 15 \\ B_{I'}(x_0,x_1) < 15 \end{matrix} \\ 100, & \text{otherwise} \end{cases}. \tag{5}$$

In the above functions, the value 100 was chosen to indicate the maximum value, which is rarely analyzed in the minimization problem. A sample image analyzed by these functions is shown in Fig. 2.

Depending on the hardware where the proposed system is implemented, the number of threads is defined as θ. This allows defining the number of parallel instances of heuristic algorithms. Each thread has one instance of the algorithm and one function to analyze. If the number of threads is smaller than the number of functions, it is the thread that has finished the computation analyzing next function. This is repeated until all defined functions will be analyzed.

Red Fox Optimization Algorithm is inspired by the behavior of foxes herd during hunting. The basic promises is the size of the herd marked as P. Each fox in the herd is represented as a one pixel $\overline{x} = (x, y)$. At the beginning, the initial herd is generated at random position as $x \in \langle 0, w-1 \rangle$

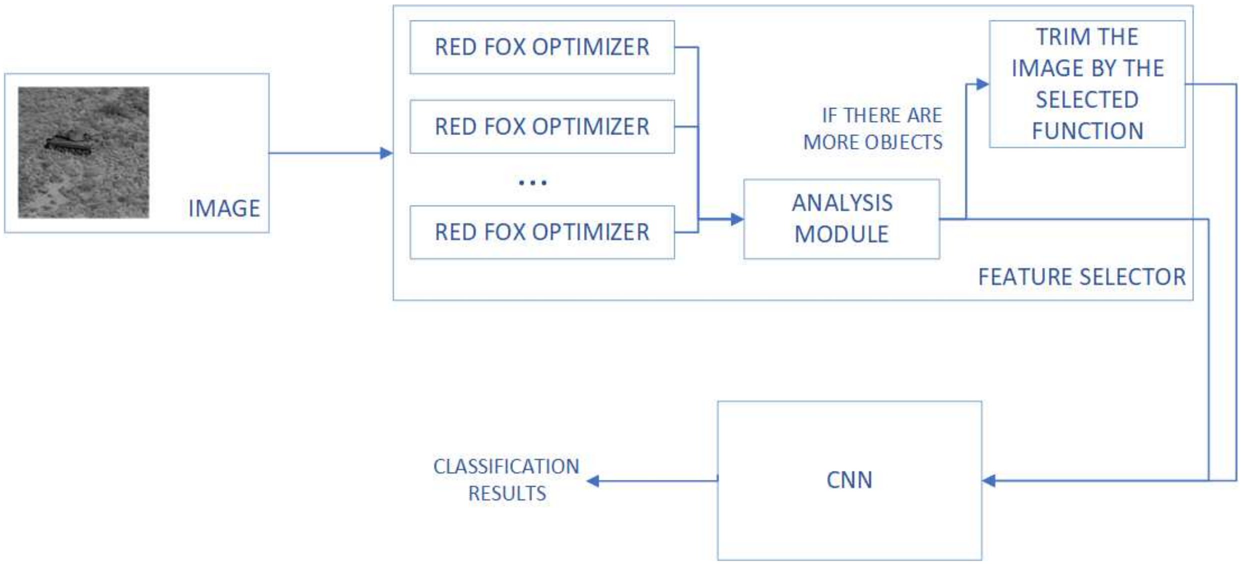


Fig. 1: A visualization of the proposed method. Incoming image is processed by Red Fox Optimizers (what is performed in a parallel way). Each of the optimizers is analyzing the image by other fitness functions and the results are sent to the analysis module which decides whether the image should be cropped. Then the image is sent to CNN for classification result.

and $y \in \langle 0, h-1\rangle$. Then, the best individual $\overline{x}^{best}$ in the whole herd is find by used fitness function. After finding the best fox in herd, the rest of them are moved by the global and local movement. The global formula for movement change the position of a given fox $\overline{x}^{act}$ over a long distance to new position $\overline{x}^{new}$ as:

$$\overline{x}^{new} = \overline{x}^{act} + \lceil \alpha \cdot \text{sign}(\overline{x}^{best} - \overline{x}^{act})\rceil, \tag{6}$$

where $\alpha \in (0, d((\overline{x}^{i})^{t}, (\overline{x}^{best})^{t})\rangle$ are understood as scaling parameters. In the next stage of algorithm, each fox is considered to make a decision about future action – move to victim, or stay and wait. This operation is made as:

$$\begin{cases} \text{Move closer} & \text{if } \mu > 0.75 \\ \text{Stay and disguise} & \text{if } \mu \leq 0.75 \end{cases}, \tag{7}$$

where $\mu \in \langle 0, 1\rangle$ is a random value. In the case when the fox moves, a new position is calculated by the following equation:

$$\begin{cases} x_0^{new} = \lceil ar \cdot \cos(\phi_1)\rceil + x_0^{act} \\ x_1^{new} = \lceil ar \cdot \sin(\phi_1)\rceil + \lceil ar \cdot \cos(\phi_2)\rceil + x_1^{act} \end{cases}, \tag{8}$$

where ϕ_1, ϕ_2 are a values generated in random way from the range $\langle 0, 2\pi\rangle$, and r is an observation angle defined by: and finally, hunt. Observation angle is calculated as:

$$r = \begin{cases} \left\lceil a\dfrac{\sin(\phi_0)}{\phi_0}\right\rceil & \text{if } \phi_0 \neq 0 \\ rand(0,1) & \text{if } \phi_0 = 0 \end{cases}, \tag{9}$$

where parameters a is scaling coefficient chosen randomly. This movement is understood as the local change position of a given fox. These movements are repeated by the T iteration. After the performance of all iterations, the best foxes are returned as the best features according to the fitness function.

B. Analysis module

The obtained results from Red Fox Optimization Algorithms are sets of points for each of the selected functions. In Fig. 2, we can see that not all points can be used in the analysis process. The main reason is the points which were placed in the area where are no important objects. For this reason, we propose that analysis is made by the evaluation of all sets. The idea is similar to the classic technique of clustering with k nearest neighbors.

Let us mark a point sets as $\Theta_1, \Theta_2, \ldots, \Theta_o$, where o is the number of used functions (and therefore the heuristic instances). The evaluation of elements in particular sets should reduce the number of points that do not indicate important areas of the image. This can be done by extending the evaluation area. The heuristic analyzes features in a grid of size 3×3. If the point has no other neighbors within a certain radius, it means that this point is not important (indicate only one pixel). Of must be noted, that the mentioned radius should be flexible to the image size. For this purpose, a radius of a neighborhood can be defined as follows:

$$r_I = \frac{1}{2}\sqrt{w+h}. \tag{10}$$

Each point $(x_1, y_1) \in \Theta_i$ is analyzed if it has a neighbor pixel $(x_2, y_2) \in \Theta_i$ using an Euclidean metric defined as

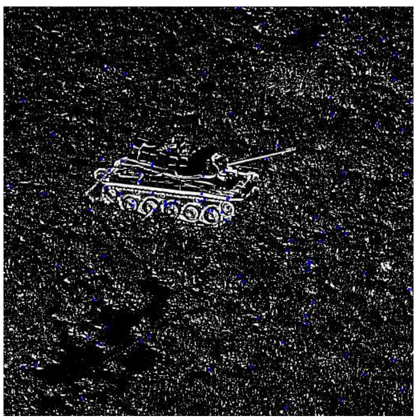
$$d((x_1,y_1),(x_2,y_2)) = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}. \tag{11}$$

If the evaluated pixel (x_1, x_2) has no neighbors in this set within r_I, it is deleted. In other cases, these points remain in the set.

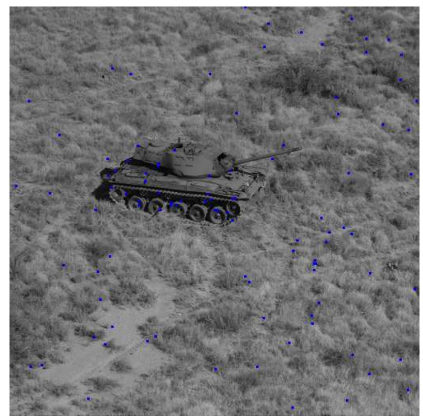
After analysis of elements in obtained sets, the remaining pixels are used for the creation of some features for each function like θ_1 (the distance between the farthest width values in the set), θ_2 (the distance between the farthest height values

(a) Original image

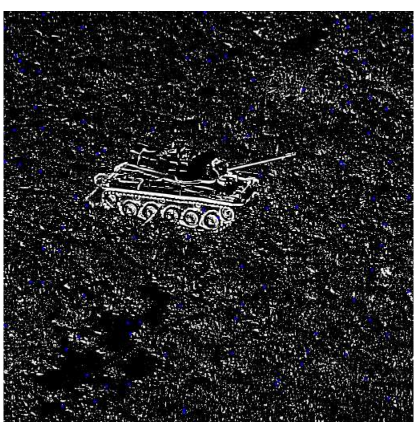
(b) Found points by the use of first function in Eq. (1).



(c) Found points by the use of first function in Eq. (2) on I'.



(d) Found points by the use of first function in Eq. (2) on original image.



(e) Found points by the use of first function in Eq. (4) on I'.

(f) Found points by the use of first function in Eq. (4) on original image.

Fig. 2: A sample image with found points by the use of a particular fitness function by the heuristic method.

in the set), θ_3 (the average value of $f_1(\cdot)$), θ_4 (the average value of $f_2(\cdot)$) and θ_5 (the average value of $f_3(\cdot)$). In the next step, these features are used for calculating the distance (using Eq. (11)) from other images in the database and finding the smallest distance. This operation can be presented as:

$$\min_j \left(\sum_{u=1}^{5} d(\theta_u, \theta_{u,j}) \right), \tag{12}$$

where $d(\theta_u, \theta_{u,j})$ is understood as a distance between features θ_u (in analyzed image) and $\theta_{u,j}$ (value of θ_u for j-th image in the database).

After finding the j-th image in the database where distance is the smallest than analyzed one, the objects on both images are compared by the evaluation of size. It is done by calculating the size factor of the object and checking if the value is greater than 20%:

$$\frac{\theta_1}{\theta_{1,j}} < 0.2 \quad \text{or} \quad \frac{\theta_2}{\theta_{2,j}} < 0.2. \tag{13}$$

If any of the above condition is satisfied, the image is cropped to the smaller size (defined by the farthest pixels in the set), then the image is resized to the input size of CNN and sent to it.

III. Experiments and discussion

In this section, we present the metrics and parameters. Then, the obtained results are presented on a benchmark image recognition task database *PASCAL Visual Object Classes Challenge* (VOC 2007) [27]. In the end, we make an analysis and discussion.

A. Evaluation metrics

We report the obtained results of classification with the average overall precision (OP) of the model and average per-class precision (CP). The prediction results were classified as true or negative and this was made in the following way: if the obtained results were greater than 0.5, it was rounded to 1, and in other cases, to 0.

B. Parameters in the evaluation of the proposed method

The proposed method uses of Red Fox Optimization Algorithm, and its parameter like the size of population and number of iteration was examined. This parameters was analyzed as $P \in \{10, 100\}$ and $\{50, 100\}$. The rest of the parameters according to the original version were generated at random for each iteration. We tested it on Intel i7-8750H CPU 2.2 GHz with 6 cores and 12 threads.

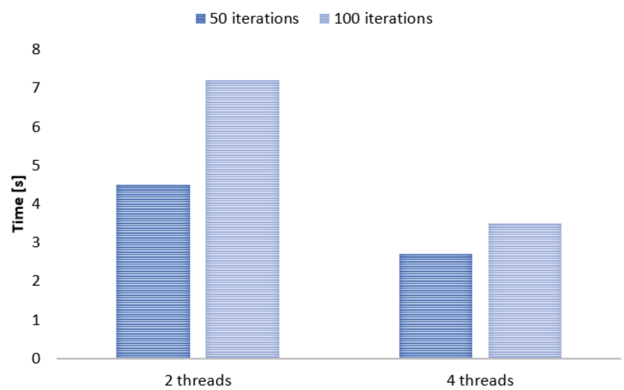
Used dataset VOC 2007 has 9963 images split into 20 classes and placed in three subsets for train, validation, and test. We used a training subset to train a CNN structure. In our research, we used two different learning transfers like VGG [28] and ResNet-101 [29] as the basic architecture of CNN.

TABLE I: Overall precision and percentage of cropped samples in subsets for VGG and Res-Net101 depending on the number of iterations and individuals in the population.

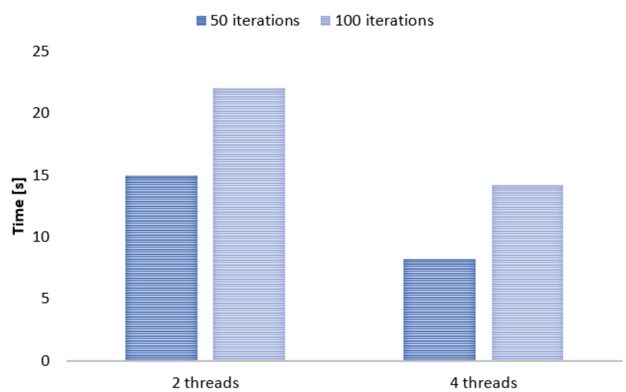
Architecture	Population	Iterations	Overall precision	Percentage of cropped samples
VGG	10	50	88,3	10
VGG	10	100	88,6	13
VGG	100	50	89,8	24
VGG	100	100	90,15	29
Res-Net101	10	50	88,7	10
Res-Net101	10	100	89,4	13
Res-Net101	100	50	89,7	24
Res-Net101	100	100	90,2	29

C. Results with discussion

At first, we retrained two used learning transfer architectures with a training set. We applied our proposal to this database and analyzed the impact of the population size and iteration number on the precision of CNN (the model was retrained by 10 iterations using the ADAM algorithm). The obtained results are presented in Tab. I. It must be noted that increasing any of these parameters increases the overall precision. The main reason for it is more chances for analyzing more possible solutions. The best result was achieved by the use of the Res-Net101 model than VGG. In the case of using the maximum values of these parameters which are 100 individuals and 100 iterations, the difference between these two models is on the level of 0.05. This is a small value that indicates that both architectures can be applied. However, the greater differences were on the level of using 10 individuals and 100 iterations. The difference value was equal to 0.8. Moreover, these values indicate that it is a better idea to use more individuals than iterations.



(a) 10 individuals in herd.



(b) 100 individuals in herd.

Fig. 3: Average time for processing an image.

During this analysis, we measured the average time for analyzing images by the heuristic approach. In Tab. 3, the average time is presented for different values of the heuristic parameter and the use of 2 and 4 threads. For this database, we used only four functions, which provided us four instances of heuristic. By the use of 2 threads, each thread has to finish analyzing the image and then change the function. The opposite situation was in the case of using 4 threads, where all functions were applied and the image was analyzed by them parallel. Increasing the number of foxes from 10 to 100 causes the time increase. In all tests, the use of 4 threads shows better results by an average of 57%. The worst result is 48%

TABLE II: Comparison between state-of-art solutions with proposed mechanism.

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table
VGG [28]	98,8	95	96,8	95,4	69,7	90,4	93,5	96	74,2	86,6	87,8
ResNet-101 [29]	99,5	97,7	97,8	96,4	65,7	91,8	96,1	97,6	74,2	80,9	85
VGG + proposed method	99	96	97,3	96,3	72,2	90,5	92	96,4	75,2	87,2	89,2
ResNet-101 + proposed method	99,5	98,1	97,8	96,4	67,2	92	96,2	97	74,8	81,2	86,6

Methods	dog	horse	motor	person	plant	sheep	sofa	train	tv	OP
VGG [28]	96	96,3	93,1	97,2	70	92,1	80,3	98,1	87	89,715
ResNet-101 [29]	98,4	96,5	95,9	98,4	70,1	88,3	80,2	98,9	89,2	89,93
VGG + proposed method	96,4	96,9	95	97,4	70,2	92,4	79	98,4	86	90,15
ResNet-101 + proposed method	98,2	96,8	96,3	98,6	70,4	89,3	79,2	98,8	89,6	90,2

shortest average time for 4 threads, which was reached for 10 individuals and 100 iterations. The highest average time gain was noted by the use of 100 foxes and 100 iterations. The results for 4 threads were better by 64% than the use of 2 threads. We notice that the used function causes many calculations because of analyzing a grid for each pixel. This can be improved by analyzing matrices and not the images. Each image is defined a set of d (here, $d = 3$) matrices for each color component.

We analyzed also the classification results for each class, and these values are shown in Tab. II. The overall precision is higher with the application heuristic technique described in this paper than without it. Moreover, except for 4 classes, in general, the proposal reached better results for the remaining 16. The worst precision was reached for car, dog, sofa, and tv. We notice, that the proposed fitness functions in some cases (especially in these four classes) found the object, and allow it to crop it. Unfortunately, the found pixels were in many cases in the middle of the object, and the cropped image has only a small area from the classified object. In consequence, CNN returns the wrong classification decision. This issue was noted also in other cases, but the number of such situations was not so large as in these four classes. The Res-Net101 shows better results than VGG because of calculating the worst cases only in 2 from 4 mentioned classes which are a sofa and a dog. VGG has lower precision in car, sofa, and tv classes. In this comparison, we used the best configuration from the previous experiments, which was the application of 100 individuals and 100 iterations.

IV. CONCLUSION

In this paper, we proposed a hybrid solution for image classification tasks by applying the heuristic method. Red Fox Optimization Algorithm was modified to analyze the image by different fitness functions (which can be modeled in each color model). The obtained results from the heuristic were examined by the analysis module inspired by the clustering idea and decided to crop the image or not. Such processed image was classified by CNN. We analyzed the impact of two main parameters in the Red Fox Optimization Algorithm which are the number of iterations and individuals on two different learning transfer architectures – VGG and Res-Net101. In general, the proposed hybrid model reached higher results in overall precision and exceeded the obtained precision in 80% cases of the classes for the used database. It is worth to notice, that combining image processing classifiers with key-points search shows good results. This approach might found the important areas and define where the object is, and then it might be cropped only to the found area. Such an action can reduce the size of the analyzed image and thus, reduce the time needed to analyze it.

In the future, we plan to analyze the impact of other fitness functions and low the time computational caused by adding more calculations by using a heuristic approach on images before classification. The next goal is to find a mechanism for the automatic size of the population according to the image size/resolution. Based on our research, we found out that the higher resolution is, then the population size also must be much bigger to analyze the area.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the contribution to this research from the Rector of the Silesian University of Technology, Gliwice, Poland under proquality grant no. 09/010/RGJ21/0056 and no. 09/010/RGJ21/0057.

REFERENCES

[1] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems*, 2020.

[2] D. Połap and M. Woźniak, "Red fox optimization algorithm," *Expert Systems with Applications*, p. 114107, 2020.

[3] V. Hayyolalam and A. A. P. Kazem, "Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103249, 2020.

[4] A. Kaveh, M. Khanzadi, and M. R. Moghaddam, "Billiards-inspired optimization algorithm; a new meta-heuristic method," in *Structures*, vol. 27. Elsevier, 2020, pp. 1722–1739.

[5] M. Abd Elaziz and S. Mirjalili, "A hyper-heuristic for improving the initial population of whale optimization algorithm," *Knowledge-Based Systems*, vol. 172, pp. 42–63, 2019.

[6] J.-S. Pan, P. Hu, and S.-C. Chu, "Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power," *Processes*, vol. 7, no. 11, p. 845, 2019.

[7] Z. Tang, G. Srivastava, and S. Liu, "Swarm intelligence and ant colony optimization in accounting model choices," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–9, 2020.

[8] A. K. Sangaiah, A. A. R. Hosseinabadi, M. B. Shareh, S. Y. Bozorgi Rad, A. Zolfagharian, and N. Chilamkurti, "Iot resource allocation and optimization based on heuristic algorithm," *Sensors*, vol. 20, no. 2, p. 539, 2020.

[9] S. T. Milan, L. Rajabion, H. Ranjbar, and N. J. Navimipour, "Nature inspired meta-heuristic algorithms for solving the load-balancing problem in cloud environments," *Computers & Operations Research*, vol. 110, pp. 159–187, 2019.

[10] Q. H. Do, T. T. Tuan, L. T. T. Ha, T. T. H. Doan, T. V. A. Nguyen *et al.*, "Development of artificial neural networks trained by heuristic algorithms for prediction of exhaust emissions and performance of a diesel engine fuelled with biodiesel blends," in *Applied Nature-Inspired Computing: Algorithms and Case Studies*. Springer, 2020, pp. 253–275.

[11] A. Banerjee, D. Ghosh, and S. Das, "Hyper-parameter tuned deep q network for area estimation of oil spills: a meta-heuristic approach," *Evolutionary Intelligence*, pp. 1–16, 2020.

[12] J. Fan, L. Wu, X. Ma, H. Zhou, and F. Zhang, "Hybrid support vector machines with heuristic algorithms for prediction of daily diffuse solar radiation in air-polluted regions," *Renewable Energy*, vol. 145, pp. 2034–2045, 2020.

[13] V. Nourani, E. Foroumandi, E. Sharghi, and D. Dabrowska, "Ecological-environmental quality estimation using remote sensing and combined artificial intelligence techniques," *Journal of Hydroinformatics*, vol. 23, no. 1, pp. 47–65, 2021.

[14] D. Dabrowska and W. Rykala, "A review of lysimeter experiments carried out on municipal landfill waste," *Toxics*, vol. 9, no. 2, p. 26, 2021.

[15] E. Protopapadakis, A. Voulodimos, A. Doulamis, N. Doulamis, and T. Stathaki, "Automatic crack detection for tunnel inspection using deep learning and heuristic image post-processing," *Applied Intelligence*, vol. 49, no. 7, pp. 2793–2806, 2019.

[16] U. Dixit, A. Mishra, A. Shukla, and R. Tiwari, "Texture classification using convolutional neural network optimized with whale optimization algorithm," *SN Applied Sciences*, vol. 1, no. 6, p. 655, 2019.

[17] T. Ozcan and A. Basturk, "Transfer learning-based convolutional neural networks with heuristic optimization for hand gesture recognition," *Neural Computing and Applications*, vol. 31, no. 12, pp. 8955–8970, 2019.

[18] B. Wang, Y. Sun, B. Xue, and M. Zhang, "A hybrid ga-pso method for evolving architecture and short connections of deep convolutional neural networks," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2019, pp. 650–663.

[19] Y. Li, J. Xiao, Y. Chen, and L. Jiao, "Evolving deep convolutional neural networks by quantum behaved particle swarm optimization with binary encoding for image classification," *Neurocomputing*, vol. 362, pp. 156–165, 2019.

[20] S. Vijh, P. Gaurav, and H. M. Pandey, "Hybrid bio-inspired algorithm and convolutional neural network for automatic lung tumor detection," *Neural Computing and Applications*, pp. 1–14, 2020.

[21] M. Wang, L. Wang, Z. Ye, and J. Yang, "Ant lion optimizer for texture classification: A moving convolutional mask," *IEEE Access*, vol. 7, pp. 61 697–61 705, 2019.

[22] J. Szymański and W. Duch, "Self organizing maps for visualization of categories," in *International Conference on Neural Information Processing*. Springer, 2012, pp. 160–167.

[23] ——, "Information retrieval with semantic memory model," *Cognitive Systems Research*, vol. 14, no. 1, pp. 84–100, 2012.

[24] A. M. Anter, Y. Wei, J. Su, Y. Yuan, B. Lei, G. Duan, W. Mai, X. Nong, B. Yu, C. Li *et al.*, "A robust swarm intelligence-based feature selection model for neuro-fuzzy recognition of mild cognitive impairment from resting-state fmri," *Information Sciences*, vol. 503, pp. 670–687, 2019.

[25] T. Hyla and N. Wawrzyniak, "Identification of vessels on inland waters using low-quality video streams," in *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021, p. 7269.

[26] M. Wlodarczyk-Sielicka and W. Blaszczak-Bak, "Processing of bathymetric data: The fusion of new reduction methods for spatial big data," *Sensors*, vol. 20, no. 21, p. 6207, 2020.

[27] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.