UDC 004.932.72
DOI https://doi.org/10.32782/IT/2024-3-2

Kyrylo ANTOSHYN
PhD Student at the Department of Computer Science, Zaporizhzhia National University, 66, Universytets'ka Str., Zaporizhzhia, Ukraine, 69600, kyrylo.antoshyn@gmail.com
***ORCID:** 0009-0001-4166-5418*

Yuliia LYMARENKO
Candidate of Technical Sciences, Associate Professor at the Department of Software Engineering, Zaporizhzhia National University, 66, Universytets'ka Str., Zaporizhzhia, Ukraine, 69600, yuliia.lymarenko@gmail.com
***ORCID:** 0000-0002-1643-6939*

To cite this article: Antoshyn, K., Lymarenko, Yu. (2024). Rozrobka metodu na osnovi rozpiznavannia obiektiv dlia vyznachennia polozhennia liudyny v obmezhenomu prostori u realnomu chasi [Development of a method based on object detection for real-time person location detection in a confined space]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 3, 14–22, doi: https://doi.org/10.32782/IT/2024-3-2

DEVELOPMENT OF A METHOD BASED ON OBJECT DETECTION FOR REAL-TIME PERSON LOCATION DETECTION IN A CONFINED SPACE

Real-time person detection provides an opportunity to solve such a complex problem as person location detection in a confined space. The solution to this issue lies in the implementation of an effective method to localize a person inside a confined space (for example, inside the room) since outdoor positioning systems like GPS do not provide high accuracy indoors. Existing computer systems that solve this problem require specialized infrastructure: devices attached to the human body, sensors, etc. This approach is not cheap and does not provide a universal solution. A device that is present in almost every building is a camera. Many existing computer systems that analyze the video stream use Kinect depth cameras, which are outdated and require additional installation. There is a limited number of solutions that analyze video stream from an RGB camera in combination with computer vision methods for person localization. Therefore, research and development of a more effective method for the above-mentioned problem using computer vision is relevant.

***The aim** of the work is to develop a method of localizing a person in a confined space that is efficient in terms of speed and accuracy, which would use the video stream of the camera in combination with the computer vision method – object detection. The method should work on an NVIDIA Jetson Nano microcomputer (which is a relatively cheap and popular solution from NVIDIA) in real-time.*

***The methodology** for solving the problem is to leverage a deep neural network to detect the person in real-time and then use a perspective transformation algorithm to estimate the person's location. A person's location is the center point of the bottom edge of the bounding box transformed from the camera perspective in a way as if the camera was positioned directly above the floor. YOLOv4-tiny neural network model was trained on the COCO and Open Images datasets using the Darknet deep learning framework.*

***The scientific novelty** is that the method for person indoor localization was developed, which is based on the combination of a person detection method using a deep convolutional neural network and perspective transformation algorithm for further location estimation in a confined space. The proposed method is more versatile than known methods that use Kinect depth cameras. The proposed method can work on a microcomputer and estimate the location of several people in one pass with an average error of 23 cm and with a speed of 16 FPS, which is superior to the known alternative approaches.*

***Conclusions.** The problem of real-time person location detection in a confined space and means of solving it based on object detection using a deep convolutional neural network are studied. A neural network, based on the YOLOv4-tiny model, was trained using the COCO and Open Images datasets, and showed an accuracy of 55.1% and 71.4%, respectively. A method has been developed that uses a trained neural network to determine a bounding box around a person in the frame, and then determines its position using a perspective transformation algorithm: the method works on an NVIDIA Jetson Nano microcomputer with an average error of 23 cm and a speed of 16 FPS, processing a video stream from the RGB camera.*

***Key words:** person indoor localization, person detection, perspective transformation, deep learning, convolutional neural network, YOLO, NVIDIA Jetson Nano.*

Кирило АНТОШИН
аспірант кафедри комп'ютерних наук, Запорізький національний університет, вул. Університетська, 66, м. Запоріжжя, Україна, 69600
***ORCID:** 0009-0001-4166-5418*

Юлія ЛИМАРЕНКО
кандидат технічних наук, доцент кафедри програмної інженерії, Запорізький національний університет, вул. Університетська, 66, м. Запоріжжя, Україна, 69600
***ORCID:** 0000-0002-1643-6939*

Бібліографічний опис статті: Антошин, К., Лимаренко, Ю. (2024). Розробка методу на основі розпізнавання об'єктів для визначення положення людини в обмеженому просторі у реальному часі. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 3, 14–22, doi: https://doi.org/10.32782/IT/2024-3-2

РОЗРОБКА МЕТОДУ НА ОСНОВІ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ДЛЯ ВИЗНАЧЕННЯ ПОЛОЖЕННЯ ЛЮДИНИ В ОБМЕЖЕНОМУ ПРОСТОРІ У РЕАЛЬНОМУ ЧАСІ

Розпізнавання людини в режимі реального часу дає можливість вирішувати таку складну проблему як визначення положення людини в обмеженому просторі. Розв'язання даної задачі полягає в реалізації ефективного методу локалізації людини в замкнутому просторі (наприклад, всередині кімнати), оскільки системи позиціонування у відкритому просторі, такі як GPS, не забезпечують високої точності у приміщенні. Існуючі комп'ютерні системи, які вирішують дану проблему, потребують спеціалізованої інфраструктури: пристроїв, прикріплених до тіла людини, датчиків тощо. Такий підхід недешевий і не дає універсального рішення. Пристрій, який наявний практично в кожній будівлі – це камера. Переважна більшість існуючих комп'ютерних систем, які аналізують відеопотік, використовують камери глибини Kinect, які є застарілими та потребують додаткового встановлення. Існує обмежена кількість рішень, які аналізують відеопотік з RGB камери у поєднанні з методами комп'ютерного зору для локалізації людини. Отже, дослідження та розробка ефективнішого методу вирішення вищезазначеної проблеми з використанням комп'ютерного зору є актуальним.

***Метою роботи** є розробка ефективного за швидкістю та точністю методу локалізації людини в приміщенні, який би використовував відеопотік камери в поєднанні з методом комп'ютерного зору – розпізнавання об'єктів. Метод повинен працювати на мікрокомп'ютері NVIDIA Jetson Nano (який є відносно дешевим і популярним рішенням від NVIDIA) в режимі реального часу.*

***Методологія** вирішення проблеми полягає у використанні глибокої нейронної мережі для розпізнавання людини в режимі реального часу разом з алгоритмом перспективного перетворення для подальшої оцінки положення людини. Положення людини – це центральна точка нижньої сторони обмежувальної рамки, трансформована з перспективи камери таким чином, ніби камера розташована прямо над підлогою. Модель нейронної мережі YOLOv4-tiny була навчена на наборах даних COCO та Open Images за допомогою фреймворку глибокого навчання Darknet.*

***Наукова новизна** полягає в тому, що було розроблено метод локалізації людини в приміщенні, який базується на поєднанні методу виявлення людини за допомогою глибокої згорткової нейронної мережі та алгоритму перспективного перетворення для подальшого визначення положення в обмеженому просторі. Запропонований метод є більш універсальним за відомі методи, які використовують камери глибини Kinect. Запропонований метод може працювати на мікрокомп'ютері та визначати положення декількох людей за один прохід із середньою похибкою 23 см та швидкістю 16 FPS, що є кращим за відомі альтернативні підходи.*

***Висновки.** Досліджена проблема визначення положення людини в обмеженому просторі у реальному часі та засоби її вирішення на основі розпізнавання об'єктів з використанням глибокої згорткової нейронної мережі. Проведено навчання нейронної мережі на основі моделі YOLOv4-tiny з використанням датасетів COCO та Open Images, яке показало точність 55.1% та 71.4% відповідно. Розроблено метод, який використовує навчену нейронну мережу для визначення обмежувальної рамки навколо людини у кадрі, а після – визначає її положення з використанням алгоритму перспективного перетворення: метод працює на мікрокомп'ютері NVIDIA Jetson Nano із середньою похибкою 23 см та швидкістю 16 FPS, обробляючи відеопотік з RGB камери.*

***Ключові слова:** локалізація людини в приміщенні, розпізнавання людини, трансформація перспективи, глибоке навчання, згорткова нейронна мережа, YOLO, NVIDIA Jetson Nano.*

The urgency of the problem. The development of artificial intelligence makes our life more optimized. One of the tasks that we have successfully automated is real-time object detection. The possibility of person detection is the cornerstone for solving more complex problems, one of which is real-time person location detection in a confined space.

A computer system that effectively solves this problem can be implemented in various areas of our life. For example, we can create a notification system that will respond to the movement of people in locations that may be dangerous or prohibited: areas with harmful substances in enterprises, areas with confidential information in security agencies, spaces with valuable exhibits in museums, etc. Such notification systems can be extended to improve the quality of life of people with disabilities: the system will help people with visual impairments to successfully move around the house by providing voice prompts depending on their location. In addition, we can create recreation areas for children and adults of a completely new level: a system in a shopping and entertainment center will project images from a projector onto the floor and reproduce effects under people in the areas they walk.

To implement such systems, we need to develop a method for person indoor localization. Outdoor positioning technologies like GPS do not provide high accuracy indoors, because signals from satellites can be affected by surroundings like walls, roofs, tunnels etc. Many of the existing computer systems that solve this problem require specialized infrastructure for their work: devices attached to the human body, sensors, etc. This approach is not cheap and does not provide a universal solution. A device that is available in almost every modern building is a camera, the video stream from which we can leverage as input data.

Many existing computer systems that analyze the video stream use Kinect depth cameras, which are outdated and require additional installation. There is a limited number of solutions that analyze video stream from an RGB camera in combination with computer vision methods for person localization. Researchers in the field of artificial intelligence create not only accurate, but also fast and lightweight models of neural networks (for example, YOLOv4-tiny), which can be trained to detect objects on relatively inexpensive graphics processors (for example, NVIDIA GeForce GTX 1050) and used on microcomputers optimized for solving high-performance tasks (for example, NVIDIA Jetson Nano). This approach guarantees a much lower price for the hardware complex, as well as data security since all calculations take place locally.

Therefore, research and development of a more effective method for the above-mentioned problem using computer vision is relevant.

Analysis of recent research and publications.

An overview of real-time person location detection in a confined space problem.

Rainer Mautz conducts an overview of the problem of determining the location of objects, including people, in a confined space, and describes available technologies for its solution as well as shows areas in which the solution to the problem has practical value (Mautz, 2012). He states that the dominating technologies for positioning in outdoor environments, called Global Navigation Satellite Systems (GNSS), perform poorly within buildings. Researcher illustrates 13 technologies for indoor positioning: cameras, Wi-Fi, Bluetooth, and others. Based on his research, cameras provide high accuracy, but computer systems based on this technology can work at room level only.

Oussama Kerdjidj et al. provide a comprehensive overview of indoor localization problem with the focus on deep learning approaches (Kerdjidj, Himeur, Sohail et al., 2024). Researchers outline recent studies that use various architectures of deep neural networks to achieve robust indoor localization: convolutional neural networks (CNNs), recurrent neural networks (RNNs) or hybrid models. They state that deep learning-based methods are resilient against noise and missing data.

An overview of solutions to real-time person location detection in a confined space problem.

Adrian Cosma et al. provide an overview of solutions to the real-time person location detection in a confined space problem (Cosma, Radoi, Radu, 2018). They claim that all existing solutions can be divided into two groups: solutions that require a specialized infrastructure for their work and solutions that use the existing infrastructure such as wireless access points, surveillance cameras in buildings and inertial sensors in mobile devices. Researchers state that most of the recent solutions of the second group use smartphone sensors and Wi-Fi, however, systems that use video stream from cameras also exist. Camera-based systems use computer vision algorithms and do not require users to wear special sensors, which simplifies the use of such systems in cases where users are not well-versed in information technologies.

Many of the existing computer vision-based systems use depth cameras (RGB-D) to solve the problem. Huan Wang et al. developed a novel RGB-D camera-based indoor occupancy

positioning system called CIOPS-RGBD (Wang, Wang, Li, 2023). Researchers were using 4 Kinect cameras to take color and depth images simultaneously. The idea was to combine results taken from various cameras: data fusion and 3D reconstruction algorithm was designed and developed. They used OpenPose library to extract keypoints (shoulders and neck) that were aligned on the depth image. Their system achieves excellent accuracy within 20 cm in a scenario when people are sitting. However, it takes 5 seconds (0.2 FPS) to localize people (CPU: 9980, GPU: 2080TI), which is not enough for real-time. The main drawback of the system, according to its authors, is that the operation distance and perspective of the RGB-D sensor are limited: when used in larger spaces, more cameras are required.

Adrian Cosma et al. developed a computer system that uses a video stream from an RGB camera as input data and works on a device with limited resources (Cosma, Radoi, Radu, 2018). The system processes the video stream using a deep neural network to estimate a person's key points. The obtained data are used to determine the person's location in a confined space. The person's location is the midpoint between their legs, transformed from the camera's perspective relative to the floor. The average error of their system is 36 cm, and the speed is 6.25 FPS.

Ángel Carro-Lagoa et al. developed a computer system that consists of several microcomputers (edge devices) with connected cameras (Carro-Lagoa, Barral, González-López et al., 2023). Each microcomputer estimates a person's location using pose estimation and then sends this information to the Real-time Location System (RTLS) server. The server performs multicamera tracking of the detected person. The average error of their system is below 40 cm, and the speed is 2 FPS.

To estimate a person's location in a confined space using computer vision, it is required to solve two problems in succession. First, we need to detect the person in the frame. Then, we need to estimate the person's location from the perspective of the camera relative to the confined space (floor). To solve the first problem, it is effective to use a deep neural network. Since there are new versions of YOLO like YOLOv4, YOLOv4-tiny etc., we can leverage object detection method instead of keypoint detection. Keypoint detection method's disadvantage is the considerable number of frames in which the neural network cannot detect a person (Cosma, Radoi, Radu, 2018). To solve the second problem, we can leverage perspective transformation.

An overview of real-time person detection problem.

Object detection in the image is object localization with bounding box and object classification. Ross Girshick et al. in 2014 developed one of the first CNN-based methods for object detection – R-CNN (Girshick, Donahue, Darrell et al., 2014). The approach consists in generating approximately 2000 regions of interest in the image using a Selective Search algorithm. The warped regions are fed into a convolutional neural network (CNN) for features extraction. After that, Support Vector Machines (SVMs) are used for objects classification. The main drawback of their approach is that it takes 49 seconds to detect objects, because CNN needs to run for each region of interest. Two improvements of R-CNN were developed: Fast R-CNN (Girshick, 2015) and Faster R-CNN (Ren, He, Girshick et al., 2016). These methods detect objects in the image 2.3 seconds and 0.2 seconds (5 FPS) respectively, which is not enough for real-time. Accuracy of Faster R-CNN on the PASCAL VOC 2007 dataset is 73.2%.

Joseph Redmon et al. in 2016 developed a much faster, but less accurate detector called YOLO (You Only Look Once) (Redmon, Divvala, Girshick et al., 2016). YOLO uses a single neural network to predict bounding boxes and class probabilities in one evaluation. This eliminates the need for a detection pipeline that the R-CNN family methods use, and the system can be optimized end-to-end directly on detection performance. YOLO works at 45 FPS with 63.4% precision on the PASCAL VOC 2007 dataset, which is only 10% less than the Faster R-CNN method.

Alex Bochkovskiy et al. in 2020 presented YOLOv4 detector, which is the next generation of YOLOv3 (Bochkovskiy, Wang, Liao, 2020). The idea behind YOLOv4 is that researchers introduced new features to the YOLO model to improve accuracy like Weighted-Residual-Connections (WRC), Mish-activation and others. They achieved 43.5% precision on the COCO dataset at a speed of ~65 FPS on Tesla V100.

The logic behind selection of model, framework, datasets, and computer vision library for real-time person detection problem.

The aim of the work is that the developed method for indoor positioning should work on the NVIDIA Jetson Nano microcomputer in real-time. The YOLOv4-tiny model is a deep CNN, smaller version of YOLOv4, which can work on the microcomputer. The main advantage of YOLOv4-tiny is its speed: it works at a speed of 371 FPS on the NVIDIA GeForce GTX 1080 Ti GPU and achieves an accuracy of 40.2% on the COCO dataset.

YOLOv4-tiny outperforms other models that can work on microcomputers: MobileNetV3 (Howard, Sandler, Chu et al., 2019) and SqueezeNet (Iandola, Han, Moskewicz et al., 2016).

For model training we have chosen Darknet framework – an open-source deep learning framework written in C/C++ that is primarily leveraged to train and use YOLOv2, YOLOv3, YOLOv4 models, and their lightweight versions. Its initial developer was Joseph Redmon, the author of the original YOLO model. Alex Bochkovskiy, author of YOLOv4, continued his work by adding support for new layers, activation functions, ability to work on Windows and more (Bochkovskiy, 2020).

Common Objects in Context (COCO) dataset has been chosen for model training. COCO is a widely used large-scale dataset containing images of complex everyday scenes. It contains annotations for solving the following tasks: object detection, key points detection, pose estimation, object segmentation and generation of text descriptions for images. COCO includes 1.5 million objects of 80 classes (categories) and 200 thousand annotated images (Lin, Maire, Belongie et al., 2015).

Open Images (The Open Images Dataset V4, OID V4) dataset has been chosen for model training too. Open Images is a large-scale dataset containing about 9.2 million annotated images. It includes 19.8 thousand object classes for classification, 600 object classes for detection, and 57 object classes for visual relationship recognition. The images contain complex scenes with an average of 8 objects (Kuznetsova, Rom, Alldrin et al., 2020).

OpenCV library has been chosen to use trained YOLO model: Alex Bochkovskiy states that YOLOv4-tiny works at 773 FPS on the NVIDIA GeForce RTX 2080 Ti, if OpenCV is used, compared to 443 FPS if Darknet is used. OpenCV is an open-source computer vision library written in C++. OpenCV includes algorithms for image processing and transformation, functionality for working with video, as well as module for using deep neural networks (dnn module).

An overview of a person coordinate estimation in a confined space problem.

After the detector has detected a person in the frame, it is necessary to determine their coordinate (middle point of the bottom edge of the bounding box) relative to the confined space (floor) from the perspective of the camera. To solve the problem, we need to select 4 points on the floor from the camera perspective, 4 points of the destination image and calculate the matrix for perspective transformation. After that, we will be able to transform any point in the projection area from the perspective of the camera. Perspective transformation can be achieved using OpenCV (Shaikh, 2020).

The purpose of the article.

The purpose of the article is to present analysis of recent research regarding real-time person indoor localization problem as well as to demonstrate a developed method that effectively solves this problem in terms of speed and accuracy: the method works on the NVIDIA Jetson Nano microcomputer by processing RGB camera video stream in real-time and outperforms existing alternatives.

Presenting main material.

The developed method is based on solving two problems step-by-step: detect a person in real-time using deep convolutional neural network and then estimate person's location using perspective transformation algorithm. Below solutions to each problem are precisely described as well as comparison with similar methods is outlined.

Training YOLOv4-tiny models to detect people.

The training was conducted on a Windows laptop with Intel Core i7-8550U CPU and NVIDIA GeForce GTX 1050 GPU. To train the YOLOv4-tiny model, the Darknet framework and its dependencies (the CUDA Toolkit, the cuDNN and OpenCV libraries) were installed.

After setting up the training environment, images of people from COCO and Open Images datasets were prepared. The developer of the Darknet framework states that the size of the training set should be between 2000 and 10000 images. There is a general recommendation that the size of the validation set should be 20% of the training set. Therefore, 10000 training images and 2000 validation images were chosen from both datasets since the model's generalization capabilities improve as the as the size of the training set grows (when data is of high quality, obviously). The use of several datasets allowed us to compare the quality of trained models.

To prepare the COCO dataset, a Python script was developed: https://github.com/KyryloAntoshyn/person-location-detector/blob/master/training/download_coco_single_class_images.py. It uses JSON annotations and COCO API to download the required number of images of the Person class from the training and validation sets. An important part of the script is convert_and_write_annotations function that converts a COCO annotation to a YOLO format (all dimensions are relative to image width and height): <object-class> <x_center> <y_center> <width> <height> where <object-class> is the class identifier, <x_center> and <y_center> are the coordinates of the center of the bounding box, and <width> and <height> are its dimensions.

To prepare the Open Images dataset, Open Images Dataset v4 Toolkit was used. An important part of the toolkit is convert_annotations.py module that converts an Open Images annotation to a YOLO format.

To train the model, training files were prepared according to recommendations of YOLOv4-tiny author. It is worth noting that the transfer learning was carried out in our scenario: the weights of the convolutional layers of the previously trained network were used: yolov4-tiny.conv.29. This approach allows to speed up the learning process and increase the probability that the neural network will learn to effectively solve the given problem.

Optimal training parameters and model architecture were chosen based on the problem being solved: neural network should detect single class (Person) as well as work on the NVIDIA Jetson Nano microcomputer in real-time. A file with the training parameters and configuration of the neural network, a file with the names of the classes of objects on which we train the neural network, a file with the paths to the corresponding files and directories necessary for training can be found at (separate files were created for both COCO and Open Images datasets): https://github.com/KyryloAntoshyn/person-location-detector/tree/master/training.

Darknet framework requires files with relative paths to the training and validation images. To create these files automatically, a Python script was developed, placed in the scripts directory, and executed (the name of the dataset directory is given as an input): https://github.com/KyryloAntoshyn/person-location-detector/blob/master/training/generate_dataset_images_relative_paths.py.

Analyzing trained YOLOv4-tiny models that detect people.

Weights with which models demonstrated the highest accuracy on the validation dataset can be found at: https://github.com/KyryloAntoshyn/person-location-detector/tree/master/person_location_detector/detection_models. These weights are used in the developed method.

Figure 1 shows the graph of YOLOv4-tiny model training on the COCO dataset.

The graph shows that the final error (in blue) of the neural network is 1.07, and the accuracy (in red) is 55.1%. It is clearly visible that at 1000 iterations an accuracy of 41% was obtained, at 3500 iterations – 54%, then it dropped to 52% and only at the end of training we got 55.1%. When the accuracy of the model drops, an overfitting takes place, a process when the model detects objects well on the training set but begins to detect poorly on the validation set. It can be concluded that training for 3500 batches is optimal for this model and dataset.

Figure 2 shows the graph of YOLOv4-tiny model training on the Open Images dataset.

The graph shows that the final error of the neural network is 0.91, and the accuracy is 71.4%. It is

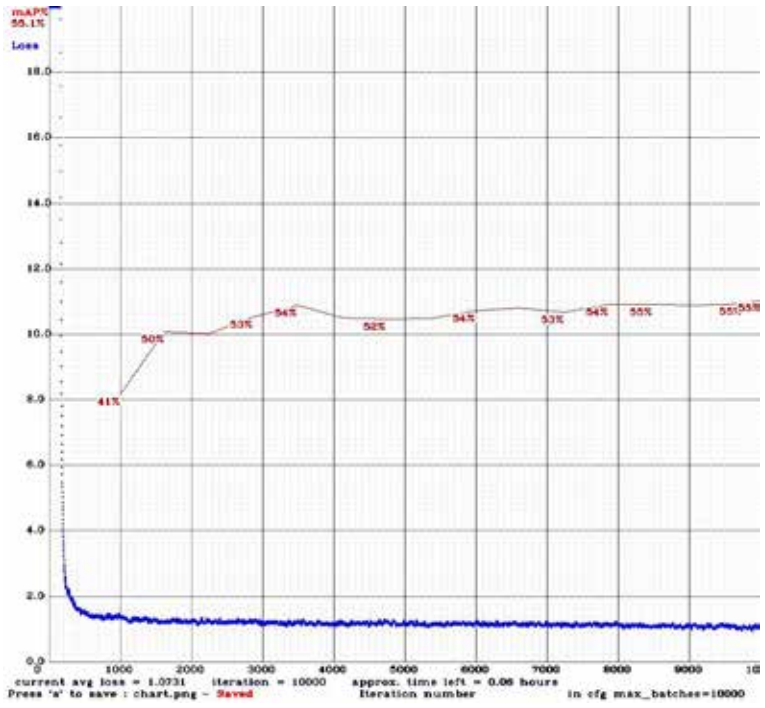

Fig. 1. The graph of YOLOv4-tiny model training on the COCO dataset

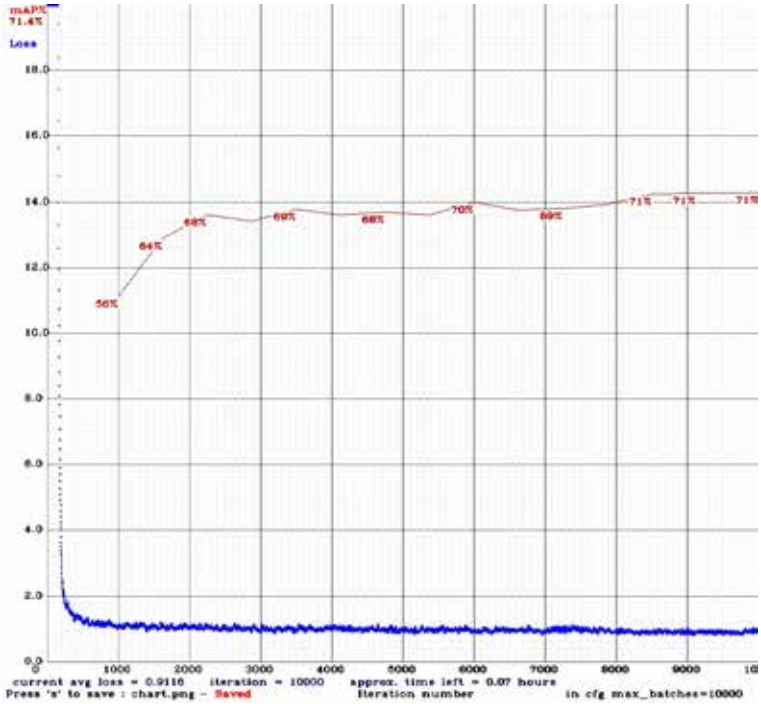

Fig. 2. The graph of YOLOv4-tiny model training on the Open Images dataset

clearly visible that at 1000 iterations an accuracy of 56% was obtained, at 3500 iterations – 69%, then it dropped to 68% and only at the end of training we got 71.4%. It can be concluded that training for 3500 batches is optimal for this model and dataset too.

At first glance, the model trained on the Open Images dataset should be more accurate due to higher precision obtained during training. However, after models' evaluation using the Oxford Town Center video, it was concluded that both models work with the same accuracy. One of the reasons is that the COCO dataset contains images with more complex scenarios, so it is more difficult for the neural network to perform generalization. Another reason is the presence of significant noise in the validation images since pre-processing of the datasets was not performed.

The training images of both datasets were checked using the LabelImg tool. The COCO dataset was found to contain annotations for a group of people, and Open Images for non-real people: mannequins, video game characters, toys, etc. To solve both problems it is required to manually remove such annotations and images. The best way, obviously, is to collect and annotate 2000-10000 images at the location where the system will be installed.

Any of the trained models can be used in the developed method, but it is required to evaluate which of them works better at the location where the system will be installed. To improve accuracy, we can try to experiment with some recommendations of YOLOv4-tiny author, some of which are: set random=1 in each yolo layer to change input image size during training randomly, increase the size of neural network to width=608 and height=608, validate quality of annotations etc.

Estimating the location of detected people.

The trained neural network model is used in the developed computer system that combines object detection approach with perspective transformation algorithm to estimate person location in the camera stream.

First, we initialize 3×3 perspective transformation matrix using OpenCV: https://github.com/KyryloAntoshyn/person-location-detector/blob/master/person_location_detector/services.py#L289. 4 pairs of points are used: coordinates of the projection area from the perspective of the camera and coordinates of the actual projection area on the floor.

Then, transformed coordinate $(\hat{x},\hat{y})$ is calculated using Expression 1, where $M_{11},M_{12},\ldots,M_{ij}$ are the elements of already initialized perspective transformation 3×3 matrix, (x,y) is the person coordinate from the camera point of view. Implementation can be found at: https://github.com/KyryloAntoshyn/person-location-detector/blob/master/person_location_detector/services.py#L327.

$$(\hat{x},\hat{y})=\left(\frac{M_{11}\times x+M_{12}\times y+M_{13}}{M_{31}\times x+M_{32}\times y+M_{33}},\frac{M_{21}\times x+M_{22}\times y+M_{23}}{M_{31}\times \boldsymbol{x}+M_{32}\times y+M_{33}}\right) \quad (1)$$

Accuracy and speed of the developed method.

Table 1

Comparison of speed of the developed method when using different devices

Device	Speed (FPS)
Training PC (NVIDIA GeForce GTX 1050)	83
NVIDIA Jetson Nano	16

The error is measured as the distance between the determined point (marked in green) and the actual point (marked in blue).

Table 2

Developed method error in various scenarios

Scenario	Error (cm)
A person is facing the camera	15
The person's back is turned to the camera	30
Part of the human body is covered by another object	25

Figure 3 shows the scenario when a person is facing the camera.

From the results obtained, for the given scenarios, the average error is 23 cm. The accuracy of the developed method primarily depends on the accuracy of trained neural network models that detect people.

The bounding box approach has two drawbacks. First, the frame does not always fit tightly to the human body, which gives an error when determining position. Then, a person can put one leg back and then its position will be determined relative to the front leg, which is also not completely accurate. These issues can be resolved using keypoints detection approach that was leveraged by Adrian Cosma et al., Ángel Carro-Lagoa et al. and Huan Wang et al. However, keypoints detection approach requires more computing resources and often does not detect a person in the frame, therefore, can be a bottleneck if the method should work on microcomputer.

To increase accuracy, we can try to combine several methods for person detection: object detection with pose estimation, combine results from several cameras set at different angles, create

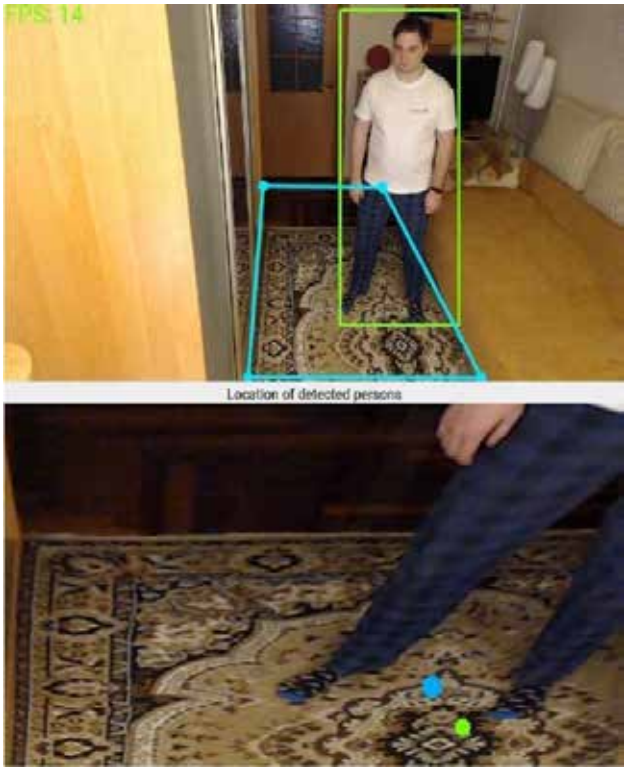

Fig. 3. Scenario when a person is facing the camera

our own dataset at the location where the system will be installed, consider using wide-angle camera viewing directly onto the floor. In fact, most of the areas of application of the developed method described in this work do not require determination of a person's location with perfect accuracy and, therefore, the approach with the bounding box is more optimal.

Conclusions. Indoor localization problem has been investigated: advantages and disadvantages of existing technologies as well as modern computer vision approaches to its solution. An effective method in terms of speed and accuracy that uses a camera video stream in combination with object detection and perspective transformation approaches has been developed. The method works on the NVIDIA Jetson Nano microcomputer in real-time with a speed of 16 FPS and accuracy of 23 cm. Therefore, it outperforms existing alternatives in terms of speed and accuracy.

BIBLIOGRAPHY:

1. Mautz R. Indoor Positioning Technologies. Zurich:Institute of Geodesy and Photogrammetry, 2012.

2. Kerdjidj O., Himeur Y., Sohail S. S., Amira A., Fadli F., Atalla S., Mansoor W., Copiaco A., Gawanmeh A., Miniaoui S., Dawoud D. W. Uncovering the Potential of Indoor Localization: Role of Deep and Transfer Learning. *IEEE Access*. 2024. Вип. 12. С. 73980–74010.

3. Cosma A., Radoi I. E., Radu V. CamLoc: Pedestrian Location Detection from Pose Estimation on Resource-constrained Smart-cameras. 2018.

4. Wang H., Wang G., Li X. An RGB-D camera-based indoor occupancy positioning system for complex and densely populated scenarios. *Indoor and Built Environment*. 2023. Вип. 32, № 6. С. 1198–1212.

5. Carro-Lagoa Á., Barral V., González-López M., Escudero C. J., Castedo L. Multicamera edge-computing system for persons indoor location and tracking. *Internet of Things*. 2023. Вип. 24.

6. Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014.

7. Girshick R. Fast R-CNN. 2015.

8. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2016.

9. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. 2016.

10. Bochkovskiy A., Wang C.-Y., Liao H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020.

11. Howard A., Sandler M., Chu G., Chen L.-C., Chen B., Tan M., Wang W., Zhu Y., Pang R., Vasudevan V., Le Q. V., Hartwig A. Searching for MobileNetV3. 2019.

12. Iandola F. N., Han S., Moskewicz M. W., Ashraf K., Dally W. J., Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. 2016.

13. Yolo v4, v3 and v2 for Windows and Linux. URL: https://github.com/AlexeyAB/darknet (дата звернення: 23.08.2024).

14. Lin T.-Y., Maire M., Belongie S., Bourdev L., Girshick R., Hays J., Perona P., Ramanan D., Zitnick C. L., Dollár P. Microsoft COCO: Common Objects in Context. 2015.

15. Kuznetsova A., Rom H., Alldrin N., Uijlings J., Krasin I., Pont-Tuset J., Kamali S., Popov S., Malloci M., Kolesnikov A., Duerig T., Ferrari V. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. 2020.

16. OpenCV Perspective Transformation. URL: https://medium.com/analytics-vidhya/opencv-perspective-transformation-9edffefb2143 (дата звернення: 23.08.2024).

REFERENCES:

1. Mautz, R. (2012). Indoor Positioning Technologies. Zurich:Institute of Geodesy and Photogrammetry,

2. Kerdjidj, O., Himeur, Y., Sohail, S. S., Amira, A., Fadli, F., Atalla, S., Mansoor, W., Copiaco, A., Gawanmeh, A., Miniaoui, S., Dawoud, D. W. (2024). Uncovering the Potential of Indoor Localization: Role of Deep and Transfer Learning. IEEE Access. Vol. 12. C. 73980–74010.

3. Cosma, A., Radoi, I. E., Radu, V. (2018). CamLoc: Pedestrian Location Detection from Pose Estimation on Resource-constrained Smart-cameras.

4. Wang, H., Wang, G., Li, X. (2023). An RGB-D camera-based indoor occupancy positioning system for complex and densely populated scenarios. Indoor and Built Environment. Vol. 32, № 6. С. 1198–1212.

5. Carro-Lagoa, Á., Barral, V., González-López, M., Escudero, C. J., Castedo, L. (2023). Multicamera edge-computing system for persons indoor location and tracking. Internet of Things. Vol. 24.

6. Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation.

7. Girshick, R. (2015). Fast R-CNN.

8. Ren, S., He, K., Girshick, R., Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.

9. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection.

10. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection.

11. Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., Hartwig, A. (2019). Searching for MobileNetV3.

12. Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size.

13. Yolo v4, v3 and v2 for Windows and Linux. Retrieved from: https://github.com/AlexeyAB/darknet (дата звернення: 23.08.2024).

14. Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., Dollár, P. (2015). Microsoft COCO: Common Objects in Context.

15. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., Ferrari, V. (2020). The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale.

16. OpenCV Perspective Transformation. Retrieved from: https://medium.com/analytics-vidhya/opencv-perspective-transformation-9edffefb2143 (дата звернення: 23.08.2024).