mathematics

MDPI

Article

CISA: Context Substitution for Image Semantics Augmentation

Sergey Nesteruk [1], Ilya Zherebtsov [2], Svetlana Illarionova [1], Dmitrii Shadrin [1,3], Andrey Somov [1,*], Sergey V. Bezzateev [4], Tatiana Yelina [4], Vladimir Denisenko [2] and Ivan Oseledets [1]

[1] Skolkovo Institute of Science and Technology (Skoltech), 121205 Moscow, Russia
[2] Voronezh State University of Engineering Technology (VSUET), 394036 Voronezh, Russia
[3] Irkutsk National Research Technical University (INRTU), 664074 Irkutsk, Russia
[4] Saint-Petrsburg State University of Aerospace Instrumentation (SUAI), 190000 Saint Petersburg, Russia
* Correspondence: a.somov@skoltech.ru

Abstract: Large datasets catalyze the rapid expansion of deep learning and computer vision. At the same time, in many domains, there is a lack of training data, which may become an obstacle for the practical application of deep computer vision models. To overcome this problem, it is popular to apply image augmentation. When a dataset contains instance segmentation masks, it is possible to apply instance-level augmentation. It operates by cutting an instance from the original image and pasting to new backgrounds. This article challenges a dataset with the same objects present in various domains. We introduce the Context Substitution for Image Semantics Augmentation framework (CISA), which is focused on choosing good background images. We compare several ways to find backgrounds that match the context of the test set, including Contrastive Language–Image Pre-Training (CLIP) image retrieval and diffusion image generation. We prove that our augmentation method is effective for classification, segmentation, and object detection with different dataset complexity and different model types. The average percentage increase in accuracy across all the tasks on a fruits and vegetables recognition dataset is 4.95%. Moreover, we show that the Fréchet Inception Distance (FID) metrics has a strong correlation with model accuracy, and it can help to choose better backgrounds without model training. The average negative correlation between model accuracy and the FID between the augmented and test datasets is 0.55 in our experiments.

Keywords: image augmentation; computer vision; data collection; image retrieval; image generation; few-shot learning

MSC: 65D19; 51N05; 68U05

Citation: Nesteruk, S.; Zherebtsov, I.; Illarionova, S.; Shadrin, D.; Somov, A.; Bezzateev, S.V.; Yelina, T.; Denisenko, V.; Oseledets, I. CISA: Context Substitution for Image Semantics Augmentation. *Mathematics* **2023**, *11*, 1818. https://doi.org/10.3390/math11081818

Academic Editors: Paolo Mercorelli, Oleg Sergiyenko and Oleksandr Tsymbal

Received: 28 February 2023
Revised: 4 April 2023
Accepted: 6 April 2023
Published: 11 April 2023

1. Introduction

Deep learning and computer vision (CV) algorithms have recently shown their capabilities in addressing various challenging industrial and scientific problems [1]. Successful application of machine learning and computer vision algorithms for solving complex tasks is impossible without relying on comprehensive and high-quality training and testing data [2,3]. CV algorithms for solving classification, object detection, and semantic and instance segmentation require a huge variety of input data to ensure robust work of the trained models [4–6]. There are two major ways to enlarge a training dataset. The first one is obvious and implies physical collection of the dataset samples in various conditions to ensure high diversity of the training data. There is a set of huge datasets that have been collected for solving computer vision problems. These datasets are commonly used as the benchmark [7–10]. One of the specifics of these datasets is that they are general-domain sets. Unfortunately, general-domain-labeled data can be almost useless for solving specific industrial problems. One of the feasible applications of such well-known datasets is that they can serve as a good basis for pre-training of neural networks (transfer learning) [11,12]. Using these pre-trained neural networks, it is possible to fine-tune them and adapt them to

address specific problems. However, in some cases, even for fine-tuning, a comprehensive dataset is in high demand. Some events are rare, and it is possible to collect only a few data samples [13–15]. Thus, a second approach for enhancing the characteristics of the dataset can help. This approach is based on artificial manipulations with the initial dataset [16,17]. One of the well-developed techniques is data augmentation, where original images are transformed according to special rules [18]. Usually, the goal of image augmentation is to make the training dataset more diverse. However, augmentation can be used to deliberately shift the data distribution. If the distribution of the original training dataset differs from the distribution of the test set, it is important to equalize them as much as possible.

The agricultural domain is part of the industrial and research areas for which the development of artificial methods for improvement of training datasets is vital [19–21]. This demand appears due to the high complexity and variability of the investigated system (plant) that has to be characterized by computer vision algorithms [22]. The difficulty of the agricultural domain makes it a good candidate for testing augmentation algorithms.

There are many different plant species, and plants grow slowly. Thus, collecting and labeling huge datasets for each specific plant growing in each specific stage is a complex task [23]. Overall, it is difficult to collect datasets [24], especially for plants, and it is expensive to annotate them [25]. Therefore, we propose a method to multiply the number of training samples. It does not require many computational resources, and it can be performed on the fly. The idea behind the algorithm is to cut instances from the original images and add them onto the new backgrounds (Figure 1).

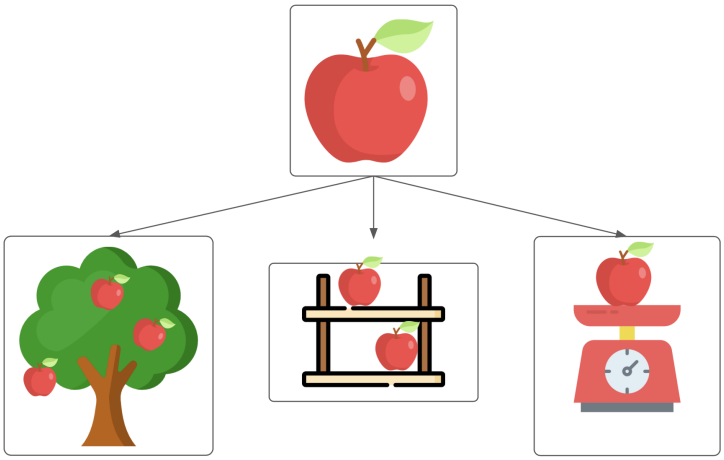


Figure 1. Context substitution showcase.

The **contribution of this study** is the following:

- we describe an efficient algorithm for instance-level image augmentation and measure its performance;
- we prove that the context is vital for instance-level augmentation;
- we propose several efficient ways to find representative background images if the test environment context is known;
- we show that it is possible to estimate which dataset variant will provide better accuracy before model training, calculating the FID between the test dataset and the training dataset variants;
- we share the dataset and generate background images and source code for augmentation.

The **novelty of this study** is as follows:

- extensive experiments with instance-level augmentation for different computer vision tasks;
- experiments with different model types;
- application of FID to choose the augmentation approach.

1.1. Image Augmentation

Computer vision models require many training data. Therefore, it becomes challenging to obtain a good model with limited datasets. Namely, a small-capacity model might not capture complex patterns, while a big capacity model tends to overfit if small datasets are used [26]. Slight changes in test data connected with surrounding and environmental conditions might also lead to a decrease in model performance [27].

To overcome this issue, we use various image augmentation techniques. Data augmentation aims to add diversity to the training set and to complicate the task for a model [28]. Among these plant image augmentation approaches, we can distinguish: basic computer vision augmentations, learned augmentation, graphical modeling, augmentation policy learning, collaging, and compositions of the ones above.

Basic computer vision augmentations are the default methods preventing overfitting in most computer vision tasks. They include image cropping, scaling, flipping, rotating, and adding noise [29]. There are also advanced augmentation methods, connected with distortion techniques and coordinate system changes [30]. Since these operations are quite generic, most popular ML frameworks support them. However, although helpful, these methods demonstrate limited use, as they bring insufficient diversity to the training data for few-shot learning cases.

Learned augmentation stands for generating training samples with an ML model. For this purpose, conditional generative adversarial networks (cGANs) and variational autoencoders (VAEs) are frequently used. In the agricultural domain, there are examples of applying GANs to *Arabidopsis* plant images for the leaf counting task [31,32]. The main drawback of this approach is that generating an image with a neural network is quite resource-intensive. Another disadvantage is the overall pipeline complexity: the errors of a model that generates training samples are accumulated with the errors of a model that solves the target task.

Learned augmentation policy is a series of techniques used to find combinations of basic augmentations that maximize model generalization. This implies hard binding of the learned policy to the ML model, the dataset, and the task. Although it is shown to provide systematic generalization improvement on object detection [33] and classification [34], its universal character as well as the ability to be performed along with multi-task learning are not supported with solid evidence.

Collaging presupposes cropping an object from an input image with the help of a manually annotated mask and pasting it to a new background with basic augmentations of each object [19]. In [35], a scene generation technique using object mask was successfully implemented for an instance detection task. It boosted model performance significantly compared with the use of only original images. The study on image augmentation for instance segmentation using a copy–paste technique with object mask was extended in [36]. The importance of scene context for image augmentation is explored in [37,38].

1.2. Image Synthesis

Graphical modeling is another popular method in plant phenomics. It involves creating a 3D model of the object of interest and rendering it. The advantage of this process is that it permits the generation of large datasets [39] with precise annotations, as the labels of each pixel are known. However, this technique is highly resource-intensive; moreover, the results obtained using the existing solutions [40,41] seem artificial. More realistic synthesis is very time-consuming. This approach is suitable when there are not many variations of the modeled object. If there are many different object types, it can be easier to collect and annotate new images.

1.3. Neural Image Generation and Image Retrieval

To gain new training images for CV tasks, one can implement GAN-based or diffusion-based models. Currently, they allow for the creation of rather realistic images and meet the demands of different domains, such as agricultural [42], manufacturing processes [43],

remote sensing [44], or medical [45]. Such models can be considered as a part of an image recognition pipeline. Moreover, recent results in Natural Language Processing (NLP) offer opportunities to extend image generation applications via textual description. For instance, an image can be generated based on a proposed prompt, namely, a phrase or a word. Such synthetic images help to extend the initial dataset. The same target image can be described by a broad variety of words and phrases that lead to diverse visual results. Another way to obtain additional training images is a data retrieval approach. It supposes to search for existing images from the Internet or some database according to a user's prompt. For instance, the CLIP model can be used to compute embedding of a text and to find images that match it better based on distance in a special embedding space [46].

2. Materials and Methods

The notation that we use in this section for describing the augmentation framework parameters is listed in Table 1.

Table 1. CISA framework internal notations.

Notation	Description
n	The number of objects per scene
m	The number of output masks
p	Average packaging overhead per input object
o	Average overhead for auxiliary data storage per object
ò	Constant system overhead
s	Objects' shrinkage ratio
θ	Orientation coefficient (width-to-height ratio)
H	The set of objects heights
$\widetilde{H}$	The set of shrinked object heights
W	The set of object heights
$\widetilde{W}$	The set of shrinked object widths
$\overline{h}$	Average over all input object heights
$\overline{w}$	Average over all input object widths
$\hat{h}$	Hard height restriction
M	Average RAM (random access memory) usage

2.1. Method Development and Description

In this paper, we introduce a method of image augmentation for a semantic segmentation task. When instance-level annotation areas are available, one can apply our method for other tasks such as classification, object detection, object counting, and semantic segmentation. Our method takes image–mask pairs and transforms them to obtain various scenes. Having a set of image–mask pairs, we can place many of them on a new background. Transformation of input data and background, accompanied by adding noise, gives the possibility for us to synthesize an infinite number of compound scenes.

This section first describes the overall augmentation pipeline and then describes the tested approaches for background image generation.

We distinguish between several types of image masks:

- **Single (*S*)**—single-channel mask that shows the object presence.
- **Multi-object (*MO*)**—multi-channel mask with a special color for each object (for each plant).
- **Multi-part (*MP*)**—multi-channel mask with a special color for each object part (for each plant leaf).
- **Semantic (*Sema*)**—multi-channel mask with a special color for each type of object (leaf, root, flower).
- **Class (*C*)**—multi-channel mask with a special color for each class (plant variety).

A single-input mask type allows us to produce more than one output mask type. Hence, multiple tasks can be solved using any dataset, even the one that was not originally designed for these tasks (see Table 2 for the possible mask transitions).

For example, an image with a multipart mask as input enables us to produce: the *S* mask, which is a Boolean representation of any other mask, the *MO* mask with unique colors for every object, the *MP* mask with a unique color for each part across all the present objects, and the *C* mask that distinguishes the classes (Figure 2). Additionally, for every generated sample, we provide bounding boxes for all objects and the number of objects of each class.

Note that we assume that each input image–mask pair includes a single object. Therefore, we can produce the *MO* mask based on any other mask. To create the *C* mask, information about input objects must be provided.

Table 2. Possible mask transitions.

Input Mask Type	S	MO	MP	Sema	C
S	+	+	-	-	+
MP	+	+	+	-	+
Sema	+	+	-	+	+

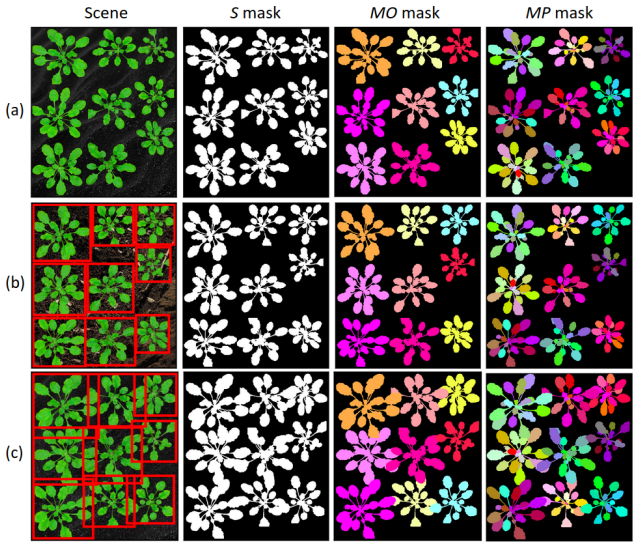


Figure 2. A MultiPartAugmentor-generated scene. (**a**) Without noise. (**b**) With added noise, blurring, and bounding boxes. (**c**) With added noise, blurring, bounding boxes, and $s = 0.1$.

2.2. System Architecture

The library with the code will be shared as an open source code with the community. The core of the presented system is the *Augmentor*. This class implements all the image and mask transformations. Such transformations as flipping or rotating are mutual for both the image and the mask. We add noise for images only.

From the main *Augmentor* class, we inherit *SingleAugmentor*, *MultiPartAugmentor* and *SemanticAugmentor* classes, helping to apply different input mask types and to treat them separately. To be more precise, *SingleAugmentor* is exploited for *S* input mask type, *MultiPartAugmentor* is for *MP* mask type, and *SemanticAugmentor* is for *Sema* mask type.

The described above classes are used in the *DataGen* class, which chooses images for each scene and balances classes if needed. Two principal ways of new scene generation are offline and online. We implement them in *SavingDataGen* and *StreamingDataGen* accordingly. Both of the classes take the path to images with corresponding masks as input. The offline data generator produces a new folder with created scenes while the online generator can be used to load data directly to a neural network.

Offline generation is more time-consuming because of additional disk access operations; at the same time, it is performed in advance and thus does not affect model training time. It also makes it easier to manually look through the obtained samples to tune the transformation parameters.

Meanwhile, the online data generator streams its results immediately to the model without saving images on the disk. Furthermore, this type of generator allows us to change parameters on the fly: for instance, the model is trained on easy samples, and then, the complexity may be manipulated based on the loss function.

2.3. Implementation Details

The present section discusses the main transformation pipeline (Figure 3).

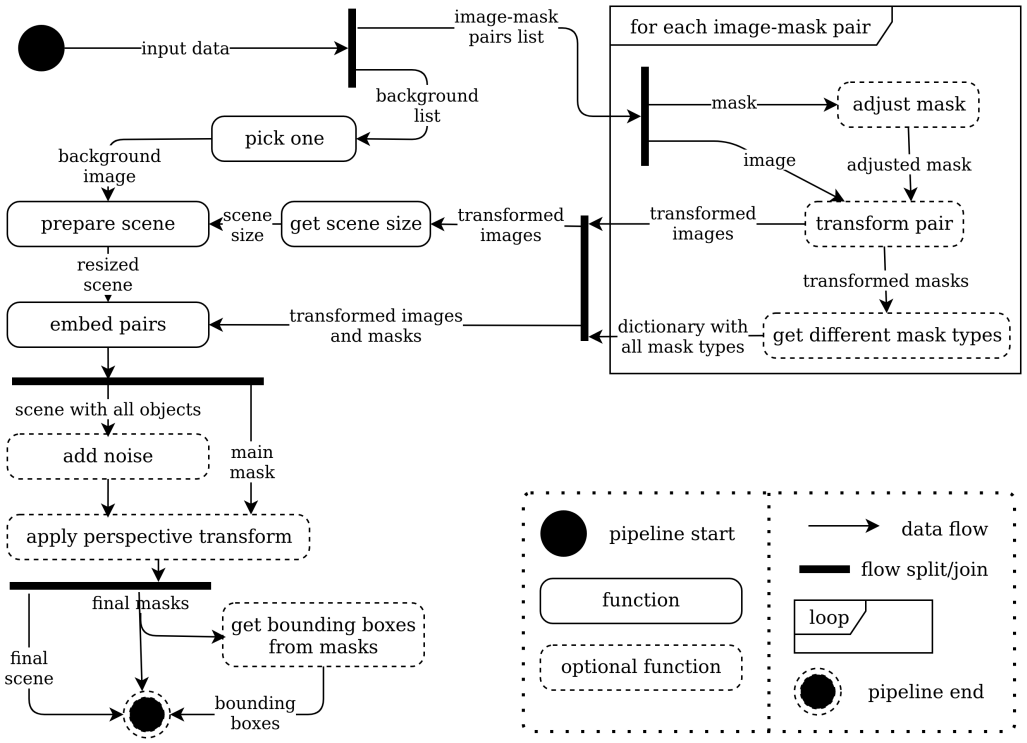


Figure 3. Transformation pipeline activity diagram.

The first step is to select the required number of image–mask pairs from a dataset. By default, we pick objects with repetitions that enable us to create scenes with a larger number of objects than present in the input data.

After that, we prepare images and masks before combining them into a single scene. The procedure is as follows:

- adjust the masks to exclude large margins;
- perform the same random transformations to both the image and mask;
- obtain all required mask types and auxiliary data.

Once all the transformations are performed and we know the sizes of all objects, the size of the output scene is calculated. Note that input objects can have different sizes and orientations; therefore, we cannot simply place objects by grid because it will lead to inefficient space usage. It is also not a good idea to place objects randomly in most cases because it will lead to uncontrollable overlapping of objects.

Within the framework of our approach, the objects are packed using the Maximal Rectangles Best Long Side Fit (MAXRECTS-BLSF) algorithm. It is a greedy algorithm that is aimed at packing rectangles of different sizes into a bin using the smallest possible area. The maximum theoretical packaging space overhead of the MAXRECTS-BLSF algorithm is

0.087. The BLSF modification of the algorithm tries to avoid a significant difference between side lengths. However, similar ot other rectangular packing algorithms, this one also tends to abuse the height dimension of the output scene, yielding a column-oriented result.

In order to control both overlapping of the objects and the orientation of output scenes, we introduce two modifications to the MAXRECTS-BLSF algorithm.

Control of the overlapping is achieved via substituting the objects' real sizes with the shrinked ones when passing them through the packing algorithm. The height and width are modified according to Equation (1):

$$\widetilde{H} = (1-s)H; \widetilde{W} = (1-s)W, \tag{1}$$

where s ranges from 0 to 1 inclusively.

The bigger the shrinkage ratio, the smaller the substituted images. It is applied to both height and width and to all of the input objects. The real overlapping area in practice will vary depending on each objects' shape and position. To perceive the overlap percentage, see Figure 4. Here, we consider the case where all input objects are squares without any holes. In other words, it is the maximum possible overlap percentage for the defined shrinkage ratio. We show this value for an object in the corner of a scene, an object on the side, and an object in the middle, separately.

We recommend choosing s between 0 and 0.3; however, taking into consideration sparse input masks, it can be slightly higher.

To control the orientation of the output scene, we set a hard limit of the scene height for the packing algorithm. Assuming that input objects will have different sizes in practice, we cannot obtain optimal packing with the fixed output image size or width-to-height ratio. To calculate the hard height limit, we use Equation (2).

$$\hat{h} = max\left(maxH, \theta\frac{\sum_{i=1}^{n}\widetilde{H}_i}{\lceil\sqrt{n}\rceil}\right) \tag{2}$$

The fraction in Equation (2) estimates the required value of height to make a square scene. We choose a maximum between it and the biggest objects' height to ensure that it is enough space for any input object. The orientation coefficient θ can be treated as the target width-to-height ratio. It will not produce the scenes with the fixed ratio, but with many samples, the average value will approach the target one. $\theta = 1$ will try to obtain square scenes. $\theta > 1$ will generate landscape scenes. In our experiments, we set θ to 1.2 to obtain close to square images with landscape preference. The average resulting width-to-height ratio over ten thousand samples was 1.1955.

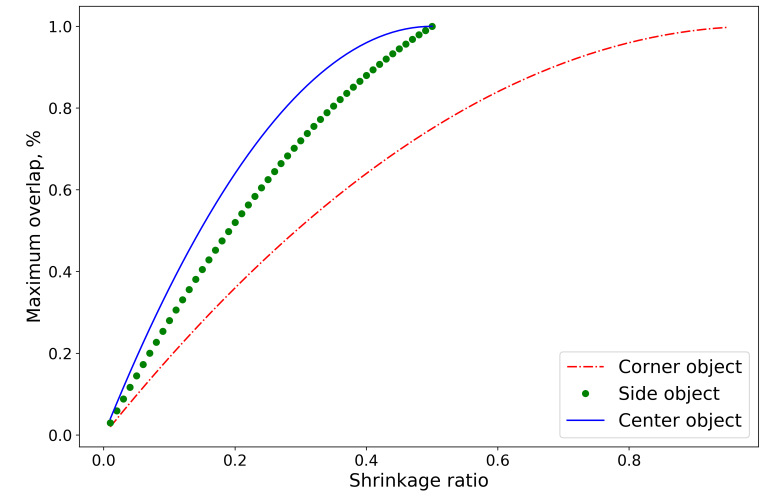
To adjust the background image size to the obtained scene size, we resize the background if it is smaller than the scene or randomly crop it if it is bigger.

We generate the required number of colors, excluding black and white, and find their Cartesian product according to Algorithm 1 for coloring the *MO* and *MP* masks.

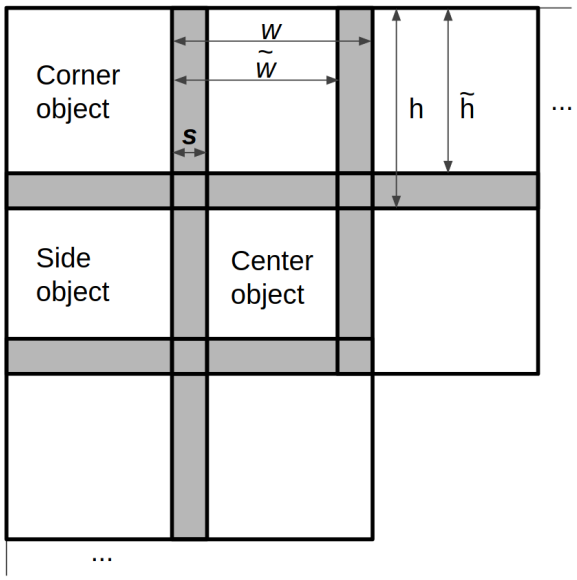
Algorithm 1: Color generation.

Input: Number of objects n;
Output: The set of colors C;
$L = \lceil\sqrt[3]{n+2}\rceil$
$s = \frac{1}{L}$
for $l = 0, ..., L-1$ **do**
    $T \leftarrow 1 - (s * l)$
end
return $C = \{(c_1, c_2, c_3) | c_1, c_2, c_3 \in T\}$

To preserve the correspondence between the input objects and their representation on the final scene, we color the objects in order of their occurrence.



(**a**)



(**b**)

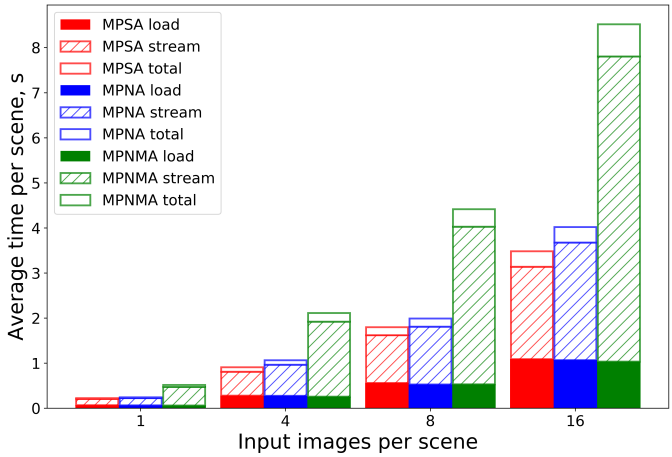
Figure 4. Shrinkage ratio effect illustration. (**a**) The dependency of maximum object overlap on shrinkage ratio. (**b**) Simplified scene generation example.

2.4. Time Performance

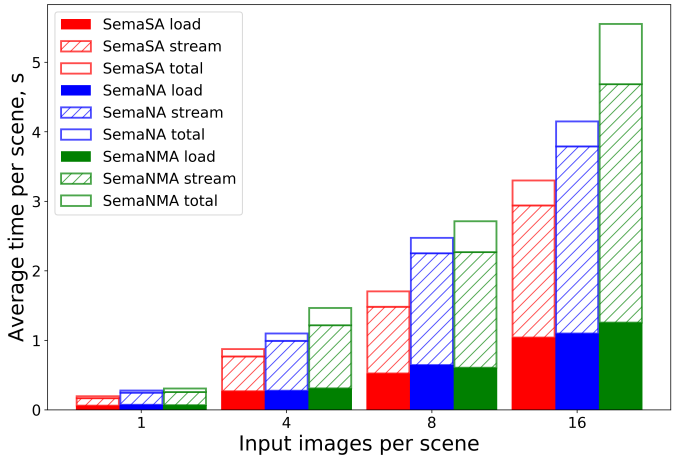
In this section, we measure the average time that is required to generate scenes of various complexity. For this experiment, we use Intel(R) Core(TM) i7-7700HQ CPU 2.80 GHz without multiprocessing. The average height of objects in the dataset is 385 pixels; the average width is 390 pixels. The results are averaged on a thousand scenes for each parameter combination and are reflected in Figure 5a for *MultiPartAugmentor* and Figure 5b for *SemanticAugmentor*.

SA (the red bar on the left) stands for *Simple Augmentor* with one type of output mask; *NA* (the blue bar in center) means adding noise and smoothing to scenes; *NMA* (the green bar on the right) means adding noise, smoothing, calculating bounding boxes, and generating all possible types of output masks. To recall possible mask types for each augmentor, refer to Table 2. The filled area in the bottom shows the time for loading input images and masks from disks. The shaded area in the middle shows the time for actual transformation. The empty area in the top shows the time for saving all the results to the disk. If every bar is accumulated with all the bars below it, the top of the shaded bar will

show the time for *StreamingDataGen*, and the top of the empty area will show the time for *SavingDataGen*.



(**a**)



(**b**)

Figure 5. Average scene-generating time with (**a**) *MultiPartAugmentor* and (**b**) *SemanticAugmentor*.

From the bar plots, you can see linear dependence between the number of input objects and the time for generating a scene.

2.5. System Parameters

Two main classes of the system where we can choose parameters are *Augmentor* and *DataGen*, or classes inherited.

The *Augmentor* parameters that define the transformations are shown in Table 3.

Table 3. Augmentor transformation parameters.

Operation	Description	Range	Default Value
Shrinkage ratio	See Figure 4 for details	[0...1)	0
Rotation	The maximum angle of image and mask rotation	[0...180]	180
Flip probability	The probability to flip the image and mask horizontally	[0...1]	0.5
Smooth	The size of the Gauss kernel applied for image smoothing	1, 3, ...	1
Perspective transform	The share of added width before perspective transform	[0...3]	0

The rest of the *Augmentor* parameters define output mask types, bounding box presence, and mask preprocessing steps.

The data generator parameters define the rules to pick samples for scenes: the number of samples per scene, picking samples for a single scene from the same class or randomly, class balancing rule, the input file structure, the output file structure.

2.6. Background Image Choosing

Making many augmented copies of objects is a very powerful tool used to increase dataset variability. However, many previous works underestimate the role of image context. The role of the context in an image plays a role in its background. In this paper, we show that the proper choice of a background is vital. For this, we experiment with methods that produce images that are similar to the test set backgrounds.

In the test set, we have five types of background. It includes: grass, floor tiles, wooden table, color blanket, and shop shelves. Therefore, we want to obtain suitable images that represent every surrounding type. The corresponding text prompts are:

- **grass**: grass, green grass, grass on the Earth, photo of grass, grass grown on the Earth;
- **floor tiles**: tile, ceramic tile, beige tile, grey tile, metal, photo of metal sheet, metal sheet, tile on the floor, close photo of tile, close photo of grey tile;
- **wooden table**: wood, wooden, wooden table, dark wooden table, light wooden table, close photo of wooden table, close photo of table in the room;
- **color blanket**: veil, cover, blanket, color blanket, dark blanket, blanket spread, bed linen, close photo of veil (cover, blanket), blanket on the bed, towel, green towel, close photo of towel on the table;
- **shop shelves**: shelves, shop shelves, close photo of shop shelves, white shop shelves, shop shelves close, table in shop, empty shelves in the shop, table with scales in front of shop shelves, scales in the shop.

We also split backgrounds into easy: wooden table, floor tiles; and complex: grass, color blanket, shop shelves. This split is manual and serves to demonstrate the difference in performance between more and less realistic images. More precisely, complex backgrounds are ones where visual augmentation looks unrealistic. Various background properties are significant not only in the agriculture domain, they represent different environmental conditions in the remote sensing domain and can be considered to boost model performance through geographical regions [47]. Background complexity in CV tasks for self-driving cars depends on urban area complexity and lighting conditions and has to be taken into account to develop robust algorithms [48]. To capture observed scenes for aerial vehicle navigation, surrounding properties are also crucial [49].

We use the described above text prompts with ruDALL-E [50] and stable diffusion [51] models to generate similar images, and with the CLIP [52] model to retrieve similar images from the LAION-400M [53] dataset. There are 100 collected backgrounds for each prompt.

For the comparison, we also add the worst-case and the best-case backgrounds. As the worst case, we propose to use random pattern images. The best case is to have real images from the same place, where a CV model will be inferenced.

Dataset

To verify the proposed approach, we conduct experiments using a set of images of various fruits and vegetables. We collect a unique dataset that comprises the following species: apple, cabbage, grape, tomato, pepper sweet, and onion. The dataset has hierarchical structure where each species includes three varieties, as is depicted in Figure 6. All species and varieties are presented in Table 4. Overall, each individual fruit or vegetable variety is represented by 150 images gained in different environmental and lighting conditions. We create a manual instance segmentation annotation for the images. Each image contains several fruits or vegetables of a single variety. Therefore, instance segmentation markup can be easily automatically converted into image classification labels. We can also obtain bounding boxes for object detection based on instance segmentation masks. Hence, we

create annotations for three CV tasks, namely, semantic segmentation, image classification, and object detection. For each task, the dataset is split into training and testing in an 80/20 ratio.

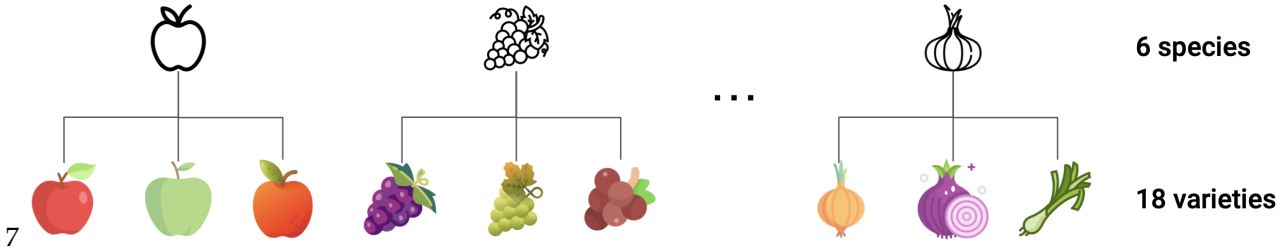


Figure 6. The hierarchical structure of the collected dataset.

Table 4. Species and varieties presented in the dataset.

Species	Varieties
Apple	Granny, Red delicious, Golden
Cabbage	Cauliflower, Peking, White
Grape	Black, Green, Pink
Tomato	Bull heart, Pink, Slivka
Pepper sweet	Green, Red, Yellow
Onion	Yellow, White, Purple

Figure 7 depicts generated images using the original dataset with instance segmentation masks.

Figure 7. Example of generated images using CISA instance-level augmentation.

2.7. Experiments

The experiment setup is as follows. We have to test the stability of our approach under various conditions. For this, we experiment with three CV tasks:

- image classification;
- semantic segmentation;
- object detection.

For each task, we compare:

- easy 6-species setup;
- complex 18-varieties setup.

For the classification task, we also compare different type of models:

- convolutional model (ResNet50 [54]);
- transformer model (SWIN [55]).

As well as models with different capacities:

- medium (ResNet50);
- small (MobileNetv3 [56]).

We set the following hyperparameters: For the ResNet50 training, we choose: a learning rate of 10^{-3} , cross-entropy loss function, SGD optimizer, exponential learning rate decay with gamma set to 0.95, and weight decay 2×10^{-3}.

For the MobileNetv3 training, we choose: a learning rate of 10^{-2}, cross-entropy loss function, SGD optimizer, exponential learning rate decay with gamma set to 0.95, and weight decay 3×10^{-4}.

For the SWIN training, we choose: a learning rate of 5×10^{-4}, cross-entropy loss function, Adam optimizer, cosine annealing learning rate decay, and weight decay 10^{-5}.

For the UNET++ training, we choose: a learning rate of 3×10^{-5}, binary cross-entropy with logits loss function, Adam optimizer, cosine annealing learning rate decay, and weight decay 10^{-5}. Images were resized to 512×512 px.

For YOLOv8 training, we choose: a learning rate of 10^{-3}, SGD optimizer, exponential learning rate decay with gamma set to 0.95, and weight decay 5×10^{-4}. Images were resized to 640×640 px.

We explicitly compare convolutional [57] and transformer [58] models. These are the two most popular types of computer vision models today. They differ in receptive field. Convolutions operate locally (Equation (3)), while transformers look at the greater scale (Equation (4)). The success of augmentation with one model type does not guarantee success with another.

$$O[x, y] = (I * K)(i, j) = \sum_{j=1} \sum_{i=1} I[x - i, y - j]K[i, j], \tag{3}$$

where O is the resulting feature map; K is a kernel.

$$A(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{4}$$

where Q, K and V are weight matrices; d is the dimensionality of an attention head.

In each experiment, we measure the model performance using five-fold cross-validation. We use early stopping to terminate model training; therefore, the number of training epochs for different models varies. Classification models are pre-trained on the ImageNet dataset. Segmentation and detection models are pre-trained on the COCO dataset.

We compare several ways to find backgrounds that match the context of the test set, including Contrastive Language–Image Pre-Training (CLIP) [52] image retrieval, VQGAN (ruDALL-E [50]) image generation, and diffusion (Stable Diffusion [51]) image generation.

In each experiment excluding the baseline, we first pre-train a model on the CISA-augmented dataset and then fine-tune the original dataset.

2.8. Evaluation Metrics

To determine the suitability of the training dataset prior to the training procedure, we propose to use the Fréchet Inception Distance (FID) metrics [59]. It is a commonly used choice to evaluate the performance of GAN models. FID measures distance between the distribution of generated images and the original natural samples. However, in our case, the idea behind FID computation is to determine the similarity and feasibility of the generated training samples and test data. A low FID value depicts the better case when we manage to obtain an artificially realistic dataset close to the original test dataset distribution. To compute FID, we use Equation (5).

$$FID = ||\mu_r - \mu_g||^2 + T_r(\sum_r + \sum_g - 2\sqrt{(\sum_r \sum_g)}), \tag{5}$$

where r and g indexes denote real and generated datasets, correspondingly; μ is the mean of the Inceptionv3 model [60] features of a dataset; $\sum_{dataset}$ is the variance matrix of a dataset; T_r is the trace operator.

For assessing classification results, we use accuracy, because the dataset is balanced.

To evaluate semantic segmentation, we calculate pixel-wise intersection over union (IoU, Equation (6)).

$$IoU = \frac{TP}{TP + FP + FN}, \tag{6}$$

where TP is the number of true positive samples; FP is the number of false positive samples; FN is the number of false negative samples.

To evaluate object detection results, we calculate mAP@0.5 (Equation (7)). It means that for the prediction, we use the threshold $IoU = 0.5$.

$$mAP@0.5 = \frac{1}{\#classes} \sum_{c \in classes} \frac{TP(c)}{TP(c) + FP(c)}, \tag{7}$$

To measure the statistical significance of our results, we calculate the Spearman rank-order correlation coefficient (Equation (8)). We choose Spearman's over Pearson's correlation because the relation between the FID and accuracy is monotonous but non-linear.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}, \tag{8}$$

where ρ is the Spearman's correlation coefficient; d_i is the distance between two ranks of each observation; n is the number of observations.

3. Results

The results of the experiments are shown in Tables 5–14.

In Table 5, one can find the results of the classification of six species with the ResNet50 model. CISA with stable diffusion backgrounds shows a 2.3% relative percentage change compared with the baseline.

In Table 6, one can find the results of the classification of 18 varieties with the ResNet50 model. CISA with stable diffusion backgrounds shows a 14.2% relative percentage change compared with the baseline.

In Table 7, one can find the results of the classification of six species with the MobileNetv3 model. CISA with stable diffusion backgrounds show a 1.2% relative percentage change compared with the baseline.

Table 5. Classification results for ResNet50 model on test images for six species.

Source of Augmentation Background	Prompts	Pre-Training Accuracy ↑	Fine-Tuned Accuracy ↑	FID ↓
Baseline	—	—	95.2 ± 0.7	—
Patterns	—	93.5 ± 1.2	94.7 ± 0.8	12.93
CLIP	easy	95 ± 0.9	97 ± 0.6	10.76
	complex	95 ± 1	96.6 ± 0.7	10.92
	all	95 ± 1	96.7 ± 0.7	9.6
ruDALL-E	all	94 ± 0.9	95.5 ± 0.8	11.1
Stable Diffusion	easy	95 ± 0.9	**97.4 ± 0.5**	9.43
	complex	94.9 ± 1	97.1 ± 0.6	9.81
	all	95 ± 1	97.3 ± 0.6	8.7
Natural backgrounds	easy	95.8 ± 0.7	98 ± 0.4	7.15
	complex	95.1 ± 0.8	97.8 ± 0.4	7.9
	all	95.3 ± 0.8	98 ± 0.4	6.14

The bold value depicts the best model, excluding models that are trained with natural backgrounds.

Table 6. Classification results for ResNet50 model on test images for 18 varieties.

Source of Augmentation Background	Prompts	Pre-Training Accuracy ↑	Fine-Tuned Accuracy ↑	FID ↓
Baseline	—	—	50 ± 2.3	—
Patterns	—	48 ± 2.5	54.9 ± 2.3	12.93
CLIP	easy	49.5 ± 3	56.4 ± 2.2	10.76
	complex	49 ± 2.7	56.1 ± 2.3	10.92
	all	49.3 ± 2.9	56.3 ± 2.1	9.6
ruDALL-E	all	49 ± 3	56 ± 2.4	11.1
Stable Diffusion	easy	50.5 ± 2.8	**57.1 ± 1.9**	9.43
	complex	50 ± 3.1	56.9 ± 2	9.81
	all	50.2 ± 2.9	**57.1 ± 1.8**	8.7
Natural backgrounds	easy	50.8 ± 2.2	57.4 ± 1.7	7.15
	complex	49.6 ± 3	56.8 ± 1.9	7.9
	all	50.1 ± 2.4	57.2 ± 1.8	6.14

Table 7. Classification results for MobileNetv3 model on test images for six species.

Source of Augmentation Background	Prompts	Pre-Training Accuracy ↑	Fine-Tuned Accuracy ↑	FID ↓
Baseline	—	—	90 ± 1.3	—
Patterns	—	88 ± 2.2	89.9 ± 1.1	12.93
CLIP	easy	90 ± 1.7	90.9 ± 1.1	10.76
	complex	89.1 ± 1.9	90.7 ± 1.2	10.92
	all	89.7 ± 1.9	90.9 ± 1	9.6
ruDALL-E	all	89 ± 2	90.8 ± 1.2	11.1
Stable Diffusion	easy	90 ± 1.5	**91.1 ± 1**	9.43
	complex	89.4 ± 1.8	90.9 ± 0.9	9.81
	all	89.8 ± 1.6	91 ± 0.9	8.7
Natural backgrounds	easy	90 ± 1.6	91.3 ± 0.9	7.15
	complex	88.9 ± 2	90.8 ± 1	7.9
	all	89.8 ± 1.4	91.2 ± 1	6.14

In Table 8, one can find the results of the classification of 18 varieties with the MobileNetv3 model. CISA with stable diffusion backgrounds show a 6.6% relative percentage change compared with the baseline.

Table 8. Classification results for MobileNetv3 model on test images for 18 varieties.

Source of Augmentation Background	Prompts	Pre-Training Accuracy ↑	Fine-Tuned Accuracy ↑	FID ↓
Baseline	—	—	38 ± 3.1	—
Patterns	—	36.5 ± 2.8	39.5 ± 2.3	12.93
CLIP	easy	37 ± 3	39.8 ± 2.7	10.76
	complex	36.8 ± 2.7	39.6 ± 2.5	10.92
	all	37 ± 2.9	39.8 ± 2.8	9.6
ruDALL-E	all	37.2 ± 3.1	39.9 ± 2.5	11.1
Stable Diffusion	easy	37.9 ± 3	40.4 ± 2.6	9.43
	complex	37.3 ± 3.2	40 ± 2.7	9.81
	all	37.7 ± 2.9	**40.5 ± 2.6**	8.7
Natural backgrounds	easy	38 ± 2.4	40.9 ± 2.1	7.15
	complex	37.2 ± 2.8	40.4 ± 2.4	7.9
	all	37.9 ± 3	40.8 ± 2.3	6.14

In Table 9, one can find the results of the classification of six species with the SWIN model. CISA with stable diffusion backgrounds show a 1% relative percentage change compared with the baseline.

Table 9. Classification results for SWIN model on test images for six species.

Source of Augmentation Background	Prompts	Pre-Training Accuracy ↑	Fine-Tuned Accuracy ↑	FID ↓
Baseline	—	—	96.8 ± 0.5	—
Patterns	—	92.8 ± 1.1	95.9 ± 0.7	12.93
CLIP	easy	93.9 ± 1	97.5 ± 0.6	10.76
	complex	94.2 ± 0.8	97.6 ± 0.5	10.92
	all	94.1 ± 0.9	97.6 ± 0.6	9.6
ruDALL-E	all	93 ± 1	96.6 ± 0.5	11.1
Stable Diffusion	easy	94.1 ± 0.8	97.7 ± 0.6	9.43
	complex	94.2 ± 0.9	97.7 ± 0.5	9.81
	all	94.3 ± 0.8	**97.8 ± 0.4**	8.7
Natural backgrounds	easy	94.7 ± 0.8	98.1 ± 0.5	7.15
	complex	94.9 ± 0.6	98.2 ± 0.4	7.9
	all	94.9 ± 0.7	98.2 ± 0.3	6.14

In Table 10, one can find the results of the classification of 18 varieties with SWIN model. CISA with stable diffusion backgrounds show a 6.4% relative percentage change compared with the baseline.

In Table 11, one can find the results of the semantic segmentation of six species with the UNET++ model. CISA with stable diffusion backgrounds show a 2.7% relative percentage change compared with the baseline.

In Table 12, one can find the results of the semantic segmentation of 18 varieties with the UNET++ model. CISA with stable diffusion backgrounds show a 6% relative percentage change compared with the baseline.

In Table 13, one can find the results of the object detection of six species with the YOLOv8 model. CISA with stable diffusion backgrounds show a 2.2% relative percentage change compared with the baseline.

In Table 14, one can find the results of the object detection of 18 varieties with the YOLOv8 model. CISA with stable diffusion backgrounds show a 6.8% relative percentage change compared with the baseline.

Table 10. Classification results for SWIN model on test images for 18 varieties.

Source of Augmentation Background	Prompts	Pre-Training Accuracy ↑	Fine-Tuned Accuracy ↑	FID ↓
Baseline	—	—	51.4 ± 2	—
Patterns	—	47.5 ± 2.6	52 ± 2	12.93
CLIP	easy	48.8 ± 2.7	53.9 ± 1.8	10.76
	complex	49.1 ± 2.5	54 ± 2	10.92
	all	49 ± 2.4	54 ± 1.9	9.6
ruDALL-E	all	48.4 ± 2.8	53 ± 2.1	11.1
Stable Diffusion	easy	49.8 ± 2.7	54.5 ± 1.8	9.43
	complex	49.9 ± 2.9	**54.7 ± 1.7**	9.81
	all	49.9 ± 2.6	54.6 ± 1.6	8.7
Natural backgrounds	easy	50.2 ± 2.1	55.1 ± 1.8	7.15
	complex	50.3 ± 2.3	55 ± 1.8	7.9
	all	50.4 ± 2.2	55.1 ± 1.6	6.14

Table 11. Segmentation results for UNET++ model on test images for six species.

Source of Augmentation Background	Prompts	Pre-Training IoU ↑	Pre-Training Accuracy ↑	Fine-Tuned IoU ↑	Fine-Tuned Accuracy ↑	FID ↓
Baseline	—	—	—	89.5 ± 0.3	95.4 ± 0.25	—
Patterns	—	85 ± 0.6	91.7 ± 0.5	91.2 ± 0.6	96.3 ± 0.3	12.93
CLIP	easy	87.3 ± 0.3	93.2 ± 0.2	93.5 ± 0.3	**98.2 ± 0.1**	10.76
	complex	86.9 ± 0.4	92.9 ± 0.4	93.4 ± 0.2	98.1 ± 0.1	10.92
	all	87.2 ± 0.4	93.1 ± 0.3	93.6 ± 0.3	98.1 ± 0.1	9.6
ruDALL-E	all	86.4 ± 0.6	92.2 ± 0.4	91.9 ± 0.5	97.7 ± 0.2	11.1
Stable Diffusion	easy	88.3 ± 0.3	94.1 ± 0.3	94.5 ± 0.2	98 ± 0.2	9.43
	complex	86.9 ± 0.5	93.8 ± 0.3	93.8 ± 0.2	97.9 ± 0.2	9.81
	all	88.2 ± 0.3	94.1 ± 0.2	94.4 ± 0.3	98 ± 0.2	8.7
Natural backgrounds	easy	88.8 ± 0.3	94.6 ± 0.3	95.3 ± 0.1	98.2 ± 0.15	7.15
	complex	88.6 ± 0.4	94.3 ± 0.2	94.8 ± 0.3	98.2 ± 0.15	7.9
	all	88.8 ± 0.4	94.5 ± 0.3	95.2 ± 0.2	98.2 ± 0.15	6.14

Table 12. Segmentation results for UNET++ model on test images for 18 varieties.

Source of Augmentation Background	Prompts	Pre-Training IoU ↑	Pre-Training Accuracy ↑	Fine-Tuned IoU ↑	Fine-Tuned Accuracy ↑	FID ↓
Baseline	—	—	—	74.5 ± 0.5	85.6 ± 0.5	—
Patterns	—	70.2 ± 0.9	81.9 ± 0.8	73.2 ± 0.6	85.8 ± 0.6	12.93
CLIP	easy	72 ± 0.5	84.7 ± 0.5	78.1 ± 0.4	89.8 ± 0.5	10.76
	complex	71.9 ± 0.8	84.6 ± 0.5	77.3 ± 0.3	89.6 ± 0.4	10.92
	all	72.1 ± 0.7	84.7 ± 0.6	77.5 ± 0.4	89.9 ± 0.4	9.6
ruDALL-E	all	71.6 ± 0.6	84.3 ± 0.7	76.1 ± 0.5	89.2 ± 0.5	11.1
Stable Diffusion	easy	72.9 ± 0.5	85.5 ± 0.35	80 ± 0.3	90.5 ± 0.4	9.43
	complex	71.4 ± 0.7	84.8 ± 0.4	78.9 ± 0.4	89.6 ± 0.5	9.81
	all	72.5 ± 0.5	85.4 ± 0.4	80.2 ± 0.4	**90.7 ± 0.4**	8.7
Natural backgrounds	easy	73.9 ± 0.6	85.5 ± 0.4	81.7 ± 0.2	91.8 ± 0.3	7.15
	complex	71.8 ± 0.7	84.6 ± 0.5	80.9 ± 0.3	91.6 ± 0.4	7.9
	all	73.5 ± 0.6	85.5 ± 0.4	81.5 ± 0.3	91.9 ± 0.3	6.14

Table 13. Object detection for YOLOv8 model on test images for six species.

Source of Augmentation Background	Prompts	Pre-Training mAP ↑	Fine-Tuned mAP ↑	FID ↓
Baseline	—	—	57.9 ± 0.5	—
Patterns	—	54.9 ± 0.4	58.2 ± 0.4	12.93
CLIP	easy	55.6 ± 0.4	59 ± 0.3	10.76
	complex	55.7 ± 0.5	58.9 ± 0.4	10.92
	all	55.6 ± 0.6	58.9 ± 0.3	9.6
ruDALL-E	all	55.2 ± 0.6	58.9 ± 0.5	11.1
Stable Diffusion	easy	55.7 ± 0.4	59.1 ± 0.3	9.43
	complex	55.5 ± 0.5	59 ± 0.5	9.81
	all	55.7 ± 0.3	**59.2 ± 0.4**	8.7
Natural backgrounds	easy	56.1 ± 0.6	60.1 ± 0.3	7.15
	complex	56.2 ± 0.4	60.2 ± 0.4	7.9
	all	56.2 ± 0.5	60.1 ± 0.3	6.14

Table 14. Object detection for YOLOv8 model on test images for 18 varieties.

Source of Augmentation Background	Prompts	Pre-Training mAP ↑	Fine-Tuned mAP ↑	FID ↓
Baseline	—	—	38.3 ± 1.1	—
Patterns	—	35.6 ± 1.2	39.2 ± 0.6	12.93
CLIP	easy	36.1 ± 0.9	40.2 ± 0.8	10.76
	complex	35.9 ± 1.2	40 ± 0.8	10.92
	all	36.1 ± 1.1	40.2 ± 0.9	9.6
ruDALL-E	all	36.2 ± 1.1	40.5 ± 1	11.1
Stable Diffusion	easy	36.7 ± 0.7	40.7 ± 0.9	9.43
	complex	36.8 ± 0.9	**40.9 ± 0.7**	9.81
	all	36.7 ± 0.8	40.9 ± 0.8	8.7
Natural backgrounds	easy	37 ± 1	41.4 ± 0.7	7.15
	complex	37.1 ± 1	41.3 ± 0.7	7.9
	all	37 ± 0.9	41.4 ± 0.6	6.14

Figure 8 shows the segmentation model predictions on the test images. The source of augmentation background for this model training is stable diffusion.

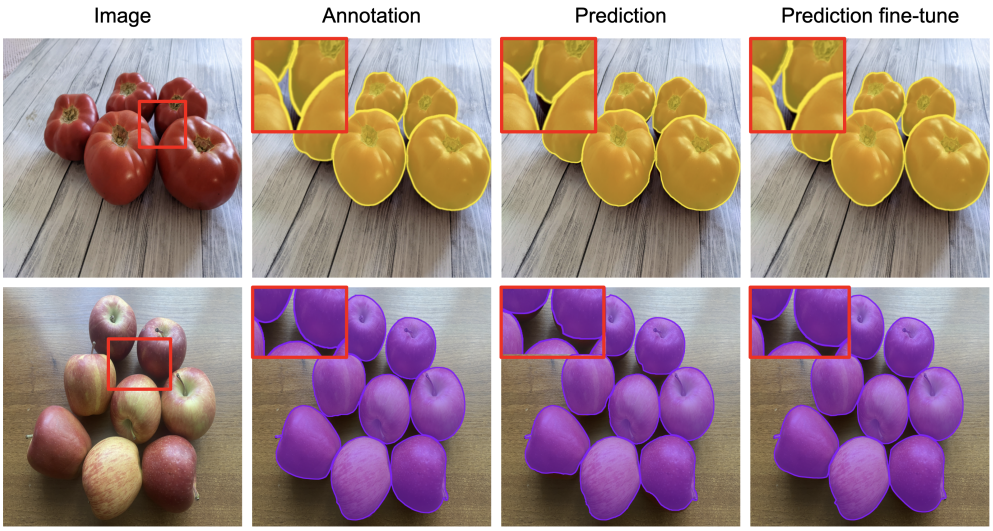


Figure 8. Example of model predictions.

4. Discussion

4.1. CISA Efficiency

Our experiments show that CISA instance-level augmentation provides a stable improvement for all of the tested CV tasks. This works both for convolutional and transformer models. The major observation is the importance of the context. Note that with random patterns, augmentation sometimes works worse than the baseline.

The best choice is to use a natural background from the location where the CV system will be used. This is possible when the camera is stationary. If there are multiple camera locations, it is better to collect background images from all of them. Recall that background images do not require any manual annotation.

Any other approach to collect similar images gives substantial improvement in comparison with other augmentation approaches. Both image retrieval and image generation show promising results. In our experiments, stable diffusion beats all other approaches for the majority of cases.

For more complex tasks, the boost is higher. The natural training dataset is still required for fine-tuning. The results from the approach without the fine-tuning are worse than the baseline.

Table 15 as well as Figures 9 and 10 show the correlation between the model performance and FID. One can see that if an augmented training set is similar to the test set, it will result in higher accuracy. It allows for choosing a better set of backgrounds without model training. For more complex tasks, the correlation seems to be lower. For segmentation and detection tasks, the correlation is very high.

Table 15. Correlation.

Model	Task	#Classes	Correlation	p Value ↓
ResNet50	classification	6	−0.64	4×10^{-13}
ResNet50	classification	18	−0.27	10^{-3}
MobileNetv3	classification	6	−0.2	2×10^{-1}
MobileNetv3	classification	18	−0.18	4×10^{-2}
SWIN	classification	6	−0.65	2×10^{-10}
SWIN	classification	18	−0.37	8×10^{-3}
UNET++	segmentation	6	−0.94	2×10^{-25}
UNET++	segmentation	18	−0.95	2×10^{-26}
YOLOv8	detection	6	−0.75	3×10^{-11}
YOLOv8	detection	18	−0.57	6×10^{-11}

The importance of context for image augmentation has been previously demonstrated in [37], where the authors created an additional neural network to select a proper location on a new background to paste the target object. In turn, we focus on the retrieval and generation of an extensive dataset using various sources of background images. Although the proposed approach does not involve additional generative models for dataset augmentation, it is a simple and powerful way to adjust recognition model performance. CISA instance-level augmentation extends the pioneering research on image augmentation [35] and recent studies [36], and it allows one to estimate dataset suitability before model training based on FID measures between original and generated datasets.

4.2. Limitations

The proposed image augmentation scheme can be used when we have masks for input images. The system can work with instance segmentation masks and semantic segmentation masks. However, if there are no instance masks available, one can try to generate pseudo-segmentation masks.

The system's primary usage involves generating complex scenes from simple input data; however, the scene can include a single object if needed. The key feature of the system is its ability to generate a huge amount of training samples even for the task for which the original dataset was not designed. For instance, having only an image and a

multi-part mask as input, we can produce samples for instance segmentation, instance parts segmentation, object detection, object counting, denoising, and classification. The described system can also be beneficial for few-shot learning when the original dataset is minimal.

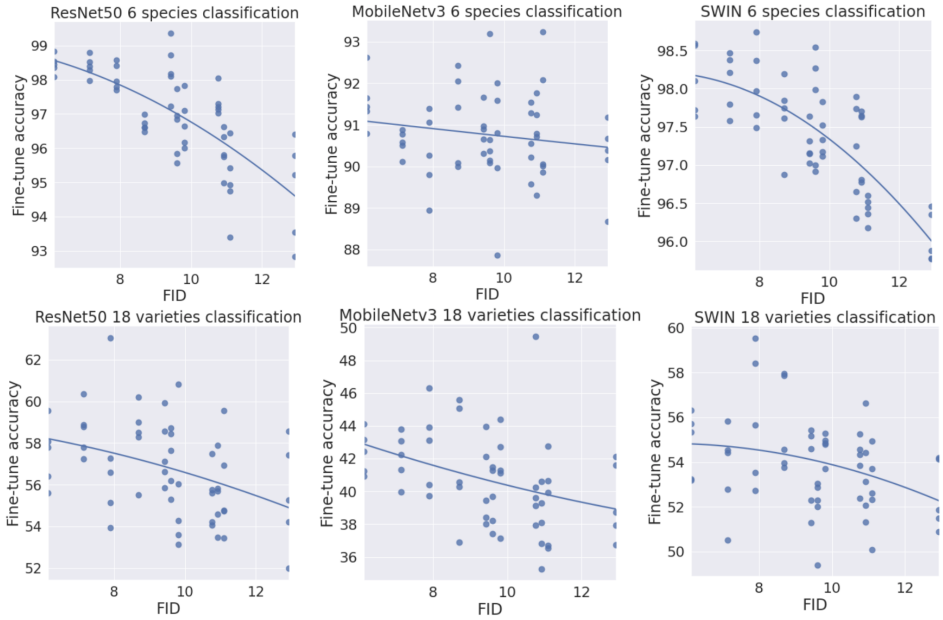


Figure 9. Relation between FID and accuracy in the classification task.

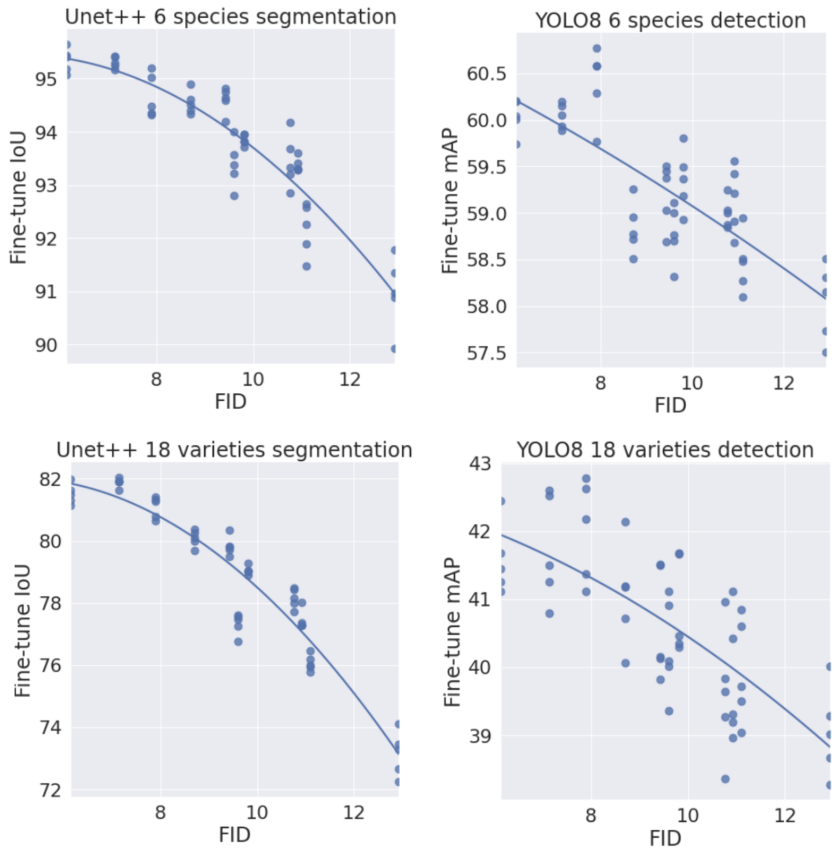


Figure 10. Relation between FID and IoU in the segmentation task and mAP in the object detection task.

To apply the proposed augmentation scheme successfully, the dataset should not be exceedingly sensitive to scene geometry, since such behavior can be undesirable in some cases. For example, if you use a dataset of people or cars, the described approach by default can place one object on top of the other. Nevertheless, we can add some extra height limitations or use perspective transformation in these cases.

Another point is that we should find appropriate background images that would fit some particular case. Retrieval-based approaches used to generate new training samples using CLIP can be significantly impeded, in particular, domains such as medical or remote sensing. For instance, in [43], the authors aimed to generate thermal images, with defective areas occurring due to the manufacturing process. It is a more complex task to retrieve such unique backgrounds using CLIP. However, there are various special data sources that do not contain annotated data but are useful as backgrounds for new samples. Another possible limitation is that if it is not possible to know the test set context, we may expect a slight performance drop.

Further study on CISA application for images derived from different sensors on different wavelengths should be conducted. Multispectral and hyperspectral data, radiography, and radar scanning have their own properties. Their artificial generation is currently under consideration in a number of works [61]. However, it is vital to take into account the nature of data, because image augmentations should not break any physical law of the studied objects.

Recall that it is important to fine-tune the model on natural images to increase the performance.

The time for scene generation is close to linear when we have enough memory to store all objects and overhead for a scene. To estimate the average required RAM per scene, we use Equation (9)

$$M = 3n\bar{h}\bar{w}[(1+m)p + o + 2] + `o \tag{9}$$

In this equation, we can neglect the overhead, $`o < o << M$, because it is considerably smaller than the data itself.

Although GAN-based image augmentation approaches are capable of providing more realistic images under certain conditions, the proposed CISA approach does not require computational resources to train an additional generative model.

5. Conclusions

In this article, we introduce an image augmentation technique for few-shot learning. The presented framework allows for generating large training datasets using only a few input samples. It also provides training data for the tasks, including instance segmentation, semantic segmentation, classification, object detection, and object counting, even if the original dataset contains annotations for the instance segmentation task only. To show our method's advantage, we compared the model performances on the tasks with different difficulties, we checked the models of different types and different capacities, and we showed the substantial improvement for all of the listed cases. The average percentage increase in accuracy across all the tasks on the fruits and vegetables recognition dataset is 4.95%. Moreover, we extensively explored approaches to collect background images, and we showed an efficient method used to choose the best background dataset without model training. WE showed that the Fréchet Inception Distance (FID) metrics has a strong correlation with model accuracy, and it can help to choose better backgrounds without model training. The average negative correlation between model accuracy and the FID between The augmented and test datasets was 0.55 in our experiments.

Author Contributions: Conceptualization, S.N.; methodology, S.N. and A.S.; software, I.Z.; validation, S.N. and S.I.; formal analysis, S.N. and I.Z.; investigation, I.Z. and S.N.; resources, S.V.B. and T.Y.; data curation, I.Z. and S.N.; writing—original draft preparation, S.N. and S.I.; writing—review and editing, A.S. and S.V.B.; visualization, S.N. and S.I.; supervision, A.S., V.D. and I.O.; project administration, S.N. and D.S.; funding acquisition, S.V.B. and T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The code is available at https://github.com/NesterukSergey/segmentation_image_augmentation, accessed on 27 February 2023. Data are shared at https://disk.yandex.com/d/VeTwxns9ncOqGA, accessed on 27 February 2023.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CISA	Context Image Semantics Augmentation framework
CLIP	Contrast Language-Image Pre-Training model
FID	Frechet Inception Distance
CV	Computer Vision
GAN	Generative Adversarial Network
cGAN	Conditional Generative Adversarial Network
VQGAN	Vector-Quantized Generative Adversarial Network
VAE	Variational Autoencoder
ML	Machine Learning
NLP	Natural Language Processing
RAM	Random Access Memory
MO	Multi-Object
MP	Multi-Part
MAXRECTS-BLSF	Maximal Rectangles Best Long Side Fit algorithm
SGD	Stochastic Gradient Descent
Adam	Adaptive Momentum Optimizer
IoU	Intersection over Union
mAP	Mean Average Precision

References

1. Kwon, O.; Sim, J.M. Effects of data set features on the performances of classification algorithms. *Expert Syst. Appl.* **2013**, *40*, 1847–1857. [CrossRef]
2. Sbai, O.; Couprie, C.; Aubry, M. Impact of base dataset design on few-shot image classification. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 597–613.
3. Zendel, O.; Murschitz, M.; Humenberger, M.; Herzner, W. How good is my test data? Introducing safety analysis for computer vision. *Int. J. Comput. Vis.* **2017**, *125*, 95–109. [CrossRef]
4. Barbedo, J.G.A. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Comput. Electron. Agric.* **2018**, *153*, 46–53. [CrossRef]
5. Zheng, S.; Song, Y.; Leung, T.; Goodfellow, I. Improving the robustness of deep neural networks via stability training. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4480–4488.
6. Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv* **2020**, arXiv:2006.16241.
7. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, USA, 20–25 June 2009; pp. 248–255.
8. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
9. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3974–3983.

10. Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; Carlos Niebles, J. Activitynet: A large-scale video benchmark for human activity understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 961–970.
11. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In *Proceedings of the International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 270–279.
12. Lemikhova, L.; Nesteruk, S.; Somov, A. Transfer Learning for Few-Shot Plants Recognition: Antarctic Station Greenhouse Use-Case. In Proceedings of the 2022 IEEE 31st International Symposium on Industrial Electronics (ISIE), Anchorage, AL, USA, 1–3 June 2022; pp. 715–720. [CrossRef]
13. Vannucci, M.; Colla, V. Classification of unbalanced datasets and detection of rare events in industry: issues and solutions. In *Proceedings of the International Conference on Engineering Applications of Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 337–351.
14. Nesteruk, S.; Shadrin, D.; Pukalchik, M.; Somov, A.; Zeidler, C.; Zabel, P.; Schubert, D. Image compression and plants classification using machine learning in controlled-environment agriculture: Antarctic station use case. *IEEE Sensors J.* **2021**, *21*, 17564–17572. [CrossRef]
15. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–34. [CrossRef]
16. Illarionova, S.; Nesteruk, S.; Shadrin, D.; Ignatiev, V.; Pukalchik, M.; Oseledets, I. Object-based augmentation for building semantic segmentation: Ventura and santa rosa case study. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1659–1668.
17. Illarionova, S.; Shadrin, D.; Ignatiev, V.; Shayakhmetov, S.; Trekin, A.; Oseledets, I. Augmentation-Based Methodology for Enhancement of Trees Map Detalization on a Large Scale. *Remote. Sens.* **2022**, *14*, 2281. [CrossRef]
18. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
19. Kuznichov, D.; Zvirin, A.; Honen, Y.; Kimmel, R. Data Augmentation for Leaf Segmentation and Counting Tasks in Rosette Plants. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 15–20 June 2019.
20. Fawakherji, M.; Potena, C.; Prevedello, I.; Pretto, A.; Bloisi, D.D.; Nardi, D. Data Augmentation Using GANs for Crop/Weed Segmentation in Precision Farming. In Proceedings of the 2020 IEEE Conference on Control Technology and Applications (CCTA), Montreal, QC, Canada, 24–26 August 2020; pp. 279–284.
21. Wu, Q.; Chen, Y.; Meng, J. DCGAN Based Data Augmentation for Tomato Leaf Disease Identification. *IEEE Access* **2020**. [CrossRef]
22. Nesteruk, S.; Shadrin, D.; Kovalenko, V.; Rodriguez-Sanchez, A.; Somov, A. Plant Growth Prediction through Intelligent Embedded Sensing. In Proceedings of the IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, The Netherlands, 17–19 June 2020; Volume 2020, pp. 411–416. [CrossRef]
23. Nesteruk, S.; Illarionova, S.; Akhtyamov, T.; Shadrin, D.; Somov, A.; Pukalchik, M.; Oseledets, I. XtremeAugment: Getting More From Your Data Through Combination of Image Collection and Image Augmentation. *IEEE Access* **2022**, *10*, 24010–24028. [CrossRef]
24. Nesteruk, S.; Bezzateev, S. Location-Based Protocol for the Pairwise Authentication in the Networks without Infrastructure. In Proceedings of the 2018 22nd Conference of Open Innovations Association (FRUCT), Jyvaskyla, Finland, 15–18 May 2018; pp. 190–197. [CrossRef]
25. Ching, T.; Himmelstein, D.S.; Beaulieu-Jones, B.K.; Kalinin, A.A.; Do, B.T.; Way, G.P.; Ferreo, E.; Agapow, P.-M.; Zirtz, M.; Hoffman, M.M.; et al. Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface* **2018**, *15*, 20170387. [CrossRef] [PubMed]
26. Feng, R.; Gu, J.; Qiao, Y.; Dong, C. Suppressing Model Overfitting for Image Super-Resolution Networks. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 15–20 June2019.
27. Illarionova, S.; Nesteruk, S.; Shadrin, D.; Ignatiev, V.; Pukalchik, M.; Oseledets, I. MixChannel: Advanced augmentation for multispectral satellite images. *Remote. Sens.* **2021**, *13*, 2181. [CrossRef]
28. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In *Proceedings of the Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 818–833.
29. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
30. Buslaev, A.; Parinov, A.; Khvedchenya, E.; Iglovikov, V.I.; Kalinin, A.A. Albumentations: Fast and flexible image augmentations. *Information* **2020**, *11*, 125. [CrossRef]
31. Zhu, Y.; Aoun, M.; Krijn, M.; Vanschoren, J.; Campus, H.T. Data Augmentation using Conditional Generative Adversarial Networks for Leaf Counting in Arabidopsis Plants. In Proceedings of the BMVC, Newcastle, UK, 3–6 September 2018; p. 324.
32. Valerio Giuffrida, M.; Scharr, H.; Tsaftaris, S.A. ARIGAN: Synthetic Arabidopsis Plants Using Generative Adversarial Network. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops, Venice, Italy, 22–29 October 2017.

33. Zoph, B.; Cubuk, E.D.; Ghiasi, G.; Lin, T.; Shlens, J.; Le, Q.V. Learning Data Augmentation Strategies for Object Detection. *arXiv* **2019**, arXiv:1906.11172.
34. Lemley, J.; Bazrafkan, S.; Corcoran, P. Smart Augmentation Learning an Optimal Data Augmentation Strategy. *IEEE Access* **2017**, *5*, 5858–5869. [CrossRef]
35. Dwibedi, D.; Misra, I.; Hebert, M. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1301–1310.
36. Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.Y.; Cubuk, E.D.; Le, Q.V.; Zoph, B. Simple copy-paste is a strong data augmentation method for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 2918–2928.
37. Dvornik, N.; Mairal, J.; Schmid, C. On the importance of visual context for data augmentation in scene understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 2014–2028. [CrossRef]
38. Su, Y.; Sun, R.; Lin, G.; Wu, Q. Context decoupling augmentation for weakly supervised semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 7004–7014.
39. Flores-Fuentes, W.; Trujillo-Hernández, G.; Alba-Corpus, I.Y.; Rodríguez-Quiñonez, J.C.; Mirada-Vega, J.E.; Hernández-Balbuena, D.; Murrieta-Rico, F.N.; Sergiyenko, O. 3D spatial measurement for model reconstruction: A review. *Measurement* **2023**, *207*, 112321. [CrossRef]
40. Barth, R.; IJsselmuiden, J.; Hemming, J.; Henten, E.V. Data synthesis methods for semantic segmentation in agriculture: A Capsicum annuum dataset. *Comput. Electron. Agric.* **2018**, *144*, 284–296. [CrossRef]
41. Ward, D.; Moghadam, P.; Hudson, N. Deep Leaf Segmentation Using Synthetic Data. *arXiv* **2018**, arXiv:1807.10931.
42. Lu, Y.; Chen, D.; Olaniyi, E.; Huang, Y. Generative adversarial networks (GANs) for image augmentation in agriculture: A systematic review. *Comput. Electron. Agric.* **2022**, *200*, 107208. [CrossRef]
43. Liu, K.; Li, Y.; Yang, J.; Liu, Y.; Yao, Y. Generative principal component thermography for enhanced defect detection and analysis. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 8261–8269. [CrossRef]
44. Illarionova, S.; Shadrin, D.; Trekin, A.; Ignatiev, V.; Oseledets, I. Generation of the nir spectral band for satellite images with convolutional neural networks. *Sensors* **2021**, *21*, 5646. [CrossRef] [PubMed]
45. Chen, Y.; Yang, X.H.; Wei, Z.; Heidari, A.A.; Zheng, N.; Li, Z.; Chen, H.; Hu, H.; Zhou, Q.; Guan, Q. Generative adversarial networks in medical image augmentation: A review. *Comput. Biol. Med.* **2022**, 105382. [CrossRef]
46. Beaumont, R. Clip Retrieval: Easily Compute Clip Embeddings and Build a Clip Retrieval System with Them. 2020 Available online: https://github.com/rom1504/clip-retrieval (accessed on 27 February 2023).
47. Illarionova, S.; Shadrin, D.; Tregubova, P.; Ignatiev, V.; Efimov, A.; Oseledets, I.; Burnaev, E. A Survey of Computer Vision Techniques for Forest Characterization and Carbon Monitoring Tasks. *Remote. Sens.* **2022**, *14*, 5861. [CrossRef]
48. Agarwal, N.; Chiang, C.W.; Sharma, A. A study on computer vision techniques for self-driving cars. In *Proceedings of the Frontier Computing: Theory, Technologies and Applications (FC 2018) 7*; Springer: Berlin/Heidelberg, Germany, 2019, pp. 629–634.
49. Lindner, L.; Sergiyenko, O.; Rivas-López, M.; Ivanov, M.; Rodríguez-Quiñonez, J.C.; Hernández-Balbuena, D.; Flores-Fuentes, W.; Tyrsa, V.; Muerrieta-Rico, F.N.; Mercorelli, P. Machine vision system errors for unmanned aerial vehicle navigation. In Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017; pp. 1615–1620.
50. Shonenkov, A. Ai-Forever/RU-Dalle: Generate images from texts. (In Russian)
51. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-resolution image synthesis with latent diffusion models. 2022 IEEE. In Proceedings of the CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 10674–10685.
52. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*; Meila, M., Zhang, T., Eds.; PMLR: New York, NY, USA, 2021; Volume 139, pp. 8748–8763.
53. Schuhmann, C.; Kaczmarczyk, R.; Komatsuzaki, A.; Katta, A.; Vencu, R.; Beaumont, R.; Jitsev, J.; Coombes, T.; Mullis, C. LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs. In Proceedings of the NeurIPS Workshop Datacentric AI. Jülich Supercomputing Center, Virtual, 13 December 2021; number FZJ-2022-00923.
54. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
55. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
56. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1314–1324.
57. Goodfellow, I.J.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: http://www.deeplearningbook.org (accessed on 27 February 2023).
58. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.

59. Bynagari, N.B. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Asian J. Appl. Sci. Eng.* **2019**, *8*, 25–34.
60. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
61. Gao, S.; Dai, Y.; Xu, Y.; Chen, J.; Liu, Y. Generative adversarial network–assisted image classification for imbalanced tire X-ray defect detection. *Trans. Inst. Meas. Control.* **2023**, 01423312221140940. [CrossRef]