IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 June 2023, accepted 13 July 2023, date of publication 19 July 2023, date of current version 28 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3296854

RESEARCH ARTICLE

PseudoAugment: Enabling Smart Checkout Adoption for New Classes Without Human Annotation

SERGEY NESTERUK[1], SVETLANA ILLARIONOVA[1], ILYA ZHEREBZOV[2], CLAIRE TRAWEEK[3], NADEZHDA MIKHAILOVA[1], ANDREY SOMOV[1], AND IVAN OSELEDETS[1]

[1]Skolkovo Institute of Science and Technology (Skoltech), 121205 Moscow, Russia
[2]Voronezh State University of Engineering Technology, 394000 Voronezh, Russia
[3]AuraBlue Corporation, Cambridge, MA 02143, USA

Corresponding author: Andrey Somov (a.somov@skoltech.ru)

The work was supported by the Analytical Center through the RF Government (Subsidy Agreement 000000D730321P5Q00 02), in 2 November 2021, under Grant 70-2021-00145.

ABSTRACT Increasingly, automation helps to minimize human involvement in many mundane aspects of life, especially retail. During the pandemic it became clear that shop automation helps not only to reduce labor and speedup service but also to reduce the spread of disease. The recognition of produce that has no barcode remains among the processes that are complicated to automate. The ability to distinguish weighted goods is necessary to correctly bill a customer at a self checkout station. A computer vision system can be deployed on either smart scales or smart cash registers. Such a system needs to recognize all the varieties of fruits, vegetables, groats and other commodities which are available for purchase unpacked. The difficulty of this problem is in the diversity of goods and visual variability of items within the same category. Furthermore, the produce at a shop frequently changes between seasons as different varieties are introduced. In this work, we present a computer vision approach that allows efficient scaling for new goods classes without any manual image labelling. To the best of our knowledge, this is the first approach that allows a smart checkout system to recognize new items without manual labelling. We provide open access to the collected dataset in conjunction with our methods. The proposed method uses top-view images of a new class, applies a pseudo-labelling algorithm to crop the samples, and uses object-based augmentation to create training data for neural networks. We test this approach to classify five fruits varieties, and show that when the number of natural training images is below 50, the baseline pipeline result is almost random guess (20% for 5 classes). PseudoAugment can achieve over 92% accuracy with only top-view images that have no pixel-level annotations. The substantial advantage of our approach remains when the number of original training images is below 250. In practice, it means that when a new fruit is introduced in a shop, we need just a handful of top-view images of containers filled with a new class for the system to start operating. The PseudoAugment method is well-suited for continual learning as it can effectively handle an ever-expanding set of classes. Other computer vision problems can be also addressed using the suggested approach.

INDEX TERMS Fruits recognition, retail automation, computer vision.

I. INTRODUCTION

Automation supports us in many aspects of daily life, from manufacturing to consumption [1]. Its goal is to reduce labor intensity and required time while increasing the accuracy of completed tasks. For instance, in healthcare, automated systems have become increasingly important and are constantly developing [2]. Studies on self-driving cars and their integration into traffic have opened up new possibilities in logistics. The use of smart city technologies, including

The associate editor coordinating the review of this manuscript and approving it for publication was Tai-Hoon Kim.

video analytics and action localization, can greatly enhance coordination and process management [3]. In the field of precision agriculture, significant transformations have taken place over the last decade, with many manual tasks now being performed automatically [4]. Automated systems also enable us to monitor natural resources and support environmental sustainability [5]. These significant results are made possible by the rapid advancements in technologies, including machine learning methods [6], data processing approaches [7], control algorithms [8], and fast computation algorithms [9].

Retail in particular benefits from many of the latest advances in automation, as employees are relieved from especially mundane or strenuous tasks. Already, machine learning has produced models that automate market analysis, product pricing, and logistics [10]. The implementation of Internet of Things (IoT) systems facilitates the automation of both human [2] and industrial monitoring processes [11], [12], [13]. The ability of e-commerce companies to rapidly adopt these technologies has led to a rapid growth of the industry and a pronounced shift in customer habits and expectations. [14]. For instance, people prefer to spend less time in lines. Additionally, the COVID-19 pandemic created a strong demand for low contact ordering and customer service. This increased the prevalence of automated checkout systems [15] and non-contact systems in general [16]. Self-service shops meet the above demands and, therefore, become widespread today [17].

We can split computer vision algorithms into classical computer vision and deep learning approaches. Classical recognition algorithms consist of two independent steps: feature selection and classification. Feature selection uses a hand-crafted algorithm such as SIFT (scale-invariant feature transform) [18] or statistical feature extraction [19]. Classification usually uses a classical machine learning algorithm such as SVM (support-vector machine) [20] or KNN (K-nearest neighbors) [21].

Deep learning approach, on the contrary, extracts features and learns to classify objects in a single step. Neural networks consist of cells that perform simple operations and recognize simple patterns. When we combine many cells, models can learn more complex relationships in a set of training data. This makes deep learning universal for the variety of tasks in different domains [22], [23]. However, to train an accurate and robust model, we need many training examples [24].

While in domains such as autopilot design, there are large-scale datasets available, and the analyzed objects lack variation [25], in agriculture-related problems, we lack fine-grained datasets with objects that differ significantly within a class [26]. Moreover, this problem is complicated because the appearance of the same object can vary with time [27].

In practical cases, it is very expensive and time-consuming to collect more training data [28]. Therefore, to increase the number of training samples artificially, we apply image augmentation. Image augmentation is a technique that makes transformed copies of the original image during the model training [29]. The basic transformations include image flipping, rotating, shifting, adding noise, changing brightness and contrast. We almost always apply simple transformations if they do not alter the essence of the image [30]. It ensures that, in every training iteration, the model sees slightly different samples, which increases data variability and makes the model more robust [31].

For the small datasets, basic image augmentation is usually not enough. A potential way to improve augmentation is to transform separate objects instead of the whole image [32]. It is referred as object-based augmentation, and results in more diverse scenes [33], [34]. The features of this approach allow for background substitution and object alteration. However, the limitation is that one needs instance-level annotations to crop objects from original images [35]. This paper proposes a pseudo-labeling approach to obtain instance-level annotations from only image-level annotations for weight goods in retail. *PseudoAugment* capitalizes upon the current delivery scheme, where different fruits and vegetables are delivered to stores in separate boxes. Thus, if we use a top-view image of a single box, all the objects in it will share the same class label. If we decompose an image into instances, we can further use them for object-based augmentation. In the result, *PseudoAugment* enables considerable image dataset augmentation with no manual instance-level annotations. For this practical application, it means that when a shop has a new fruit or vegetable variety, it is possible to expand smart scales system within minimal time and labor.

Below we summarize our main contributions:

- We introduce the novel concept *PseudoAugment* that combines instance-level pseudo-labeling and object-based augmentation;
- We provide open access to the dataset we collected, enabling the replication of our research and facilitating the conduct of similar experiments;
- We propose approaches to utilize *PseudoAugment* in both learning from scratch and incremental learning scenarios;
- We provide detailed explanation for each step of our approach, enabling the selection of the optimal implementation for a given computer vision problem;
- We implement and thoroughly evaluate the performance of *PseudoAugment* with different amount of training data to show the limitations.

II. RELATED WORK

A product recognition in retail is an essential problem in computer vision that includes particular challenges. In a number of studies, attempts are made to create an effective system [36].

In [37], they assess the computer vision algorithms for packaged products identification in vending machines applying a transfer learning technique. The authors open sourced the dataset with 300 annotated images. They

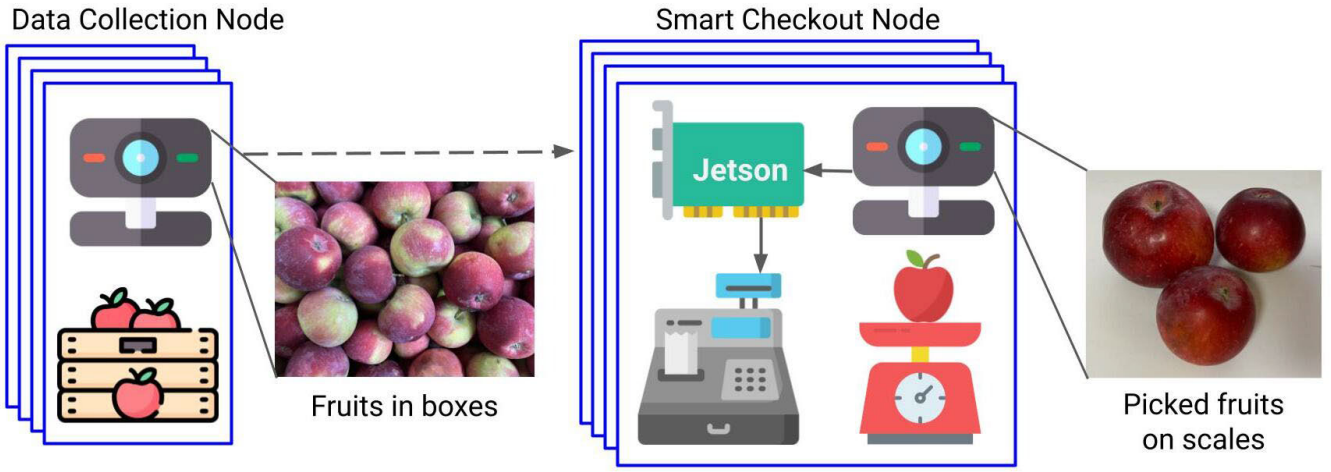


FIGURE 1. Smart Checkout System Principal Scheme.

study the problem of sparse label data and achieve the accuracy of 90% using as little as six images per class for product classification. Transfer learning approach for grocery product recognition is also presented in [38] and for fruits classification in [39].

In retail tasks, a common issue for new product launches arises when a machine learning algorithm needs to be retrained for a new previously unseen class. The research question of efficient model training for new classes is addressed in [40]. They highlight the importance of data augmentation, fine-grained classification, and one-shot learning techniques to deal with the following challenges: data limitations, intra-class variation, and flexibility. Another powerful tool that the authors mention is incremental learning when minimal retraining is required. It shows promising results in the general domain [41]. Another approach to prevent old classes being forgotten when a new one is introduced is continual learning techniques [42]. The authors use the generative adversarial network (GAN) model to create a memory of the old tasks for agriculture applications. However, this approach is very resource-intensive for on-edge implementation.

The similar appearance of retail products motivates the following research question: what is the best way to distinguish between a large quantity of varying classes? In [43], they propose an improved convolutional neural network and adjust the model robustness using mosaic data transformation.

An insufficient quantity of labelled datasets leads to training from scratch in most fruit classification tasks using custom collected samples [44]. Therefore, the authors suggest focusing more on the unsupervised algorithms and applying augmentation techniques to reduce expensive and time-consuming manual labelling. The lack of labelled data for fruits varieties in retail stores is also emphasized in [45]. They create a fruit dataset with three fruits types including items in plastic bags to train a deep neural network. Although promising results are obtained, the authors propose future data augmentation exploration and minimum number of training images estimation for sufficient prediction accuracy.

Algorithms onboard implementation is a required stage of a real-world retail application developing. In [46], the authors demonstrate a prototype for fruits classification system using a deep neural network implemented using a RaspberryPi. A RaspberryPi module for a neural network-based fruits inspection system is also implemented in [47]. The system enables accurate real time performance.

In scenarios where there is a dearth of training data, it is essential to adopt a comprehensive approach that encompasses both model-centric and data-centric methodologies. The latter focuses on optimizing data collection, annotation, and processing, thereby enhancing the quantity and quality of data. This approach entails the removal of irrelevant and spurious samples, leading to an increase in the availability of high-quality data that facilitates the training of generic models. Consequently, this results in improved accuracy and stable performance.

Data augmentation plays a central role in data-centric methods, particularly with image datasets, where both images and their annotations can be augmented. Annotation augmentation involves adding noise to enhance model robustness or creating different types of annotations to train different models. Image augmentations encompass a range of generic color and geometry transformations, image mixing [48], [49], [50], [51], neural network-based augmentations [52], [53], and modeling [54], [55]. This study focuses on instance-level augmentation, which augments individual objects rather than the entire image, enabling background substitution and greater flexibility. Existing instance-level augmentation approaches either rely on costly and challenging manual instance-level annotations [56] or perform coarse image manipulations [57]. In contrast, this study proposes an

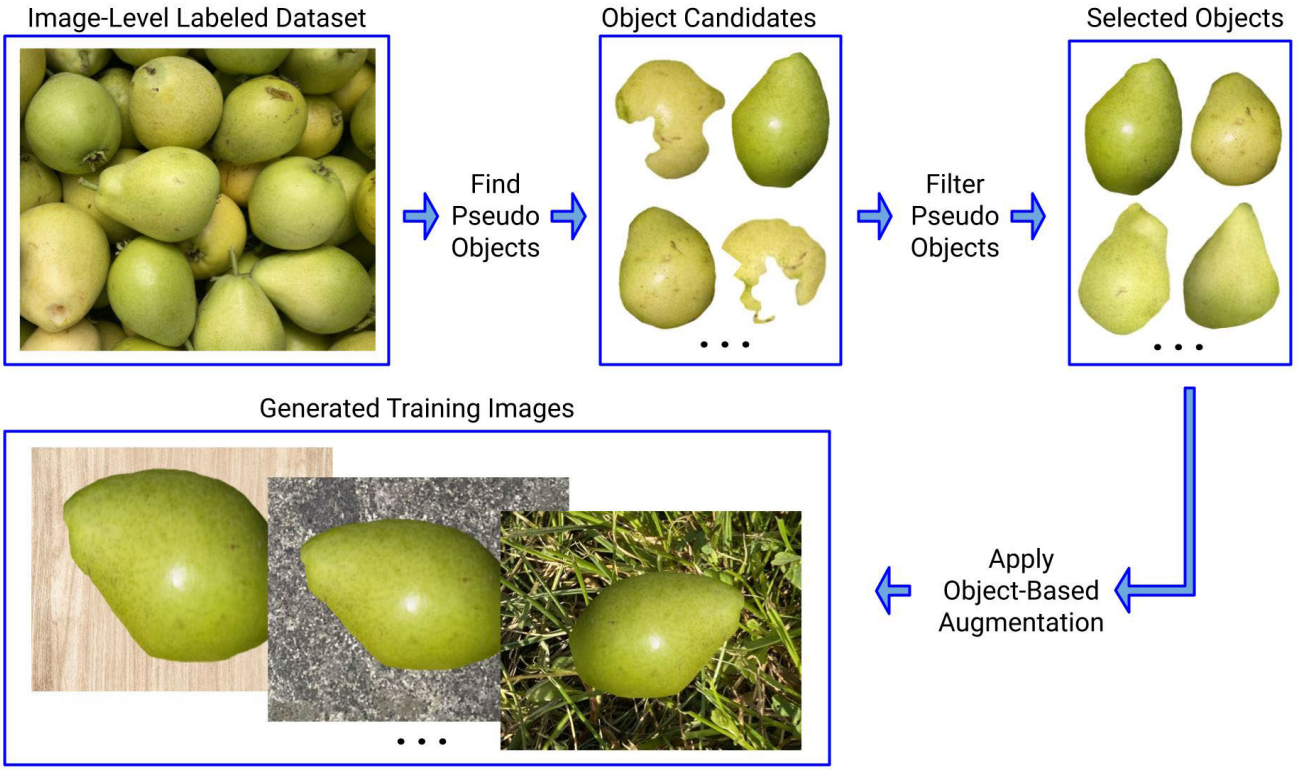


FIGURE 2. PseudoAugment Principal Scheme. PseudoAugment finds object candidates in the single-class top-view image, filters defective candidates, and uses the rest for object-based augmentation.

approach that identifies cases where instance-level augmentation can be performed at high quality without manual annotation. Although this is a specialized case, it has potential applications in various fields, as demonstrated by the example problem presented in this study.

III. METHODS

A. PSEUDOAUGMENT

In computer vision problems that we aim to solve with deep learning tools, one of the main limitations is in the inapproachability of large and well-annotated datasets. In practical cases, we sometimes can have a big dataset, but with several underrepresented classes. Despite the availability of numerous open source datasets tailored to specific tasks, the data distribution typically varies from that of the target problems, necessitating programmatic solutions [58]. The scaling of such systems in a very challenging task. One of the solutions is to augment the original data. Image augmentation applies various transformations to an image that changes its visual representation, but preserves that semantics. Since understanding image semantics is a tough task itself, a generic solution is just to apply small transformations. However, this approach limits possible generated images significantly.

Another approach is to apply object-based augmentation (OBA) instead of image-based. It means that we want to transform each object individually. OBA is very efficient for data-poor data sets. However, the limitation of OBA is the need for pixel-level annotation. Here, we describe a specific case where one can obtain approximate pixel-level annotations automatically. Further, we show that the approximate pseudo mask solution is good enough for image augmentation purpose. Moreover, we show that OBA, even with inaccurate annotations, can improve accuracy of the model sufficiently.

PseudoAugment is an improved OBA pipeline that is suitable for the cases where it is possible to acquire supplementary images with instances of a single class. Usually, those are top-view images. PseudoAugment consists of three steps: instance finding, instance filtering, and OBA (Figure 2).

The PseudoAugment methodology offers multiple implementation options for each step. Practitioners may select and replace any step with a model or algorithm that aligns with their task requirements. The primary consideration lies in balancing accuracy with computational resources.

B. INSTANCE FINDING

The goal of an instance finding algorithm is to extract object proposals. The concept of object proposal pertains to identifying image regions that are probable to encompass a solitary pertinent object. Essentially, this stage involves segmenting an image into subregions that solely comprise

objects while discarding the surrounding background. The attainment of spurious or erroneous objects is acceptable in this step.

We explore the performance of PseudoAugment with different algorithms that find instances. We provide a visual comparison of several algorithms we use to retrieve instances in Figure 3.

We can split such algorithms into two groups: DL-based models that require GPU, and other algorithms that can run fast on a CPU. In general, DL-based approaches have better accuracy, but they are more computationally intensive. Therefore, we compare results separately in these groups. Depending on available resources for a project, one can choose capitalizes upon the current delivery scheme, where different fruits and vegetables are delivered to stores in separate boxes.

Among classical computer vision algorithms, we test: Compact Watershed [59], Morphological Snakes [60], Border Following Algorithm [61], and Quick Shift [62].

The scientific concept of the watershed algorithm involves the computation of catchment basins within an image that has been flooded from specific markers. This process involves the allocation of pixels into marked basins, utilizing a grayscale gradient image as input, which represents the image as a landscape. Bright pixels in the image indicate boundaries between different regions, forming high peaks. The landscape is then flooded from the given markers until different basins meet at the peaks, resulting in the formation of distinct image segments for each basin. We use the Compact Watershed variant of the algorithm, which used to ensure that the resulting regions are more localized and well-shaped.

The Morphological Snakes algorithm is a segmentation technique used in image processing that involves the use of an energy function to deform a contour or boundary of an object in the image. This algorithm is based on the concept of active contours, which are curves or boundaries that can move and adjust their shape to fit the edges of an object in the image. The energy function used in the Morphological Snakes algorithm is composed of two terms: an internal energy term that controls the smoothness and regularity of the contour, and an external energy term that attracts the contour towards the edges of the object in the image [63]. The external energy term is calculated using a gradient or edge map of the image, which provides information about the location and intensity of edges in the image. The internal energy term is calculated based on the curvature and length of the contour, ensuring that it maintains a smooth and regular shape. The Morphological Snakes algorithm iteratively updates the position and shape of the contour based on the energy function until it converges to the edges of the object in the image. This results in a segmented image where the object is separated from the background [64]. We set the smoothing parameter to one.

The Border Following Algorithm is a technique used in image processing to extract the boundary or contour of an object in an image. This algorithm works by starting at a known point on the boundary of the object and following the edges of the object to trace its outline. In the Moore-Neighbor Tracing implementation, the algorithm starts at a known point on the boundary and moves in a clockwise direction, checking neighboring pixels to determine the next direction to follow [61].

Quickshift is a fast and efficient algorithm used for image segmentation. It works by grouping pixels that have similar color and texture characteristics into regions, which can then be used to identify objects or boundaries within an image. Quickshift uses a hierarchical approach to clustering, starting with small regions and gradually merging them together to form larger regions [65]. We set the sigma to one for smoothing, and choose fifty for a cut-off point for data distances.

Among deep learning models, we test: Feature Pyramid Network with ResNet101 backbone (ResNet101-FPN) [66], MaskRCNN [67], and DeepLabv3 [68]. Currently, these architectures are widely used and show high performance in various computer vision domains, such as medical imaging [69], precision agriculture [70], and remote sensing [71].

Feature Pyramid Network (FPN) is a neural network architecture used for object detection in images. It consists of a bottom-up pathway that extracts features from the input image at different scales, and a top-down pathway that combines these features to generate a pyramid of feature maps. The FPN architecture uses lateral connections to merge features from different scales, and a top-down pathway to propagate high-level information to lower scales. This allows the network to detect objects at different scales and resolutions, making it more robust to variations in object size and location [66]. We set the hyperparameners for the FPN as follows: 5 feature maps (FMs) with the smallest FM resolution of 32, and 256 channels for each FM. Max pooling to downsample FMs, and nearest-neighbor interpolation to upsample them. Learning rate is 0.01. Learning rate decay is polynomial. Weight decay is 0.0001.

Mask R-CNN builds upon the Faster R-CNN framework by adding a branch for predicting object masks in parallel with the existing branch for bounding box detection. This allows the network to not only detect objects, but also segment them at the pixel level. The Mask R-CNN architecture uses a Region Proposal Network (RPN) to generate candidate object regions, and then applies a series of convolutional layers to extract features from these regions. These features are then used to predict the class label, bounding box coordinates, and mask for each object instance [67]. We set the hyperparameners for the Mask R-CNN as follows: initial learning rate: 0.001. Learning rate decay is linear. Weight decay: 0.0005, Stochastic Gradient Descent (SGD) optimizer with Nesterov momentum of 0.9.

DeepLabv3 is a deep neural network architecture used for semantic image segmentation. It uses a modified version of the ResNet architecture as a backbone, and adds atrous spatial pyramid pooling (ASPP) modules to capture multi-scale

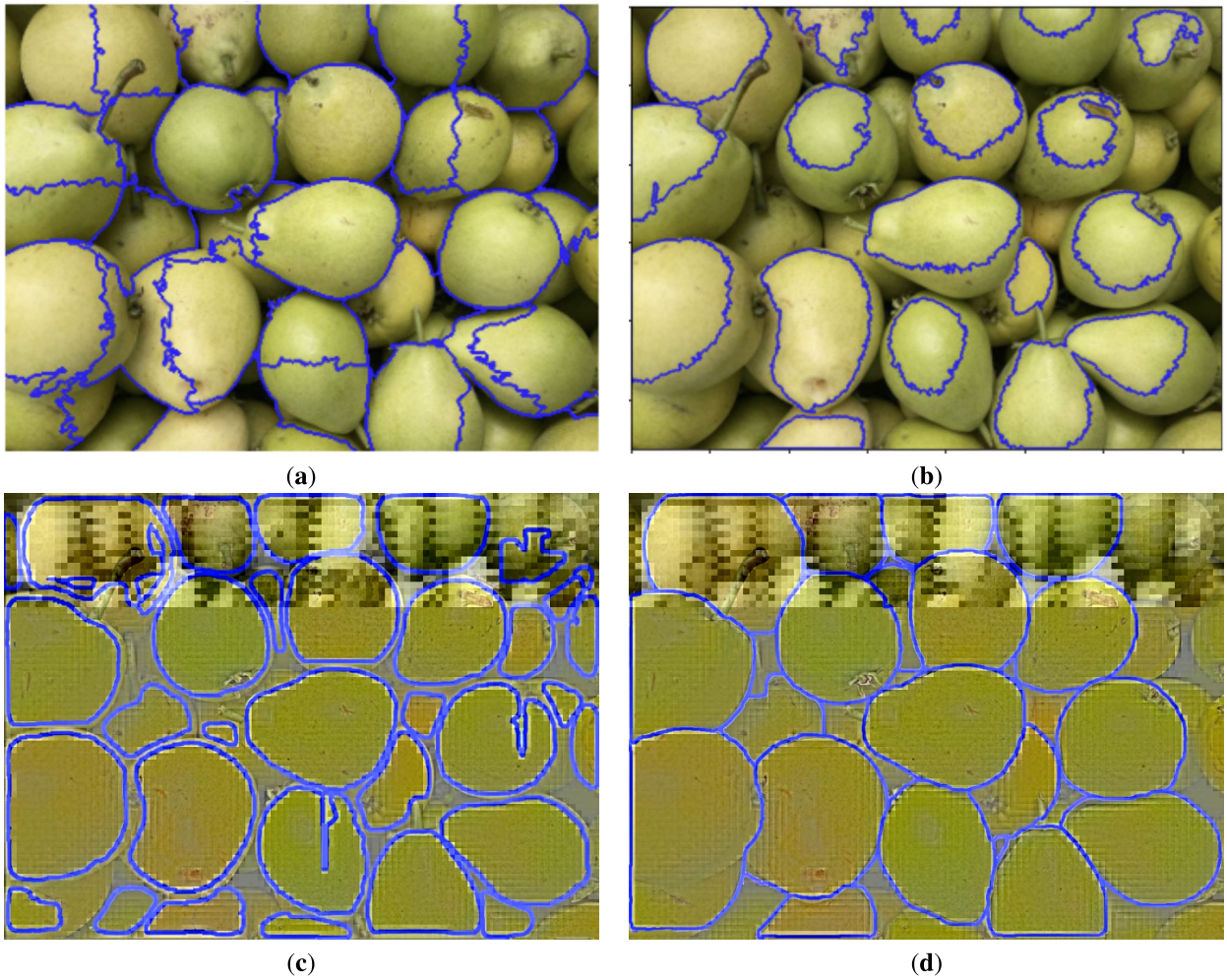


FIGURE 3. Examples of the extracted instances before filtering. Compact Watershed algorithm (a) , Border Following algorithm (b), MaskRCNN model (c), ResNet101-FPN model (d).

contextual information. The ASPP modules use dilated convolutions at different rates to extract features at different scales, which are then combined to generate a final segmentation map. DeepLabv3 also uses a decoder module to refine the segmentation map and produce sharper boundaries [68]. We set the hyperparameners for the DeepLabv3 as follows: initial learning rate: 0.007. Learning rate decay is linear. Weight decay: 0.0005, Stochastic Gradient Descent (SGD) optimizer with Nesterov momentum of 0.9.

To measure the results of the instance finding stage quantitatively, we manually count the number of instances on all top-view images separately for each class. For the practical application, this manual work is not required. In Table 1 we compare the performance of different algorithms for instance finding task. We use recall metrics calculated as 1 [72].

$$Recall = \frac{N_{found}}{N_{GT}}, \tag{1}$$

where N_{found} is the number of object candidates found by algorithm, N_{GT} is the number of ground truth objects. This metric shows the percent of the objects that an algorithm successfully found. Note that this metric is approximate because, for this step, we ignore false objects. We evaluate recall per class, and then apply macro-averaging to aggregate the result. The higher the value, the better. As one can see, deep learning approaches work better with the best result of 97.2% with ResNet101-FPN. The best method among ones that do not require a GPU to run is Compact Watershed with 80.6% recall.

C. INSTANCE FILTERING

Some of the found object candidates have very poor shapes, hence we apply instance filtering to eliminate them. For this purpose, we manually check whether a cropped object is good or not. To show the results quantitatively, we make a small dataset of good and bad crops. For the practical application, this manual work is not required. Note that for the training we use only good and bag samples for apples. No manual work is required for the new classes.

We compare the performance of two classification models for the instance filtering stage. The features that we use to

TABLE 1. Pseudo-instance detection recall.

Pseudo-Label Algorithm	Pears	Apples		Peaches		Average ↑
	Duchesse	Orlik	Zhigulevskoe	Golden Jubilee	Vladimir	
Compact Watershed	85	80	81	79	78	**80.6**
Active Contour	80	73	75	77	73	75.6
Border Following algorithm	73	66	71	71	67	69.6
Quick Shift	51	53	52	48	51	51
ResNet101-FPN	98	98	98	95	97	**97.2**
MaskRCNN	97	94	96	93	94	94.8
DeepLabv3	96	92	93	88	91	92

TABLE 2. Instance filtering false negative rate.

Pseudo-Label Algorithm	Filter Algorithm	Pears	Apples		Peaches		Average ↓
		Duchesse	Orlik	Zhigulevskoe	Golden Jubilee	Vladimir	
Compact watershed	Logistic Regression	8	9	7	9	9	8.4
	Random Forest	6	8	5	6	7	**6.4**
ResNet101-FPN	Logistic Regression	5	7	5	5	5	5.4
	Random Forest	4	5	4	3	5	**4.2**

train a classifier are based on the contours of the found objects and include: contour perimeter, contour area, the ration between the area and the perimeter, the 'roundness' of the contour approximated with circular Hough transform [73]. One can find the results in Table 2. We use the metric calculated with equation 2 [74].

$$FNR = \frac{N_{bad}}{N_{found}}, \tag{2}$$

where FNR is false negative rate, N_{bad} is the number of bad-looking object candidates after the filtering algorithm. The lower the value, the better. We evaluate the FNR per class, and then apply macro-averaging to aggregate the result.

D. OBJECT-BASED AUGMENTATION

Object-based augmentation (OBA) is the general image augmentation concept where we crop object from original image and paste to the new background.

In our work, we try several approaches. First of all, we can choose different sources of the background images. We can use uniform white background, backgrounds with random patterns, and background images from the same domain as the test images have. Also, we can paste either one or multiple instances per image. Moreover, we can mix OBA-generated with natural images if any are available. The results of the experiments with different approaches are reflected in section V.

IV. SMART SELF CHECKOUT USE CASE

A. USE CASE DESCRIPTION

In this work, we showcase how PseudoAugment can be applied in smart checkout systems. Smart checkout allows customers to pay in shops with no human cashier required. The advantages of such a system is that it lowers labor cost, speeds up the checkout process, and eliminates human interaction during pandemics.

To function, a smart checkout system must be able to all goods, including bulk goods missing barcodes. Recall that we do not cover the recognition of barcodes in this work because this problem is already solved with high accuracy [75].

Although several solutions for smart checkout systems exist, their common bottleneck is in the ability to adopt to the new classes. We show how an existing solution can be improved to work in a data-poor environment and to learn new classes rapidly. The main feature that we utilize is that top-view images of the delivered containers with fruits and vegetables are easy to collect, and they contain many class instances which we can retrieve automatically.

B. DATASET

Our dataset consists of five classes. They are:

- Pirum 'Duchesse' (Duchesse pear).
- Malum 'Orlik' (Orlik apple).
- Malum 'Zhigulevskoe' (Zhigulevskoe apple).
- Prunus persica 'Golden Jubilee' (Golden Jubilee peach).
- Prunus persica 'Vladimir' (Vladimir peach).

We collect two types of images. Images in which several objects are present under normal conditions, we call natural. One or several objects appear in every image. The images were taken in a variety of locations: in a garden, at a local grocery store, and in a lab setting. The location of each image is present in the dataset. It allows us to calculate the metrics for each location type separately, and compare the performance with different background types. We have 1000 natural images per class, and a total of 5000 natural images.

Another type of images contains top-view container images. Many objects are present in every top-view image. It makes them both very informative and easy to collect. We use 70 top-view images per class with 7.1 objects per image on average. That gives us approximately 500 auxiliary objects per class. Note that by combining different images and backgrounds and applying various transformations, we can generate more training images than the number of cropped objects.

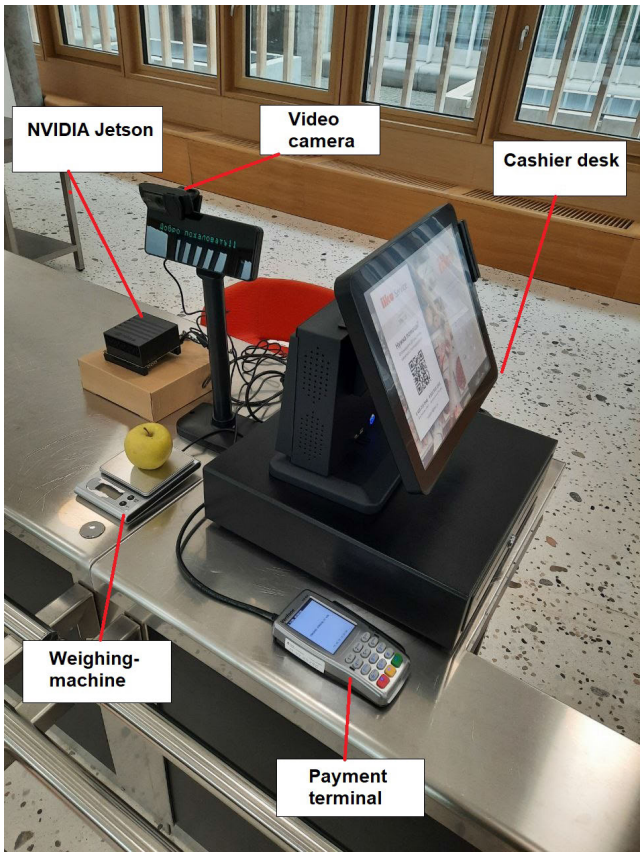


FIGURE 4. Experimental deployment.

We have also collected images without any objects of interest to use them as a background for object-based augmentation. We use 100 images with artificial patterns, and 100 natural images that correspond to the backgrounds of the main dataset.

C. IMPLEMENTATION

The principal scheme of the system is shown in Figure 1. As one can see, it consists of two nodes. We call them a data collection node and a smart checkout node.

The purpose of the data collection node is to collect new training data. It requires only a camera that sends images to the smart checkout node. In our design, we use the fact that fruits and vegetables are delivered to shops in big boxes, and each box is filled with a single variety. This implies that a top-view image of a single box contains instances of a single fruit or vegetable variety. The data collection node can be deployed in shops' warehouse. If we embed it to the goods reception pipeline, we can automatically update our dataset with new images. We can use one or many imaging stations in a warehouse.

The smart checkout node is a node that interacts with customers. In our implementation (Figure 4), it consists of *Jetson AGX Xavier* board with *Logitech C920s PRO HD Webcam* attached. The system can be integrated to a regular self checkout hardware. Also, an existing smart checkout system can be used. The only requirement is that it needs to have a GPU onboard.

The smart checkout node solves two tasks. The primary task is to recognize customers' goods. In the inference mode, a node makes images when anything is placed in the dedicated area, and runs a trained neural network to recognize an item. The model must be trained to recognize all the goods that are available for purchase. However, it is not possible to recognize a variety directly if it was not present in a training set. In this case, a model can predict a class from the training set that is visually close to the desired one.

But to be able to predict the exact class, we need to retrain a model. The complication with adding a new class to a model appears in data management steps. Usually, one needs to collect new training images and manually label them. Recall that, to obtain a robust model, it is vital to have many training samples.

In our solution, we improve the pipeline described above. Our smart checkout node is able to retrain itself for the new classes without manual annotations. First, it receives top-view images from a data collection node. Then, it sequentially applies algorithms described in sections III-B and III-C. In our system, we choose a DL-based approach, for the instance finding, more precisely ResNet101-FPN. However, if computational resources are limited, one can choose another method.

After extracting instances for new classes, we generate new training images, and fine-tune our model. We freeze the backbone of the model, and update only four last layers to reduce the training time. Our main solution takes only 20 minutes to tune for 5 new classes on *Jetson AGX Xavier*. Due to our observations, this is fast enough for the described use case because it usually takes more time to place new goods on shelves after their delivery.

We can use one or multiple inference stations in a shop. That is possible to use only one node for model fine tuning. After the training, models can share the weights.

For the pipeline implementation we have used: Scikit-image [76] and OpenCV [77] for classical computer vision algorithms, Scikit-learn [78] for machine learning algorithms, Pytorch [79] for deep learning models training, TensorRT [80] for models inference acceleration.

V. RESULTS

A. BASELINE

For the baseline solution, we use ResNet50 model [81]. We train it with the batch size of 64, SGD optimizer, cross entropy loss function and the learning rate of 0.001. The number of training samples is specified for every experiment individually. The model is pre-trained with the ImageNet dataset [82]. For the baseline, image augmentations include image horizontal flipping, random cropping, color jittering.

The baseline model does not use any auxiliary data from the top-view images. The result is shown as a dotted line on Figure 5.

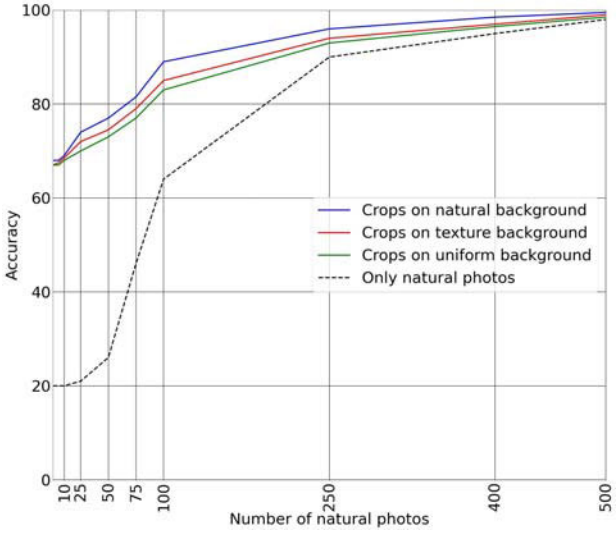


FIGURE 5. The comparison of the baseline accuracy with the accuracy of PseudoAugment algorithm using different substitution backgrounds.

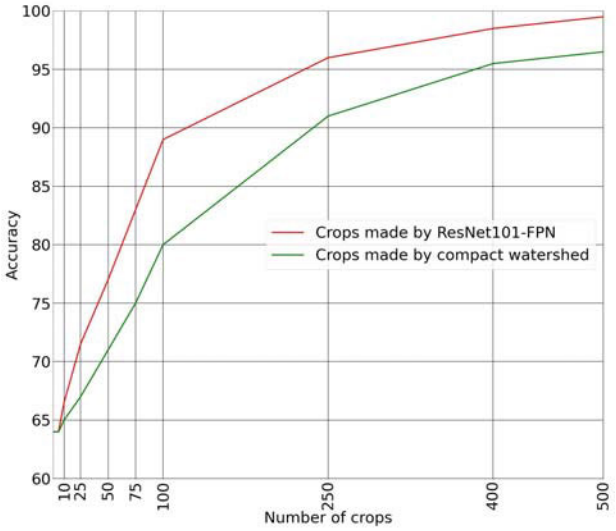


FIGURE 6. The performance comparison with ML-based (Compact watershed) and DL-based (R101-FPN) algorithms for instances extraction.

We calculate accuracy according to Equation 3 [72].

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N} I[C_i == \hat{C}_i], \tag{3}$$

where N is the number of images in the dataset, C_i is the correct class of the i^{th} sample in the dataset, $\hat{C}_i$ is the predicted class of the i^{th} sample in the dataset, $I[\dots]$ is the indicator function, which returns 1 if the value inside is true and zero otherwise.

As one can see, the accuracy of the baseline model drops dramatically when only few learning samples are available.

B. LEARNING FROM SCRATCH

In the learning from scratch scenario, we learn a model only once. It uses all five classes. The model and its hyperparameters are the same that we use for the baseline.

In Figure 5, we compare the accuracy of the baseline with several variations of PseudoAugment. More precisely, we show how the resulting accuracy depends on the type of the substituted background. For this experiment, all available objects from 35 top-view images were used. It is clear that even no background (uniform white background) OBA performs much better than the baseline.

Random patterns improve the results because a model has to learn how to distinguish an object from the background. If one can find images similar to the locations where the system is deployed, it is possible to improve the result even further. For the purpose of utilizing ''natural'' backgrounds, we employ images acquired within the confines of the same room where the system under development is employed. Although it is not imperative to utilize images exclusively from the same location, it is often convenient to do so in numerous practical scenarios. Specifically, this holds true for

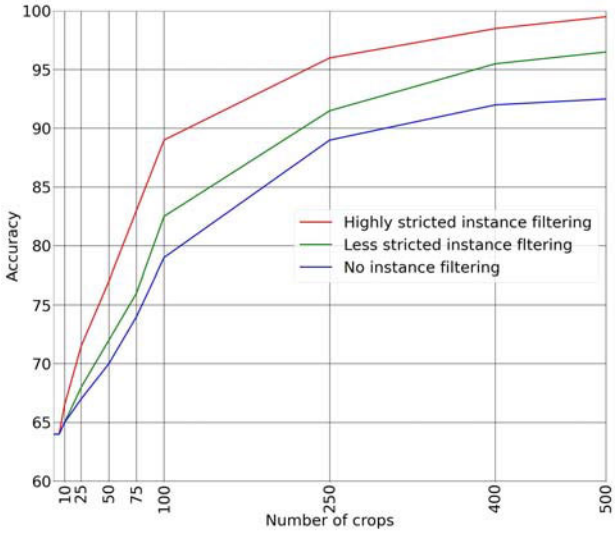


FIGURE 7. PseudoAugment performance comparison with different strictness of instance filtering.

situations involving stationary cameras. These images require no manual annotation. In the event that acquiring such images proves unfeasible, they may be substituted with other similar images. For instance, if the system is intended for use in a warehouse, it is possible to utilize images sourced from other warehouses available within open access datasets. Note that PseudoAugment gives us reasonable result even with no natural images at all.

In Figure 6, we show the difference in accuracy when different instance finding algorithms are used. As expected, we see that the DL-based approach has a better result. However, if computational resources are limited, one can still use a classical computer vision approach, and get an improvement.

In Figure 7, we show the effect of instance filtering. While training a model with badly shaped objects still gives an improvement, it is beneficial to screen them out.

Our findings indicate that the optimal outcome measurements for pseudo-labeling are achieved using ResNet101-FPN, while instance filtering is most effectively accomplished with Random Forest.

Our primary conclusion is that PseudoAugment represents a viable few-shot learning algorithm, which is particularly useful when the available training samples are limited. However, if the training dataset is sufficiently large, PseudoAugment may not be necessary. The frequent scenario is when certain classes have an abundance of samples, while others are underrepresented, such as when new classes are introduced. We delve into this specific case in greater detail in the subsequent section.

C. CONTINUAL LEARNING

In this section, we show our results in a continuous learning scenario. We train a baseline model with an incomplete set of classes and 1000 natural images per class. We choose the Naïve Continual Learning approach [83] that uses only new images for fine tuning and freeze the model backbone to minimize computational resources.

Then we apply PseudoAugment to fine tune the model to the new classes using top-view images only. Recall that it means that no manual annotation is used for the new classes.

Upon freezing a majority of the layers within our model, the resultant learning process is streamlined to solely incorporate final feature map representations. This approach yields expedited results, as backpropagation through fewer layers incurs less computational overhead and parameter convergence is achieved more rapidly. Our fine-tuning methodology involves utilizing augmented images generated by our algorithm, wherein objects belonging to the designated class are randomly selected and superimposed onto a novel background.

In Figure 8, one can see how the accuracy drops when we add new classes and fine tune the model. PseudoAugment prevents accuracy degradation. However, it still needs top-view training samples. If not enough auxiliary instances are provided, the model is not as efficient. For the fruits classification use case we achieve 98.3% accuracy without any natural images, and with only 70 top-view images per class.

D. ABLATION STUDY

In this section, we show some additions results. In Figure 9, one can see the results of the experiment with two approaches for object-based augmentation. We compare the accuracy of the model trained with only a single object per generated image and the accuracy of the model trained with several objects per generated image. We see no significant difference between these approaches for the classification task. However, it worth to further research the performance of

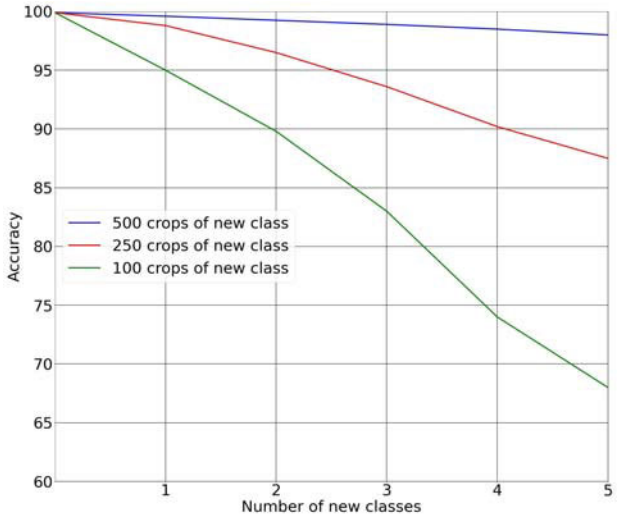


FIGURE 8. PseudoAugment performance comparison with different number of crops of new classes. Initial classes are trained with 1000 images per class. New classes are trained with generated samples only.

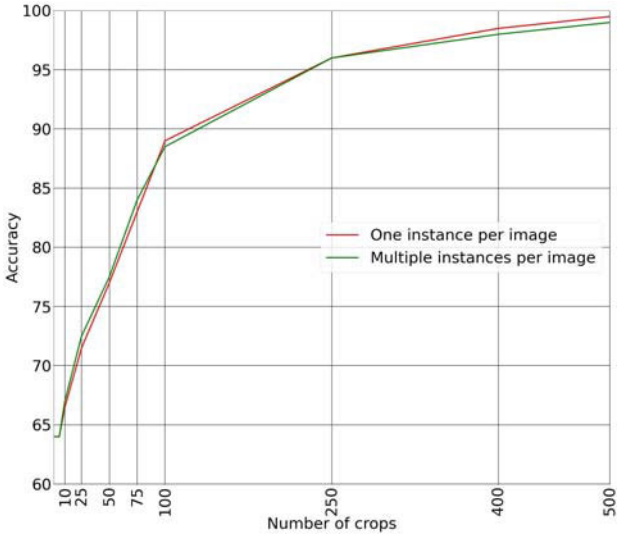


FIGURE 9. PseudoAugment performance comparison with single and multiple instances per generated image.

such computer vision problems as object detection as instance segmentation. Also, we do not overlap objects on generated images. A prospective research direction is to use a dynamic change of OBA for curriculum learning. The adaptive change of samples difficulty can provide a more robust model.

In Figure 10, we separate our original dataset according to the background and compare the results with different backgrounds of training samples. The colors in the legend represent the type of training data. Blank means the baseline approach. Colors correspond to the OBA with different backgrounds.

There are several conclusions from this figure. First, we see that it is beneficial to have background images for

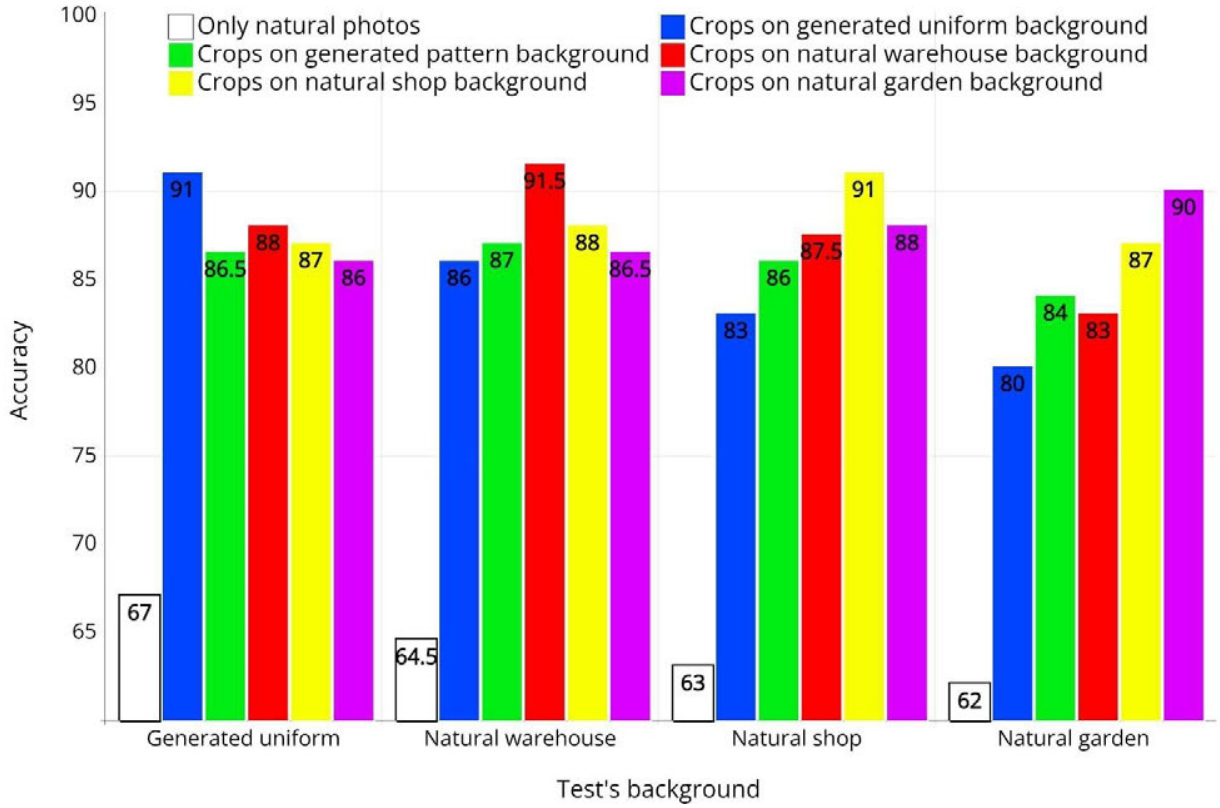


FIGURE 10. The examination of the PseudoAugment sensitivity for different backgrounds. Different holdout set backgrounds are used for accuracy calculation, and different substitution backgrounds are compared for training samples generation.

augmentation that are similar to the testing conditions. This allows a model to learn the context. Often, in practical applications, it is easy to make a dozen of plain images in the area where the system will be deployed. Moreover, these images do not require any annotations.

Second, as we see that the accuracy on generated test samples is considerably higher that the accuracy on natural images when we use generated train samples, it is essential to use only natural images for test leaderboard.

Moreover, the proposed approach can be integrated with other advanced augmentation methods that address various spectral bands [51] and sensing techniques [84].

VI. CONCLUSION

Deep learning shows high results in computer vision problems. It allows us to automate many routine tasks. In the agri-food domain, the main bottleneck that limits DL application is in the low availability of well-annotated training data. Training samples usually do not cover the diversity of the studied objects. Another challenging problem is to adopt a model to new classes.

In this paper, we describe a novel approach to image augmentation that aims to solve the above problems. Our PseudoAugment algorithm uses top-view images of containers with fruits or vegetables to extract pseudo-labelled instances without manual annotation. Then, we use object-based augmentation to generate new training images.

Our pipeline can be easily embedded into an existing smart checkout system to increase its performance. There is no need to chance model architecture or training procedure. We show that on fruit classification problem, our solution can reach 98.3% accuracy with no natural training images. If we have only 14 top-view images for a single new class, we can reach 95% accuracy. It is possible to combine natural images and generated images to achieve better results.

We also show that PseudoAugment can efficiently work in a continuous learning environment. Our system is able to fine tune to a new class in under 20 minutes on a Jetson Xavier board with no manually annotated images.

The utilization of the PseudoAugment algorithm is not constrained to smart checkout systems, as it can be efficaciously employed for any task that involves the acquisition of top-view images featuring readily distinguishable target objects. This algorithm has potential applications in industrial automation scenarios where objects are transported on conveyors with uniform backgrounds.

DATA AVAILABILITY

The dataset analysed during the current study is available to download via the following link: https://drive.google.com/drive/folders/1AohWL3j3Y1LbboaYKUVPsTVeE2BCNoCE.

REFERENCES

[1] P. Fröhlich, M. Baldauf, T. Meneweger, M. Tscheligi, B. de Ruyter, and F. Paternó, "Everyday automation experience: A research agenda," *Pers. Ubiquitous Comput.*, vol. 24, no. 6, pp. 725–734, Dec. 2020.

[2] W. Qi and A. Aliverti, "A multimodal wearable system for continuous and real-time breathing pattern monitoring during daily activity," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 8, pp. 2199–2207, Aug. 2020.

[3] G. Burdukovskaya, G. Ovchinnikov, M. Fedorov, and D. Shadrin, "Improving of action localization in videos using the novel feature extraction," in *Proc. IEEE 30th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2021, pp. 1–6.

[4] L. Lemikhova, S. Nesteruk, and A. Somov, "Transfer learning for few-shot plants recognition: Antarctic station greenhouse use-case," in *Proc. IEEE 31st Int. Symp. Ind. Electron. (ISIE)*, Jun. 2022, pp. 715–720.

[5] S. Illarionova, D. Shadrin, P. Tregubova, V. Ignatiev, A. Efimov, I. Oseledets, and E. Burnaev, "A survey of computer vision techniques for forest characterization and carbon monitoring tasks," *Remote Sens.*, vol. 14, no. 22, p. 5861, Nov. 2022.

[6] B. Mahesh, "Machine learning algorithms—A review," *Int. J. Sci. Res. (IJSR)*, vol. 9, pp. 381–386, Jan. 2020.

[7] O. Voynov, G. Bobrovskikh, P. Karpyshev, S. Galochkin, A.-T. Ardelean, A. Bozhenko, E. Karmanova, P. Kopanev, Y. Labutin-Rymsho, and R. Rakhimov, "Multi-sensor large-scale dataset for multi-view 3D reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 21392–21403.

[8] G. Yang, "Asymptotic tracking with novel integral robust schemes for mismatched uncertain nonlinear systems," *Int. J. Robust Nonlinear Control*, vol. 33, no. 3, pp. 1988–2002, Feb. 2023.

[9] J. Gusak, D. Cherniuk, A. Shilova, A. Katrutsa, D. Bershatsky, X. Zhao, L. Eyraud-Dubois, O. Shlyazhko, D. Dimitrov, I. Oseledets, and O. Beaumont, "Survey on large scale neural network training," 2022, *arXiv:2202.10435*.

[10] M. E. Isharyani, B. M. Sopha, M. A. Wibisono, and B. Tjahjono, "Smart retail adaptation framework for traditional retailers: A systematical review of literature," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manag. (IEEM)*, Dec. 2021, pp. 143–147.

[11] C. Tian, Z. Xu, L. Wang, and Y. Liu, "Arc fault detection using artificial intelligence: Challenges and benefits," *Math. Biosci. Eng.*, vol. 20, no. 7, pp. 12404–12432, 2023.

[12] Y. Shi, H. Li, X. Fu, R. Luan, Y. Wang, N. Wang, Z. Sun, Y. Niu, C. Wang, C. Zhang, and Z. L. Wang, "Self-powered difunctional sensors based on sliding contact-electrification and tribovoltaic effects for pneumatic monitoring and controlling," *Nano Energy*, vol. 110, Jun. 2023, Art. no. 108339.

[13] S. Nesteruk and S. Bezzateev, "Location-based protocol for the pairwise authentication in the networks without infrastructure," in *Proc. 22nd Conf. Open Innov. Assoc. (FRUCT)*, May 2018, pp. 190–197.

[14] B. Ratchford, G. Soysal, A. Zentner, and D. K. Gauri, "Online and offline retailing: What we know and directions for future research," *J. Retailing*, vol. 98, no. 1, pp. 152–177, Mar. 2022.

[15] F. Fortuna, M. Risso, and F. Musso, "Omnichannelling and the predominance of big retailers in the post-COVID era," *Symphonya. Emerg. Issues Manag.*, vol. 2, pp. 142–157, Nov. 2021.

[16] W. Qi, S. E. Ovur, Z. Li, A. Marzullo, and R. Song, "Multi-sensor guided hand gesture recognition for a teleoperated robot using a recurrent neural network," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 6039–6045, Jul. 2021.

[17] P. Sharma, A. Ueno, and R. Kingshott, "Self-service technology in supermarkets—Do frontline staff still matter?" *J. Retailing Consum. Services*, vol. 59, Mar. 2021, Art. no. 102356.

[18] S. Purohit, R. Viroja, S. Gandhi, and N. Chaudhary, "Automatic plant species recognition technique using machine learning approaches," in *Proc. Int. Conf. Comput. Netw. Commun. (CoCoNet)*, Dec. 2015, pp. 710–719.

[19] A. Nosseir and S. E. A. Ahmed, "Automatic identification and classifications for fruits using k-NN," in *Proc. 7th Int. Conf. Softw. Inf. Eng.*, May 2018, pp. 62–67.

[20] M. T. Habib, A. Majumder, A. Z. M. Jakaria, M. Akter, M. S. Uddin, and F. Ahmed, "Machine vision based papaya disease recognition," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 3, pp. 300–309, Mar. 2020.

[21] S. R. Dubey and A. S. Jalal, "Species and variety detection of fruits and vegetables from images," *Int. J. Appl. Pattern Recognit.*, vol. 1, no. 1, pp. 108–126, 2013.

[22] P. Ongsulee, "Artificial intelligence, machine learning and deep learning," in *Proc. 15th Int. Conf. ICT Knowl. Eng.*, Nov. 2017, pp. 1–6.

[23] S. Illarionova, D. Shadrin, V. Ignatiev, S. Shayakhmetov, A. Trekin, and I. Oseledets, "Estimation of the canopy height model from multispectral satellite imagery with convolutional neural networks," *IEEE Access*, vol. 10, pp. 34116–34132, 2022.

[24] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 843–852.

[25] J. Janai, F. Guney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Found. Trends Comput. Graph. Vis.*, vol. 12, pp. 1–308, Jul. 2020.

[26] Y. Lu and S. Young, "A survey of public datasets for computer vision tasks in precision agriculture," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105760.

[27] S. Nesteruk, D. Shadrin, V. Kovalenko, A. Rodríguez-Sánchez, and A. Somov, "Plant growth prediction through intelligent embedded sensing," in *Proc. IEEE 29th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2020, pp. 411–416.

[28] S. Nesteruk, D. Shadrin, M. Pukalchik, A. Somov, C. Zeidler, P. Zabel, and D. Schubert, "Image compression and plants classification using machine learning in controlled-environment agriculture: Antarctic station use case," *IEEE Sensors J.*, vol. 21, no. 16, pp. 17564–17572, Aug. 2021.

[29] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[30] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 558–567.

[31] S. Wu, H. Zhang, G. Valiant, and C. Ré, "On the generalization effects of linear transformations in data augmentation," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10410–10420.

[32] S. Nesteruk, I. Zherebtsov, S. Illarionova, D. Shadrin, A. Somov, S. V. Bezzateev, T. Yelina, V. Denisenko, and I. Oseledets, "CISA: Context substitution for image semantics augmentation," *Mathematics*, vol. 11, no. 8, p. 1818, Apr. 2023.

[33] S. Nesteruk, S. Illarionova, T. Akhtyamov, D. Shadrin, A. Somov, M. Pukalchik, and I. Oseledets, "XtremeAugment: Getting more from your data through combination of image collection and image augmentation," *IEEE Access*, vol. 10, pp. 24010–24028, 2022.

[34] S. Illarionova, S. Nesteruk, D. Shadrin, V. Ignatiev, M. Pukalchik, and I. Oseledets, "Object-based augmentation for building semantic segmentation: Ventura and Santa Rosa case study," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1659–1668.

[35] S. Illarionova, D. Shadrin, V. Ignatiev, S. Shayakhmetov, A. Trekin, and I. Oseledets, "Augmentation-based methodology for enhancement of trees map detalization on a large scale," *Remote Sens.*, vol. 14, no. 9, p. 2281, May 2022.

[36] B. Santra and D. P. Mukherjee, "A comprehensive survey on computer vision based approaches for automatic identification of products in retail store," *Image Vis. Comput.*, vol. 86, pp. 45–63, Jun. 2019.

[37] K. Fuchs, T. Grundmann, and E. Fleisch, "Towards identification of packaged products via computer vision: Convolutional neural networks for object detection and image classification in retail environments," in *Proc. 9th Int. Conf. Internet Things*, Oct. 2019, pp. 1–8.

[38] W. Geng, F. Han, J. Lin, L. Zhu, J. Bai, S. Wang, L. He, Q. Xiao, and Z. Lai, "Fine-grained grocery product recognition by one-shot learning," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 1706–1714.

[39] R. Verma and A. K. Verma, "Fruit classification using deep convolutional neural network and transfer learning," in *Proc. Int. Conf. Emerg. Technol. Comput. Eng.* Cham, Switzerland: Springer, 2022, pp. 290–301.

[40] Y. Wei, S. Tran, S. Xu, B. Kang, and M. Springer, "Deep learning for retail product recognition: Challenges and techniques," *Comput. Intell. Neurosci.*, vol. 2020, pp. 1–23, Nov. 2020.

[41] M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi, "Constrained few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9047–9057.

[42] Y. Li and X. Chao, "ANN-based continual classification in agriculture," *Agriculture*, vol. 10, no. 5, p. 178, May 2020.

[43] J. Wang, C. Huang, L. Zhao, and Z. Li, "Lightweight identification of retail products based on improved convolutional neural network," *Multimedia Tools Appl.*, vol. 81, no. 22, pp. 31313–31328, 2022.

[44] C. C. Ukwuoma, Q. Zhiguang, M. B. Bin Heyat, L. Ali, Z. Almaspoor, and H. N. Monday, "Recent advancements in fruit detection and classification using deep learning techniques," *Math. Problems Eng.*, vol. 2022, pp. 1–29, Jan. 2022.

[45] J. L. Rojas-Aranda, J. I. Nunez-Varela, J. C. Cuevas-Tello, and G. Rangel-Ramirez, "Fruit classification for retail stores using deep learning," in *Proc. Mex. Conf. Pattern Recognit.* Cham, Switzerland: Springer, 2020, pp. 3–13.

[46] M. Sugadev, K. Sucharitha, I. R. Sheeba, and B. Velan, "Computer vision based automated billing system for fruit stores," in *Proc. 3rd Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Aug. 2020, pp. 1337–1342.

[47] N. Ismail and O. A. Malik, "Real-time visual inspection system for grading fruits using computer vision and deep learning techniques," *Inf. Process. Agricult.*, vol. 9, no. 1, pp. 24–37, Mar. 2022.

[48] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.

[49] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6022–6031.

[50] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[51] S. Illarionova, S. Nesteruk, D. Shadrin, V. Ignatiev, M. Pukalchik, and I. Oseledets, "MixChannel: Advanced augmentation for multispectral satellite images," *Remote Sens.*, vol. 13, no. 11, p. 2181, 2021.

[52] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," 2015, *arXiv:1508.06576*.

[53] C.-T. Lin, S.-W. Huang, Y.-Y. Wu, and S.-H. Lai, "GAN-based day-to-night image style transfer for nighttime vehicle detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 951–963, Feb. 2021.

[54] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian, "Building generalizable agents with a realistic and rich 3D environment," 2018, *arXiv:1801.02209*.

[55] D. Misra, A. Bennett, V. Blukis, E. Niklasson, M. Shatkhin, and Y. Artzi, "Mapping instructions to actions in 3D environments with visual goal prediction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 2667–2678.

[56] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2917–2927.

[57] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1310–1319.

[58] Y. Shi, L. Li, J. Yang, Y. Wang, and S. Hao, "Center-based transfer feature learning with classifier adaptation for surface defect recognition," *Mech. Syst. Signal Process.*, vol. 188, Apr. 2023, Art. no. 110001.

[59] P. Neubert and P. Protzel, "Compact watershed and preemptive SLIC: On improving trade-offs of superpixel segmentation algorithms," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 996–1001.

[60] L. Alvarez, L. Baumela, P. Henriquez, and P. Marquez-Neila, "Morphological snakes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2197–2202.

[61] S. Suzuki and K. Be, "Topological structural analysis of digitized binary images by border following," *Comput. Vis., Graph., Image Process.*, vol. 30, no. 1, pp. 32–46, Apr. 1985.

[62] B. Fulkerson and S. Soatto, "Really quick shift: Image segmentation on a GPU," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2010, pp. 350–358.

[63] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vis.*, vol. 22, no. 1, p. 61, 1997.

[64] L. Alvarez, L. Baumela, P. Márquez-Neila, and P. Henríquez, "A real time morphological snakes algorithm," *Image Process. Line*, vol. 2, pp. 1–7, Mar. 2012.

[65] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Computer Vision—ECCV*. Marseille, France: Springer, Oct. 2008, pp. 705–718.

[66] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[67] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[68] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[69] X. Chen, X. Wang, K. Zhang, K.-M. Fung, T. C. Thai, K. Moore, R. S. Mannel, H. Liu, B. Zheng, and Y. Qiu, "Recent advances and clinical applications of deep learning in medical image analysis," *Med. Image Anal.*, vol. 79, Jul. 2022, Art. no. 102444.

[70] E. K. Raptis, G. D. Karatzinis, M. Krestenitis, A. C. Kapoutsis, K. Ioannidis, S. Vrochidis, I. Kompatsiaris, and E. B. Kosmatopoulos, "Multimodal data collection system for UAV-based precision agriculture applications," in *Proc. 6th IEEE Int. Conf. Robotic Comput. (IRC)*, Dec. 2022, pp. 1–7.

[71] S. Illarionova, D. Shadrin, I. Shukhratov, K. Evteeva, G. Popandopulo, N. Sotiriadi, I. Oseledets, and E. Burnaev, "Benchmark for building segmentation on up-scaled Sentinel-2 imagery," *Remote Sens.*, vol. 15, no. 9, p. 2347, Apr. 2023.

[72] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," 2020, *arXiv:2010.16061*.

[73] H. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of Hough transform methods for circle finding," *Image Vis. Comput.*, vol. 8, no. 1, pp. 71–77, Feb. 1990.

[74] D. S. Burke, J. F. Brundage, R. R. Redfield, J. J. Damato, C. A. Schable, P. Putman, R. Visintine, and H. I. Kim, "Measurement of the false positive rate in a screening program for human immunodeficiency virus infections," *New England J. Med.*, vol. 319, no. 15, pp. 961–964, Oct. 1988.

[75] R. Wudhikarn, P. Charoenkwan, and K. Malang, "Deep learning in barcode recognition: A systematic literature review," *IEEE Access*, vol. 10, pp. 8049–8072, 2022.

[76] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "Scikit-image: Image processing in Python," *PeerJ*, vol. 2, p. e453, Jun. 2014.

[77] G. Bradski, "The OpenCV library," *Softw. Tools Prof. Programmer*, vol. 25, no. 11, pp. 120–123, Nov. 2000.

[78] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 10, pp. 2825–2830, Jul. 2017.

[79] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8024–8035.

[80] O. Shafi, C. Rai, R. Sen, and G. Ananthanarayanan, "Demystifying TensorRT: Characterizing neural network inference engine on Nvidia edge devices," in *Proc. IEEE Int. Symp. Workload Characterization (IISWC)*, Nov. 2021, pp. 226–237.

[81] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

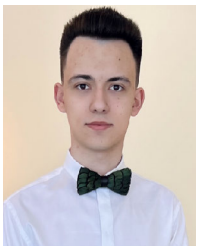
[82] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[83] V. Lomonaco, D. Maltoni, and L. Pellegrini, "Rehearsal-free continual learning over small non-i.i.d. batches," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1–3.

[84] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, "A review of medical image data augmentation techniques for deep learning applications," *J. Med. Imag. Radiat. Oncol.*, vol. 65, no. 5, pp. 545–563, Aug. 2021.

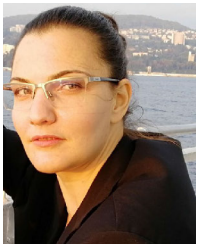
SERGEY NESTERUK received the B.S. and M.S. degrees in information security from the Saint Petersburg University of Aerospace Instrumentation, in 2018 and 2020, respectively, and the M.S. degree in information science and technology from the Skolkovo Institute of Science and Technology (Skoltech), Russia, in 2020, where he is currently pursuing the Ph.D. degree. His research are related to monitoring systems and applying machine learning methods to the collected data. He is involved in the development of the Precision Agriculture Laboratory, Skoltech, and is responsible for the development of greenhouse image collecting systems, the development of image augmentation framework, and computer vision research.

SVETLANA ILLARIONOVA received the bachelor's and master's degrees in computer science from Lomonosov Moscow State University, Moscow, Russia, in 2017 and 2019, respectively, and the Ph.D. degree in computer science from the Skolkovo Institute of Science and Technology (Skoltech), Moscow, in 2023. Her research interests include computer vision, deep neural networks, and remote sensing.



ILYA ZHEREBZOV pursuing the B.S. degree with the Voronezh State University of Engineering Technology. His research interests include data processing, machine learning, deep learning, and computer vision.

CLAIRE TRAWEEK received the bachelor's degree in mechanical engineering and computation from MIT. She is currently pursuing the Ph.D. degree in materials science and mechanical engineering. She is the Head of Product with AuraBlue Corporation, a biosensor startup.



NADEZHDA MIKHAILOVA received the Ph.D. degree from the Saint Petersburg Mining Institute. She is currently an Expert with the Skolkovo Institute of Science and Technology (Skoltech). Her research interests include waste management, AI for circular economy and waste treatment, and LCA.

ANDREY SOMOV received the degree in information and communication technology and the Diploma degree in electronics engineering from Russian State Technological University (MATI), Moscow, Russia, in 2004 and 2006, respectively, and the Ph.D. degree from the University of Trento, Trento, Italy, in 2009, with a focus on power management in wireless sensor networks (WSN). He is currently an Associate Professor with the Skolkovo Institute of Science and Technology (Skoltech), Russia. Before joining Skoltech in 2017, he was a Senior Researcher with the CREATE-NET Research Center, Trento, Italy, from 2010 to 2015; and a Research Fellow with the University of Exeter, Exeter, U.K., from 2016 to 2017. He has published more than 100 papers in peer-reviewed international journals and conference proceedings. His current research interests include machine learning, precision agriculture, and associated proof-of-concept implementation. He holds some awards in the fields of WSN and the IoT, including the Google IoT Technology Research Award, in 2016, and the Best Paper Award from the IEEE Internet of People (IoP) Conference, in 2019.

IVAN OSELEDETS received the degree from the Moscow Institute of Physics and Technology, in 2006, and the Candidate of Sciences and D.Sc. degrees from the Marchuk Institute of Numerical Mathematics of Russian Academy of Sciences, in 2007 and 2012, respectively. In 2013, he joined the Skolkovo Institute of Science and Technology, (Skoltech), where he is currently the Director of the Center for Artificial Intelligence Technology. His research covers a broad range of topics. He proposed a new decomposition of high-dimensional arrays (tensors)—tensor-train decomposition and developed many efficient algorithms for solving high-dimensional problems. His current research interests include the development of new algorithms in machine learning and artificial intelligence, such as the construction of adversarial examples, theory of generative adversarial networks, and compression of neural networks. It resulted in publications in top computer science conferences, such as ICML, NIPS, ICLR, CVPR, RecSys, ACL, and ICDM. He is an Associate Editor of *SIAM Journal on Mathematics of Data Science*, *SIAM Journal on Scientific Computing*, and *Advances in Computational Mathematics* (Springer).