YOLOv4 for Urban Object Detection: Case of Electronic Inventory in St. Petersburg

Ebrahim Najafi Kajabad, Petr Begen, Boris Nizomutdinov, Sergey Ivanov
ITMO University
St. Petersburg, Russia
{e.najafi, begen, boris, svivanov}@itmo.ru

Abstract—**The paper presents the results of preparing a labeled dataset from open sources for 11 object classes and the analysis of two well-known object detection methods in the task of urban electronic inventory in Saint Petersburg in Russia under the concept of Smart City methods and technologies. We proposed YOLOv4 for urban object detection such as Windows, Doors, Adv Billboards, Ramps, etc. To do that the first step is data collection from the environment, and data augmentation techniques are employed to generate data. Then the transfer learning method is used to train our dataset with both algorithms YOLOv3 & YOLOv4 and finally the NMS method is used to remove overlaps bounding boxes. To evaluate the performance of both methods RMSE used as a metric. The YOLOv4 method showed better results in object detecting and classifying than YOLOv3 in total and in the context of each class. Based on RMSE metric formula average classification loss after the training model for YOLOv3 is 0.66 and against for YOLOv4 is 0.33. Using YOLOv4 helped us to develop the first version of web-service for automated urban object detection and recognition in real-time that can be scaled and distributed to other districts of the city.**

I. Introduction

Modern cities are developing faster than ever before and though each city has a unique structure and environment they face similar environmental, economic, and social challenges such as climate adaptation, economic problems, or population growth through global migration and urbanization [1], [2]. Using information technologies and their implementation for solving urban problems and management in certain areas of the urban economy, which has been termed as “Smart City”, is becoming widespread and popular worldwide.

The goal of the “Smart city” is not only digital transformation and automation of processes, but it is also a comprehensive increase in urban infrastructure efficiency. In this paper, the Petrogradsky administration district in Saint-Petersburg is chosen as a specific case for research in the urban field. Some problematic areas of city development in St. Petersburg are related to the state of the urban environment and its elements [3], such as the urban environment (cleaning, local improvement, state of backyards and adjacent areas), the state of vehicles, and the functioning of the transport system, the processes of city inventory (state of houses, illegal and questionable advertising), etc. These processes are executed by a human in most cases and rather do not have automation tools, which can optimize such routine activity and increase efficiency. Otherwise even having an electronic tool (a portal) for carrying out, for instance, city inventory still demands huge people's efforts to maintain the system in the actual state by making a thousand photos of objects and manually checking information about them. The application of artificial intelligence technologies in public administration has an increasingly strong position on today's agenda [4], [5].

The toolkit of Artificial Intelligence (AI) uses principles and approaches similar to human intelligence allowing for automatic processing of significant amounts of data, which provides a more rapid and relevant solution to state management problems [6]. In Russia, the improvement of quality and efficiency of public administration, social area development, critically dependent on the digital economy formation was specifically emphasized in the framework of the Strategy of information society development in the Russian Federation for 2017–2030, approved by the Decree of the President of the Russian Federation and the national program “Digital economy” approved in July 2017 and National Strategy for the development of Artificial Intelligence in the Russian Federation for the period up to 2030.

In this paper, the authors consider issues of using AI tools such as deep learning and computer vision method, e.g. object detection, in a process of urban electronic inventory in St. Petersburg. AI tools are aimed to solve several problems and shortcomings of inventory process made by a human, for instance, a routine process of observing urban areas by each worker and manual data collection about city objects and elements. Such designed and the developed tool can be scaled to other districts in the city or can be an example for other cities that are going to implement the “Smart city” concept and do effective urban inventory. The “smart” classifier can determine the main objects (elements) of the urban economy from a photo image (e.g. doors, windows, entrance visors, ramps, etc.), group certain elements with dependencies in order of nesting (for example, Krasnogvardeysky district → Belorusskaya street → house 6 → entrance group → door → code lock).

As the main approach of this research is to develop a “smart” classifier based on Artificial Intelligence tools particularly for urban object detection by using the YOLOv4 algorithm. Data augmentation is used to improve the size of the dataset. And Transfer Learning (TL) is applied on pretraining weight which already has been trained with an MS COCO dataset to improve the performance of detection.

This research is arranged as follows. Section 2 describes the related works. Part 3 talks about the dataset and proposed method. Section 4 explains the experimental results and finally, we display a discussion and conclusion in section 5.

II. LITERATURE REVIEW

Artificial Intelligent (AI) systems and Machine Learning (ML) have emerged as significant tools in various areas such as industry, robotics, medicine, transportation, business, and technologies over the past few years. Deep learning and computer vision approach provide convenient tools to obtain expansions in an industrial space like object detection, object tracking, improve safety, and security [7], [8], [9].

Generally, CNN (convolutional neural network) is the most representative model of deep learning. It has been widely applied to many applications, such as image super-resolution reconstruction, image classification, face recognition, pedestrian detection, and video analysis. But also, various network architectures such as AlexNet, VGG [10], SSD [11], YOLO [12], and R-CNN [13], have been extensively studied to develop performance and accuracy. Some various algorithms and methods already have been worked to detect objects. Here we survey related approaches that already suggested.

In [14] presented computer vision techniques to detect doors in the environment, they proposed a method to detect doors based on door edges and corners instead of using color or texture. In [15] introduced the Mask-RCNN method to extract important features of the windows and doors from aerial texture files of CityGML models. Two different databases are utilized for training to improve performance. the DBSCAN clustering was used to correct the results and also to improve the predictions using the texture coordinates from the 3D CityGML. This technique works well but the Mask-RCNN method is not quite fast to detect objects in real-time as compared to the YOLO family. Authors in this [16] paper presented a new method to detect windows and evaluate their parameters in 3D LiDAR points clouds which they collected from the mobile terrestrial data system from the urban environment. the principal object of this procedure is to connect information from both symmetrical and temporal correlations by ANOVA estimation. It can help to complete missing features because of occlusion. By [17] proposed a method based on CNN which can be able to detect doors, cabinets, and handles. CNN implemented to detect the ROI area from the related images. Segmentation techniques based on K-means clustering proposed to distinct door or cabinet from the handle.

In [18] presented a method based on Deep Convolutional Neural Network (DCNN) for advertisement billboard detection in the city environment. In this work, AlexNet's (DCNN) utilized as a pre-training method and to improve the performance of the model to detect billboards used the transform learning techniques. Detecting advertisement billboards in the video frame was proposed by [19] in which the ADNet used as a deep neural network architecture which includes pre-training weights of VGG network and modified the last layers for training. This model used positive and negative images from advertising billboards for training and was able to classify the billboards with an accuracy of 0.94 percentages. Unsupervised feature learning network presented by [20] to detect house numbers from street photos by using a dataset that collected from Google Street View images. The dataset was included 600,000-digit images. In [21] YOLOv3 has been proposed to detect stairs by the robot. In this work, transfer learning is utilized to detect stairs in real-time. The dataset included 848 stair images for testing and training the network.

III. DATASET AND OBJECT DETECTION METHODS CHOOSING

A. Data Collection

Having custom data for most computer vision tasks such as classification, segmentation, and object detection plays a significant role in the industrial part and it is always challenging to prepare enough data for training. Although there are many different online datasets to collect data, however, is still challenged in matching the precision of human perception. Deep learning algorithms like CNN or YOLO are largely reliant on big data to prevent overfitting. To train our own custom object detection model we need to have the training and validation sets of images with their ground truth bounding boxes and labels. In this research, transfer learning was utilized to train the dataset on the YOLOv3 and YOLOv4.

For the correct work in the classification task, we take under the consideration the Classifier of the internet portal “Passportization of improved facilities in St. Petersburg” that includes thousands of urban objects and elements. In the context of our research, we chose 11 of the main ones (we focused only on the main ones for the first pilot of our research) such as: Windows, Doors, House numbers, Entrance Visors, Ramps, Balconies, Billboard Advertisements, Door locks, Garbage boxes, Benches and Stairs.

In this research due to the lack of enough data for some special objects such as Ramps, Entrance Visors, and Door locks, we have used three ways to collect and improve the size of our dataset. Firstly, we utilized Open Images dataset v6 which is one of the largest datasets that includes 600 categories with about 9 million images with object bounding boxes, object segmentation masks, and annotation. Secondly, we collected images via internet searching and manually annotated them. The annotation technique manually creates regions in an image and assigns a label. Finally, the image data augmentation technique utilized that better deep learning models can be built using them by enhances the size of the dataset by geometric transformations, cropping images, random rotating images [22]. Generally, our dataset includes 11 different classes with 13,315 images in total.

B. Data Augmentation

Deep convolutional neural networks are excessively dependent on big data to avoid overfitting. As a result, data collection is one of the most crucial parts of machine learning. In a recent development in deep learning models have been mostly assigned to the amount and variety of data gathered in recent years. Also, many application domains suffering from access to big data. So, data augmentation is an approach that makes able to significantly raise the variety of the data available for training models, without practically fetching new data to overcome the problem of fewer data [22]. Cropping, padding, horizontal flipping, geometric transformation image are data augmentation techniques in which commonly used to train large neural networks. The geometric transformation includes image translation, rotation, and zoom. Image translation means that changing the position of pixels on the image and moving the image to the new position.

C. Transfer Learning (TL)

In this experiment, we have used a transfer learning method for training our dataset on YOLOv3 and YOLOv4 networks. For this purpose, we are applied pretraining weight which already has been trained with an MS COCO dataset to accomplish urban economy object detection.

Transfer learning (TL) is a method to improve a learner from one area by transferring data from a related area [23]. In other words, it is an approach that initializes the CNN model parameters which can be utilized for the related task that already trained on the other similar challenges. TL allowing the use of pre-trained models directly, as feature extraction preprocessing, and integrated into entirely new models. In transfer learning, one or more hidden layers from the trained model will be used in a new model to train on the problem of interest. This technique has the benefit of reducing the training time for a neural network model and result in high accuracy with a lower error.

D. Brief Review of YOLOv3&v4 Architectures

The YOLO family is one of the most significant algorithms in the way of object detection, the principal benefits behind this algorithm are speed and accuracy which makes it significantly important than the CNN family. YOLO algorithm can be able to predict the bounding boxes and the class probabilities in one shot step instead of the CNN family which works based on two-shot steps.

The input of the YOLO network is a fixed RGB image of 416*416. YOLO family architecture is different from each other, for example, YOLOv3 mainly consists of 106 layers, of which 75 layers are convolutional layers and the other 31 layers are included shortcut, route, upsample, and YOLO layers. For detecting objects, YOLOv3 used 3 different scales at layer 82, 94, and 106. These scales help the network to extract important features like Feature Pyramid Network (FPS), which combines low-resolution, semantically strong features with high-resolution, semantically weak features via a top-down pathway and lateral connections [24]. On the COCO dataset, each scales include 3 boxes, as a result, the output tensor is N*N*[3 boxes * (4 bounding box + 1 objectness prediction + C number of classes predictions)], YOLOv3 still uses K-means clustering to determine better bounding box priors [25], [26].

In YOLOv4 architecture suppose that a model with a larger number of convolution layers 3*3 and a larger number of values should be selected as the main structure. Where the YOLOv4 with Cross-Stage-Patial-connections CSPDarknet53 contains 29 convolution layers 3*3, a 725*725 size, and 27.6 M parameters. In the YOLOv4 PANet path-aggregation neck used as the method for increasing the receptive field and for various backbone levels for different detector levels instead of using FPN which was used in YOLOv3. The CSPDarknet53 backbone model shows higher accuracy in terms of object detection and classifier than other models like CSPResNeXt50.

In YOLOv4 utilized different new features such as YOLOv3 (anchor-based) head as the architecture of YOLOv4, Weighted-Residual-Connections (WRC), CSP, Cross-mini-Batch Normalization (CmBN), Mish activation function and Mosaic data augmentation and other methods which by combining some of them this network was able to obtain state-of-the-art results 43.5 for the MS COCO dataset at a real-time speed of 65 FPS on Tesla V100 [27].

Generally, Object detection techniques is combination of several components such as: Input image, Backbone which is a network to extracts feature map from image by using different method like VGG16, Resnet-50, ResNext50 etc. The Neck and head which are sub-sets of the backbone to raise the feature distinguish and robustness by using FPN, PAN, RFB and other methods and Head that is responsible for the prediction by using detector algorithms like YOLO, SSD, RCNN, FRCNN etc. The YOLOv4 uses CSPDarknet53 neural network as a backbone and Spatial pyramid pooling (SPP) block as Neck due to it can be able to separates out the most remarkable context features and nearly will not causes to decrease of the network operation speed. In YOLOv4 PANet uses as a method for parameter aggregation from different backbone levels for different detector levels vs YOLOv3 which used the FPN method. and Finally, YOLOv3 utilized as the Head of network for YOLOv4. YOLOv3 is quite popular, powerful, and fast. However, YOLOv4 is more robustness in terms of speed and performance [27], [30].

IV. Experimental Results

Darknet is a framework to train neural networks, it is open source and written in C/CUDA and serves as the basis for YOLO. After installing Darknet53 on Google Colab, the dataset trained on both YOLOv3 and v4 algorithms to evaluate the performance of both algorithms. For training our data we used the GPU system to improve the speed and performance of training, so Google Colab Pro is utilized for this purpose. It is a free online cloud-based Jupyter notebook environment that allows us to train our machine learning and deep learning models on GPUs. We have used the Tesla P100-PCIE GPU system with 24GB memory on Google Colab. To implement the urban object detection system Python 3.5 with OpenCV 3.4 library is used.

As explained before, we used the data augmentation method to improve the size of the data set and improve the performance of the training system. For this purpose, rotate, crop, pad, and zoom techniques have been used as data augmentation for three objects such as Ramps, Door Locks, and Doors. The below images Fig. 2 shows the samples of data augmentation in different ways. Augmentation operations allow users to increase the size of data significantly. For example, only 40 images from the door lock were available, however by using data augmentation was able to produce 250 images for training that can be very beneficial to improve the accuracy of the detection system.

For this aim transfer learning method is utilized to train our data on pre-training weight which already was trained by the MS COCO dataset. The configuration of default hyperparameters changed as follows: the size of the image is 416*416 and the training steps are 22000 iterations, the initial learning rate is 0.01 and then multiply with a factor 0.1 at the 80% steps and the 90% steps of total iterations respectively. All the networks use a single GPU to execute multi-scale training

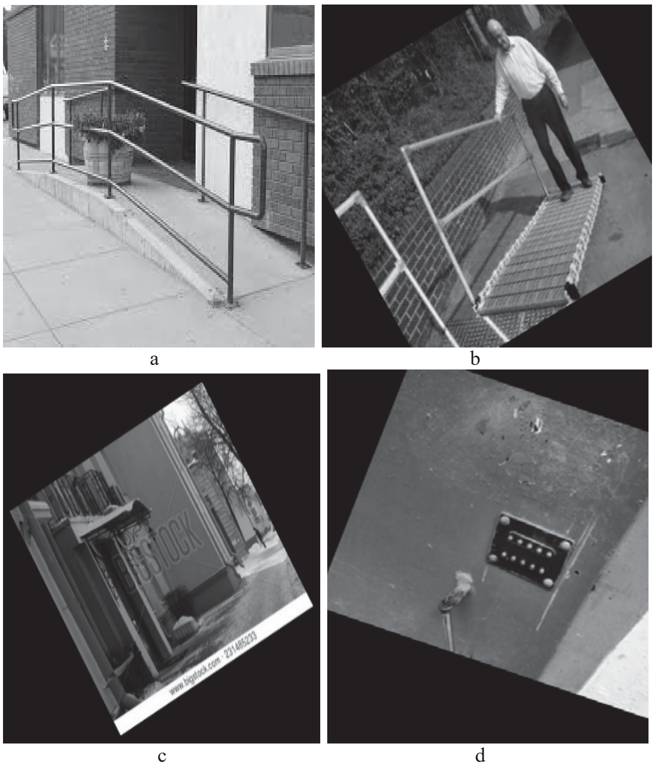


Fig. 1. Examples of different image augmentation. (a) The original image. (b) Result of image rotation. (c) Results of image zoom. (d) Result of image translation and rotation.

in the batch size of 64 means that the system will be using 64 images for every training step, and mini-batch size of 16 or 32 depend on the power of GPU memory limitation where the batch size will be divided by mini-batch size to reduce GPU VRAM requirements.

The number of classes in the YOLO layers is 11 and the size of the filter is 48 which is calculated by this formula filters = (classes + 5) * 3, where 5 includes 4 bounding boxes plus 1 object prediction and 3 boxes. After training our data with both YOLOv3 and v4 algorithms, we obtained the average losses of 0.66 and 0.33, respectively. The loss function estimates how closely the distribution of predictions made by a model matches the distribution of target variables in the training data. Although both algorithms achieved lower average losses, however, YOLOv4 was able to estimate and match better the target variables in the training data than YOLOv3.

Non-maximum suppression (NMS) is an important step in object detection work. Where object detection methods that work by sliding windows techniques usually have multiple overlap windows with high confidence scores that are near to the location of the objects. By removing neighborhood windows that have similar scores are considered as candidate regions. So, it makes sure that in an object detection system, an object is detected only once [28]. As a result, we utilized the NMS method to remove overlap windows from a detected object with the 0.2 threshold value.

We evaluated our approach both quantitatively and qualitatively. For this purpose, we collected 10 images from each separate class from the environment and thrown each image into the system to measure the performance and the accuracy of the urban object detection system.

TABLE I. RESULTS OF OBJECT DETECTION BASED ON YOLOV3 ALGORITHM

YOLOv3				
Object Name	**Detected**	**Not Detected**	**Total Objects**	**RMSE**
Window	136	164	300	21.55
Door	6	8	12	1.0
House_number	4	9	13	1.22
Entrance_Visor	3	7	10	0.83
Ramp	6	4	10	0.63
Balcony	18	19	37	2.07
Billboard_Adv	8	3	11	0.77
Door_lock	3	7	10	0.83
Garbage_box	13	7	20	0.94
Bench	11	3	14	0.70
Stairs	7	3	10	0.54

TABLE II. RESULTS OF OBJECT DETECTION BASED ON YOLOV4 ALGORITHM

YOLOv4				
Object Name	**Detected**	**Not Detected**	**Total Objects**	**RMSE**
Window	183	108	300	14.48
Door	8	4	12	0.77
House_number	5	8	13	1.05
Entrance_Visor	3	7	10	0.83
Ramp	5	5	10	0.70
Balcony	24	13	37	1.37
Billboard_Adv	10	1	11	0.44
Door_lock	3	7	10	0.83
Garbage_box	16	4	20	0.63
Bench	13	1	14	0.31
Stairs	9	1	10	0.31

In this assessment, we have manually counted the number of objects which were detected by the system and then considered several factors such as True positive detection (TP), False positive detection (FP), and object not detected. In assessing the accuracy, the root means of the squared difference between the estimated values and what is estimated are used as a metric to evaluate the number of objects detected. The estimated number of the object have constantly been aggregated and calculated mean square error according to the RMSE formula (1).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}^{n}(A_i - \widehat{D}_i)^2} \quad (1)$$

Where n represents the number of total images for each class and A_i shows the number of the actual object in each image and $\widehat{D}_i$ is the number of the detected object respectively [29].

The above two tables display the comparison results of both YOLOv3 and v4 algorithms. Each class of object is checked by 10 images for evaluation. Analysis results show that the performance and the accuracy of YOLOv4 were better as compare to YOLOv3 and during the test, it was able to detect more objects than YOLOv3, especially when the objects are located in the far distance in the image. However, in some classes, both algorithms were not able to do well detect objects, especially when the objects are quite small, or we have not to feed enough data for training such as Lock door and Entrance Visor.

Fig. 2 represents the output of error estimation based on the RMSE method for YOLOv3 and v4 algorithms that measures

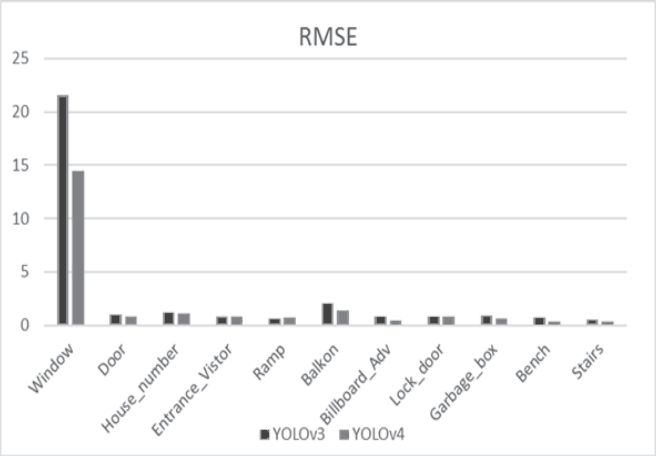
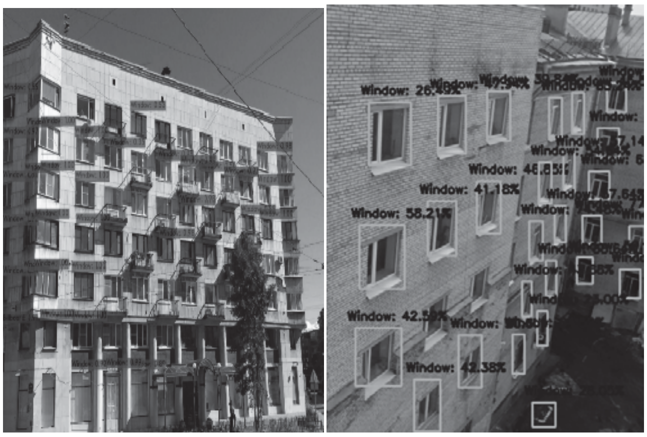


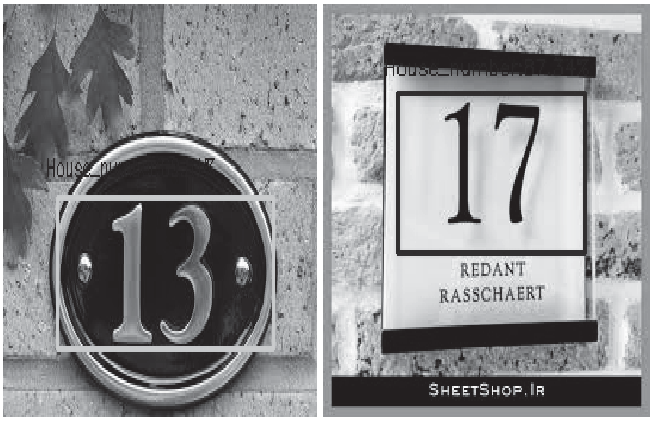
Fig. 2. The comparison RMSE results of 10 classes from YOLO v3 and v4

the number of objects which were detected and or not detected for each class. The results illustrate that YOLOv4 was able to detect more objects in each class and it obtained a lower error estimation value for most of the classes. The below images Fig. 3 demonstrates the results of object detection algorithm YOLOv4 for some of the classes. The YOLO model works well with big objects, however, it is not quite strong to detect small objects in the image, for training we used 416*416 image size but to detect small objects we have to increase the size of images which takes a long time for training and the speed of detection will decrease.

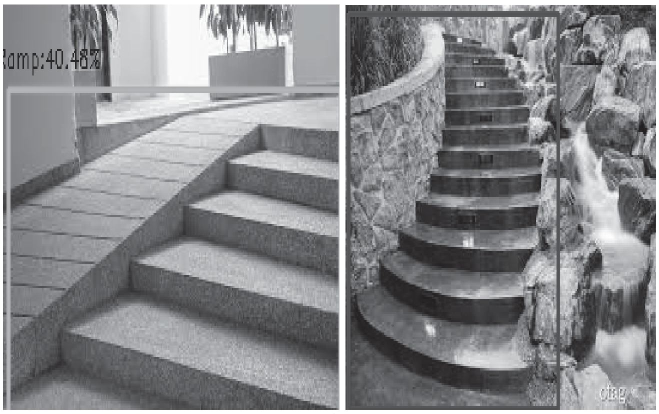


a

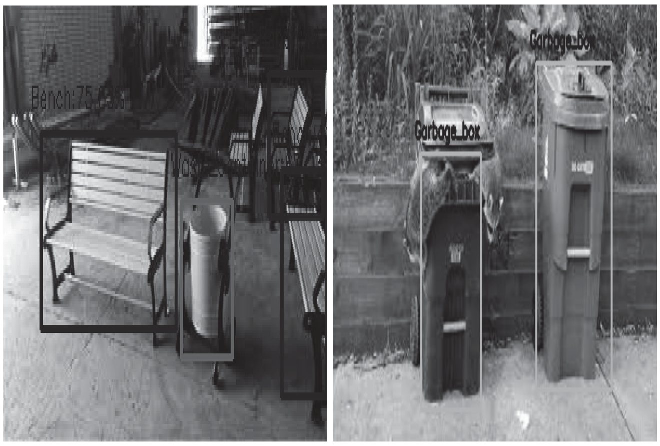
b



c



d



e

Fig. 3. The results of the YOLOv4 algorithm. (a) shows the results of window detection. (b) Billboard_Adv detection. (c) the results of house number detection (d) the results of ramp and stairs detection and (f) Garbage box and bench detection.

V. Discussion and conclusions

In this paper digital technologies and approaches to automatize routine complex urban processes under the concepts of Smart City strategies and best practices are considered. We propose a novel method for urban object detection using YOLOv3 and YOLOv4 algorithms that allow us to detect objects in real-time with high confidence of classification. We are used 11 different classes of urban objects for detection. To obtain and prepare enough training data we utilized various data augmentation techniques such as rotation, zoom, and transferring images. Furthermore, the transfer learning method presents to train collected data where it applies weight learned previously for different objects that can be used to solve new problems faster or with better solutions. To do that the pre-trained model as feature extractors is used and then trained on our database.

The RMSE method presented as a formula metric to evaluate the performance of both algorithms YOLOv3 and v4, in Tables 1 and 2. The experimental results indicate that the performance and the accuracy of the YOLOv4 algorithm were better to detect objects in different classes than YOLOv3. Also, YOLOv4 was able to detect most of the objects in each class of image. YOLOv4 was used in development as a fundamental method for web-service that can detect, classify, and locate urban objects in real-time with high fidelity. Such a tool can automatize the process of electronic inventory and can minimize or even replace human activity in this routine.

Prospects of future research work and development are in improving the size of the dataset and the number of classes to detect many various urban objects and using different machine learning algorithms with more accurate and concise approaches such as Faster-RCNN and SSD to evaluate our proposed method. Despite these approaches can have more confidence precision they are slower than YOLO methods, so the comparative analysis is actual.

ACKNOWLEDGMENT

This work was financially supported by the grant from Sberbank for ITMO University scientific projects under the theme № 50454 “Smart City technologies in providing banking services to the people”.

REFERENCES

[1] T. Bakici, E. Almirall and J. Wareham, “A Smart City Initiative: The Case of Barcelona”, *Journal of Knowledge Economy*, vol. 4(2), 2013, pp. 135–148, Web: https://doi.org/10.1007/s13132-012-0084-9.

[2] P. Neirotti, A.D. Marco, A.C. Cagliano, G. Mangano and F. Scorrano, “Current trends in Smart City initiatives: Some stylized facts”, 2014, pp. 25–36, Web: https://doi.org/10.1016/j.cities.2013.12.010.

[3] S.I. Drozhzhin, A.V. Shiyan and S.A. Mityagin, “Smart City Implementation and Aspects: The Case of St. Petersburg”, *in Electronic Governance and Open Society: Challenges in Eurasia’. EGOSE 2018. Communications in Computer and Information Science*, vol. 947, 2019, Web: https://doi.org/10.1007/978-3-030-13283-5_2.

[4] K.C. Desouza, “Delivering Artificial Intelligence in Government: Challenges and Opportunities”, *IBM Center for The Business of Government*, 2018.

[5] S.G. Vasin, “Artificial Intelligence in State Management”, *Upravlenie*, no. 3, 2017, pp. 5–10.

[6] A.A. Kosorukov, “Artificial intelligence technologies in the modern public administration”, *Sociodynamics*, 2019, pp. 43–58, Web: https://doi.org/10.25136/2409-7144.2019.5.29714.

[7] Z. Zhao, P.S. Zheng, S. Xu and X. Wu, “Object Detection with Deep Learning: A Review”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, 2019, pp. 3212–3232, Web: https://doi.org/10.1109/TNNLS.2018.2876865.

[8] Z. Zou, Z. Shi, Y. Guo and J. Ye, “Object Detection in 20 Years: A Survey”, 2019, Web: arXiv:1905.05055.

[9] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu and M. Pietikäinen, “Deep Learning for Generic Object Detection: A Survey”, *International Journal of Computer Vision*, no. 128, 2020, pp. 261–318, Web: https://doi.org/10.1007/s11263-019-01247-4.

[10] W. Yu, K. Yang, Y. Bai, T. Xiao, H. Yao and Y. Rui, “Visualizing and comparing AlexNet and VGG using deconvolutional layers”, *in Proceedings of the 33 rd International Conference on Machine Learning*, 2016.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu and A.C. Berg, “SSD: Single shot multibox detector”, *European conference on computer vision*, 2016, pp. 21–37, Web: arXiv: 1512.02325.

[12] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger”, *in Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[13] K. He, G. Gkioxari, P. Dollár and R. Girshick, “Mask R-CNN”, *in Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[14] Yingli Tian and Aries Arditi, “Computer Vision-Based Door Detection for Accessibility of Unfamiliar Environments to Blind Persons”, Researchgate, 2010, Web: DOI: 10.1007/978-3-642-14100-3_39.

[15] Franziska Lippoldt, “Window Detection in Facades for Aerial Texture Files of 3D CityGML Models”, *IEEE Xplore*, 2019.

[16] A. K. Aijazi, P. Checchin and L. Trassoudaine, “AUTOMATIC DETECTION AND FEATURE ESTIMATION OF WINDOWS FOR REFINING BUILDING FAC¸ ADES IN 3D URBAN POINT CLOUDS”, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Science*, vol. II-3, 2014.

[17] Adrian Llopart, Ole Ravn and Nils. A. Andersen, “Door and Cabinet Recognition Using Convolutional Neural Nets and Real-Time Method Fusion for Handle Detection and Grasping”, *IEEE, 3rd International Conference on Control, Automation and Robotics*, 2017, Web: DOI: 978-1-5090-6088-7117/$31.00.

[18] Romi Fadillah Rahmat, Dennis Opim, Salim Sitompul, Sarah Purnamawati and Rahmat Budiarto, “Advertisement billboard detection and geotagging system with inductive transfer learning in deep convolutional neural network”, *Researchgate TELKOMNIKA*, vol. 17, no. 5, 2019, pp. 2659–2666, Web: DOI: 10.12928/TELKOMNIKA.v17i5.11276.

[19] Murhaf Hossari and Soumyabrata Dev, “ADNet: A Deep Network for Detecting Adverts”, *CEUR-WS.org.*, vol. 2259, 2018, Web: ArXiv:1811.04115.

[20] Yuval Netzer, Tao Wang and Adam Coates, “Reading Digits in Natural Images with Unsupervised Feature Learning”, *Researchgate*, 2011.

[21] Unmesh Patil et al, “Deep Learning-based Stair Detection and Statistical Image Filtering for Autonomous Stair Climbing”, *in Third IEEE International Conference on Robotic Computing (IRC)*, 2019, Web: DOI 10.1109/IRC.2019.00031.

[22] C. Shorten and T.M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning”, *Journal of Big Data*, 2019, Web: https://doi.org/10.1186/s40537-019-0197-0.

[23] K. Weiss, T.M. Khoshgoftaar and D. Wang, “A survey of transfer learning”, *Journal of Big Data*, 2016, Web: https://doi.org/10.1186/s40537-016-0043-6.

[24] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, “Feature Pyramid Networks for Object Detection”, 2017, Web: arXiv:1612.03144v2.

[25] E. Najafi and S. Ivanov, “People Detection and Finding Attractive Areas by the use of Movement Detection Analysis and Deep Learning Approach”, *in 8th International Young Scientist Conference on Computational Science*, 2019, pp. 327–337, Web: https://doi.org/10.1016/j.procs.2019.08.209.

[26] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement”, 2018, Web: ArXiv:1804.02767v1.

[27] A. Bochkovskiy, C.-Y. Wan and H.-Y.M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection”, 2020, Web: ArXiv:2004.10934v1.

[28] R. Rothe, M. Guillaumin and L. Van Gool, “Non-maximum Suppression for Object Detection by Passing Messages Between Windows”, *Computer Vision - ACCV 2014. ACCV 2014. Lecture Notes in Computer Science*, vol. 9003, 2015, Web: https://doi.org/10.1007/978-3-319-16865-4_19.

[29] A.V. Kurilkin, S.V. Ivanov, “A comparison of methods to detect people flow using video processing”, *YSC 2016. 5th International Young Scientist Conference on Computational Science*, 2016, pp. 125–134, Web: https://doi.org/10.1016/j.procs.2016.11.016.

[30] YOLOv4, 2020, Web: https://riteshkanjee.medium.com/yolov4-superior-faster-more-accurate-object-detection-7e8194bf1872.