Models and means of object recognition using artificial neural networks

Vasyl Teslyuk[1], Bohdan Borkivskyi[1], *Hamza Ali Alshawabkeh*[2]

[1] *Lviv Polytechnic National University, Lviv, Ukraine*
[2] *Al-Baha University, Kingdom of Saudi Arabia*

Abstract

The purpose of this work is to present a generalized approach for object detection using neural networks in various environments.

The novelty of the research is an offered combination of the machine learning method – neural network for object detection and the classical method – custom programmed algorithm for selecting objects of our interest. The effectiveness of the approach is achieved by combining the two methods and using their strengths – ability of neural networks to learn from images dataset and work with new images and ability to deliver unique business value using custom programmed algorithms. This paper describes solving this problem, finding the necessary methods and algorithms.

Keywords

Neural network, convolutional network, object detection.

1. Introduction

Today the task of object recognition can not be considered a perfectly accomplished task. Different approaches try to optimize specific aspects, such as the quality or speed of recognition, but in the process suffer from other properties, such as the ability of algorithms to generalize. Examples of such products are the system of recording intruders, which finds cars very well, but can not perform any other functions; or a system that recognizes hundreds of different objects but is slow. Therefore, finding a software solution that simultaneously tries to optimize all aspects is an urgent task.

Object recognition systems are very useful in various fields, such as: security systems, violation detection systems, social robotic systems. In order to improve the quality of object recognition, systems are designed to perform a specific task that involves the use of a small amount of data. In this case, to solve another task, it is necessary to design a separate system. That is, this approach is not universal and cannot be used in the case of a large amount of input data.

An example of a task that requires generalized action is a robotic system to help visually impaired people find things. If such a system can find only a few objects, its practical value will be low. But if such a system can find many commonly used things, and can work both indoors and outdoors, it will be much better able to help people.

[1]MoMLeT+DS-2022: 4th International Workshop on Modern Machine Learning Technologies and Data Science, November 25-26, 2022, Leiden-Lviv, The Netherlands-Ukraine

EMAIL: vasyl.m.teslyuk@lpnu.ua (V. Teslyuk); bohdan.p.borkivskyi@lpnu.ua (B. Borkivskyi); Halshwabkah@bu.edu.sa (Hamza A. Alshawabkeh)

ORCID: 0000-0002-5974-9310 (V. Teslyuk), 0000-0003-3301-476X (B. Borkivskyi), 0000-0003-3859-8055 (Hamza Ali Alshawabkeh)

But there is problem in using systems, that know how to work with a lot of different objects. Under some circumstances we need to account only for specific set of objects. For example, when we process footage from road camera – it can be used in system to keep track of people on crossroads, and we should account for pedestrians and cyclists, or it can be used to look for vehicles and we need to search cars and buses only. That is why developing an algorithm that can process the same information in different ways is a relevant task.

2. Model selection

When developing a computer vision system, great attention should be paid to the choice of the model that will be the basis of the algorithm, because the quality and speed of the system depends on the characteristics of the selected model.

Developers in the field of artificial intelligence and computer vision are constantly improving existing and trying to invent new approaches [1], and there are several that have already become widely used around the world, because they have proven their effectiveness in applied tasks. However, each of the algorithms has both strengths and weaknesses that need to be investigated to determine which of the algorithms is best suited to solve our problem.

The strength of the Faster R-CNN model [2] is the accuracy of object recognition, since the architectural feature of this model is the presence of two subnets [3, 4] at once (Figure 1). One performs the usual task of using a convolutional network to extract a feature map from the input data. The role of the second subnet is the region proposal network (Region Proposal Network), which, instead of pre-defined regions, searches for suitable regions on its own, which, in combination with a large training sample, gives a high result on test images.

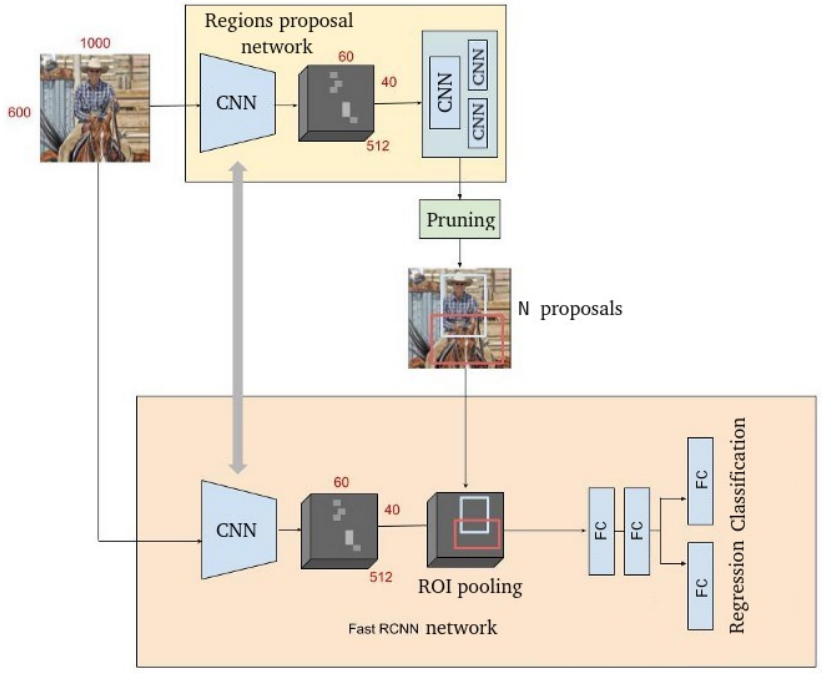

Figure 1: Faster R-CNN model structure

But these architectural features are also weakness of the network. Since two sub-models are involved in the operation of the model, this reduces the speed of the entire model and, accordingly, the processing time of each sample.

YOLO is another network that is often used to solve object search problems. The strength of this model [5] is reflected in the name - "you only look once", that is, it does not have the weakness of

Faster R-CNN. The model is built mainly using convolutional layers, with the exception of a few FC layers (Figure 2).

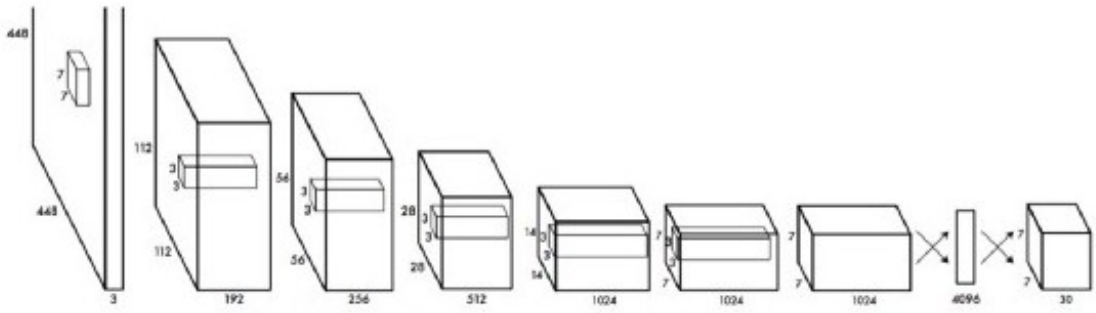

Figure 2: YOLO model architecture

Since the image passes through the network layers only once, it reduces the running time of the model and increases performance. But this model is not without its drawbacks - due to the peculiarity of the model's operation and the organization of its layers, in particular the last layer with a size of 7x7 and the possibility of building only 2 regions for each location - the total number of proposed regions is only 98. Due to these spatial limitations of the model, it is difficult to find small objects , or objects that appear in groups.

SSD [6] has a somewhat similar methodology to YOLO, which is also reflected in its name - "single frame detector". The network consists of convolutional layers (Figure 3), divided into two parts. The first part, which is based on the VGG-16 convolutional neural network [7], is a typical classification model, and is used to obtain a feature map, after which, using additional convolutional layers of different sizes, additional feature maps of a smaller size are obtained.

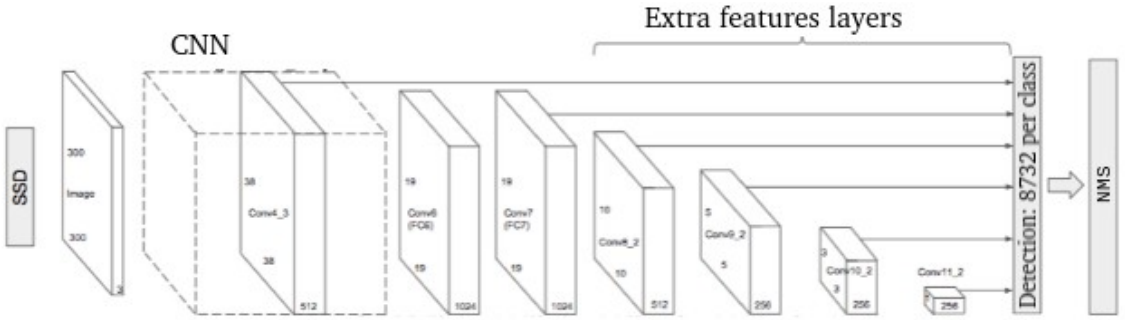

Figure 3: SSD model architecture

Due to the peculiarity of the organization of layers in this network, at the output we get 8732 regions (against 98 in YOLO), which significantly increases the number of potential candidates for recognition.

For a better understanding of the effectiveness of each model, let's get acquainted with the results [8] of testing the models (Table 1) on the same data.

Table 1: Models comparison

Model	Speed (fps)	Accuracy (mAP)
SSD	59	74.3
Faster R-CNN	7	73.2
YOLO	45	63.4

Taking into account the reviewed features of the models, their advantages and disadvantages, which are confirmed by the results of comparative testing, the best variant of the model is SSD.

3. Dataset selection

When implementing a system using artificial intelligence methods of the "learning with a teacher" type, it is important to collect a high-quality array of data, which will later be used to train the model. The main criterion when choosing a data set is its diversity, both in terms of object type and the number of unique instances. Since there will be no practical use in systems for recognizing several objects, the number of classes should start from several tens of units.

The VOC dataset was introduced in 2005 and was used in the PASCAL VOC Challenge from 2005 to 2012 [9]. During this time, the variability of the data increased from 4 to 20 classes, which are divided into 4 superclasses. In total, the data contains 10,000 images with 24,000 objects on them.

A significant drawback of this data set when creating a generalized object recognition system is the presence of only six classes for marking household items.

CIFAR-100, as well as its simplified version CIFAR-10, are datasets presented by researchers at the Department of Computer Science at the University of Toronto [10]. This dataset has 100 classes divided into 20 larger superclasses. Each class contains 500 images for training and 100 images for testing the model, giving a total of 60,000 images.

ImageNet is a set of images that is ordered according to the WordNet hierarchy, in which thousands of images correspond to each link of the system [11]. Two research needs in the field of computer vision inspired the creation of this dataset. The first is the growing demand for highly accurate metrics for evaluating object classification systems, and the second is the critical need for large volumes of data to create more generalizable machine learning methods.

The most widely used subset of this set is ILSVRC, which was created during 2012-2017 and contains 1.2 million images, divided into a thousand classes, with a size of 166Gb. Such a volume of data is very valuable for powerful laboratories, and at the same time is not appropriate for creating compact computer vision systems [12, 13], since training models outside of laboratory conditions becomes a very long process.

COCO is a large-scale data set for solving the problems of object recognition and segmentation, image caption generation, which was created with the support of such IT companies as Microsoft and Facebook [14].

The updated version of this set, released in 2017, contains 120 thousand images, which are divided into 80 classes. Unlike the previous sets, this dataset maintains a balance between different types of classes, and around 30 classes are available to label household objects, which allows for a well-generalized recognition system.

4. Results processing

A typical object detection pipeline has one component for creating proposals for classification. Proposals are nothing more than candidate regions for the object we are interested in. Most approaches use a sliding window over the feature map and assign confidence scores depending on the features computed in that window. Nearby windows have somewhat similar scores and are considered candidate regions. This leads to hundreds of offers. Since the proposal generation method must have high completeness (recall), we keep loose constraints at this stage. However, processing these multiple propositions through a classification network is cumbersome. This leads to a technique that filters offers based on some criteria called Non-maximum Suppression [15].

The main steps of the algorithm (Figure 4):

Step 1. As input, the algorithm accepts an array B with a list of proposals, corresponding confidence estimates S, and a certain threshold value N.

Step 2. We choose the proposal with the highest confidence value and transfer it from the input array to the output array D.

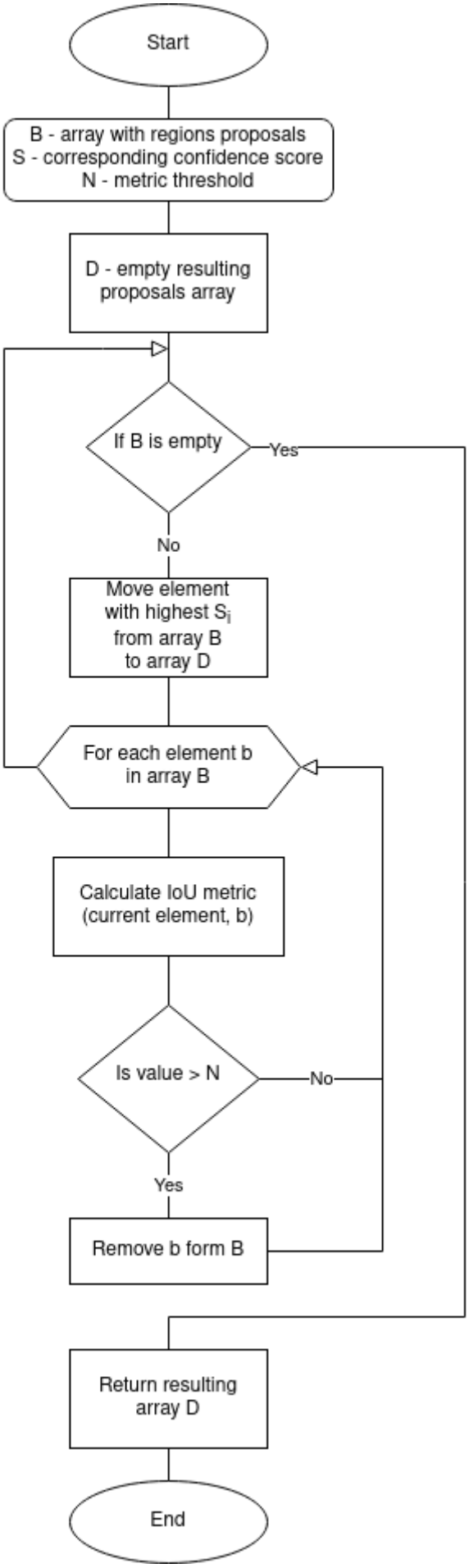

Figure 4: NMS algorithm scheme

Step 3. We compare this entry with the rest of the entries of the input array by calculating the value of the IoU (Intersection over Union) metric. If the received value is greater than the specified threshold value (elements are very similar) - we delete this offer.

Step 4. Repeat steps 2-3 until the input array becomes empty.

Step 5. We return the original array D with unique offers of regions.

To ensure the generalized operation of the system, a model capable of recognizing different types of objects is used. But in some situations, it is not advisable to search for all possible objects: a car that can be seen far away in the window of the room, or household appliances when the automated system moves around the room.

To ensure the search of only relevant objects, a filtering algorithm was developed according to the mode (Figure 5), which compares information about each separately taken classified object, and filters it according to information about the classes of the specified mode.

At the core of this filtering algorithm is creation of proper configuration, based on what type of detection is performed, e.g. for keeping track of pedestrians relevant classes would be person, cyclist, etc, for keeping track of transport relevant classes would be car, bus, van, scooter, etc. The input of the algorithm is list of previously found objects, and based on operating mode of the system, only corresponding objects from configuration are considered.

Figure 5: Object detection result with filtering (“road” mode applied)

As can be seen from Figure 6, although image contains variety of different objects (cars, pedestrians) and model can identify all of them, based on “road” mode setup (blue color) only vehicles are displayed as recognized objects. In this example “road” mode stands for vehicles.

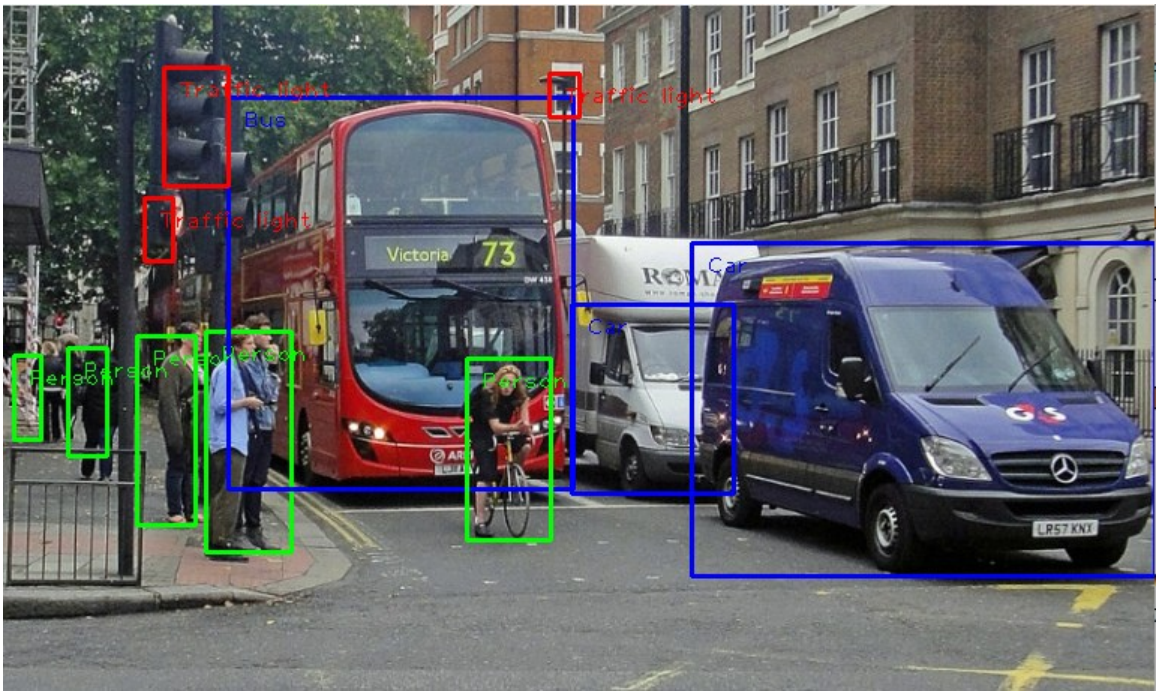

Figure 6: Object detection result with no filtering

5. Experiments

Testing took place on computers of various capacities. The weakest computer on which the operating system was tested is the Raspberry Pi 4 Model B [16], according to the minimum hardware requirements formed based on the characteristics of this device (Table 2). Although a device with 8Gb of RAM was used during system testing, the program files themselves and additional configuration files are small in size, so a model with 2Gb of RAM will be enough for work.

Table 2: Minimum hardware requirements

Parameter	Value
CPU	Quad core Cortex-A72 64-bit 1.5GHz
RAM	2GB LPDDR4-3200 SDRAM
Storage size	8GB
Camera port	Availability of one USB/Micro USB/CSI-2 port

A short video of the operation of the robotic system for finding household items was recorded to check the functionality of the system. During the operation of the system, various actions were performed on the user interface - switching the available operating modes, turning on and off the debugging mode.

During the testing, no problems were found in the operation of the system, no errors in the recognition of objects or the application of the selected modes.

After checking the correct operation of the system, experiments were carried out on several devices of different power in order to find out the speed of the system in conditions of different configuration of devices.

The first experiment was conducted on a Dell Latitude E6330 computer, the characteristics of which are presented in Table 3. Thanks to the combination of these characteristics, this model copes well with the role of an office computer and can be used for solving problems using finding objects in offices.

Table 3: Dell computer specifications

Parameter	Value
Name	Dell Latitude E6330
CPU	Intel(R) Core(TM) i7-3520M CPU @ 2.90GHz
System bit rate	64 bit
RAM	16Gb DDR3-1600
Storage size	128GB

The second experiment was conducted using a portable Raspberry Pi 4 Model B computer, the characteristics of which are presented in Table 4:

Table 4: Raspberry Pi computer specifications

Parameter	Value
Name	Raspberry Pi 4 Model B
CPU	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) SoC @ 1.5GHz
System bit rate	64 bit
RAM	8GB LPDDR4-3200 SDRAM
Storage size	8GB

The third experiment was conducted using a more powerful computer to reveal the potential capabilities of the system. The experiment used a modern Lenovo ThinkPad E14 Gen 2 laptop, the characteristics of which are presented in Table 5:

Table 5: Lenovo computer specifications

Parameter	Value
Name	Lenovo ThinkPad E14 Gen 2
CPU	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz
System bit rate	64 bit
RAM	32Gb DDR4-3200
Storage size	512GB

The experiments consisted in determining the speed of the system on different devices by calculating the time spent on processing one frame. Having obtained the average value of the time spent, we obtained the average number of processed frames per second. After conducting several experiments on both devices, the performance results presented in Table 6 were obtained:

Table 6: Experiments results

Device	Processing time (s)	Speed (fps)
Dell	0.09	11.1
Raspberry Pi	0.34	2.94
Lenovo	0.05	20

From the obtained results, it follows that none of the devices can be used in real-time systems with a minimum average fps of 30 frames per second.

However, a modern Lenovo computer with an indicator of 20fps can be used to solve most tasks, not only recognizing objects in a confined space, but also recognizing objects that move at high speed, in particular, streams of cars.

Although the Dell computer loses almost 2 times in terms of speed, the result of 10fps is also good, and the system on such a device can also be used to solve many tasks that involve tracking not such fast objects.

According to the results of the experiments, the Raspberry Pi computer showed the worst results, which corresponds to its technical characteristics. The image processing speed of 3fps does not allow using this device to recognize objects that move even at an average speed. But the device can be used to implement the object search technology for a robotic system, since mostly such systems move at a low speed, which is sufficient for high-quality image processing with a long processing time.

6. Conclusions

As a result of the work, the existing software solutions for solving the problem of finding the object in the image were investigated, available approaches to the design and construction of neural networks for object recognition were analyzed. A comparative characterization of known models for object recognition was carried out, namely SSD, YOLO and Faster R-CNN, and experiments were conducted to determine the qualitative characteristics of the proposed models. As a result, the choice was made in favor of the SSD model.

A comparative characterization of different datasets to use in supervised learning was conducted. COCO dataset was selected as the main one.

An algorithm for filtering model results according to the selected operating mode was developed and a program with a graphical interface was developed to demonstrate its work.

Combination of fast SSD model, extensive COCO dataset with 80 different object classes, and filtering algorithm produces efficient object detection system, that allows processing information as fast as 20 frames per second and find only relevant objects in different environments based on the working setup.

The developed system can be used in different situations. Example of case where this technology can be useful is surveillance system. Based on what we need surveillance for, we can configure corresponding mode: detecting people in office or detecting vehicles in logistic company.

Another example of a situation where a solution could be useful is assistive robots. If they work outdoors they can help navigate person across city. If they work indoors, the same system can navigate person across building and help locating relevant things as well.

7. References

[1] Jiuxiang Gu et al. Recent Advances in Convolutional Neural Networks. 2015. arXiv: 1512.07108 [cs.CV].

[2] Shaoqing Ren and Kaiming He and Ross Girshick and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2016. arXiv: 1506.01497 [cs.CV]

[3] Ross Girshick. Fast R-CNN. 2015. arXiv: 1504.08083 [cs.CV]

[4] Teslyuk V., Kazarian A., Kryvinska N., Tsmots I. Optimal artificial neural network type selection method for usage in smart house systems. Sensors, 2021, 21(1), pp. 1–14, 47

[5] Joseph Redmon and Santosh Divvala and Ross Girshick and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. 2016. arXiv: 1506.02640 [cs.CV]

[6] Liu, Wei and Anguelov, Dragomir and Erhan, Dumitru and Szegedy, Christian and Reed, Scott and Fu, Cheng-Yang and Berg, Alexander C. SSD: Single Shot MultiBox Detector. 2016. arXiv: 1512.02325 [cs.CV]

[7] Hussam Qassim and David Feinzimer and Abhishek Verma. Residual Squeeze VGG16. 2017. arXiv: 1705.03004 [cs.CV]

[8] Sik-Ho Tsang. SSD — Single Shot Detector (Object Detection). 2018. URL: https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11

[9] Everingham, M. and Eslami, S. M. A. and Van~Gool, L. and Williams, C. K. I. and Winn, J. and Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. International Journal of Computer Vision (Jan 2015) p. 98-136.
[10] Alex Krizhevsky. CIFAR-10 and CIFAR-100 datasets. [Web resource]: http://www.cs.toronto.edu/~kriz/cifar.html
[11] Olga Russakovsky and Jia Deng and Hao Su and Jonathan Krause and Sanjeev Satheesh and Sean Ma and Zhiheng Huang and Andrej Karpathy and Aditya Khosla and Michael Bernstein and Alexander C. Berg and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. 2015. arXiv: 1409.0575 [cs.CV]
[12] Hrytsyk V., Nazarkevych M. Real-Time Sensing, Reasoning and Adaptation for Computer Vision Systems. Lecture Notes on Data Engineering and Communications Technologies, 2022, 77, pp. 573–585.
[13] Berezsky O., Zarichnyi M., Pitsun O. Development of a metric and the methods for quantitative estimation of the segmentation of biomedical images. Eastern-European Journal of Enterprise Technologies, 2017, 6(4-90), pp. 4–11.
[14] Tsung-Yi Lin and Michael Maire and Serge Belongie and Lubomir Bourdev and Ross Girshick and James Hays and Pietro Perona and Deva Ramanan and C. Lawrence Zitnick and Piotr Dollár. Microsoft COCO: Common Objects in Context. 2015. arXiv: 1405.0312 [cs.CV]
[15] Zekun Luo, Zheng Fang, Sixiao Zheng, Yabiao Wang, Yanwei Fu. NMS-Loss: Learning with Non-Maximum Suppression for Crowded Pedestrian Detection. 2021. arXiv: 2106.02426 [cs.CV]
[16] Dayal A, Paluru N, Cenkeramaddi LR, J. S, Yalavarthy PK. Design and Implementation of Deep Learning Based Contactless Authentication System Using Hand Gestures. Electronics. 2021; 10(2):182. https://doi.org/10.3390/electronics10020182.