TITLE: Deep Learning Methods in Unmanned Underwater Vehicles

AUTHORS: Ercan ATANER,Büsra ÖZDES,Gamze ÖZTÜRK,Taha Yasin Can ÇELIK,Akif

DURDU,Hakan TERZIOGLU

PAGES: 345-350

ORIGINAL PDF URL: https://dergipark.org.tr/tr/download/article-file/1325431

<u>*Araştırma Makalesi*</u>   <u>*Research Article*</u>

Deep Learning Methods in Unmanned Underwater Vehicles

Ercan Ataner [1*], Büşra Özdeş [2], Gamze Öztürk[3], Taha Yasin Can Çelik[4], Akif Durdu [5], Hakan Terzioğlu[6]

[1]Konya Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Konya, Türkiye (ORCID: 0000-0002-4548-9968)
[2]Konya Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Konya, Türkiye (ORCID: 0000-0002-3902-3053)
[3]Konya Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Konya, Türkiye (ORCID: 0000-0003-3278-0016)
[4]Konya Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Konya, Türkiye (ORCID: 0000-0002-3955-0953)
[5]Konya Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Konya, Türkiye (ORCID: 0000-0002-5611-2322)
[6]Konya Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Konya, Türkiye (ORCID: 0000-0001-5928-8457)

Abstract

Unmanned underwater vehicles (ROV/AUV) are robotic systems that can float underwater, are autonomous and remotely controlled. Nowadays, the Navy has focused on the operational use of unmanned underwater vehicles in the defense industry and in many areas, and has increased interest in this issue. Unmanned underwater vehicles. Unmanned underwater vehicles are carried out in civilian and military applications for different and varied purposes like protection of national sources, protection of environmental sources and researchs about that, miscellaneous construction activities, police of coastal and country. Also they can use civil and military applications and they helped they have helped with much of the academic and industrial research done in recent years. To sum up they are remotely controlled vehicles with observation and exploration features. This article discusses image processing and deep learning techniques in unmanned underwater vehicles. Also it presents an in-depth review of the artificial intelligence technique and aims to contribute to our country's defense industry. The options that will enable the vehicle to succeed in autonomous missions are mentioned. The Raspberry Pi 3 microprocessor was used in autonomous missions. The Raspberry Pi Camera Module, which is compatible with the Raspberry Pi 3, is preferred. Python was used as a programming language during software process. Objects in the images taken from the camera have been identified using the OpenCV library and deep learning. The TensorFlow library which deep learning library, was used for object detection and tracking. At the beginning The Faster-RCNN-Inception-V2 model was used as the Model. However, Faster-RCNN-Inception-V2 model and Raspberry Pi 3 FPS cooperation working did not show a good performance. For this reason, the SSDLite-MobileNet-V2 model, which is fast enough for most real-time object detection applications, is preferred.

Keywords: Deep Learning, Image Processing, Object Detection and Tracking, Raspberry Pi, Unmanned Underwater Vehicles.

İnsansız Su Altı Araçlarında Derin Öğrenme Yöntemleri

Öz

İnsansız su altı araçları (ROV/AUV) su altında yüzebilen, otonom ve uzaktan kontrol edilebilen robotik sistemlerdir. Günümüzde deniz kuvvetleri, savunma sanayii ve birçok alanda insansız su altı araçlarının operasyonel kullanımına olan ilgi giderek artmıştır. İnsansız su altı araçları doğal kaynakların korunması, çevresel kaynakların korunması ve bunların incelenmesi, muhtelif inşaat faaliyetleri, kıyı ve ülke güvenliğinin sağlanması gibi farklı ve çeşitli amaçlarla sivil ve askeri uygulamalarda yürütülmekte olup, son yıllarda yapılan akademik ve endüstriyel araştırmaların büyük bir kısmına yardım eden, gözlem ve keşif özelliğine sahip, uzaktan kontrol edilebilen araçlardır. Bu çalışmada insansız su altı araçlarında görüntü işleme ve derin öğrenme yöntemlerinden bahsedilmektedir. Yapay zeka tekniğinin kapsamlı bir incelemesini sunmaktadır ve ülkemiz savunma sanayiine katkıda bulunmak amaçlanmaktadır. Otonom sürüş için Raspberry Pi 3 mikroişlemcisi kullanılmış olup kamera olarak ise Raspberry Pi 3 ile uyumlu olan Raspberry Pi Kamera Modülü tercih edilmiştir. Yazılım dili olarak ise Python kullanılmıştır. Kameradan alınan görüntülerdeki nesneler, OpenCV kütüphanesi ve derin öğrenme kullanılarak tespit edilmiştir. Nesne tespiti ve takibi için derin öğrenme kütüphanesi olan TensorFlow kütüphanesi kullanılmıştır. Model olarak öncelikle Faster-RCNN-Inception-V2 modeli kullanılmıştır. Fakat

---

* Corresponding Author: Konya Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Konya, Türkiye, ORCID: 0000-0002-4548-9968, ercanataner@gmail.com

FasterRCNN-Inception-V2 modeli ile Raspberry Pi 3 FPS bakımından iyi bir performans gösterememiştir. Bu nedenle çoğu gerçek zamanlı nesne algılama uygulaması için yeterince hızlı olan SSDLite-MobileNet-V2 modeli tercih edilmiştir.

Anahtar Kelimeler: Derin Öğrenme, Görüntü işleme, İnsansız Su Altı Araçları, Nesne Tespiti ve Takibi, Raspberry Pi.

1. Introduction

Before the 20th century, people dreamed of exploring the Seas, the deep oceans. Unaware of the presence of crude oil under the sea, they were vulnerable to hazards that could come from the water. They didn't have the technology to investigate the sinking of ships. Nowadays, with the advancement of technology and the development of autonomous underwater vehicles, surveillance and reconnaissance, safety of critical areas, pipeline welding, mining, fisheries, archaeology studies, and environmental pollution biological resources in the river can be used for the detection of (Alam, Ray, & Anavatti, 2014).

Underwater studies are more troublesome than other environments. The difference in density and the fact that water is more open to the effect of distorting the data also complicates deep learning and image processing. Therefore, unmanned underwater vehicles provide ease of operation due to their suitability of dimensions and high maneuverability. Unmanned underwater vehicles mainly consist of two main groups, Joystick controlled and autonomous. Joystick-controlled ones are called "ROV (Remote Operating Vehicle)", while autonomous ones are called" AUV (Atonomus Underwater Vehicle)" (CANLI, KURTOĞLU, CANLI, & TUNA).

ROV (Remote Operating Vehicle) is an underwater vehicle that is controlled by the operator via a control and performs underwater work, if it is necessary to be defined in its most general form. ROVs can be relatively small and simple tools in size and function, only for monitoring purposes, to take images and make some measurements via underwater cameras. İf necessary as well as many sensor, cameras, sonar, etc. that will be located on them, with this help, there can also be large systems that have a large degree of autonomous operation competence and perform quite complex functions using manipulators (CANLI et al.). Despite the increase in the number of hidden layers and nodes in artificial neural networks, artificial intelligence methods were no longer used in the early 2000s due to insufficient hardware developments. However, thanks to the GPU and other hardware developments, artificial neural networks, which consist of many hidden layers, have started to be used again, as their computational costs have decreased (Schmidhuber, 2015).

The phrase "Deep Learning" was first introduced in the context of artificial neural networks in 2000 by Igor Aizenberg et al (Aizenberg, Aizenberg, & Vandewalle, 2000). Artificial neural networks are complex systems formed by connecting artificial neurons with different topology and network models, which are similar to neurons, which are the basic unit of the human brain. An artificial neural network is a parallel connected hierarchical organization of many artificial neurons that interact with each other. The mathematical infrastructure of deep learning, which was founded in the 1960s - contrary to what is known - is not new. For deep artificial neural networks, it can be said that classic artificial neural networks are a special version of multi-layer and multi-neuron. The most important feature of deep learning networks, designed in different models depending on the field in which they are used, is that they do not need to do 'attribute engineering' to extract attributes appropriate to the problem (KIZRAK & BOLAT, 2018). Convolutional neural networks, a subspecies of multilayer forward-fed Artificial Neural Networks (Ysa), were introduced by Yann LeCun in 1998. (LeCun, Bottou, Bengio, & Haffner, 1998). It plays an active role in object detection and classification. Convolutional neural networks (CNN) are a type of Multi Layer Perceptron (MLP). Cells in the visual center are divided into subregions to cover the entire visual, simple cells are thought to have edge-like features, while complex cells are thought to concentrate on the entire visual with wider receivers. The cnn algorithm, an advanced neural network, was also inspired by the visual center of animals (Şeker, Diri, & Balık, 2017). The CNN consists of one or more fully connected layers, such as one or more convolutional layers, subsampling layers followed by a standard multilayer neural network (Zhang, Yang, Zhang, & Zhu, 2016).

Due to hardware developments, there have been significant developments in the fields of computer vision, machine learning and mobile robotics research in recent years. Developments in the field of deep learning, in particular, contain promising results (Wason, 2018). In recent years, deep learning methods have emerged as powerful machine learning methods for object detection and recognition (Deng & Yu, 2014). Object detection and object recognition, which are important elements of digital image processing applications, are topics that have been studied for many years. A large number of different algorithms have been developed for object detection and object recognition. Viola Jones was the first algorithm to effectively perform fast detection of objects (Viola & Jones, 2001). In recent years, thanks to advances in graphics processing units and deep learning, methods that can detect and identify objects with greater accuracy have been developed. In the literature, object tracking is generally discussed in four different stages. These stages are preliminary operations, object detection, object classification and object tracking (Özbaysar & Borandağ, 2018). Among these stages, object detection is of great importance and greatly affects the success of the stages after object detection. Popular libraries used for object detection and recognition when required literature scanning is performed include Single Shot Multibox Detactor (SSD), Region Based Convolutional Networks(R-CNN), Fast R-CNN, Faster R-CNN, and Mask R-CNN (Özbaysar & Borandağ, 2018).

In a study conducted in 2012, the performance of artificial pattern recognition algorithms, which were designed by Google's research team and consist of 16000 processors and more than one billion connections, reached the human level (Lohr, 2012). In 2014, Facebook used deep-learning technology called DeepFace, which performs facial recognition tasks by adding 120 million parameters to automatically tag its users in photos (Taigman, Yang, Ranzato, & Wolf, 2014). Deep learning uses algorithms known as artificial neural networks inspired by the information processing methods of biological nervous systems. Thus, it allows computers to define what each data represents and learn models (Lee & Son, 2017).

In this study, Raspberry Pi 3 was used for autonomous driving in unmanned underwater vehicles. Python was used as the software language. The objects in the images taken from the camera were identified using the OpenCV library and deep learning. Deep learning, based on representing learning data, allows you to achieve highly successful results in the field of image processing and easily solve complex image processing problems.

The TensorFlow library was used for object detection and tracking. TensorFlow uses data flow graphs to create models and allows programmers to create multi-layered and large-scale artificial neural networks (Tokui, Oono, Hido, & Clayton, 2015). This framework, which was developed using Python, supports many languages such as Javascript, R, Swift as well as Python today. It also offers many object detection models (classifiers pre-trained with specific neural network architectures) in the TensorFlow detection model zoo collection.

2. Material and Method

2.1. The Materials Used

The Raspberry Pi 3 microprocessor was used for autonomous driving in the unmanned underwater vehicle. Raspberry Pi Camera Module, which is compatible with Raspberry Pi 3, was preferred as the camera. Python is used as the software language. Figure 1 shows the raspberry Pi 3 module, Figure 2 also shows the Raspberry Pi Camera Module.

Figure 1. Raspberry Pi 3
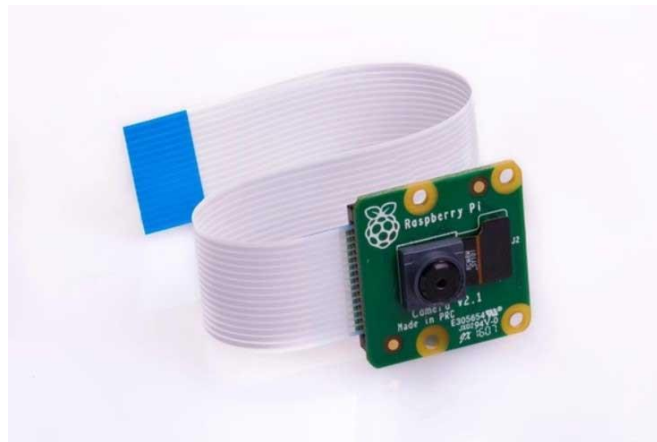
Figure 2. Raspberry Pi Camera Module

2.2. The Methods Used

The objects in the images taken from the camera were identified using the OpenCV library and deep learning. Deep learning, based on representing learning data, allows you to achieve highly successful results in the field of image processing and easily solve complex image processing problems. Although the training times of deep learning methods developed in recent years are long, the success rates achieved at the test stage have increased confidence in deep learning methods (Daş, Polat, & Tuna).

TensorFlow, Google's open source library, was used for object detection and tracking. TensorFlow is an open source deep learning library used for numerical computation using data flow graphs. The nodes in the graph represent mathematical operations, while the edges of the graph represent multidimensional data sequences (tensors) transmitted between them. Thanks to its flexible structure, it

allows calculations to be distributed to one or more CPUs on a desktop, server, or mobile device with a single API. This framework, which was developed using Python, supports many languages such as Python Javascript, R, Swift. It also offers many object detection models (classifiers pre-trained with specific neural network architectures) in the TensorFlow detection model zoo collection. SSD, R-CNN, Fast R-CNN, Faster R-CNN and Mask R-CNN models are widely used in object detection and tracking. Models such as the SSD-MobileNet model have an architecture that allows faster detection but with less accuracy. Models such as the Faster-RCNN model provide slower detection but greater accuracy (Galvez, Bandala, Dadios, Vicerra, & Maningo, 2018).

The R-CNN model is the most basic model that uses the region proposal approach. Fast R-CNN is a model developed to eliminate the slowness of the R-CNN model. This model passes the corresponding image through a conventional neural network only once. Instead of passing each region through a conventional neural network separately. A weakness of the Fast R-CNN model is the fact that it spends most of its time making region proposals during the testing phase. The model developed for faster implementation of regional proposals is Faster R-CNN. Faster R-CNN, which fully solves the slowness problem seen in R-CNN, can gain speed by making these suggestions within the network instead of receiving zone suggestions with selective search. In the Mask R-CNN model, a rectangle is drawn where the object is located, and all the pixels that this object occupies in the image are detected. The SSD model works faster than the Faster R-CNN and performs object detection at once. In R-CNN, the region proposal is made in two different stages, first determining the regions that are expected to be objects in the picture, and then classifying them with a fully connected layer. In the SSD model, these two stages occur at once. The models were compared according to different parameters and the results are given in Table 1.

Table 1. Comparison of models

	R-CNN	**Fast R-CNN**	**Faster R-CNN**
Test time per image	50 seconds	2 seconds	0.2 seconds
Speedup	1x	25x	250x
mAP (VOC 2007)	66.0	66.9	66.9

In order to detect an object or objects, we must first have data belonging to those classes. It is important that the data in our training kit matches the data in our test kit. Ensuring the same environmental conditions will allow the model to produce more successful results. In the similarity of ambient conditions in the trained and tested data, the model detects that object or objects with higher scores. For this reason, data collection was carried out underwater. The obtained 583 data is divided into 20% test folder and 80% train folder. The fact that the test and train data are very similar to each other causes the model to have an overfit (memorization) condition. In order to prevent this situation, attention has been paid to the diversity of data while collecting data. After data is collected, it is necessary to specify what the objects are. For this purpose, a labeling program called LabelImg was used. LabelImg produces files in xml format. The file contains the class name of the tagged object and the coordinates of the location where the object is located. Figure 3 shows the tagging of data with labellmg.

Figure 3. Tagging of data with labellmg

Xml files must be converted to a format that TensorFlow understands. For this purpose, first xml was converted to CSV, and then csv was converted to tfrecord, the format of TensorFlow. Then Labelmap is created. Labelmap defines the mapping of class names to class ID numbers, telling the tutorial what each object is. Then the object detection training line is configured. Finally, the training of the model was carried out. Faster-RCNN-Inception-V2 model was used as a model. However, with the Faster-RCNN-Inception-V2 model, Raspberry Pi 3 could not perform well in terms of FPS. For this reason, the SSDLite-MobileNet-V2 model is preferred, which is fast enough for most real-time object detection applications. The Python script in Object_detection_picamera. PY in this model detects real-time objects from a Picamera or USB webcam. When the script is run, a window opens showing the objects detected in the image. After the training, the model was tested and successful results were obtained.

3. Results and Discussion

Many different approaches and applications exist on object recognition or face recognition through image or image. The methods used in these approaches are different, and their success rates are low compared to the latest approaches in the literature (Baykara & Daş, 2013). However, the rate of success with deep learning was higher than other approaches. Object detection and tracking was performed using Faster-RCNN-Inception-V2 and SSDLite-MobileNet-V2. The 583 data obtained from the underwater object were divided into 20% test and 80% train folder and the original data set was created. In order to prevent the occurrence of a memorization situation, attention has been paid to the diversity of the data when collecting data. After collecting data, a labeling program called LabelImg was used, since objects must be specified what they are.

Xml was converted to CSV, and then csv was converted to tfrecord, the format of TensorFlow. Then Labelmap is created. Then the object detection training line is configured. Finally, the training of the model was carried out. Faster-RCNN-Inception-V2 model was used as a model. However, with the Faster-RCNN-Inception-V2 model, Raspberry Pi 3 could not perform well in terms of FPS. For this reason, the SSDLite-MobileNet-V2 model is preferred, which is fast enough for most real-time object detection applications. After the training, the model was tested and successful results were obtained. Figure 4 shows the detection and tracking of the circle under water with the trained model.

Figure 4. Detection and tracking of the circle under water with the trained model

The most important feature of the CNN model is that it is faster, more efficient in image processing, and most importantly, it automatically extracts attributes and reflects them to the result (Cömert, Kocamaz, & Subha, 2018).

As can be seen from Tests with the trained model, the Faster R-CNN model should be used to have a high success rate in object detection and tracking. however the Faster R-CNN needs more powerful hardware compared to other models, and it is also a slower model than the SSD model. In addition, the SSD model can detect objects quickly even in devices with low hardware. The success rate of the SSD model is lower than that of Faster R-CNN. For this reason, if the speed factor is important when performing object detection and tracking, the SSD model should be preferred.

4. Conclusions and Recommendations

The Raspberry Pi 3 was chosen as a microprocessor, considering the possibilities we have in order to detect and track objects in unmanned underwater vehicles. Python was used as the programming language. Deep learning, which enables the expansion of artificial intelligence and machine learning applications, provides highly successful results in the field of image processing, unlike algorithms for tasks, and allows complex image processing problems to be easily solved. In this study, TensorFlow library was used for deep learning and object detection and tracking. As a Model, the Faster-RCNN-Inception-V2 model was used, but the faster-RCNN-Inception-V2 model and the Raspberry Pi 3 did not perform well in terms of FPS. For this reason, the SSDLite-MobileNet-V2 model is preferred, which is fast enough for most real-time object detection applications. With these models, real-time object detection and tracking were performed and the strengths and weaknesses of the models were revealed.

5. Acknowledge

References

Aizenberg, I. N., Aizenberg, N. N., & Vandewalle, J. (2000). Multiple-Valued Threshold Logic and Multi-Valued Neurons. In *Multi-Valued and Universal Binary Neurons* (pp. 25-80): Springer.

Alam, K., Ray, T., & Anavatti, S. G. (2014). Design and construction of an autonomous underwater vehicle. *Neurocomputing, 142*, 16-29.

Baykara, M., & Daş, R. (2013). *Real time face recognition and tracking system.* Paper presented at the 2013 International Conference on Electronics, Computer and Computation (ICECCO).

CANLI, G. A., KURTOĞLU, İ., CANLI, M. O., & TUNA, Ö. S. DÜNYADA VE ÜLKEMİZDE İNSANSIZ SUALTI ARAÇLARI İSAA-AUV & ROV TASARIM VE UYGULAMALARI. *GİDB Dergi*(04), 43-75.

Cömert, Z., Kocamaz, A. F., & Subha, V. (2018). Prognostic model based on image-based time-frequency features and genetic algorithm for fetal hypoxia assessment. *Computers in Biology and Medicine, 99*, 85-97.

Daş, R., Polat, B., & Tuna, G. Derin Öğrenme ile Resim ve Videolarda Nesnelerin Tanınması ve Takibi. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 31*(2), 571-581.

Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends in signal processing, 7*(3–4), 197-387.

Galvez, R. L., Bandala, A. A., Dadios, E. P., Vicerra, R. R. P., & Maningo, J. M. Z. (2018). *Object detection using convolutional neural networks.* Paper presented at the TENCON 2018-2018 IEEE Region 10 Conference.

KIZRAK, M. A., & BOLAT, B. (2018). Derin öğrenme ile kalabalık analizi üzerine detaylı bir araştırma. *Bilişim Teknolojileri Dergisi, 11*(3), 263-286.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.

Lee, K., & Son, M. (2017). *Deepspotcloud: leveraging cross-region gpu spot instances for deep learning.* Paper presented at the 2017 IEEE 10th International Conference on Cloud Computing (CLOUD).

Lohr, S. (2012). The age of big data. *New York Times, 11*(2012).

Özbaysar, E., & Borandağ, E. (2018). *Vehicle plate tracking system.* Paper presented at the 2018 26th Signal Processing and Communications Applications Conference (SIU).

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks, 61*, 85-117.

Şeker, A., Diri, B., & Balık, H. H. (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. *Gazi Mühendislik Bilimleri Dergisi, 3*(3), 47-64.

Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). *Deepface: Closing the gap to human-level performance in face verification.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Tokui, S., Oono, K., Hido, S., & Clayton, J. (2015). *Chainer: a next-generation open source framework for deep learning.* Paper presented at the Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS).

Viola, P., & Jones, M. (2001). *Rapid object detection using a boosted cascade of simple features.* Paper presented at the Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001.

Wason, R. (2018). Deep learning: Evolution and expansion. *Cognitive Systems Research, 52*, 701-708.

Zhang, L., Yang, F., Zhang, Y. D., & Zhu, Y. J. (2016). *Road crack detection using deep convolutional neural network.* Paper presented at the 2016 IEEE international conference on image processing (ICIP).