

# Using Pretrained AlexNet Deep Learning Neural Network for Recognition of Underwater Objects

## Uporaba pretreniranih AlexNet neuronskih mreža dubokog učenja za prepoznavanje podvodnih objekata

Piotr Szymak

Institute of Electrical Engineering  
and Automatics  
Polish Naval Academy Gdynia  
Poland  
e-mail: p.szymak@amw.gdynia.pl

Marek Gasiorowski

Institute of Electrical Engineering and  
Automatics  
Polish Naval Academy Gdynia  
Poland  
e-mail: m.gasiorowski@amw.gdynia.pl

DOI 10.17818/NM/2020/1.2

UDK 004.43\*004.896 (204.1)

Preliminary communication/ Prethodno priopćenje  
Paper accepted / Rukopis primljen: 7. 8. 2019.

### Summary

Recently, the growing number of Autonomous Underwater Vehicles (AUVs) can be seen. These vehicles are power supplied and controlled from the sources located on their boards. To operate autonomously underwater robots have to be equipped with the different sensors and software for making decision based on the signals from these sensors. The goal of the paper is to show initial research carried out for underwater objects recognition based on video images. Based on several examples included in the literature, the object recognition algorithm proposed in the paper is based on the deep neural network. In the research, the network and training algorithms accessible in the Matlab have been used. The final software will be implemented on board of the Biomimetic Autonomous Underwater Vehicle (BAUV), driven by undulating propulsion imitating oscillating motion of fins, e.g. of a fish.

### KEY WORDS

deep learning  
underwater objects recognition  
Matlab

### Sažetak

U posljednje vrijeme može se uočiti sve veći broj autonomnih podvodnih plovila (AUV). Ova plovila imaju pogon i kontroliraju ih izvori koji se nalaze na njima. Da bi radili, autonomno podvodni roboti moraju biti opremljeni raznim sensorima i softverom kako bi se donosile odluke na temelju signala primljenih s pomoću senzora. Cilj je ovoga rada prikazati početno istraživanje provedeno na prepoznavanju podvodnih objekata putem videa. Na temelju nekoliko primjera koji se mogu naći u literaturi, algoritam za prepoznavanje objekata koji se predlaže u ovome radu temelji se na dubokim neuronskim mrežama. U istraživanju koristili su se dostupna mreža i algoritmi za obuku u Matlabu. Konačno dobiveni softver primijenit će se na biomimetrijskom autonomnom podvodnom plovilu (BAUV), pogonjenom valovitim pogonom koji imitira oscilirajuće gibanje peraja, npr. ribljih peraja.

### KLJUČNE RIJEČI

duboko učenje  
prepoznavanje podvodnih objekata  
Matlab

## 1. INTRODUCTION / Uvod

In the recent years, a dynamical development of underwater robotics has been noticed. One of the latest innovative constructions in this field are biomimetic autonomous underwater vehicles (BAUVs). They imitate underwater living organisms, e.g. fishes, marine mammals, etc. They can imitate both the construction and kinematics of motion. BAUVs are driven by undulating propulsion imitating real fins, e.g. of a fish – Figure 1 [7] or of a seal – Figure 2 [15].

The BAUVs are driven by undulating propulsion consisting usually of two side fins and one tail (Figure 1) or two tail fins (Figure 2). This propulsion system is usually supported by an artificial swim bladder, i.e. a ballast tank for changing buoyancy of the BAUV [9]. Additionally, the BAUVs are equipped with different sensors and communication and navigation devices. Generally, the sensors can be divided into two groups: hydroacoustic and photosensitive sensors. The first ones use hydroacoustic waves sent and received (active) or only received

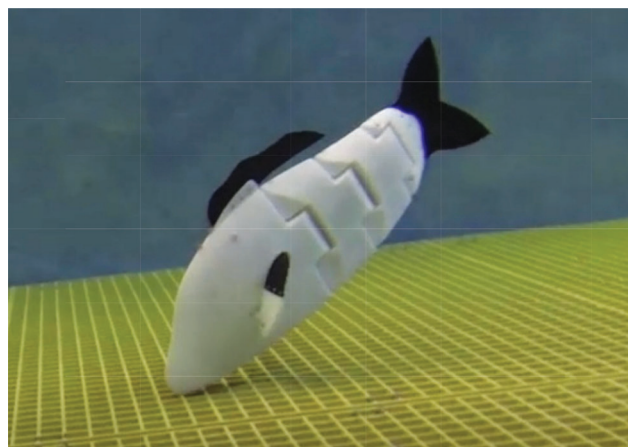


Figure 1 CyberFish ver. 5 close to the bottom in the swimming pool  
Slika 1. CyberFish ver. 5 blizu dna u bazenu

(passive). The second sensors receive light rays (passive). Both sensors can create images, i.e. sonar and video images. One of the sensors of the BAUV (Figure 2) is video camera mounted on the fixed mast. Usually, it is used for video image registration for further analysis by human expert. It can be used for different other tasks in autonomous way after digital image processing.

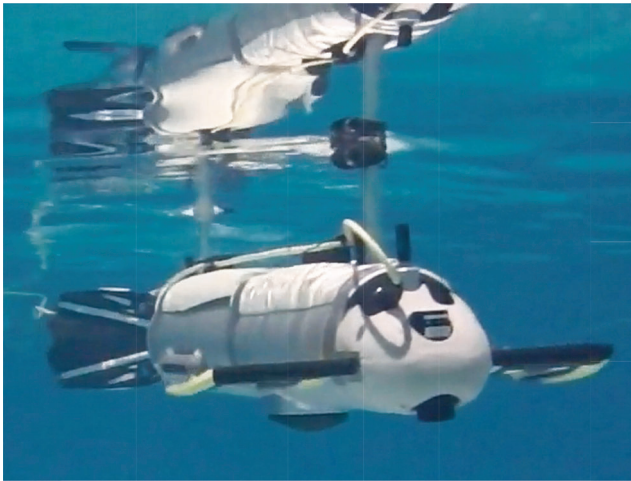


Figure 2 CyberSeal close to the surface of water in the swimming pool

*Slika 2. CyberSeal blizu površine vode u bazenu*

The whole sensors can support autonomy of the BAUV, e.g. video camera and hydrophones can be used for the passive detection of the obstacles [13], adequately at short and long distances. While the forward looking sonar and the echo-sounders can be used for active obstacle detection. Moreover, all the sensors can be used for the other tasks after digital image processing, e.g. object recognition. Regarding the obstacle detection, the general scheme of the autonomy system operation is to detect obstacle and then to avoid it. This scheme can differ depended on the type of carried out mission and the type of obstacle, e.g. if the BAUV operate in the enemy area, it should submerge on the higher depth when it recognize diver or another underwater vehicle, but it can continue its mission even without maneuver of obstacle avoidance, if it recognizes swarm of fishes. Therefore, the problem of recognition of underwater objects seems to be quite important, especially in the case of military application of BAUV or classical AUV.

After initial literature analysis [3][4][6][8][16], it was concluded that the deep learning seems to be modern and promising techniques for underwater obstacle recognition. Deep learning is a part of broader family of machine learning methods based on learning data representations [11]. The process of learning can be supervised, semi-supervised or unsupervised [1]. Deep learning architectures can be deep neural networks, deep belief networks and recurrent neural networks [12]. In the result of the learning process, the deep learning architectures should work similar to human expert in some learned cases. In general, the deep network means artificial neural network with multiply layers between input

and output layers. Deep learning is used in many application connected with data processing such as voice recognition, image recognition, drug detection, etc. Deep learning-based software more often produces more accurate results than human experts [2].

One of the software toolbox for deep learning is Neural Network Toolbox included in Matlab 2018 [10]. This paper undertakes problem of recognition of underwater objects using deep learning networks and training algorithms included in Matlab 2018. In the case of obtaining positive final result of the research, the deep neural network properly recognizing underwater objects will be used in the future research on BAUV (Figure 2). The vehicle has installed inside Nvidia Jetson TX2 and Matlab it allows you to generate the code on this hardware platform.

This paper includes results of numerical research using deep learning techniques from Matlab. In the next section, the structure and training algorithms of deep neural network is described. Then, the research problem is formulated. In the next section, the results of numerical research carried out in Matlab are included. At the end, the conclusions from the research and the schedule of future research are presented.

## 2. DEEP LEARNING IN MATLAB / *Duboko učenje u Matlabu*

The Matlab environment contains 9 deep networks pre-trained using ImageNet database, which is used in the ImageNet Large-Scale Visual Recognition Challenge [14]. The networks consist of dozens of layers. They are trained on more than a million images, therefore they can classify images into 1000 object categories, such as keyboard, mouse, cat, etc. One of the networks is AlexNet containing 25 layers (Table 1). This pre-trained network was used in the numerical research.

There are 3 following gradient methods of training deep networks accessible in Matlab:

- SGDM - the stochastic gradient descent with momentum optimizer.
- RMSProp - the root mean square propagation optimizer.
- Adam - the derived from adaptive moment estimation optimizer.

The stochastic gradient descent algorithm updates the network parameters (weights and biases) to minimize the loss function by taking small steps in the direction of the negative gradient of the loss. The momentum term added to the network parameters update helps to reduce the oscillation, which may appear along the path of steepest descent towards the optimum [11]. The stochastic gradient descent with momentum algorithm uses a single learning rate for all the parameters. This algorithm can be defined as [10]:

$$\theta_{n+1} = \theta_n - \alpha \nabla E(\theta_n) + \gamma (\theta_n - \theta_{n-1}) \quad (1)$$

where  $n$  means the following steps of iterative process of training,  $\alpha$  is the learning rate,  $\theta$  is the parameter vector, and  $E(\theta)$  is the loss function,  $\gamma$  is the momentum factor determining how much the previous step influences on the current step of iteration.

Table 1 Structure of AlexNet deep network [9]

Tablica 1. Struktura AlexNet duboke mreže

No	Layers	
	Name	Description
1	'data'	227x227x3 images with 'zerocenter' normalization
2	'conv1'	96 11x11x3 convolutions with stride and padding
3	'relu1'	Rectified Linear Unit (ReLU)
4	'norm1'	cross channel normalization with 5 channels per element
5	'pool1'	3x3 max pooling with stride and padding
6	'conv2'	256 5x5x48 convolutions with stride and padding
7	'relu2'	ReLU
8	'norm2'	cross channel normalization with 5 channels per element
9	'pool2'	3x3 max pooling with stride and padding
10	'conv3'	384 3x3x256 convolutions with stride and padding
11	'relu3'	ReLU
12	'conv4'	384 3x3x192 convolutions with stride and padding
13	'relu4'	ReLU
14	'conv5'	256 3x3x192 convolutions with stride and padding
15	'relu5'	ReLU
16	'pool5'	3x3 max pooling with stride and padding
17	'fc6'	4096 fully connected layer
18	'relu6'	ReLU
19	'drop6'	50% dropout method
20	'fc7'	4096 fully connected layer
21	'relu7'	ReLU
22	'drop7'	50% dropout
23	'fc8'	1000 fully connected layer
24	'prob'	softmax function
25	'output'	cross-entropy ex with 'tench' and 999 other classes

The root mean square propagation algorithm uses a learning rates that are different for different parameters and that can automatically adapt to the loss function being optimized. This algorithm can be defined as [10]:

$$\theta_{n+1} = \theta_n - (\alpha \nabla E(\theta_n) / \sqrt{v_n + \epsilon}) \quad (2)$$

where

$$v_n = \beta_2 v_{n-1} + (1 - \beta_2) [\nabla E(\theta_n)]^2 \quad (3)$$

where

$\beta_2$  is the decay rate of the moving average for squared gradient and  $\epsilon$  is the constant less or equal to zero.

The derived from adaptive moment estimation (Adam) uses a parameter update that is similar to RMSProp with momentum [5]. The update is calculated based on the following equation [10]:

$$\theta_{n+1} = \theta_n - (\alpha m_n / \sqrt{v_n + \epsilon}) \quad (4)$$

where

$$m_n = \beta_1 m_{n-1} + (1 - \beta_1) \nabla E(\theta_n) \quad (5)$$

and

$$v_n = \beta_2 v_{n-1} + (1 - \beta_2) [\nabla E(\theta_n)]^2 \quad (6)$$

where

$\beta_1$  is the decay rate of the moving average for gradient.

If gradients over many iterations are similar, the moving average of the gradient lets the parameter updates establish momentum in a certain direction. If the gradients contain mostly noise, then the moving average of the gradient and the parameter updates become smaller [10].

Training deep networks is extremely computationally intensive and Matlab enables us to make the calculations on GPU or CPU, if the GPU is not accessible. Moreover, the iterative calculations connected with training deep neural network can

be performed on multiple GPUs or CPU cores, in parallel on a cluster. To compare computational efficiency of CPU and GPU hardwares, the same numerical research presented in the further part of paper was performed on single CPU and single GPU.

### 3. RESEARCH PROBLEM / Istraživački problem

The research problem was formulated in the following way. Due to the specific military purposes and carried out missions, the BAUV has to recognize underwater objects based on the video images and it has to classify detected objects to the one of three classes: divers, fishes, AUVs.

To solve this problem deep learning algorithms included in Matlab will be examined, especially pretrained AlexNet deep network and three learning methods described in the previous section. The examined variants of deep network are described in the next section.

To train networks and then to verify them, 150 images were downloaded from the internet: 50 images with divers, 50 with fishes and 50 with Remotely Operated Vehicles ROVs and AUVs. The first 50 images contain one or two divers in different shots with background of the green-yellow or blue water and yellow-brown and grey bottom (Figure 3). Sometimes the divers were visualized on the background of underwater infrastructure. The next 50 images include single fishes or swarm of fishes (Figure 4). Similarly to the previous pictures there were taken in different waters (colour, visibility, etc.). The whole images of the last group present different constructions of ROVs/AUVs with various equipment (lighting, manipulator, sonar, etc.) (Figure 5).

As we can see, the collected photos do not present simple examples of divers, fishes and ROVs/AUVs. They were taken in different waters, by different photographers and in various scale and number. It seems that it will be hard to obtain optimal solution of deep network, taking additionally into account the fact that some people can also have problems with classification of the images.

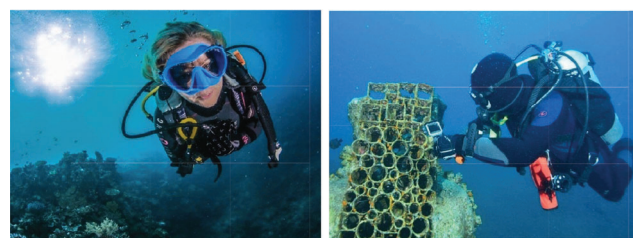


Figure 3 Examples of the images with divers (courtesy of Gdynia Dive Centre DTS Piotr Niewiński)

Slika 3. Primjeri slika s ronjocima (zahvaljujući ljubaznosti Gdynia centra za ronjenje DTS Piotr Niewiński)

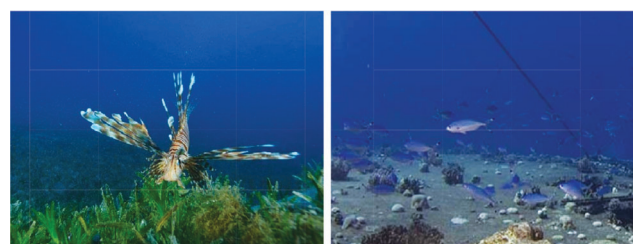


Figure 4 Examples of the images with fishes (courtesy of Gdynia Dive Centre DTS Piotr Niewiński)

Slika 4. Primjeri slika s ribama (zahvaljujući ljubaznosti Gdynia centra za ronjenje DTS Piotr Niewiński)

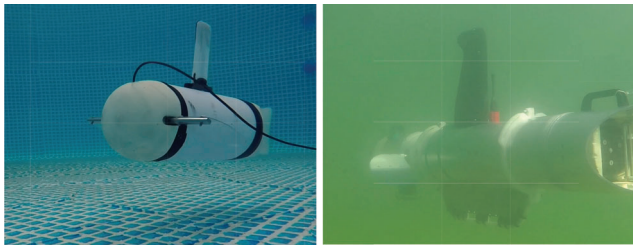


Figure 5 Examples of the images with ROVs/AUVs [15]  
Slika 5. Primjeri slika s ROV/AUV [15]

#### 4. NUMERICAL RESEARCH / Numeričko istraživanje

The research was carried out in two stages. In each stage 12 variants of deep neural network were trained and then verified. The goal of the first stage was to examine training methods. Therefore, in this stage only two simple training parameters were changed (Table II). All the others parameters were default, especially for the RMSProp and Adam training methods: squared gradient decay factor equal to 0.999 and gradient decay factor equal to 0.9 [10].

The goal of the second stage was to examine additional training parameters (Table III). These parameters influence on learning rate, which should adapt to the present state of the training process. Additional goal of this stage was to achieve the better results of training and verification process. Taking into consideration SGDM training method, the initial learn rate equal to 0.001 was accepted. This value allows us to achieve 0.0001 value of learn rate after the first learning rate drop assuming that the learning rate drop factor is equal to 0.1.

All the variants of networks were trained and verified 30 times. It allowed statistical analysis of the obtained results of the numerical research. Each time the training and verification images were randomly chosen from the 150 images. It was assumed that 120 pictures were destined for the training purposes and 30 for the verification of the trained networks.

Each training process lasts 20 epochs. This number of epoch was achieved in the result of initial research.

Table 2 12 variants of deep neural network for the first stage of the research

Tablica 2. 12 varijanti dubokih neuronskih mreža za prvu fazu istraživanja

Variant	Training method	Initial learning rate	Mini batch size
1	SGDM	0.001	10
2	SGDM	0.0001	10
3	SGDM	0.001	20
4	SGDM	0.0001	20
5	RMSProp	0.001	10
6	RMSProp	0.0001	10
7	RMSProp	0.001	20
8	RMSProp	0.0001	20
9	Adam	0.001	10
10	Adam	0.0001	10
11	Adam	0.001	20
12	Adam	0.0001	20

The first stage of research was performed using two hardware platforms:

- CPU Intel Core i7-6500U 2.5 GHz.
- GPU NVIDIA Quadro m2000.

The research enables us to compare the computational

efficiency of CPU and GPU platforms. The second stage of research was only performed using GPU microprocessor system.

To estimate training and validation of the networks the following indicators were accepted:

- $A_{avr}, A_{sd}$  – average value and standard deviation of accuracies obtained during 30 validation trials (accuracy indicate what part of the images has been recognized; '1' means that all the images were recognized properly).

Table 3 12 variants of deep neural network for the second stage of the research

Tablica 3. 12 varijanti dubokih neuronskih mreža za drugu fazu istraživanja

Variant	Training method	Learning rate drop factor	Learning rate drop period	Momentum	Mini batch size
1	SGDM	0.1	5	0.1	20
2	SGDM	0.1	5	0.2	20
3	SGDM	0.3	5	0.1	20
4	SGDM	0.3	5	0.2	20
		Initial Learn Rate	Gradient decay factor	Squared gradient decay factor	Mini batch size
5	RMSProp	0.0001	-	0.2	20
6	RMSProp	0.0001	-	0.4	20
7	RMSProp	0.0001	-	0.9	20
8	RMSProp	0.0001	-	0.99	20
9	Adam	0.0001	0.1	0.1	20
10	Adam	0.0001	0.1	0.9	20
11	Adam	0.0001	0.9	0.1	20
12	Adam	0.0001	0.9	0.9	20

In the Table IV, results of numerical research for the first 12 variants of deep neural network are presented. As it can be seen, the best estimation was received by the networks trained by SGDM method, especially the mean value and standard deviation of verification accuracy.

Table 4 Results of research for the first 12 variants of deep neural network

Tablica 4. Rezultati istraživanja prvih 12 varijanti dubokih neuronskih mreža

Variant	Training accuracy		Verification accuracy	
	$A_{vm}$	$A_{vd}$	$A_{vm}$	$A_{vd}$
1	0.973	0.083	0.803	0.066
2	1	0	0.926	0.031
3	1	0	0.895	0.029
4	0.998	0.009	0.927	0.026
5	0.507	0.144	0.457	0.083
6	0.986	0.035	0.861	0.059
7	0.493	0.120	0.453	0.085
8	0.985	0.073	0.880	0.060
9	0.523	0.206	0.431	0.085
10	0.980	0.076	0.857	0.059
11	0.435	0.148	0.419	0.091
12	0.985	0.064	0.883	0.063

Also quite good estimation was received for the 8<sup>th</sup> and 12<sup>th</sup> variants of network trained adequately with RMSProp and Adam training methods with the initial learning rate equal to 0.0001 and mini batch size equal to 20. Variant no. 8 is the best taking into

consideration verification process, i.e. 5 of 30 validation trials were finished with recognizing all the images.

During all the trials calculation time also was registered and archived. The first stage of research was performed over the time 73.8 h using CPU and over the time 3.7 h using GPU. It indicates that the 19.5 times faster execution of numerical research by GPU than CPU was obtained.

In the Table 5, results of numerical research for the second 12 variants of deep neural network are illustrated. As it can be seen, no variant of the network learned by the SGDM method achieved the accuracy comparable to the accuracies obtained for the variants trained in the first stage of research.

Considering the next training method (variants no. 5-8), the parameters accepted in the second stage give better results than received in the first stage of the research. Taking into accounts this variants no. 7 and 8, it can be concluded that the gradient decay factor in RMSprop method should be quite large equal or greater than 0.9.

Considering the last tested method (variants no. 9-12), the both coefficients: squared gradient decay factor and gradient decay factor should be larger than 0.9 because better verification accuracy was received for the greater values default in the first stage of research than the values used in the second stage of research.

The results of research included in the Table 4 and 5 show that the selection of proper training method and its parameters is very important and has large influence on the final results.

Table 5 Results of research for the second 12 variants of deep neural network

Tablica 5. Rezultati istraživanja drugih 12 varijanti dubokih neuronskih mreža

Variant	Training accuracy		Verification accuracy	
	$A_{vm}$	$A_{vd}$	$A_{vm}$	$A_{vd}$
1	0.863	0.240	0.656	0.234
2	0.762	0.269	0.558	0.323
3	0.792	0.236	0.571	0.310
4	0.673	0.251	0.479	0.287
5	0.977	0.061	0.873	0.090
6	0.997	0.013	0.890	0.067
7	1	0	0.907	0.060
8	1	0	0.630	0.080
9	0.975	0.068	0.841	0.126
10	0.982	0.091	0.896	0.078
11	0.375	0.103	0.307	0.071
12	0.995	0.020	0.836	0.069

## 5. CONCLUSION / Zaključak

In the paper, the pretrained AlexNet deep neural network and 3 gradient training methods accessible in Matlab were tested for the problem of underwater objects recognition. The AlexNet network and training methods in Matlab allows us to obtain correctly recognizing underwater objects deep neural network. All the training methods give the comparable results. Each of the method needs accurate tuning all the training parameters. Using GPU hardware gives you almost 20 times faster

Based on the obtained results from the second stage of research comparing to the results received in the first stage, it can be underlined that the optimization method, e.g. Genetic Algorithm GA should be applied for searching the best values of training parameters. Moreover, to counteract overfitting the training data

the database with larger number of underwater images is needed. Partial solution of too small database is the data augmentation, what also will be applied together with increase of the underwater images' number in the future research.

In the final step of the research, the underwater images registered in the likely area of the BAUV operation should be used for training process. Then, the verification in real environment with the deep neural network implemented on Nvidia Jetson TX2 should be performed.

## Acknowledgements / Zahvale

The results of research presented in the paper were obtained within European Defence Agency project category B called SABUVIS [9]. The project was carried out in the years 2015-2018 by the consortium consisting of the following Polish partners: Polish Naval Academy AMW – the leader, Cracow University of Technology PK, Industrial Institute of Automatics and Measurement PIAP, Forkos Company, German partners: Bundeswehr Technical Center for Ships and Naval Weapons WTD 71 in Eckernförde, Fraunhofer Institute, and Portuguese partners: R&D center CINA, Engineering Faculty from Porto University and OceanScan company.

## REFERENCES / Literatura

- [1] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. New York: Springer.
- [2] Cireşan, D., Meier, U., Masci, J., Schmidhuber, J. (2011). "Multi-column deep neural network for traffic sign classification". Neural Networks, Conference IJCNN, pp. 333-338. <https://doi.org/10.1109/ijcnn.2012.02.023>
- [3] Demos, K., Ravaut, M., Fagette, A., Hock-Siong, L. "Deep Learning applied to Underwater Mine Warfare". IEEE, OCEANS 2017 – Aberdeen, pp. 1-7. <https://doi.org/10.1109/oceanse.2017.8084910>
- [4] Druzhkov, P. N., Kustikova, V. D. (2016). "A survey of deep learning methods and software tools for image classification and object detection". Journal Pattern Recognition and Image Analysis, Vol. 26, No. 1, pp. 9-15. <https://doi.org/10.1134/s1054661816010065>
- [5] Jin, L., Liang, H. (2017). "Deep learning for underwater image recognition in small sample size situations". IEEE, OCEANS 2017 – Aberdeen. <https://doi.org/10.1109/oceanse.2017.8084645>
- [6] Kingma, D., Jimmy, B. (2014). "Adam: A method for stochastic optimization". <https://arxiv.org/abs/1412.6980>
- [7] Malec, M., Morawski, M., Szymak, P., Trzmiel, A. (2014). "Analysis of Parameters of Traveling Wave Impact on the Speed of Biomimetic Underwater Vehicle". Trans Tech Publications, Solid State Phenomena, Vol. 210, pp. 273-279. <https://doi.org/10.4028/www.scientific.net/ssp.210.273>
- [8] Mathworks (2018). Matlab documentation for Deep Learning Toolbox. <https://www.mathworks.com/help/matlab>
- [9] Moniruzzaman, M., Islam, S., Bennamoun, M., Lavery, P. (2017). "Deep Learning on Underwater Marine Object Detection: A Survey". International Conference on Advanced Concepts for Intelligent Vision Systems, November 2017. [https://doi.org/10.1007/978-3-319-70353-4\\_13](https://doi.org/10.1007/978-3-319-70353-4_13)
- [10] Morawski, M., Słota, A., Zając, J., Malec, M., Krupa, K. (2017). "Hardware and low-level control of biomimetic underwater vehicle designed to perform ISR tasks". Journal of Marine Engineering & Technology, Vol. 16, No. 4, pp. 227-237. <https://doi.org/10.1080/20464177.2017.1387089>
- [11] Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. Cambridge, Massachusetts: The MIT Press.
- [12] Pascanu, R., Mikolov, T., Bengio, Y. (2013). "On the difficulty of training recurrent neural networks". Proceedings of the 30th International Conference on Machine Learning, Vol. 28, No. 3, pp. 1310-1318.
- [13] Piskur, P., Szymak, P. (2017). "Algorithms for passive detection of moving vessels in marine environment". Journal of Marine Engineering & Technology, Vol. 16, No. 4, pp. 377-385. <https://doi.org/10.1080/20464177.2017.1398483>
- [14] Russakovsky, O., Deng, J., Su, H. et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". International Journal of Computer Vision, Vol. 115, No. 3, pp. 211-252. <https://doi.org/10.1007/s11263-015-0816-y>
- [15] Szymak, P., Praczyk, T., Naus, K., Szturomski, B., Malec, M., Morawski, M., Szymak, P. (2016). "Research on biomimetic underwater vehicles for underwater ISR". Proc. SPIE, Vol. 9831, Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VII. <https://doi.org/10.1117/12.2225587>
- [16] Zhu, P., Isaacs, J., Fu, B., Ferrari, S. (2017). "Deep Learning Feature Extraction for Target Recognition and Classification in Underwater Sonar Images". IEEE 56th Annual Conference on Decision and Control (CDC), 2017, Melbourne, Australia, pp. 2724-2731. <https://doi.org/10.1109/cdc.2017.8264055>