WareVision: CNN Barcode Detection-Based UAV Trajectory Optimization for Autonomous Warehouse Stocktaking

Ivan Kalinov, Alexander Petrovsky, Valeriy Ilin, Egor Pristanskiy, Mikhail Kurenkov, Vladimir Ramzhaev, Ildar Idrisov, and Dzmitry Tsetserukou

***Abstract*—This letter presents a heterogeneous Unmanned Aerial Vehicle (UAV)-based robotic system for real-time barcode detection and scanning using Convolutional Neural Networks (CNN). The proposed approach improves the UAV's localization using scanned barcodes as landmarks in a real warehouse with low-light conditions. Instead of using the standard overlapping snake-based grid (OSBG) trajectory, we implement a novel approach for flight-path optimization based on barcode locations. This approach reduces the time of warehouse stocktaking and decreases the number of mistakes in barcode scanning.**

***Index Terms*—Inventory management, computer vision for automation, multi-robot systems, AI-based methods, object detection, segmentation and categorization.**

I. Introduction

A. Motivation

Vertical take-off and landing (VTOL) UAVs have been gaining popularity and being used in numerous domains for the last ten years. The tendency of using drone technologies in the industry completely changes the business model and redraws enterprise landscapes. The growing e-commerce market has led to an increase in warehouse space and stricter tracking requirements at all stages [1]. Due to these requirements and the latest technological achievements in UAV indoor navigation, drones have become widely used in warehousing operations, e.g., for automated warehouse stocktaking [2]. UAV-based system for this application can help warehouse employees to eliminate tedious and dangerous procedures. However, one of the fundamental aspects of the application of drones for warehouse stocktaking is the computer vision system's quality and the error rate. UAVs should work autonomously and provide appropriate accuracy. Simultaneously, the equipment installed on the UAV has to have as little weight as possible and low power consumption for increasing operation time.

Fig. 1. The heterogeneous autonomous system of two robots.

B. Problem Statement

A typical warehouse is a vast building with a significant number of shelves over 10 meters in height. In such warehouses, every pallet and pallet-place has a unique 1D-barcode, rarely a radio frequency identification (RFID) marker, or a QR-marker. Barcodes and especially 1D-barcodes are the most common type of identifiers that are being used in 87% of warehouses [1]; that is why our method for an autonomous heterogeneous system (Fig. 1) has a focus on barcode scanning. Most of the inventory processes are performed manually. Warehouse workers put down every pallet manually, then scan it and put it back on the rack. It is a very routine and time-consuming procedure. Full inventory checking deducts more than three days of the regular warehouse operation [3]. Therefore, a UAV would be an excellent solution for this task. Still, the main problem of all UAVs for indoor flight is a lack of precise localization and navigation methods for reliable barcode scanning, thus, current commercial systems [4], [5] use piloted UAVs to scan the tags.

Scanning information from barcodes using a camera is a complicated and ineffective method. Existing camera-based methods for barcode scanning [2] are not suitable for usage in real warehouse conditions because of lack of lighting, image artifacts of rolling shutter cameras, and small size of barcodes.

Manuscript received March 15, 2020; accepted July 1, 2020. Date of publication July 21, 2020; date of current version August 26, 2020. This letter was recommended for publication by Associate Editor J. Morrison and Editor J. Yi upon evaluation of the reviewers' comments. *(Corresponding author: Ivan Kalinov.)*

The authors are with the Skolkovo Institute of Science and Technology, Moscow 143026, Russia (e-mail: ivan.kalinov@skolkovotech.ru; aleksandr.petrovskii@skolkovotech.ru; valery.ilin@skoltech.ru; pristanskiyee@gmail.com; mikhail.kurenkov@skoltech.ru; vladimir.ramzhaev@skoltech.ru; i.idrisov@skoltech.ru; d.tsetserukou@skoltech.ru).

This letter has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/LRA.2020.3010733

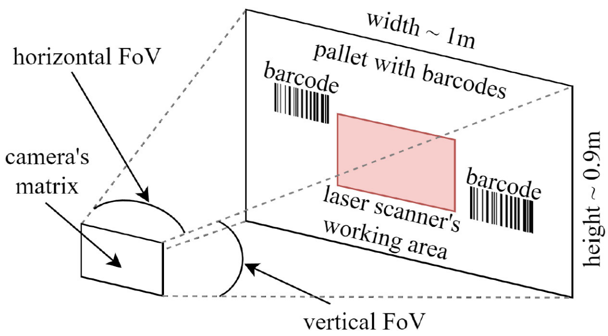


Fig. 2. Schematic representation of a real scene from the UAVs point of view, including camera frame and working area of laser scanner.

In order to get reliable information from the barcode during the flight, the UAV should have a global shutter camera and an on-board computer with enough computational performance for processing images. The real size of a pallet on a rack is usually about 1 meter wide and 0.9 meter high (Fig. 2). According to IC Barcode Tool Recommendations by Imagine Source company [6], the on-board global shutter camera should have 55.8 Mpx image resolution and 53° field of view (FoV) to scan barcodes from a distance of 1 meter. For stable scanning, we should have at least 2 pixels for 1 bar of a barcode with a standard width of 10 mil [6]. This requirement is impossible due to the unavailability of such a camera in mass production. To solve this problem, we developed an approach based on a cheap rolling shutter camera for barcode detection and a laser scanner for reading. A detailed description is presented in Section II.

C. Related Work

Over the past few years, many systems related to drone-assisted inventory management in warehouses were presented. The most relevant of them introduced frameworks for a multi-UAV solution or a UAV along with ground robot solution. However, they describe either half-autonomous systems with piloted UAVs or autonomous systems with high positioning error (up to one meter) or unsuitable system configurations, which are not applicable for stable flight in narrow passages between racks in warehouses [7]–[9]. Fernández-Caramés *et al.* [10] presented a design of the UAV and blockchain-based system for Industry 4.0 inventory. Their work focuses on an industrial inspection with RFID tags, but the UAV flight is not autonomous and requires a pilot, unlike our system.

Beul *et al.* [11] presented a UAV capable of fast autonomous indoor flight and scanning RFID tags in a warehouse. Kwon *et al.* [12] proposed an autonomous UAV with a low-cost sensing system to be used effectively for narrow and dark warehouse environments, but the approach for barcode scanning was not mentioned.

Cho *et al.* [13] introduced a 2D-barcode detection system based on neural networks. The work states the precision and recall are about 95% during post-processing. The system's configuration was not mentioned in the article, as well as the results of flight experiments. Besides, the system requires more than 12,000 images for training and testing, while we achieved similar results with only 537 images in training data set, and our system works in real-time.

Hansen *et al.* [14] presented an approach for real-time barcode detection and classification using deep learning, but their system works with high-quality images which are difficult to obtain during the flight, and this method requires high computing power for high performance which makes it impossible to use on the drone for inventory purposes.

Xu *et al.* [15] introduced algorithms for the automatic extraction of barcodes from video data. For a known barcode region, a Harris corner detector and Hough transform-based algorithm were applied to estimate the bars' orientation angle quickly. No flight experiments were presented in the article, and all results were obtained during post-processing.

Initially, the problem of barcode localization in the image was solved by complicated transformations of the image matrices, distance search, and pattern matching [16]. With the release of the OpenCV library [17], the solution to this problem was simplified: the researchers began to add image offset, blur filters, and perform the other transformations to solve the problem of barcode detection [18]. Creusot *et al.* [19] used the optimal detected parameters in the Maximally stable extremal regions (MSER) algorithm to highlight the dark areas of the barcode in the image [20]. Combining the obtained areas, he received a reliable barcode mask. This method requires a minimum amount of noise in the image and the big size of the barcodes, which is impossible in real warehouse conditions. Zhou *et al.* [21] introduced an angle-robust method for multiple 1D-barcode detection based on a line extracting algorithm. The algorithm is resistant to image rotation. However, it shows low accuracy in photos with a small barcode size.

Katona *et al.* [22] presented a robust solution based on OpenCV methods composition that allows detecting barcodes with high precision. The main feature of the method is the usage of a template match algorithm [23]. After implementing the presented approach, we discovered that it is not suitable for our solution. In the images from a distance of 1 meter, even with big barcodes, it gave poor results. Also, their approach requires a large number of specific patterns for every new environment.

Works described above [16]–[22] did not use machine learning models for barcode detection. Redmon *et al.* [24] presented one of the most popular CNN for object detection. Despite the high accuracy of this method it requires the high consumption of computing resources. In addition, CNNs for semantic segmentation are more sensitive to noise, which may be the barcode, and also gives more accurate coordinates, which is important for position adjustment.

Also, let us consider existing path optimization methods for UAVs since our robot has to build an optimal trajectory to cover all the barcodes. F. Cheng *et al.* [25] presented the UAV trajectory optimization method for data offloading in the edge area of multiple adjacent cells. However, their results are obtained only in simulation and not applicable to indoor flight in narrow spaces. H. Oleynikova *et al.* [26] proposed a continuous-time trajectory optimization method for real-time collision avoidance of UAVs. Although this approach can work indoor in real-time, it requires using a visual-inertial stereo sensor for obstacle recognition and Intel i7 CPU with high power consumption for calculations. Andrew T. Klesh *et al.* [27] presented a new information-based formulation for optimal exploration of a given environment by UAVs. The presented approach solves Traveling Salesman Problem for the exploration of three objects of interest only in a simulation. Yong-bo Chen *et al.* [28] introduced an additional control force into the artificial potential field planning method for the UAV path planning problem without real-environment tests.

D. Contributions

To exclude humans from stocktaking completely, we have developed an autonomous heterogeneous robotic system of two robots: the UGR and the UAV. This combination allows keeping an always-up-to-date inventory record of the contents within the warehouse (Fig. 1).

This system is capable of autonomous navigation and precise localization in an indoor environment. We solve the problem of robust system operation by dividing localization into two parts for each subsystem, i.e., the UAV and the UGR. The UGR performs global localization and navigation in a warehouse, while the UAV always flies above the platform, detects and scans barcodes on the racks and pallets.

For drone localization, we have developed the method of pose estimation relative to the UGR. This approach enables us to calculate coordinates of the UAV relative to the UGR and then to obtain the global coordinates of the UAV. Also, our method does not imply the utilization of any additional infrastructure for navigation, as opposed to motion capture systems, since all necessary equipment is installed on the UAV and the UGR. This setup is described in detail in our previous work [29].

Our main contributions for this work are:

- UAV-based, robust, and lightweight system for real-time barcode detection and scanning;
- active perception by the UAV based on the detected barcodes, which improves its localization in comparison with the previous method [29]. By active perception we imply real-time position adjustment of the UAV using standard objects of warehouse environment (1D-barcodes);
- proposed approach with CNN for barcode detection reduces fly-by time optimizing the UAV's trajectory in comparison with the standard OSBG fly-by method;
- the robot is able to get global coordinates of each detected and scanned barcode for warehouse planogram and further analytics.

II. Methodology

A. Heterogeneous Robot Overview

Our autonomous robotic system consists of the UGR and the UAV. The main objective of the UGR is to determine its coordinates relative to the surrounding objects in the warehouse. Such an approach allows us to calculate the position of each pallet accurately. In order to determine the pallet's location in 2D space, the robot uses SLAM algorithm based on graph-based approaches [30]. For map building of the surveyed area, the robot uses a Light Detection and Ranging device (LIDAR). The robot receives two-dimensional scans and then compares them with local submaps. Thus, the robot finds its position and complements the map with new obstacles. After that, the UGR combines the submaps into one graph and finds the final terrain map using nonlinear optimization methods.

For the UAV localization system, we use a camera on the ground robot and two concentric patterns of active IR (infrared) markers on the UAV. All the key hardware components and communication between the UAV and the UGR are represented in Fig. 3. As we stated, the UAV's goal is to fly in a small cylindrical working area above the mobile robot. We measure the altitude of the UAV using the recognized IR pattern. In our setup, we use an Imaging Source DFK33UX250 2448x2048 camera with a 137.9° FoV lens. We have an IR-passing 950 nm filter on the camera to obscure all the light in the visible range and leave only IR-markers in the image. The detailed explanation of the system setup is described in our previous article [29].

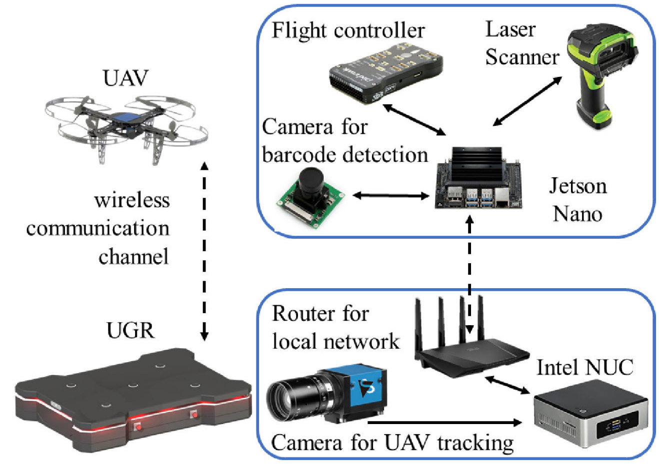


Fig. 3. Communication system setup.

B. Barcode Detection and Scanning System

In stocktaking, the most critical parameters are the duration of the inventory process, the number of tags correctly identified, and the coordinates of scanned tags in the warehouse. Barcodes are the most common type of tags for labeling in warehouses [1]. In subsection I-B, we stated that it is impossible to use a global shutter camera in our conditions. Even if we had a global shutter monochrome camera with resolution 55.8 Mpx, the video received with such a camera with 8-10 Frames Per Second (FPS) could not be transmitted over a local wireless network for processing due to minimal size of frame equal to 6.65 MB. It would have to be processed directly onboard the UAV. Such an approach directly affects its flight-time by increasing the weight of onboard computer and drone's energy consumption. All these factors increase the duration of stocktaking. Besides, for increasing the percentage of correctly scanned barcodes, it is necessary to fly several times in front of each pallet to increase the overlapping area of the image frames.

Therefore, we propose a novel approach. The core idea is to install a small Pi NoIR Camera V2 for barcodes detection and determine their position relative to UAV using CNN, then read them with the Zebra DS3608ER barcode laser scanner. Such a combination is necessary for precise barcode detection since the laser scanner has 12° FoV, and the scanning band has a width of around 0.21 m from a distance of 1.0 m, which is not enough to cover the whole pallet. Combining the camera with the scanner, the UAV can detect barcodes and read them.

Taking into account the previous works [18]–[23] and the small size of barcodes in the frame, it makes sense to use a CNN based algorithm to detect all barcodes in the image and solve the segmentation problem for barcode areas detection. Since we have only one class for recognition, we have chosen the most efficient binary segmentation method based on U-net architecture [31]. The architecture of the implemented CNN, similar to U-net, is presented in Fig. 4. The main module in our network is "Down convolutional" operation" which includes a sequence of the following layers: convolution, activation (ReLU), and batch normalization. "Up convolutional operation" uses the

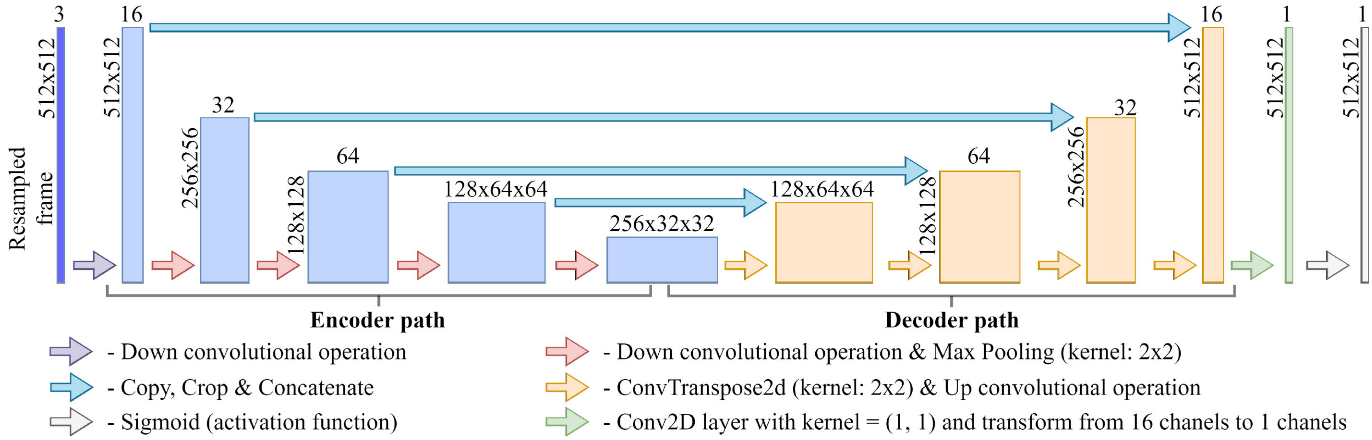
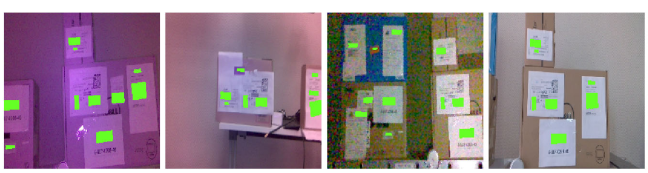


Fig. 4. U-net based architecture of our system.

(a) Source images.



(b) Detected barcodes.

Fig. 5. Segmentation results on validation data.

same layers but instead of increasing the number of channels "Up convolutional operation" decreases it. For feature restoration, we also use "Copy, Crop, and Concatenate" operations. The final layer of the network is "Sigmoid" which allows us to detect a barcode class probability for each pixel in the image.

We have trained our CNN with a relatively small dataset consisting of 537 images. The validation dataset consists of 67 images; the test dataset has 66 images. Each image contains 7 barcode instances on average. These datasets include image frames from Raspberry Pi Camera v1.0 and v2.0 (with rate 50/50). The frames were taken with different lighting conditions: street, room, and IR (in darkness). The collected and marked dataset that is used in our approach is open and available via link.[1]

For the training, we used original 3-channel images (RGB). During the training, we randomly applied one of the following augmentation tricks (flip, rotation, contrast, and brightness change; cropping, gamma changing, and channel shuffling) to each image of the trained dataset in every epoch. The possibilities of each trick are equal, and the total number of epochs is 500. The source image with segmentation results of our CNN is presented in Fig. 5. Then we processed the CNN output with standard OpenCV morphological transformations to erase noise and applied contours detection to get the coordinates of areas with barcodes.

[1]https://box.skoltech.ru/index.php/s/jnoqSwZpes4CnuO

C. Active Perception Method

To localize the drone, we use IR markers [29]. However, this IR-based localization was not enough for flying on altitudes of more than 8 meters since it yields a 10 – 15 cm error in altitude. For localization improvement, we decided to use the surrounding objects of a warehouse, for example, barcodes. This section describes in detail how UAV's active perception works. We simultaneously plan the drone exploration path, detect barcodes, make a barcode map, and estimate the UAV position on this map using verified barcodes as an additional source of information.

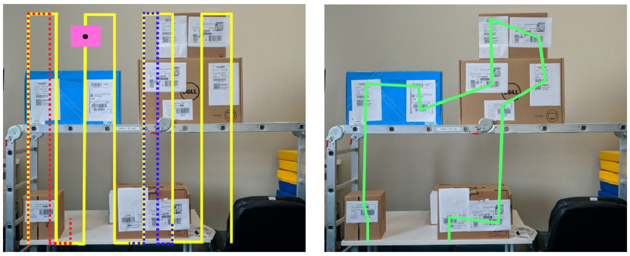
Initially, we generate a list of waypoints with an interval of 0.5 meters, taking into account the maximal and minimal height of the rack. The drone starts following the generated fly-path along these waypoints. When the CNN detects a new region in the UAV camera image as a barcode (Fig. 5(b)), our algorithm adds this region to the factor graph, marks it as a new waypoint and optimizes the path by solving traveling salesman problem. On the next step, the drone flies to the barcode and verifies it with the laser scanner:

- If the barcode gets verified - the algorithm adds it to the barcode database. Then, the UAV updates its position relative to the barcode map. At the last step, the robotic system performs simultaneous optimization of the previous UAV trajectory and the barcode positions using methods of graph-based SLAM [32].
- If the laser scanner cannot verify the barcode - our algorithm deletes this waypoint from the graph, rebuilds the path, and gives the drone the next waypoint.

As the final result, the drone continues to follow the planned path, as shown in Fig. 6(b).

It is important to note that all detected regions from CNN go to the factor graph. The drone updates and specifies its localization relative to each detected barcode region in every UAV camera image even before the laser scanner's verification. The pseudo-code of our algorithm is presented in three blocks below (Algorithms 1–3).

We called the described approach of improving the UAV localization "active perception". During this process, we get the verified location of the pallet in a warehouse with an accuracy of 10–15 cm, which is undoubtedly essential for warehouse stocktaking. The result of this work is an accurate map of the detected and scanned barcodes and their position in a warehouse (Fig. 7).



(a) OSBG trajectories. (b) Optimized trajectory.

Fig. 6. Trajectories of the UAV. For OSBG trajectories, three examples are presented: yellow line - 0% overlapping; red dotted line - 25%; blue dotted line - 50%. The pink rectangle indicates the working area of the laser scanner, the black dot in the middle is the position of the UAV.

Algorithm 1: Scanning Process.

WayPoints $= \{x_1, x_2, x_3, \ldots, x_n\}$
BarcodeDataBase $= \emptyset$
FactorGraph $= \emptyset$
while *WayPoints* **do**
  NextPoint = *WayPoints*.pop()
  FlightPath = OptimizePath(*NextPoint*, *FactorGraph*)
  while *FlightPath* **do**
    GoToPoint(*FlightPath*.pop())
    Barcode = GetBarcodeInfo(*FactorGraph*)
    if *Barcode*.IsDetected = True **then**
      BarcodeDataBase.append(*Barcode*)
    else
      FactorGraph.remove(*Barcode*)
    end if
    FlightPath = OptimizePath(*NextPoint*, *FactorGraph*)
  end while
end while

Algorithm 2: OptimizePath(*FinalPoint*, *FactorGraph*).

Input: *FinalPoint*, *FactorGraph*
Output: *Path*
Barcodes = GetBarcodeFromCNN()
FactorGraph.addBarcodesMeasurements(*Barcodes*)
FactorGraph.addIMUMeasurement(*IMU*)
FactorGraph.optimize()
UAVPosition = *FactorGraph*.GetUAVPosition()
BarcodePositions = *FactorGraph*.GetPosition(*Barcodes*)
Path = SolveTravellingSalesmanProblem(*FinalPoint*, *UAVPosition*, *BarcodePositions*)
return *Path*

Algorithm 3: GetBarcodeInfo(*UAVPosition*).

Input: *UAVPosition*
Output: *Barcode*
Barcode = BarcodeStucture()
Barcode.ID = LaserScannerData()
if *Barcode.ID* is None **then**
  Barcode.IsDetected = False
else
  Barcode.IsDetected = True
end if
Barcode.Position = *FactorGraph*.getPosition(*Barcode*)
return *Barcode*

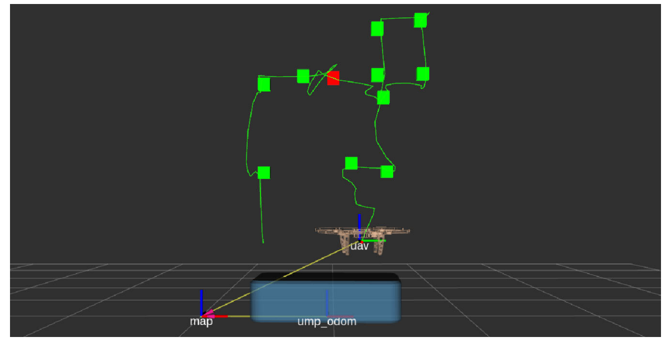


Fig. 7. The accurate map of barcodes. The green boxes represent verified barcodes. The red box represents false-positive barcode detected by the CNN.

TABLE I
COMPARISON OF COMPUTER VISION METHODS

	Objects			Pixels metrics, %			FPS
	TP	**FP**	**FN**	**IoU**	**Precision**	**Recall**	
OpenCV based [15]	30	439	416	9.1	41.40	10.50	29
Multiple barcodes [21]	61	217	385	20.3	52.63	26.73	26
Template matching [22]	87	125	359	57.2	63.20	43.51	20
Yolo v.3 [24]	438	45	8	81.7	89.13	88.70	3
Our approach	**445**	**17**	**1**	**90.7**	**91.56**	**96.96**	13

III. EXPERIMENTS

A. Comparison of Computer Vision Methods

We mentioned several methods for detecting barcodes in the introduction section. Before implementing the Unet-like CNN in our system, we conducted an experiment and compared all these methods in Table I. For comparison, we chose two classes of metrics. The first class is the standard metrics for the object detection - Intersection over Union (IoU), Precision, and Recall. It is worth noting that this class of metrics involves pixel-by-pixel comparison and gives not entirely accurate data on the number of barcodes. Therefore, the second class of metrics, we decided to show quantitatively: correctly recognized barcodes (True positive - TP) with intersection more or equal to 50% of source barcode, mistakenly detected (False Positive - FP) with intersection less than 50%, and undetected (False Negative - FN), which we calculated as the difference between the total number of barcodes and TP. It is important to note that our system checks any detected region for the presence of a barcode with the laser scanner during the flight. FP errors are not so fundamental and will only increase the flyby time, in contrast to FN errors, which lead to skipping the inventory object. Also, as another comparison criterion, we added resulting FPS of Nvidia Jetson Nano (2019). The total number of barcodes in the test data set equals 446.

We received all parameters for our approach with a threshold of 0.909 (Table I). This threshold value is optimal since its reduction does not change the number of FN cases but significantly impairs the other parameters. The comparison shows that standard methods have insufficient results for barcodes recognition. Yolo v.3 CNN has comparable results to our approach; however, it has much lower FPS.

B. Flight-Path Correction

Since the heterogeneous robotic system is designed to automate the stocktaking procedure in warehouses, we need to

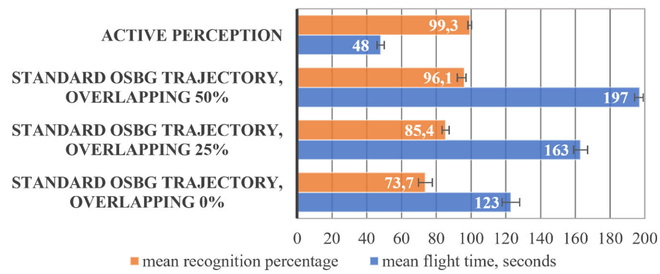


Fig. 8. Comparison of the methods.

detect and scan as many barcodes as possible. The goal of this experiment is to check the performance of the algorithm responsible for the UAV flight-path correction. We compare the results of barcode detection and scanning using two different strategies for the UAV:

1) flight with standard OSBG trajectory (Fig. 6(a));
2) flight with adjusted trajectory using the active perception module (Fig. 6(b)).

For both setups, we used the UGR and the UAV for the barcodes detection in laboratory conditions. Barcodes have been adhered to each box in random order to simulate the real warehouse. In the first setup, CNN is not enabled, and stocktaking procedure goes as following:

- the robot starts stocktaking at the beginning of an alley;
- the UAV flies up to the highest desired point (2.3 meters above the UGR) scanning the first row of boxes;
- the UGR moves to the second row, the UAV follows it;
- the UAV descends until it reaches an altitude of 0.3 meters above the UGR, scanning the barcodes of the second row of boxes.

For the second setup, we use the active perception module described in II-C to recognize the area in front of the UAV. The UAV starts to take-off along the first row of boxes to the highest point where a barcode is detected, adjusting its trajectory if needed. Then, the UAV follows the platform repeating the same procedure. The result of trajectory adjustment using the active perception module is presented in Fig. 6(b).

Since stocktaking is the main task of the developed system, we need to count as many barcodes as possible. To get the maximum percentage of detected and scanned barcodes, we need to cover the entire front of the stand during the flight with steps no more than 0.21 meters. This constraint comes from the scanner's FoV, which is described in subsection II-B. Also, we evaluated the percentage of scanned barcodes depending on the percentage of working area overlapping of the laser scanner for the first setup (Fig. 6). The experiment was conducted twenty times for each of three different percentages of overlapping and proposed active perception approach. Fig. 8 shows a result of comparison between them.

We used one order of boxes arrangement, as well as barcodes in both setups. The predefined flight speed was 0.5 m/s, and the laser scanner rate was 30 FPS in all setups. Fig. 8, representing the result of this experiment, reveals that mean flight time drastically decreased with a flight-path correction method, while the recognition percentage is comparable with standard OSBG flight trajectory with 50% overlapping.

Taking acquired results into consideration, we can conclude that the active perception method for barcodes detection and scanning has much better performance (barcode recognition rate and flight time) than the standard OSBG method with a simple flight trajectory of the UAV.

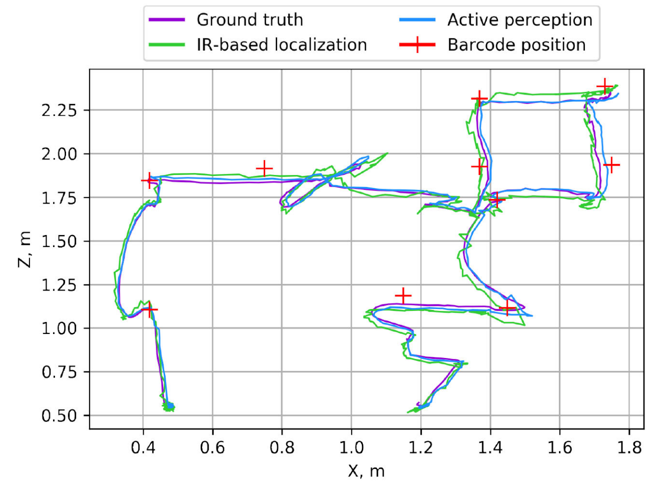


Fig. 9. Visual comparison of three trajectories with the real positions of barcodes. RMSE for IR-based localization is 2.5 cm, and RMSE for active perception localization is 1.8 cm.

C. *Localization Accuracy During Active Perception*

As a part of the experiment, we used the second setup from the accuracy section to estimate the active perception localization accuracy, i.e., how supplemented localization data from the UAV camera influences the drone's flight-path. The positions of all barcodes on the experimental bench were measured using the Vicon motion capture system (mo-cap). The resolution of the camera on the UGR was reduced linearly by a factor of 4, so the dots per inch (dpi) were reduced by a factor of 16. We did it due to the limitations of our experimental setup: the maximum height of the mo-cap system was 3 meters. The actual flights were carried out at the altitudes of 12 meters, therefore, the resolution was reduced to recreate the real warehouse conditions. The flight-path of the UAV was measured by mo-cap as ground truth, along with IR-based localization from the UGR, and active perception localization. Fig. 9 shows a graph of the comparison of three trajectories with the real positions of barcodes. The green line shows the drone's flight-path with the IR-based localization for barcode detection. The violet line indicates ground truth obtained by the Vicon mo-cap system. The blue line represents the flight-path with the use of barcode coordinates for active perception localization, and red crosses indicate barcode positions.

Fig. 9 reveals that active perception localization is more accurate relative to the ground truth data and shows 1.38 times better accuracy on high altitudes than the IR-based localization system.

IV. Conclusion

In this letter, we have presented a CNN barcode detection-based system for UAV trajectory optimization implemented in a heterogeneous robot for autonomous warehouse stocktaking. The proposed solution has three significant advantages. Firstly, the flight-path correction using developed approach with CNN allowed us to detect and scan barcodes with higher precision compared with the standard OSBG flight trajectory, even with 50% overlap. Secondly, the active perception localization system makes the UAV's flight more stable and 1.38 times more accurate than the previously developed localization method. Thirdly, the proposed approach decreases the duration of an inventory process without any loss in barcode recognition percentage, which is crucial for a real warehouse stocktaking. Moreover, the proposed approach used standard objects of a warehouse environment (1D-barcodes) for UAV localization improvement.

The advantages mentioned above allow our robot to get global coordinates of each scanned barcode for warehouse planograms and further analytics.

References

[1] P. Thanapal, J. Prabhu, and M. Jakhar, "A survey on barcode RFID and NFC," *Mater. Sci. Eng. Conf. Ser.*, vol. 263, no. 4, 2017, Art. no. 042049.

[2] M. Maurer, J. Pestana, F. Fraundorfer, and H. Bischof, "Towards an autonomous vision-based inventory drone," Tech. Rep. 2018. [Online]. Available: https://github.com/glassechidna/zxing-cpp

[3] B. K. Sett, B. Sarkar, and A. Goswami, "A two-warehouse inventory model with increasing demand and time varying deterioration," *Scientia Iranica*, vol. 19, no. 6, pp. 1969–1977, 2012.

[4] "J. pons, dronescan," 2020. [Online]. Available: www.dronescan.co

[5] "Hardis group, eyesee," 2020. [Online]. Available: eyesee-drone.com

[6] "Ic barcode recommendations," 2020. [Online]. Available: https://www.theimagingsource.com/support/documentation/ic-barcode/tech_BarcodeRegion.htm

[7] Y. Choi, M. Martel, S. Briceno, and D. Mavris, "Multi-UAV trajectory optimization and deep learning-based imagery analysis for a UAC-based inventory tracking solution," in *Proc. AIAA Scitech Forum*, 2019, pp. 1569–1582.

[8] S. Barlow, Y. Choi, S. Briceno, and D. N. Mavris, "A multi-UAS trajectory optimization methodology for complex enclosed environments," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, Jun. 2019, pp. 596–604.

[9] E. H. C. Harik, F. Guerin, F. Guinand, J. F. Brethe, and H. Pelvillain, "Towards an autonomous warehouse inventory scheme," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Feb. 2017, pp. 1–8.

[10] T. M. Fernández-Caramés, O. Blanco-Novoa, I. Froiz-Míguez, and P. Fraga-Lamas, "Towards an autonomous industry 4.0 warehouse: A UAV and blockchain-based system for inventory and traceability applications in big data-driven supply chain management," *Sensors (Basel, Switzerland)*, vol. 19, no. 10, pp. 1–31, May 2019.

[11] M. Beul, D. Droeschel, M. Nieuwenhuisen, J. Quenzel, S. Houben, and S. Behnke, "Fast autonomous flight in warehouses for inventory applications," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3121–3128, Oct. 2018.

[12] W. Kwon, J. H. Park, M. Lee, J. Her, S.-H. Kim, and J.-W. Seo, "Robust autonomous navigation of unmanned aerial vehicles (UAVs) for warehouses' inventory application," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 243–249, Jan. 2020.

[13] H. Cho, D. Kim, J. Park, K. Roh, and W. Hwang, "2D barcode detection using images for drone-assisted inventory management," in *Proc. 15th Int. Conf. Ubiquitous Robots*, Aug. 2018, pp. 461–465.

[14] D. K. Hansen, K. Nasrollahi, C. B. Rasmusen, and T. B. Moeslund, "Real-time barcode detection and classification using deep learning," 2017. [Online]. Available: https://pjreddie.com/darknet/imagenet/

[15] L. Xu, V. R. Kamat, and C. C. Menassa, "Automatic extraction of 1D barcodes from video scans for drone-assisted inventory management in warehousing applications," *Int. J. Logistics Res. Appl.*, vol. 21, no. 3, pp. 243–258, May 2018.

[16] E. Ohbuchi, H. Hanaizumi, and L. A. Hock, "Barcode readers using the camera device in mobile phones," in *Proc. Int. Conf. Cyberworlds*, 2004, pp. 260–265.

[17] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2008.

[18] W. Xu and S. McCloskey, "2d barcode localization and motion deblurring using a flutter shutter camera," in *Proc. IEEE Workshop Appl. Comput. Vision*, 2011, pp. 159–165.

[19] C. Creusot and A. Munawar, "Real-time barcode detection in the wild," in *Proc. IEEE Winter Conf. Appl. Comput. Vision*, 2015, pp. 239–245.

[20] M. Donoser and H. Bischof, "Efficient maximally stable extremal region (mser) tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2006, vol. 1, pp. 553–560.

[21] R. Zhou and X. Guo, "A new method of angle-robust multiple 1d-barcodes detection," in *Proc. 2nd IEEE Int. Conf. Comput. Commun.*, 2016, pp. 433–438.

[22] M. Katona, P. Bodnár, and L. G. Nyúl, "Distance transform and template matching based methods for localization of barcodes and QR codes," *Comput. Sci. Inf. Syst.*, no. 00, pp. 20–20, 2019.

[23] F. Jurie and M. Dhome, "A simple and efficient template matching algorithm," in *Proc. 8th IEEE Int. Conf. Comput. Vision*, 2001, vol. 2, pp. 544–549.

[24] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018, *arXiv:1804.02767*.

[25] F. Cheng *et al.*, "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, Jul. 2018.

[26] H. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, and E. Galceran, "Continuous-time trajectory optimization for online UAV replanning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 5332–5339.

[27] A. T. Klesh, P. T. Kabamba, and A. R. Girard, "Path planning for cooperative time-optimal information collection," in *Proc. Amer. Control Conf.*, 2008, pp. 1991–1996.

[28] Y.-B. Chen, G.-C. Luo, Y.-S. Mei, J.-Q. Yu, and X.-L. Su, "UAV path planning using artificial potential field method updated by optimal control theory," *Int. J. Syst. Sci.*, vol. 47, no. 6, pp. 1407–1420, 2016.

[29] I. Kalinov, E. Safronov, R. Agishev, M. Kurenkov, and D. Tsetserukou, "High-precision UAV localization system for landing on a mobile collaborative robot based on an IR marker pattern recognition," in *Proc. IEEE Veh. Technol. Conf.*, Apr. 2019, vol. 2019, pp. 1–6.

[30] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 1271–1278.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.

[32] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep. 1, 2012.