YOLO-Barcode: towards universal real-time barcode detection on mobile devices

D.M. Ershova [1,2], A.V. Gayer [1,3], P.V. Bezmaternykh [1,3], V.V. Arlazarov [1,3]
[1] Smart Engines Service LLC, 117312, Russia, Moscow, Prospect 60-Letiya Oktyabrya 9;
[2] Moscow Institute of Physics and Technology,
141701, Russia, Moscow Region, Dolgoprudny, Institutskiy per. 9;
[3] Federal Research Center "Computer Science and Control" of RAS,
119333, Moscow, Russia, Vavilova str. 44, corp. 2

Abstract

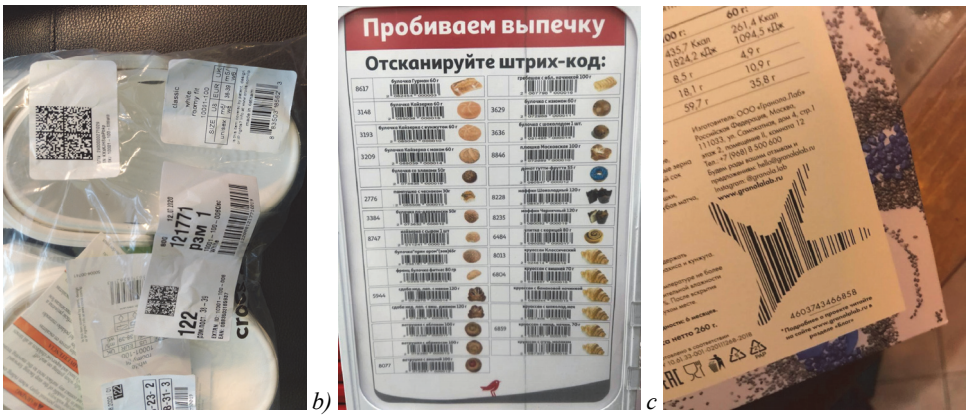
Existing approaches to barcode detection have a number of disadvantages, including being tied to specific types of barcodes, computational complexity or low detection accuracy. In this paper, we propose YOLO-Barcode – a deep learning model inspired by the You Only Look Once approach that allows to achieve high detection accuracy with real-time performance on mobile devices. The proposed model copes well with a large number of densely spaced barcodes, as well as highly elongated one-dimensional barcodes. YOLO-Barcode not only successfully detects the huge variety of barcode types, but also classifies them. Comparing with the previous universal barcode detector DilatedModel based on semantic segmentation, the YOLO-Barcode is 4 times faster and achieves state-of-the-art accuracy on the ZVZ-real public dataset: 98.6 % versus 88.9 % by F1-score. The analysis of existing publicly available datasets reveals the absence of many real-life scenarios of mobile barcode reading. To fill this gap, the new "SE-barcode" dataset is presented. The proposed model, used as a baseline, achieves a 92.11 % by F1-score on this dataset.

Keywords: barcode detection, barcode dataset, object detection, convolutional neural networks, deep learning.

Citation: Ershova DM, Gayer AV, Bezmaternykh PV, Arlazarov VV. YOLO-Barcode: towards universal real-time barcode detection on mobile devices. Computer Optics 2024; 48(4): 592-600. DOI: 10.18287/2412-6179-CO-1424.

Introduction

Today barcodes are used in various industries such as retail, logistics, travel, and healthcare. They offer a fast, accurate, and efficient method of tracking and managing information especially compared to other methods like manual data entry or RFID technology. Linear (1D) barcodes, such as the UPC and EAN codes, offer a simple representation of data using a series of parallel lines of different heights and thicknesses. Two-dimensional (2D) barcodes, such as QR and Data Matrix codes, store information in both horizontal and vertical dimensions and provide more capacity than linear ones. Warehouses, post offices and delivery services actively use mobile devices in their work, including for reading barcodes using a built-in camera (Fig. 1). Computer vision algorithms read all the barcodes in a photo at one time, while most scanners can only scan one barcode at a time (Fig. 1*b*), which is physically more difficult and takes longer.



a) b) c

Fig. 1. Examples of complex code detection cases: a) several barcodes of different types and shapes with projective distortions; b) densely spaced linear codes; c) "designed" code

The problem of barcode detection on images obtained from mobile devices has become very actual over the last 15 years. The vast majority of barcode detection approaches are based on classical image processing techniques. Some of them are "barcode-type independent", which means that they rely only on

commonly shared barcode traits. The list of traits includes the high contrast of minimal logical barcode elements or “modules”, the presence of a quiet zone around the code and the absence of large monotonous regions inside the code. To simplify the problem of barcode detection, many types introduce special groups of modules named “finder patterns” into their structure (Fig. 2*a*). Therefore, there are approaches that try to identify these patterns on the image in order to detect and segment the code itself. Despite the fact that such algorithms perform rather well on high-quality images with a focus on a specific barcode type, in real life they are not so effective due to certain constraints. Nowadays, several barcodes of different types and scales can be used simultaneously in a small area (Fig. 2*b*), thus requiring to analyze the same image several times using the appropriate method for every type, which has a negative impact on performance. Moreover, barcodes may be partially damaged or printed on reflective surfaces, what complicates the task of their detection. In the case of barcode capture using a mobile device camera, the environmental conditions are often uncontrolled, leading to barcode images that may suffer from projective distortions, uneven lighting, and flares [1]. Another problem with these methods is the diversity of “designed” barcodes, which are increasingly used for marketing needs. They are characterized by the presence of decorative features in the code structure or they mimic some object and are commonly embedded into complex backgrounds (Fig. 2*c*, *d*).

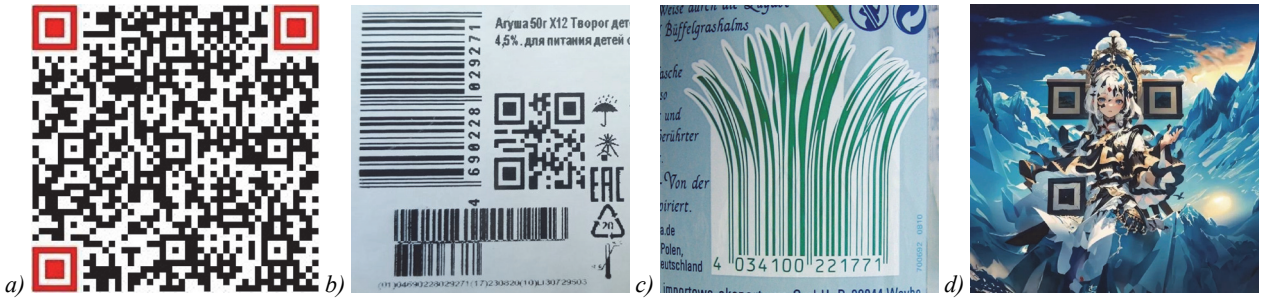


Fig. 2. Real barcode usage samples: (a) “finder patterns” of QR code type, (b) many barcodes of different types in a small area, (c) “designed” code sample in complex background, (d) “designed” QR code generated via Stable Diffusion

In recent years, approaches based on deep convolutional neural networks (D-CNN) have been significantly developed. According to evaluations on public datasets [2, 3], they outperform algorithmic methods in the problem of barcode detection. D-CNN provide resistance to capturing conditions and at the same time can detect several barcodes with different patterns on a single image.

In this paper, we propose YOLO-Barcode - a compact and computationally efficient anchor-based D-CNN for the problem of barcode detection and classification. This proposed model effectively handles with a large quantity of closely positioned barcodes, including long-shaped linear barcodes. The YOLO-Barcode was tested on four public datasets. The first one is ZVZ – a public dataset containing both real and synthetic images of 18 different barcode types. Others are WWU Muenster BarcodeDB, ArTe-Lab 1D Medium, and ParcelBar. These datasets contain images with linear codes only. In addition, an analysis was made of existing public datasets of barcodes with ground truth. It showed that not all real-life scenarios are present in them. To fill this gap, we present SE-barcode - novel dataset of synthesized barcodes in real-life scenes. This dataset consists of 1813 images with different types of barcodes with varying shapes, scales, and number of barcodes per image. By creating this dataset, we provide a new benchmark for barcode detection, which will help to estimate an ability of the detectors in solving hard real-life problems.

1. Related work

The pioneering methods for detecting barcodes captured via mobile device cameras were based on different combinations of classical digital image processing techniques. They typically include binarization [2, 4], morphological filtering [4 – 8], edges, and lines extraction [2, 3, 7]. For mobile devices, it is essential to reduce the computational complexity of the used methods. Choosing an appropriate binarization method can drastically simplify all the subsequent stages of image analysis, but that choice is not a trivial task. As for barcodes, many specialized methods have already been developed [8, 9, 10, 11]. Moreover, the performance of the binarization method depends heavily on its parameters fine tuning. Morphological methods also suffer from this problem: one needs not only to choose the sequence of filters but also select the proper shape of the structural elements. To overcome this issue and be able process barcodes of different scales, some iterative schemes were proposed [13]. The major drawback of such methods is their performance in terms of speed. Edge and line detectors are sensitive to image quality and barcode scale which often leads to problems on blurred or low-contrast images [14].

Barcode type-dependent detection methods are great, but only when someone can guarantee the absence of other types, thus severely limiting their usage.

Eventually, these methods require manual parameters tuning and generally cannot deal with barcodes in different scales with appropriate performance. Another disadvantage of such methods is their limited understanding of context – usually they rely on knowledge of barcode structure. That can lead to detection errors in cases of complex background. Finally,

false detection is another issue of these methods. Every wrongly detected region with a barcode candidate forces to analyze it for every allowed barcode type, thus severely slowing the whole code processing. Unfortunately, the false detection rate is not commonly presented in the studies related to classic methods. Another approach to this problem is to use methods based on training procedures that eliminate manual fine-tuning.

Machine learning approaches, particularly deep convolutional neural networks, have been widely used in barcode detection tasks over the last years and provide significant advantages over classical algorithms. Most existing methods consider barcode detection as an object detection task. Among all the D-CNN methods, "**Y**ou **O**nly **L**ook **O**nce" or YOLO [15] is one of the most applied approach [16]. The main advantage of YOLO-based models is that they find all objects in the entire image by applying a network to it only once. Coupled with the fact that YOLOv4 [17] and YOLOv5 [18] have been significantly optimized since the first version, YOLO-based networks became the best solution for use on mobile devices. In the paper [19] a model is shown that combined YOLOv1 as a detector and CNN for angle prediction of the found barcodes for better decoding. This model is able to detect both 1D and QR barcodes and shows good results on the Muenster BarcodeDB, but have low computational effectiveness and requires training two separate neural networks. In the paper [20] a modified method for QR codes detection based on the YOLO model was proposed. YOLO model was used as a QR code pattern finder. Paper [21] describes a model based on ThinYOLOv4 [21], which allows to detect barcodes and separate them into 1D and 2D barcodes by applying a simple classifier to each found bounding box. This method is significantly faster compared to method proposed In the paper [19] and shows solid performance on the author's private test dataset; however, it has not been tested on publicly available datasets.

In the paper [22] barcodes detection problem is considered as instance segmentation task. The authors proposed DilatedModel – lightweight model based on dilated convolutions, which achieved strong results on the Muenster BarcodeDB dataset and state-of-the-art result on the ArTe-Lab 1D Medium dataset. This approach not only detects but also classifies barcodes using a single network. In the paper [23] two-stage method for detecting barcodes in high-resolution images is suggested. This method is based on a modified Region Proposal Network (RPN) and a newly proposed Y-Net segmentation network with additional post-processing for creating bounding box for each segmented barcode mask. The implementation of such approach outperforms existing detection methods on the Muenster BarcodeDB, ArTe-Lab 1D Medium and the synthetic dataset published by the authors. However, this method requires a training of two separate networks, does not classify barcodes and was not tested on a competitive dataset with several barcodes on real images. In the paper [24] a method for synthetic data generation and a new ZVZ-dataset, which contains 920 real images with different barcode types, were proposed. The authors used the ZVZ-dataset to evaluate the U-Net model based on the ResNet18 backbone and their own lightweight DilatedModel architecture and provided strong baseline results.

The widespread use of mobile devices with various computational power requires the creation of both a computationally efficient and universal model for detecting barcodes of various scales, shapes and types. At the same time, the preliminary classification of barcodes facilitates their decoding and greatly reduces the recognition time spent on one frame. Despite the fact that modern mobile devices use relatively powerful hardware, their lifespan is limited by the battery. Thus, there is a need for optimized detection models whose computational complexity will be minimal at high performance. In this paper, we consider barcode detection as an object detection task and propose a new deep learning model based on the YOLO approach. The proposed model is capable to predict not only the bounding box but also the class of the barcode. It performs well with long-shaped objects, has 4 times fewer operations than DilatedModel [22] and shows state-of-the-art quality on ZVZ dataset.

2. Suggested method

YOLO-Barcode is based on the YOLO approach. In this approach, the input image is divided by a fixed $S\times S$ grid. Each cell in this grid is responsible for predicting the bounding box of the object whose center falls within it. Each bounding box predicts confidence, (x,y) coordinates of the box's center and its width and height. Each grid cell in the proposed model can also predict an object class, namely the barcode type. Total loss for the detection part is defined by equation:

$$L=\sum_{i=0}^{S^2}\sum_{j=0}^{A}(I_{ij}^{obj}\cdot(L_{obj}+L_{xy}+L_{wh})+I_{ij}^{noobj}\cdot L_{noobj}),$$

where I_{ij}^{obj} and I_{ij}^{noobj} are indicator functions that denotes object presence in cell (i,j), S is a size of grid and A is a number of anchors. L_{obj}, L_{noobj}, L_{xy} and L_{wh} are taken from paper [25]:

$$L_{obj}=-log(c),\ L_{noobj}=-\lambda_{noobj}\cdot log(1-c),$$
$$L_{xy}=\lambda_{xy}\cdot((x-\hat{x})^2+(y-\hat{y})^2),$$
$$L_{wh}=\lambda_{wh}\cdot((w_s-\hat{w}_s)^2+(h_s-\hat{h}_s)^2),$$
$$w=w_{anchor}\cdot exp(w_s);\ \ h=h_{anchor}\cdot exp(h_s).$$

There, c is the predicted confidence for the object, (x,y) – predicted coordinates of the object center which are related to cell, w and h are offsets predicted relative to chosen anchor. The values $\hat{x},\hat{y},\hat{w}_s,\hat{h}_s$ are taken from **g**round **t**ruth (GT). The hyperparameters λ_{noobj}, λ_{xy} and λ_{wh} should be adjusted manually according to the training

data, chosen anchors and grid size. Anchors are predefined bounding boxes with different aspect ratios and sizes. During labels generation stage, the most suitable anchor for each object is usually assigned using the **I**ntersection **O**ver **U**nion (IOU) score. Since IOU does not directly take into account the aspect ratio of the object, which is essential for different types of barcodes, we used the geometric method, which was proposed in YOLOv7 approach [26]. This method considers the most suitable anchor in terms of maximum of minimum between the normalized ratios of the widths and heights of the object and the anchor. For the classification part we applied softmax on the last *n* channels of output tensor and used cross-entropy as a loss, where *n* is the number of classes.

The YOLO-Barcode consists of 31 layers, the architecture is presented in Fig. 3. The ReLU was used as an activation function. We use a 512×512 grayscale image as input to improve the classification of small barcodes of different types but similar shapes. Due to the depth of the neural network, some low-level features can be lost, so to preserve them the architecture consists of a bottom-top part and top-down pathway part with lateral connections. Combining feature maps from different levels enhances the detector's ability to capture multi-scale information, contextual understanding, and robustness to scale variations [24]. In Tab. 1 we compare different D-CNN architectures used for barcode detection in terms of performance.

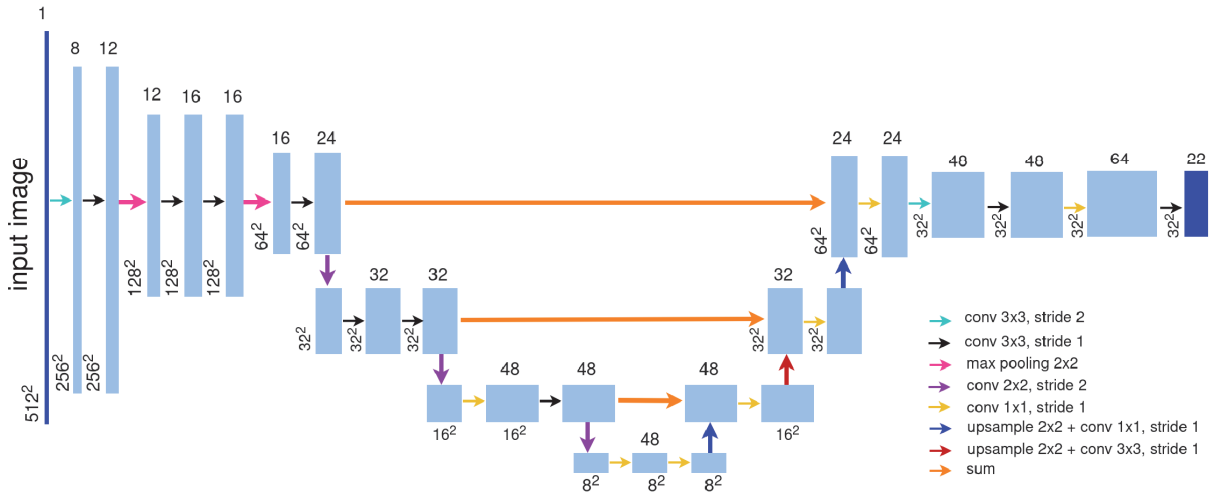


Fig. 3. YOLO-Barcode architecture

Tab. 1. Comparison of D-CNN models used for barcode detection in terms of performance

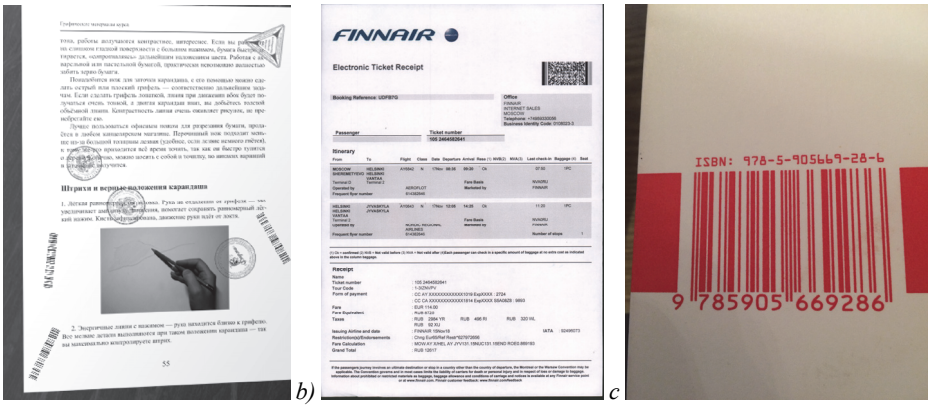
Model	Input size (px)	Parameters (M)	Performance (GFLOP)
Tiny YOLO v3	416×416	8.55	5.16
Tiny YOLO v4	416×416	6.06	6.1
DilatedModel [22]	512×512	0.04	1.9
YOLO-Barcode	512×512	0.14	**0.46**

In real-time recognition systems, the speed of a neural network holds greater significance than its size. The size of the neural network mostly affects initialization time, while its inference time affects the total processing time of each frame. Since the initialization of a neural network occurs only once, the ongoing processing speed is crucial for efficient and timely recognition, particularly when dealing with a huge amount of data.

3. Experimental setup

Our experiments are based on the ZVZ-dataset [24], which consists of two parts: ZVZ-real and ZVZ-synth. ZVZ-real is the dataset of 971 real images with 1214 barcodes in total, the maximum number of barcodes per image is 7. ZVZ-real is divided into 512 training images, 102 validation images and 306 test images. There are 18 different types of barcodes in total: QRCode, Aztec, DataMatrix, MaxiCode, PDF417, Non-postal-1D-Barcodes (Code128, Patch, Industrial25, EAN8, EAN13, Interleaved25, Standard2of5, Code32, UCC128, FullASCIICode, MATRIX25, Code39, IATA25, UPCA, UPCE, CODABAR, Code93, 2-Digit), and Postal-1D-Barcodes (Postnet, AustraliaPost, Kix, IntelligentMail, Royal-MailCode, JapanPost). ZVZ-synth is a synthesized dataset consisting of 30 thousand 512×512 samples of document images with barcodes. Some examples from ZVZ are shown in (Fig. 4). For experiments on real data, we used the train part of ZVZ-real and randomly chosen 2000 images from ZVZ-synth. Training images were grayscaled and resized to 512×512 input resolution, maintaining aspect ratio. To avoid an overfitting and increase the variety of train data, real-time augmentation [28] was used. During augmentation, first we apply mosaic augmentation with a grid 2×2 or 3×3 to 15 % of images. It was guaranteed that each inserted image would have no more than two barcodes. Then, 95 % of the data

was augmented with projective and lighting distortions, blur and noise, which were applied in random order. Each projective distortion was guaranteed to leave all the barcode corners within the image.



a) b) c

Fig. 4. Examples from ZVZ dataset: (a) from ZVZ-synth, (b) from ZVZ-real with small PDF-417 on the complex background, (c) from ZVZ-real with big linear barcode on the simple background

We train our model for 5000 epochs with a learning rate of 0.0005, a momentum of 0.9 and a batch size of 64. Stochastic gradient descent (SGD) was chosen as an optimization algorithm. We used the *k*-mean procedure on the training data to get the following anchor boxes (width and height relative to input size): [0.2, 0.05], [0.05, 0.2] and [0.1, 0.1]. The coefficients in the YOLO loss function were $\lambda_{xy} = 1.0$, $\lambda_{wh} = 2.0$ and $\lambda_{noobj} = 0.45$. In total, 7 classes are predicted as in [24]: QRCode, Aztec, DataMatrix, MaxiCode, PDF417, Non-postal-1D-Barcodes and Postal-1D-Barcodes.

For experiments on synthetic data, we used the train part of ZVZ-synth, which consists of 27 thousand images. We used the same preprocessing and augmentation as for real data excluding mosaic one. The model was trained for 500 epochs with the same parameters as for real data.

4. Results

4.1. Evaluation on ZVZ-real and ZVZ-synth

As metrics we used object-based precision, recall, and F-measure [22, 24]. To filter out the bounding boxes found by our detector, we fixed a confidence threshold of 0.5 and used non-maximum suppression with a threshold of 0.3 to remove duplicate responses for the same object. As in [22, 24], all metrics above are computed for the detected object and do not take into account the barcode class. To evaluate the classification of the detected objects by type, we used the accuracy metric (number of correctly guessed objects divided by the number of correctly detected objects). Thus, classification errors only affect classification accuracy. The results for the ZVZ-real dataset are shown in Tab. 2.

Tab. 2. Results obtained on the ZVZ-real. FULL states for entire real dataset (train + validation + test). P = precision, R = recall, F = F1-score, Acc - classification accuracy

Model	Pretrained on	FULL				TEST			
		P	R	F	Acc	P	R	F	Acc
ResNet18U-Net [24]	ImageNet	90.5	97.2	93.7	n/a	83.0	95.9	88.9	n/a
ResNet18U-Net [24]	-	87.8	94.8	91.2	n/a	80.9	91.5	85.8	n/a
DilatedModel [24]	ZVZ-synth	81.2	96.4	88.1	n/a	73.0	95.1	82.7	n/a
DilatedModel [24]	-	79.9	94.4	86.6	n/a	73.1	93.2	82.0	n/a
YOLO-Barcode	-	**98.0**	**98.4**	**98.2**	85.1	**98.3**	**98.8**	**98.6**	85.2

Fig. 5 demonstrates the performance of YOLO-Barcode in the difficult test cases from ZVZ-real (magenta denotes ground truth, green – correct detection result, red – false positive answer).

We also evaluated the model trained on the ZVZ-synth train part on the ZVZ-synth validation part. ZVZ-synth has a large number of barcodes printed at different angles; however the proposed model was trained to predict bounding boxes for objects without slope.

Therefore, we compared YOLO-Barcode detections with ground truth quadrangles (*) and with bounding boxes for ground truth quadrangles (**). For evaluation we used the same thresholds as for the model trained on real data. The results are shown in Tab. 3.

Examples of detections on ZVZ-synth are shown in Fig. 6. The most frequent error is that of two very close barcodes, only one is detected. In this case, the second one has much lower confidence and does not pass the confidence threshold.

4.2. Evaluation on other public datasets

We evaluated YOLO-Barcode trained on ZVZ-real and ZVZ-synth on popular barcode detection benchmarks – WWU Muenster BarcodeDB [2], ArTe-Lab Medium Barcode Dataset [3] and ParcelBar [29]. The Muenster dataset consists of 595 real images with ground truth. Barcodes on most of the images are linear, large and centered; there are also images with

several barcodes. The ArTeLab dataset consists of 365 images with ground truth, each image contains only one large linear barcode. There are images of damaged or deformed barcodes, as well as barcodes with glare. Since only segment masks were provided as the ground truth for the ArTeLab dataset, we converted them into bounding boxes. The ParcelBar dataset consists of 844 images with bounding boxes as ground truth. It contains post-box barcodes captured on mobile camera. Each image includes either one or several barcode tags. This dataset differs from the previous ones in greater variability of barcodes scales, complex backgrounds and different image distortions. Since there were some ground truth errors and 2D barcodes were not marked, we re-marked this dataset. For evaluation we chose a confidence threshold of 0.4 and used non-maximum suppression with a 0.3 threshold. The results are shown in Tab. 4.

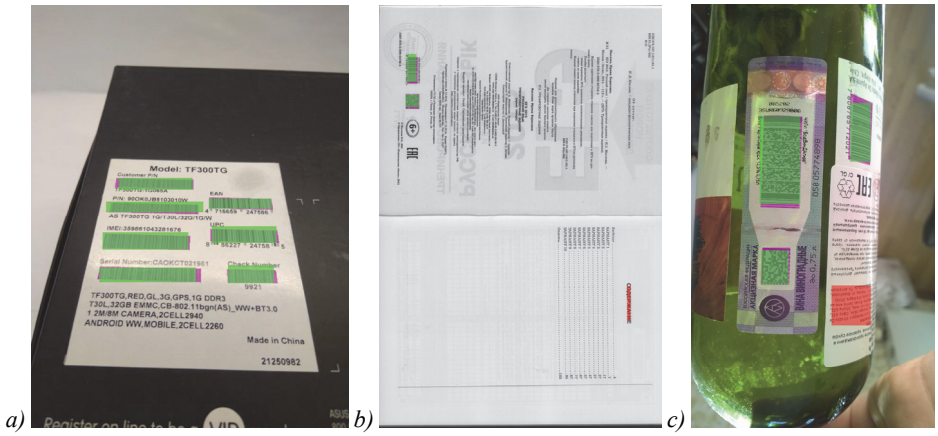


Fig 5. Detection results of YOLO-Barcode on images with: (a) many long-shaped barcodes, (b) three barcodes of different types, (c) two small barcodes on the complex background

Tab. 3. Results obtained on ZVZ-synth validation part

Model	Pretrained on	P	R	F	Acc
ResNet18U-Net [24]	ImageNet	97.1	96.3	96.7	n/a
ResNet18U-Net [24]	-	97.0	95.6	96.4	n/a
DilatedModel [24]	-	88.6	94.7	91.5	n/a
YOLO-Barcode (*)	-	92.3	91.5	91.86	79.9
YOLO-Barcode (**)	-	**98.7**	**97.8**	**98.3**	**80.5**

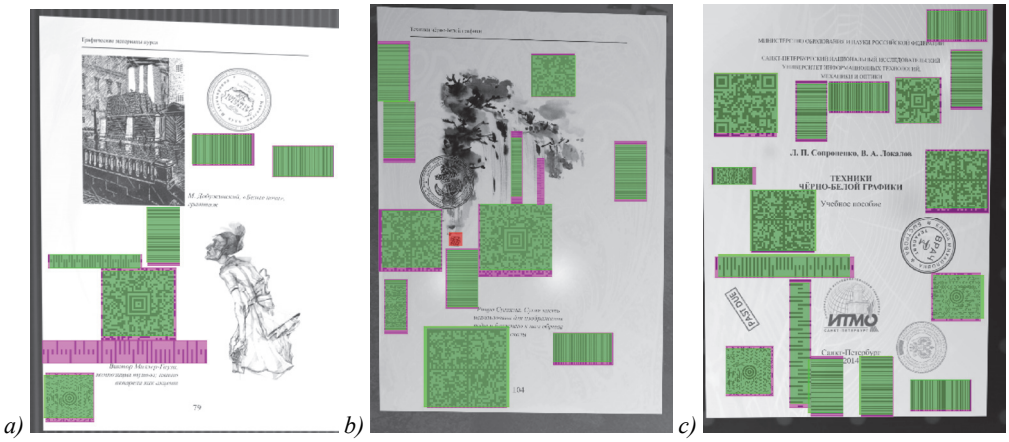


Fig 6. Detection results of YOLO-Barcode on ZVZ-synth: (a) typical false negative error, (b) typical false positive error on the complex background, (c) correct result on image with many barcodes of different types and sizes

Tab. 4. Results obtained on the popular benchmarks for barcode detection.
P = precision, R = recall, Acc – classification accuracy. (*) + classfication, (**) EAN13 only

Model	Trained on	Muenster			ArTeLab			ParcelBar	
		P	R	Acc	P	R	Acc	P	R
DilatedModel(*) [22]	Private dataset [22]	80.5	98.7	n/a	83.9	99.7	n/a	n/a	n/a
DilatedModel(**) [22]	Private dataset [22]	75.9	100.0	n/a	83.9	99.7	n/a	n/a	n/a
YOLO-Barcode	ZVZ	**88.1**	99.2	99.71	**84.6**	97.8	99.7	91.03	93.2

The proposed model shows worthy results on WWU Muenster and slightly worse results on ArTeLab in terms of recall. It is related to the presence of partially covered barcodes. Also there are some GT errors in both benchmarks (Fig. 7).

4.3. SE-barcode dataset

Analyzing existing public datasets, we noticed a significant absence of images containing a large number of barcodes of different types, as well as a lack of images with very small barcodes. Such situations are widely common in various industries, such as retail and logistics, where it is very important that all codes are correctly detected and decoded. To fill the gap, we present the SE-barcode dataset. It consists of 1813 images with ground truth. Some sample images from the SE-barcode dataset are shown in Fig. 8. The dataset is available for download at ftp://smartengines.com/yolo-barcode.

It contains very small barcodes, faded and designed barcodes, and barcodes with different projective distortions on complex background. For synthesis, we collected real images and replaced all barcodes with synthetic ones. The dataset consists of commonly used barcode types: UPCA, Code128, Code39, EAN13, UPCE, EAN8, ITF, DataMatrix, QR, Aztec and PDF417.

Comparison of the proposed synthesized dataset with the other public datasets is presented on Fig. 9 and Tab. 5.

Fig. 7. Detection results of the YOLO-Barcode on Muenster, ArTeLab and ParcelBar datasets. (a) ParcelBar, small barcodes on the complex background, (b) ArTeLab, big barcodes on the simple background, incorrect GT, (c) MuensterDB, partially closed barcode, (d) ArTeLab, many long-shaped barcodes, incorrect GT. Magenta denotes ground truth, green – correct detection result, red – false positive answer

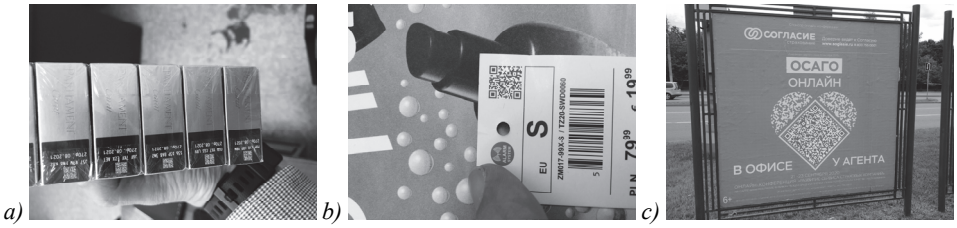


Fig.8. Examples from SE-barcode dataset: (a) many small codes of the same type, (b) codes of different types and shapes, (c) "designed" code

Tab. 5. Comparison of all datasets by the number of barcodes per image

Dataset \ Number of objects	1	2	3–5	6–10	>11
ZVZ-real [24]	738	135	33	15	0
MuensterDB [2]	595	0	0	0	0
ArTeLab [3]	364	1	0	0	0
ParcelBar [29]	530	257	56	1	0
SE-barcode	1033	531	169	61	19

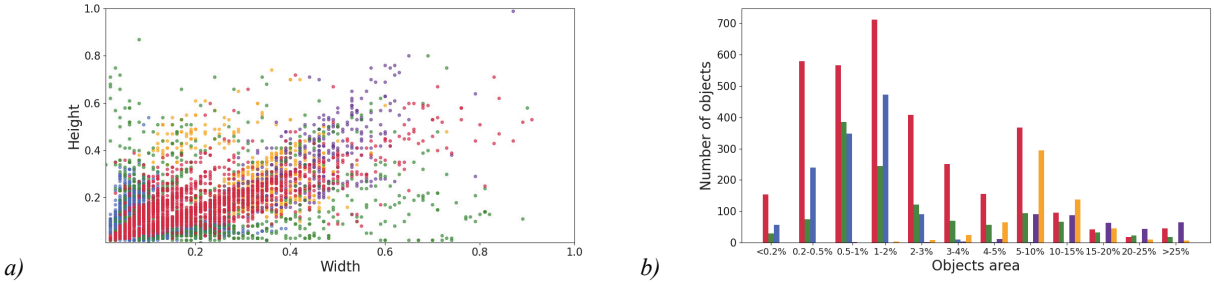


Fig. 9. Distributions of (a) objects shapes and (b) objects areas in public barcode datasets. Red denotes SE-barcode dataset, green – ZVZ, blue – ParcelBar, purple – ArTeLab, orange – Muenster

We evaluated YOLO-Barcode trained on ZVZ-real and ZVZ-synth on SE-barcode dataset with confidence threshold 0.4 and non-maximum suppression threshold 0.3. Detection result were compared with bounding boxes of barcodes. Results are shown in Tab. 6.

Typical precision errors are associated with false detections on various textures resembling barcodes. Typical recall error is a missing one barcode that is surrounded by a large number (more than 6) of other barcodes.

Tab. 6. Results obtained on SE-barcode dataset

Model	Trained on	P	R	F
YOLO-Barcode	ZVZ	91.83	92.4	92.11

5. Performance

To demonstrate the performance of YOLO-Barcode, we measured the running time on the desktop and mobile processors. Running time is averaged over 100 images. As the desktop processor, AMD Ryzen Threadripper PRO 5975WX was used. As the mobile processors, we used Apple A12 (Apple iPhone XR, 2018) and Apple A16 (Apple iPhone 14 Pro Max, 2022). The proposed model demonstrates the real-time detection rate on all the processors in the experiment. The results are presented in Tab. 7.

Tab. 7. YOLO-Barcode performance result

CPU	Time (ms)
AMD Ryzen Threadripper PRO 5975WX	27
Apple A12	32
Apple A16	20

Conclusion

In this paper, we presented the YOLO-Barcode - YOLO-based model for detecting barcodes of any types, shapes and scales, and their classification based on type. The proposed model is computationally efficient and can be applied on mobile devices for real-time barcode detection. It is approximately 4 times faster than DilatedModel [22], previous universal barcode detector for mobile devices based on semantic segmentation. We demonstrated that a YOLO-based approach is well suited for barcodes detection in hard scenes, which includes long-shaped and small barcodes. YOLO-Barcode can also correctly separate densely spaced barcodes, which are a bad case for the DilatedModel. The use of a deep learning model also ensures that the approach is easily scalable in the case of new types of barcodes appearing. To estimate quality, we used four public datasets: ZVZ, Muenster, ArTeLab and ParcelBar. The proposed model demonstrates high accuracy on all of them, and achieves state-of-the-art quality for the ZVZ-real dataset. To address the limitations of existing public datasets, we introduced the SE-barcode dataset, which contains images with hard real-life cases such as small barcodes or a large number of barcodes in a single image. The dataset is available for download at ftp://smartengines.com/yolo-barcode. According to evaluation results on this dataset, the proposed model confirms solid performance in various capture conditions. The SE-barcode benchmark serves as a valuable resource for advancing barcode detection technology.

It is important to note the low number of false positives, which is coupled with the ability to determine barcode types. This fact leads to an acceleration of frame processing time. The performance of the proposed model was measured on the Apple iPhone XR, Apple iPhone 14 Pro Max and Ryzen Threadripper PRO 5975WX – the estimated time was 32 ms, 20 ms and 27 ms respectively. The model size is equal to 580 KB and suits well for mobile devices.

The accuracy of the YOLO-Barcode on public datasets exceeds 97% in terms of recall. In order to further improve the accuracy of barcode detection, it is necessary to develop new datasets that sufficiently reflect complex real-world cases. The preparation of such a dataset can be a topic for future work. Another direction for further research is the reduction of the resulting detector's size. Despite the superiority in accuracy and performance over the DilatedModel, proposed model contains approximately 4.24 times more weights. The size of 580 KB is small enough for a deep learning model, but may still be too big for some mobile applications. Weights quantization for low-precision calculations or knowledge distillation may be a solution to create a lighter detector, but this requires further study, including how exactly such methods affect accuracy.

References

[1] Arlazarov VV, Zhukovsky A, Krivtsov V, Nikolaev D, Polevoy D. Analysis of using stationary and mobile small-scale digital video cameras for document recognition [In Russian]. Information Technologies and Computation Systems 2014; (3): 71-78.

[2] Wachenfeld S, Terlunen S, Jiang X. Robust recognition of 1-D barcodes using camera phones. 2008 19th Int Conf on Pattern Recognition 2008: 1-4. DOI: 10.1109/ICPR.2008.4761085.

[3] A. Zamberletti, I. Gallo and S. Albertini, Robust angle invariant 1D barcode detection. 2013 2nd IAPR Asian Conf on Pattern Recognition 2013: 160-164. DOI: 10.1109/ACPR.2013.17.

[4] Chai D, Hock F. Locating and decoding EAN-13 barcodes from images captured by digital cameras. 2005 5th Int Conf on Information Communications and Signal Processing 2005: 1595-1599. DOI: 10.1109/ICICS.2005.1689328.

[5] Chen C, He B, ZhangL, Yan P-Q. Autonomous recognition system for barcode detection in complex scenes. ITM Web of Conferences 2017; 12: 04016. DOI: 10.1051/itmconf/20171204016.

[6] Katona M, Nyúl LG. Efficient 1D and 2D barcode detection using mathematical morphology. In Book: Hendriks CLL, Borgefors G, Strand R, eds. Mathematical morphology and its applications to signal and image processing. Berlin, Heidelberg: Springer-Verlag; 2013: 464-475. DOI: 10.1007/978-3-642-38294-9_39.

[7] Bodnár P, Nyúl L. Barcode detection using local analysis, mathematical morphology, and clustering. Acta Cybernetica 2013; 21(1): 21-35. DOI: 10.14232/actacyb.21.1.2013.3.
[8] Bodnár P, Nyúl LG. Barcode detection with uniform partitioning and distance transformation. Proc 14th IASTED Int Conf on Computer Graphics and Imaging (CGIM 2013) 2013; 48-53. DOI: 10.2316/P.2013.797-022.
[9] Yang H, Kot AC, Jiang X. Binarization of low-quality barcode images captured by mobile phones using local window of adaptive location and size. IEEE Trans Image Process 2012; 21(1): 418-425. DOI: 10.1109/TIP.2011.2155074.
[10] Chen R, Yu Y, Xu X, Wang L, Zhao H, Tan HZ. Adaptive binarization of QR code images for fast automatic sorting in warehouse systems. Sensors 2019; 19(24): 5466. DOI: 10.3390/s19245466.
[11] Usilin SA, Bezmaternykh PV, Arlazarov VV. Fast approach for QR code localization on images using Viola-Jones method. Proc SPIE 2020; 11433: 114333G. DOI: 10.1117/12.2559386.
[12] Kruchinin AY. Industrial datamatrix barcode recognition with random tilt and rotate the camera. Computer Optics 2014; 38(4): 856-864. DOI: 10.18287/0134-2452-2014-38-4-865-870.
[13] Li JH, Wang WH, Rao TT, Zhu WB, Liu CJ. Morphological segmentation of 2-D barcode gray scale image. Int Conf on Information System and Artificial Intelligence (ISAI 2016) 2017: 62-68. DOI: 10.1109/ISAI.2016.0022
[14] Krešić-Jurić S. Analysis of edge detection in bar code symbols: An overview and open problem. J Appl Mathematics 2012; 2012: 758657. DOI: 10.1155/2012/758657.
[15] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. 2016 IEEE Conf on Computer Vision and Pattern Recognition (CVPR) 2016; 779-788. DOI: 10.1109/CVPR.2016.91.
[16] Wudhikarn R, Charoenkwan P, Malang K. Deep learning in barcode recognition: a systematic literature review. IEEE Access 2022; 10: 8049-8072. DOI: 10.1109/ACCESS.2022.3143033.
[17] Bochkovskiy A, Wang C, Liao HM. YOLOv4: Optimal speed and accuracy of object detection. arXiv Preprint. 2020. Source: <https://arxiv.org/abs/2004.10934>.
[18] ultralytics/yolov5. 2024. Source: <https://github.com/ultralytics/yolov5>.
[19] Hansen D, Nasrollahi K, Rasmusen C, Moeslund T. Real-time barcode detection and classification using deep learning. Proc 9th Int Joint Conf on Computational Intelligence 2017: 321-327. DOI: 10.5220/0006508203210327.
[20] Hussain N, Finelli C. KP-YOLO: A modification of YOLO algorithm for the keypoint-based detection of QR codes. In Book: Schilling F-P, Stadelmann T, eds. Artificial neural networks in pattern recognition. Cham: Springer Nature Switzerland AG; 2020: 211-222. DOI: 10.1007/978-3-030-58309-5_17.
[21] Zhang L, Sui Y, Zhu F, Zhu M, He B, Deng Z. Fast barcode detection method based on thinYOLOv4. In Book: Sun F, Liu H, Fang B, eds. Cognitive systems and signal processing (ICCSIP 2020). Singapore: Springer; 2021: 41-55. DOI: 10.1007/978-981-16-2336-3_4.
[22] Zharkov A, Zagaynov I. Universal barcode detector via semantic segmentation. 2019 Int Conf on Document Analysis and Recognition (ICDAR) 2019: 837-843. DOI: 10.1109/ICDAR.2019.00139.
[23] Quenum J, Wang K, Zakhor A. Fast, accurate barcode detection in ultra high-resolution images. IEEE Int Conf on Image Processing (ICIP) 2021: 1019-1023. DOI: 10.1109/ICIP42928.2021.9506134.
[24] Zharkov A, Vavilin A, Zagaynov I. New benchmarks for barcode detection using both synthetic and real data. In Book: Bai X, Karatzas D, Lopresti D, eds. Document analysis systems (DAS 2020). Cham: Springer; 2020: 481-493. DOI: 10.1007/978-3-030-57058-3_34.
[25] Redmon J, Farhadi A. YOLOv3: An incremental improvement. arXiv Preprint. 2018. Source: <http://arxiv.org/abs/1804.02767>.
[26] Wang C-Y, Bochkovskiy A, Liao H-YM. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv Preprint. 2022. Source: <https://arxiv.org/abs/2207.02696>.
[27] Lin T-Y, Dollar P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. 2017 IEEE Conf on Computer Vision and Pattern Recognition (CVPR) 2017: 936-944. DOI: 10.1109/CVPR.2017.106.
[28] Gayer AV, Chernyshova YS, Sheshkus AV. Effective real-time augmentation of training dataset for the neural networks learning. Proc SPIE 2019; 11041: 110411I. DOI: 10.1117/12.2522969.
[29] Kamnardsiri T, Charoenkwan P, Malang C, Wudhikarn R. 1D barcode detection: Novel benchmark datasets and comprehensive comparison of deep convolutional neural network approaches. Sensors 2022; 22(22): 8788. DOI: 10.3390/s22228788.

Authors' information

Daria Mikhailovna Ershova, (b. 2000), PhD student at Moscow Institute of Physics and Technology. Graduated from Lomonosov Moscow State University in 2023, is a developer at Smart Engines Service LLC since 2021. Research interests: machine learning, computer vision, object detection. E-mail: *d.ershova@smartengines.com*

Alexander Vyacheslavovich Gayer, (b. 1995), received a master degree in Applied Informatics from the National University of Science and Technology "MISiS" in 2019. Since 2017, he has been working at Smart Engines Service LLC as a developer, since 2019 he has also been working at the FRC "Computer Science and Control" of RAS as a junior researcher. Research interests: computer vision, deep learning, object detection.
E-mail: *agayer@smartengines.com*

Pavel Vladimirovich Bezmaternykh, (b. 1987), received a specialist degree in Applied Mathematics from the Moscow Institute of Steel and Alloys in 2009. Since 2016 he is employed at Smart Engines Service LLC, and since 2019 he is employed at the FRC "Computer Science and Control" of RAS. He is an author of more than 10 scientific publications. Research interests: image processing, document recognition. E-mail: *bezmaternyh@isa.ru*

Vladimir Viktorovich Arlazarov, (b. 1976), received a specialist degree in Applied Mathematics from the Moscow Institute of Steel Ph.D in Computer Science, the Ph.D. degree in computer science in 2005, and the Doctor of technical sciences degree in 2023 at the Federal Research Center "Computer Science and Control", Russian Academy of Science, CEO of Smart Engines Service LLC. Research interests: computer vision and document analysis systems E-mail: *vva@smartengines.com*

Received September 14, 2023. The final version – December 02, 2023.