ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 5, No.1, 2020

THE ALGORITHM OF CYBER-PHYSICAL SYSTEM TARGETING ON A MOVABLE OBJECT USING THE SMART SENSOR UNIT

Dmytro Kushnir and Yaroslav Paramud

Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine.
Authors' e-mail*: dimakush1@gmail.com, paramud.yar@gmail.com*

Submitted on 21.04.2020

***Abstract*: It is known that smart sensor units are one of the main components of the cyber-physical system. One of the tasks, which have been entrusted to such units, are targeting and tracking of movable objects. The algorithm of targeting on such objects using observation equipment has been considered. This algorithm is able to continuously monitor observation results, predict the direction with the highest probability of movement and form a set of commands to maximize the approximation of a moving object to the center of an information frame. The algorithm has been verified on an experimental physical model using a drone. The object recognition module has been developed using YOLOv3 architecture. iOS application has been developed in order to communicate with the drone through WIFI hotspot using UDP commands. Advanced filters have been added to increase the quality of recognition results. The results of experimental research on the mobile platform confirmed the functioning of the targeting algorithm in real-time.**

***Index Terms:* Cyber-physical system, drones, CNN, iOS, targeting algorithm, mobile system, smart sensor unit, YOLOv3.**

I. INTRODUCTION

Nowadays Cyber-Physical Systems (CPS) development becomes more and more popular. Cyber-Physical Systems are integration of physical processes and cybernetic tools which provide the organization of measuring and computing, secure storage and sharing of measuring and service information, organization and implementation of impacts on the physical processes [1]. Integrating such components into one system allows us to create complex and efficient technical and service tools.

At the same time, an important part of any CPS could be Smart Sensor Unit (SSU). Generally, CPS structure is a set of SSU, which interacts with the real-world and is equipped with observance and targeting elements [2].

The tracking and targeting task of movable objects can be assigned to one of the SSU. Research in that area is a topical task, which can be used in any movement tracking system like drones. Accordingly, the aim of this research is the development of a targeting algorithm for SSU on a movable object, that could be applied to a mobile drone system.

Drones are impacting our society in many ways. We can now accomplish tasks that weren't possible before or that required a lot of human intervention. They have completely changed the way we gather and process information on remote areas. Most of the drones are human-controlled but with the rise of artificial intelligence we are entering a whole new world.

There are a lot of applied tasks that drones can track or guide [3-7]. From tracking person, using GPS coordinates on huge distances [3] up to systems that's are trying to encourage animals to be playful [6].

Fig. 1. An example of a GPS drone tracking system using a personal mobile device as a locator

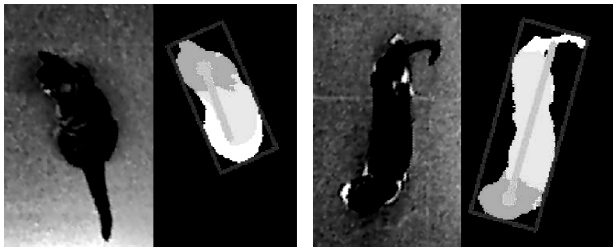


Fig. 2. Animal playful pose analysis drone system [6]

Such systems, depicted as an example in Fig. 1 and Fig. 2 require the availability of huge computation machines or at least cloud computing system which requires high network latency.

Nowadays embedded mobile solutions for object processing are getting increasingly popular. Such mobile frameworks like CoreML [8] allow us to use convolutional neural networks (CNN) model's architecture like You Only Look Once (YOLO) to resolve different applied tasks. One of such tasks is object recognition. There are many ways to perform object recognition. We can divide it into two main

groups: Machine Learning (ML) and Deep Learning (DL) approaches.

For ML, on the one hand, it becomes necessary to first define its features (for example, Haar-like features), then to use a technique such as support vector machine to do the classification. On the other hand, DL techniques are able to do end-to-end object detection without specifically defining features, and are typically based on CNN.

YOLOv3 architecture belongs to the last type of neural networks. This architecture is easy to use and quite effective among the other CNN network [9]. The result of YOLOv3 recognition is the bounding box depicted on the screen with a probability of recognition. Such recognition results of quality could be increased by using minimization and smoothing filters [10]. The example of such system is depicted in Fig. 3.

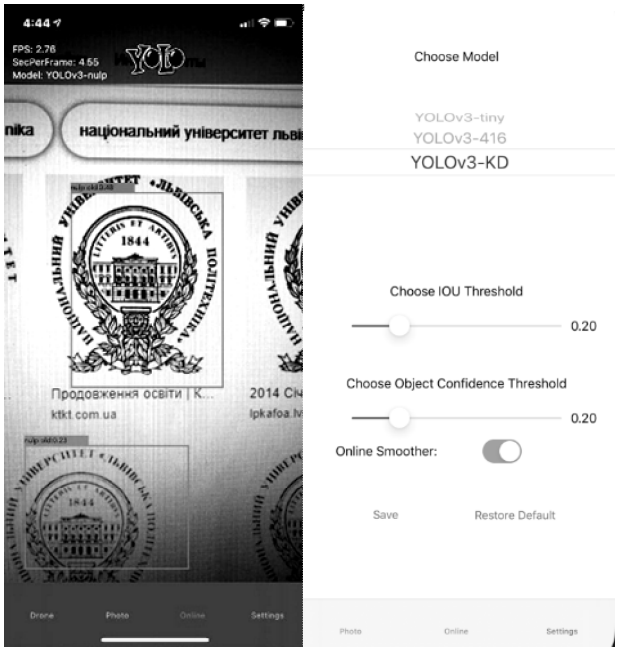


Fig. 3. Object recognition system with applied smoothing (IOU) and minimization (o. confidence) filters

This system uses its own YOLOv3 model, to recognize custom objects in real-time on the mobile device. The smoothing and minimization filters help reduce the time of search and recognition of objects.

Combining result analysis from mobile drone object targeting, recognition systems, CPS and SSU, we can create a new approach that will allow us to create an embedded targeting algorithm.

II. TARGETING ALGORITHM ARCHITECTURE

The schematic representation of the targeting algorithm as a part of SSU is shown in Fig. 4.

The tracking algorithm is divided into two parts: distance and angle to the recognized object calculation.

Distance between information frame center and the recognized bounding box is calculated by the formula:

$$\begin{aligned} hypot &= \sum_{i=1}^{\sqrt{n}} v_i^2 = \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2} \\ d_x &= A_x - B_x \\ d_y &= A_y - B_y \\ dist &= hypot(d_x, d_y) \end{aligned} \quad (1)$$

where: v – the input vector; $A_{x,y}$ – information frame center of coordinates; $B_{x,y}$ – recognized bounding box center of coordinates; $d_{x,y}$ – coordinates to compute the distance; *hypot* – function, hypotension or square root sum of squares of input vectors; *dist* – distance resultant vector. The distance between two points in the Cartesian coordinate system.

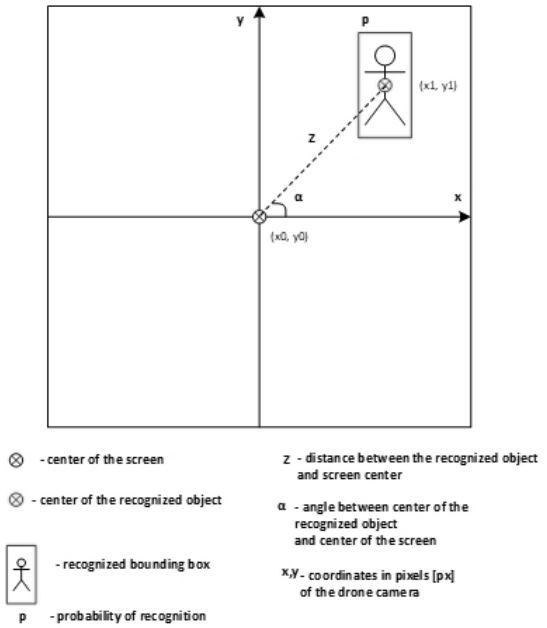


Fig. 4. Object targeting algorithm schematic representation.

To determine the distance coefficient, it is advisable to calculate the value representing a percentage of the area, which occupies the bounding box recognition of the object concerning the size of the observable area.

$$area\_p = \left(\frac{area}{frame\_h \times frame\_w}\right) \times 100 \quad (2)$$

where *area* – rectangle with the recognized bounding box; *frame h* – information frame height; *frame w* – information frame width; *area p* – percent ratio of the device screen size and the size of the rectangle with the recognized bounding box (distance coefficient).

From here we get the drone motion algorithm on the back and forth:

- If area_p value is less than 25%, move the drone forward.
- If area_p value is greater than 25%, move the drone back.

- If the area_p value is between 25% and 50%, the drone standing still.

Besides, the distance resultant vector will be used to assist with targeting in the right direction. This vector should strive for the information frame center of coordinates. This is a helper value that assists distance and angle calculation modules choose the right movement solution, with regard to the Cartesian square.

The deviation angle between the center point of the screen and the center of the rectangle with the recognized bounding box is calculated by the formula:

$$\begin{aligned} d_x &= B_x - A_x \\ d_y &= A_y - B_y \\ angle &= \frac{\arctan(d_x, d_y) \times 360}{2 \times \pi} \end{aligned} \quad (3)$$

where: $A_{x,y}$ – information frame center of coordinates; $B_{x,y}$ – recognized bounding box center of coordinates; $d_{x,y}$ – coordinates to compute the angle between two points for the arctangent in Cartesian square; *angle* – The angle in radians between the center of the screen and the recognized bounding box.

The tracking algorithm diagram is shown in Fig. 5.

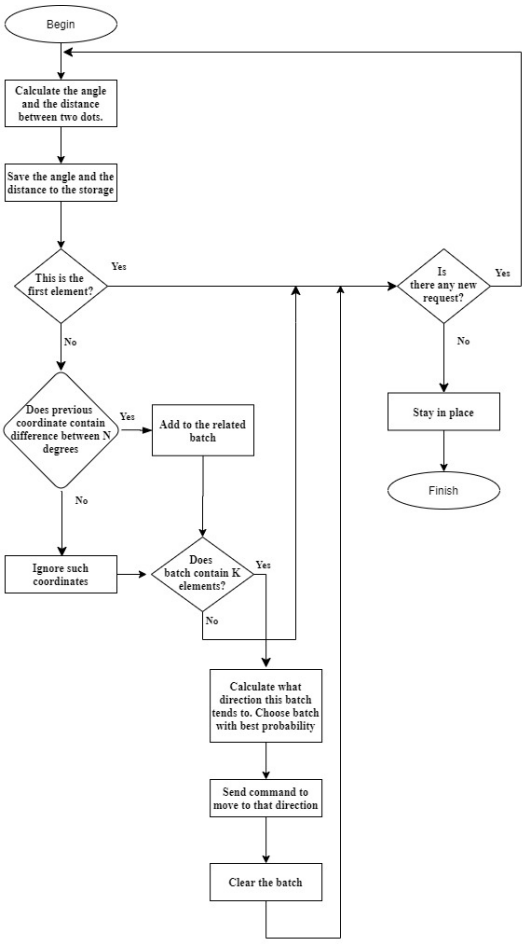


Fig. 5. Tracking algorithm diagram

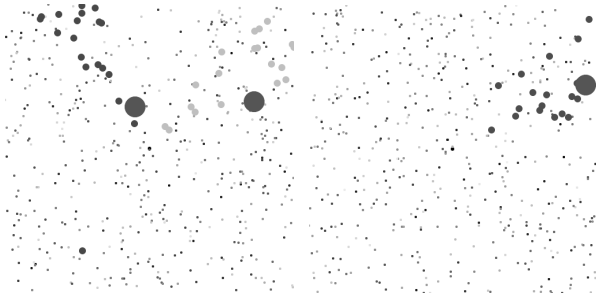


Fig. 6. Tracking algorithm simulation for 500 randomly recognized objects. The biggest dot represents the final direction point

The tracking algorithm uses values of *angle* and *distance* to save it into the batch. Then, if there are differences in N degrees with the previous batch in some direction – it means there are possible movements on a particular side. If the same direction behavioral repeats K times, we send some set of movement commands to the drone. The value of movement depends on the position of the last batch in the batches array. For example, we can rotate drone clockwise in 20 degrees and move it down to 6 centimeters.

The algorithm was tested in a simulated environment. For screen with dimensions 960x720 pixels and 500 randomly located recognized objects. The appearance of such objects performed consistently one by one. Some objects could relate to the common class. The results are shown in Fig. 6.

Here, the biggest dot represents the last batch in the array. The drone will move in that direction. Implementation can be seen in Listing 1.

Listing 1

```
func predictMovement(box: (angle: CGFloat, distance:
 CGFloat, score: Float, classIndex: Int)) {
if (predictedBoxes.count <= 1) {
 predictedBoxes.append(box)
} else {
let previousBox = predictedBoxes.last
let maximumDegreeDifference: CGFloat = 20.0
let delta = compareTwoValues(a: box.angle, b: previou
sBox!.angle)
if (delta < maximumDegreeDifference) {
predictedBoxes.append(box)
}
// remove observation history after some period
if (predictedBoxes.count == 30) {
let lastBox = predictedBoxes.last
performMovement(box: lastBox!)
predictedBoxes.removeAll()
}
}
```

III. RECOGNITION MODULE ARCHITECTURE

The recognition module is based on the YOLOv3 model with two input layers and additional filters.

YOLOv3 divides the input image into an S×S grid. Each grid cell predicts only one object.

For the developed model YOLOv3-KD included 2 source layers, each of which provides regions for different objects in the form of a two-dimensional array. The dimensions of the array cells are as follows: 8, 16 and 32.

Suppose we have an image of 416x416 pixels at the input. Then the original matrices (grids) will have the sizes: 52x52, 26x26 and 13x13 (416/8 = 52, 416/16 = 26 and 416/32 = 13).

After launching the downloaded Core ML code for the developed neural network model, the output will yield the following results: [1, 1, 18, 25, 25]. Where 18 is the vector obtained by the formula:

$$z = b * ((t_x + t_y + t_w + t_h + p_o) + (p_1 + p_2 + \ldots + p_n) \quad (4)$$

where z is the resultant vector b – source network layers (regions); t_x, t_v, t_w, t_h – coordinates x, y, width and height of the predicted region; p_o – region appearance prediction factor (Object confidence minimization filter); $p_1..p_n$ are input classes, where n is the number

Now we need to process this vector. For each predicted region, a probability distribution is required for a given range of classes. We can do this with SoftMax function. It helps classify the probability of recognitions for some vector of objects. Then it chooses the most valuable probability:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \quad (5)$$

where $\sigma(z)_i$ – smoothed vector with the biggest probability; z – a resultant vector that has been passed to SoftMax; K – count of classes.

To obtain the coordinates and sizes of regions we need to use the following formula:

$$\begin{aligned} b_x &= s(t_x) + c_x \\ b_y &= s(t_y) + c_y \\ b_w &= p_w e^{tw} \\ b_h &= p_h e^{th} \end{aligned} \quad (6)$$

where b_x, b_v, b_w, b_h are the predicted x, y coordinates, width, and height respectively, t_x, t_v, tw, th outputs of the neural network, s – sigmoid function, c_x and c_v are the upper leftmost initial grid coordinates, and p_w and p_h – the value of the anchors for the three regions.

After we have obtained the coordinates and sizes of the regions and the corresponding probabilities for all the objects found in the image, we can start drawing them on top of the image. Each prediction is processed in the separate thread to retrieve the stream frame asynchronously (Listing 2).

The YOLOv3 base model is showing good recognition results. We can see that it is outperforming most of his competitors on the Common Objects In Context (COCO) dataset (Fig. 7) [9] .

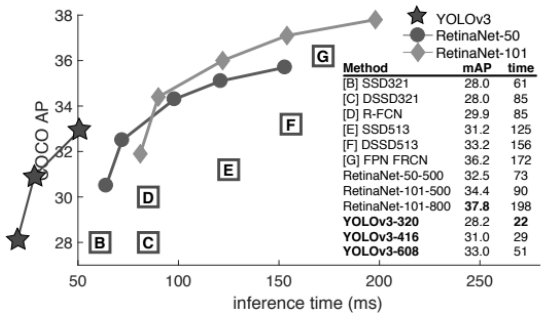


Fig. 7. YOLOv3 model performance comparison with other CNN models where mAP – mean average precision

Listing 2

```
func predict(frame: UIImage) {
DispatchQueue.global().async {
do {
let startTime = CACurrentMediaTime()
let predictions = try self.model.predict(frame: frame)
let elapsed = CACurrentMediaTime() – startTime
self.showResultOnMain(predictions: predictions, elaps
ed: Float(elapsed), error: nil)
} catch let error as YOLOError {
self.showResultOnMain(predictions: nil, elapsed: -
1, error: error)
} catch {
self.showResultOnMain(predictions: nil, elapsed: -
1, error: YOLOError.unknownError)
}
}
}
```

IV. MOBILE DRONE SYSTEM STRUCTURE OVERVIEW

The proposed mobile drone system is depicted in Fig. 8. It is an example of how we could use such an algorithm.

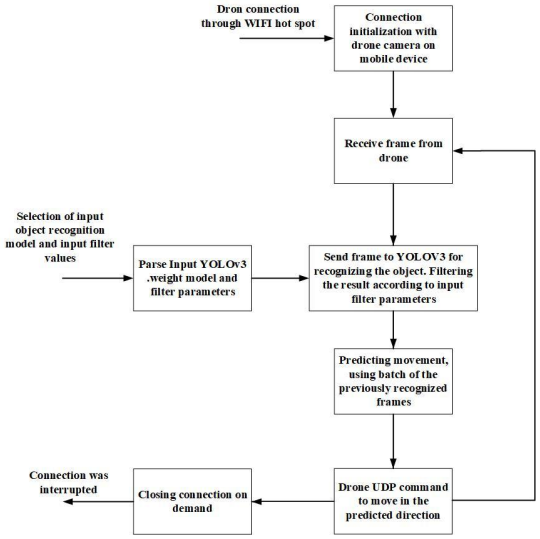


Fig. 8. General system structure

The mobile system asynchronously awaits stream frame, that could come from drone directly via WIFI or through any wireless router. Once it received the frame, it passes it consistently to the recognition and tracking modules.

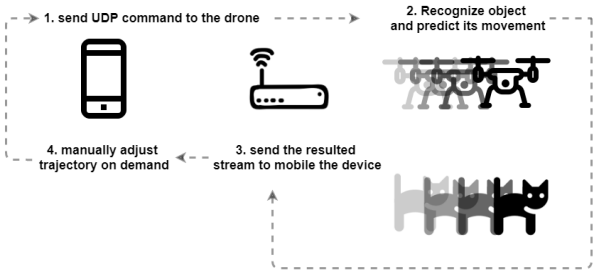


Fig. 9. Schematic representation of movement tracking

According to the diagram from Fig. 9, the mobile device sends UDP commands to the drone via an access point. In return, the drone may send some specific info like a video-stream frame. In that work TELLO drone from DJI company has been used. As this drone doesn't have any powerful processor inside, all computations will be made on a mobile device. The list of commands that have been used to control the drone is provided in Table 1.

Table 1

Tello UDP commands

Command	Description
command	entry SDK mode
takeoff	Tello auto takeoff
land	Tello auto land
stream on	Set video stream on
stream off	Set video stream off
emergency	Stop all motors immediately
up x	Tello fly up with distance x cm x: 20-500
down x	Tello fly down with distance x cm x: 20-500
left x	Tello fly left with distance x cm x: 20-500
right x	Tello fly right with distance x cm x: 20-500
forward x	Tello fly forward with distance x cm x: 20-500
back x	Tello fly back with distance x cm x: 20-500
cw x	Tello rotate x degree clockwise x: 1-3600
ccw x	Tello rotate x degree counterclockwise x: 1-3600
wifi ssid pass	Set Wi-Fi with SSID password

The stream has been parsed in the separate program thread to decrease network latency (Listing 3). Additionally, this allows us to process information frames in recognition module in parallel, using other threads.

Listing 3

```
// Send Stream
DispatchQueue.global(qos: .userInteractive).async {
 var currentImg: [UInt8] = []
 while self.isConnected {
 let data = self.tello.getStream()
 if let d = data {
 currentImg = currentImg + d
 if d.count < 1460 && currentImg.count > 40 {
self.frameDecoder.interpretRawFrameData(&currentI
mg)
 currentImg = []
 }}}}
// Retrieve Stream in callback
 var cgImage: CGImage?VTCreateCGImageFromCVP
ixelBuffer(frame, options: nil, imageOut: &cgImage)
if let cgImage = cgImage {
DispatchQueue.main.async {
self.videoView.image = UIImage(cgImage: cgImage)
if !self.processed {
guard let image = self.videoView.image else {
self.showAlert(title: "Warning!", msg: "Image from dr
one can't be obtained") return }
self.modelProvider.predict(frame: image)
} else {
self.predictionLayer.clear()}}
} else {
print("Video stream fail")
}
```

V. MOBILE SYSTEM ALGORITHM INTEGRATION

The tracking algorithm was integrated into the mobile system, written in the Swift language. Core ML framework was used to handle object recognition using the YOLOv3 model.

If several object paths will be recognized simultaneously, the algorithm will choose one with the best probability.

As it can be seen in Fig. 10, drone has moved from one position (white cross shifted to the left and down) to another which is closer to the center of information frame (commands clockwise rotation to 4 degrees and move up to 8 centimeters were sent).

There may be some issues with network latency which may reduce the quality of tracking on large distances. This could be fixed by applying a more powerful network adapter or hardwire algorithm into drone cheap. The accuracy of the correct algorithm work in normal conditions is about 70-80%.

Fig. 10. Example of recognition and tracking modules work on mobile system. The big white cross represents the center of the screen, white dot – a center of the recognized object

VI.CONCLUSION

The paper presents the application of the developed algorithm of object targeting as a part of SSU on a mobile drone system. This algorithm was integrated and tested in a real and simulated environment. In the current stage algorithm is able to send several commands to guide the device to the observable object direction, depending on its location. In normal conditions the accuracy of correct algorithm work is about 80%. It was shown that it is possible to use the developed algorithm in a real-time mobile drone system to track a movable object.

Dmytro Kushnir. Ph.D. student at Computer Engineering Department of Lviv Polytechnic National University, Institute of Computer Technologies, Automation and Metrology (ICTA). Received Master degree of Computer Engineering in 2018 at Lviv Polytechnic National University. He has worked as a Soft-ware Engineer in web development since 2017 in GlobalLogic IT company.

Scientific interests include machine learning, IoT, mobile and web development.

REFERENCES

[1] A. Melnyk, (2016, November). Cyber-physical systems multilayer platform and research framework. Advances in Cyber-Physical Systems [Online]. Available: http://science.lpnu.ua/acps/all-volumes-and-issues/volume-1-number-1-2016/cyber-physical-systems-multilayer-platform-and

[2] O. Botchkaryov, V. Golembo, Y. Paramud, V. Yazuk, Cyber-physical systems: technologies of data collection [Text]: monography – O. Botchkaryov, V. Golembo, Y. Paramud, V. Yazuk. Editorial chiev: prof. A. Melnyk, Lviv: "Magnolia 2006", 2019. – pp.10–12 (in Ukrainian)

[3] A. Koubaa, B. Qureshi, (2018, March). DroneTrack: Cloud-Based Real-Time Object Tracking using Unmanned Aerial Vehicles, IEEE Access [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8306379

[4] A. Koubaa, B. Qureshi, (2018, March). DroneTrack: Cloud-Based Real-Time Object Tracking using Unmanned Aerial Vehicles, IEEE Access [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8306379

[5] G. Ding, L. Zhang, Y. Lin, T. Tsiftsis, Y. Yao (2018, January). An Amateur Drone Surveillance System Based on the Cognitive Internet of Things, IEEE Communications Magazine [Online]. Available: https://ieeexplore.ieee.org/document/8255734

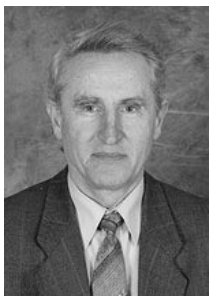
[6] P. Pons, J. Jaen, A. Catala (2015, November). Developing a depth-based tracking system for interactive playful environments with animals, ACE '15: Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology [Online]. Available: https://dl.acm.org/doi/10.1145/2832932.2837007

[7] R. Canals, A. Roussel, J. Famechon (2002, August). A biprocessor-oriented vision-based target tracking system, IEEE Transactions on Industrial Electronics [Online]. Available: https://ieeexplore.ieee.org/abstract/document/993283

[8] Apple (2019, October), "CoreMl documentation" [Online]. Available: https://developer.apple.com/documentation/ coreml

[9] J. Redmon, A. Farhadi (2018, April) "YOLOv3: An Incremental Improvement" arXiv 2018 [Online]. Available: https://pjreddie.com/media/files/papers/YOLOv3.pdf, Apr.

[10] D. Kushnir, Y. Paramud, (2019, November). Methods for real-time object searching and recognizing in video images on ios mobile platform. Computer Systems and Networks Volume 1, Number 1. [Online]. 1(1), pp. 24–34 Available: https://doi.org/10.23939/csn2019.01.024 (in Ukrainian)



Yaroslav Paramud. Ph.D., Assoc. Prof at Computer Engineering Department of Lviv Polytechnic National University.

Scientific interests include processing radar information, research algorithms, and structures of specialized computing devices and systems. The number of scientific publications – more than 30, has 13 Inventor's Certificates.