# Interactive Generation of Image Variations for Copy-Paste Data Augmentation

Keita Higuchi
khiguchi@acm.org
Preferred Networks Inc.
Japan

Taiyo Mizuhashi
taiyo-mizuhashi@g.ecc.u-tokyo.ac.jp
The University of Tokyo
Japan

Fabrice Matulic
fmatulic@preferred.jp
Preferred Networks Inc.
Japan

Takeo Igarashi
takeo@acm.org
The University of Tokyo
Japan

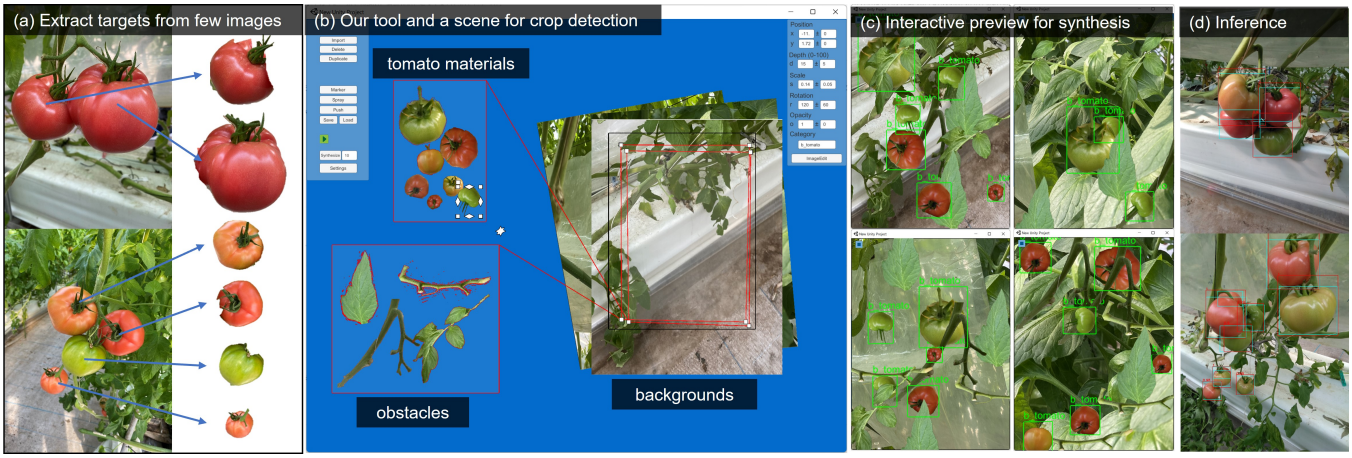


**Figure 1: Workflow of VisAugment: (a) extract image elements from few captured data, (b) editing a scene for a task with VisAugment, (c) reviewing synthesized training dataset with Interactive preview, and (c) training a model with synthesized dataset and running inference.**

## ABSTRACT

In machine learning, data augmentation is an important technique to artificially increase the amount of training data by generating variations, e.g., geometric and colour transformations. Simple data augmentation such as scaling and rotation is already provided by existing tools, but advanced data augmentation such as copy-paste image composition requires coding. Such composition operations are difficult to intuitively define in coding environments as typically there is no visual confirmation of generated images. Therefore, composition-based augmentations are not frequently used by developers. To address this issue, we propose a dedicated graphical tool. Contrary to image operations of standard graphics editors designed to produce a *single* image, our tool creates *multiple image variations* to be used as training data. The editor allows the user to visually and interactively set parameter *ranges* for transformations, and quickly review synthesized images based on the parameters. We report performance evaluations and user studies with machine learning experts.

## CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; **Visualization systems and tools**.

## KEYWORDS

interactive machine learning, visualization, data augmentations

**ACM Reference Format:**
Keita Higuchi, Taiyo Mizuhashi, Fabrice Matulic, and Takeo Igarashi. 2023. Interactive Generation of Image Variations for Copy-Paste Data Augmentation. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23), April 23–28, 2023, Hamburg, Germany.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3544549.3585856


*CHI EA '23, April 23–28, 2023, Hamburg, Germany*

ACM ISBN 978-1-4503-9422-2/23/04.
https://doi.org/10.1145/3544549.3585856

## 1 INTRODUCTION

Computer vision tasks using supervised deep learning, such as object detection and segmentation in images, typically require a large

amount of training data, i.e., labeled images, to achieve high performance. However, acquiring and annotating hundreds of images can be costly or unfeasible, for instance, when source data is rare or expensive. Data augmentation is a common technique used in machine learning to address this issue, where the amount of input data is artificially increased by generating slightly modified versions of the real input images to introduce more variety. Augmentation is typically achieved by applying simple transformations such as changing brightness and contrast, cropping, rotating, translating, and scaling the images. Although these methods are accessible in common libraries for deep learning [2, 3, 13, 22], the variation of generated images is limited.

Recent studies have investigated synthesis-based image augmentation using copy-paste image composition and realistic computer graphic models. These methods generate novel images with synthetic image elements and annotations and increase the variations of training data beyond simple transformations. They are particularly effective when a limited number of source images is available. Those techniques have been used in several contexts, including crop seed phenotyping [28], and cancer detection [31]. In those kinds of scenarios, augmented images are generated by compositing elements to be detected into source images using computer programs, i.e. code [28]. This can be hit-and-miss as the developer cannot easily confirm that chosen parameters are adequate by visually checking the resulting images. Furthermore, advanced operations such as image composition are inherently difficult to specify with code. As a result of this complexity, data augmentation can be daunting for novices and wrongly used by inexperienced developers, while more elaborate techniques remain mostly impractical.

To make such type of advanced data augmentation accessible, we propose a graphical image editing tool for composition-based data augmentation that we call VisAugment (Figure 1). The user interface is similar to that of standard graphical image editors (e.g., Photoshop, GIMP, etc.) but while standard image editors typically produce a *single* image, VisAugment is designed to generate *several* variations of input images, ready to use as training data. Our editor defines every augmentation parameter, such as translation, rotation, and scale, within a uniformly distributed sample space with user-specified range, from which values can be randomly chosen to generate augmented images.

We developed a prototype of VisAugment and demonstrated its practical value in a performance evaluation and expert review. For the performance evaluation, we trained models by augmenting training data which includes only a small number of target image elements using our tool. Those models show performance comparable to models trained with natural images containing many more instances of image elements to detect. In a user study, six machine learning experts used our tool to successfully build neural network models for detecting tomatoes using only small amounts of input data. The experts further improved detection performance by repeatedly editing input scenes and reviewing performance with a few validation images. In our supplemental material, we show several application scenarios of the proposed system such as object counting, stock monitoring, and visual inspection [1].

## 2 RELATED WORK

In training machine learning models, image data augmentation helps avoid overfitting by increasing the complexity and diversity of training data [25]. Many techniques are used in data augmentation, including basic geometric transformations such as rotation, translation, flipping, and cropping. Colour enhancement [19, 32] and noise injection [20] are also common techniques for such tasks. Selecting appropriate augmentation techniques and their parameters for a given dataset is not trivial because some operations might result in unrealistic or useless images (e.g., vertically flipping an image of a face or choosing cropping parameters that remove the object to detect). Cubuk et al. proposed AutoAugment that learns data augmentation strategies and finds effective augmentation operations for input datasets [5]. Recent extensions of that work further optimized the operation-finding process [6, 10, 17]. Such simple augmentations can be used to introduce variety to images in large datasets, but are limited to a certain type of operations.

Recent work proposed advanced data augmentation techniques based on image composition, which generate new composite images from multiple source samples [4, 9, 12, 16, 26, 27, 29]. Datasets can also be generated from entirely synthetic data, for example using 3D computer graphics [11, 18, 21, 28, 30]. In those cases, variations are typically defined using code, which can be time-consuming if parameters have to be manually tuned for machine learning experts. This may also be a daunting task for domain experts with little programming experience. VisAugment introduces more interactivity and visual feedback of the augmentations to facilitate image synthesizing.

## 3 PROPOSED SYSTEM

VisAugment generates training images by layered compositions of image elements and is specifically designed for scenarios where only a limited number of input data is available to train a machine learning model (e.g., for object detection). Composite images are created by cutting out elements or objects to detect and pasting them using different transformations on backgrounds. The tool provides interactive features to easily specify how those image collages are generated. Our current implementation supports simple geometric transformations (translation, rotation, scale, and duplication) of image elements. Other common operations, such as modifying colours, adding noise etc., can easily be integrated if needed.

### 3.1 VisAugment

Figure 1 (b) shows a screenshot of the system. It basically works like a standard image editor. The central area shows the image being edited. The left panel shows tools, and the right panel shows the properties of the element being selected. The user starts editing by importing one or more source image elements into the scene (our current implementation does not support image cut-out, so extracting image elements to be detected needs to be done beforehand in another tool such as Photoshop or GIMP). The user can then define various editing operations for each image element (translation, rotation, and scaling) using mouse dragging or typing specific values into the property panel.

On of the key features of VisAugment is that each property is defined as a random variable within a uniform distribution and

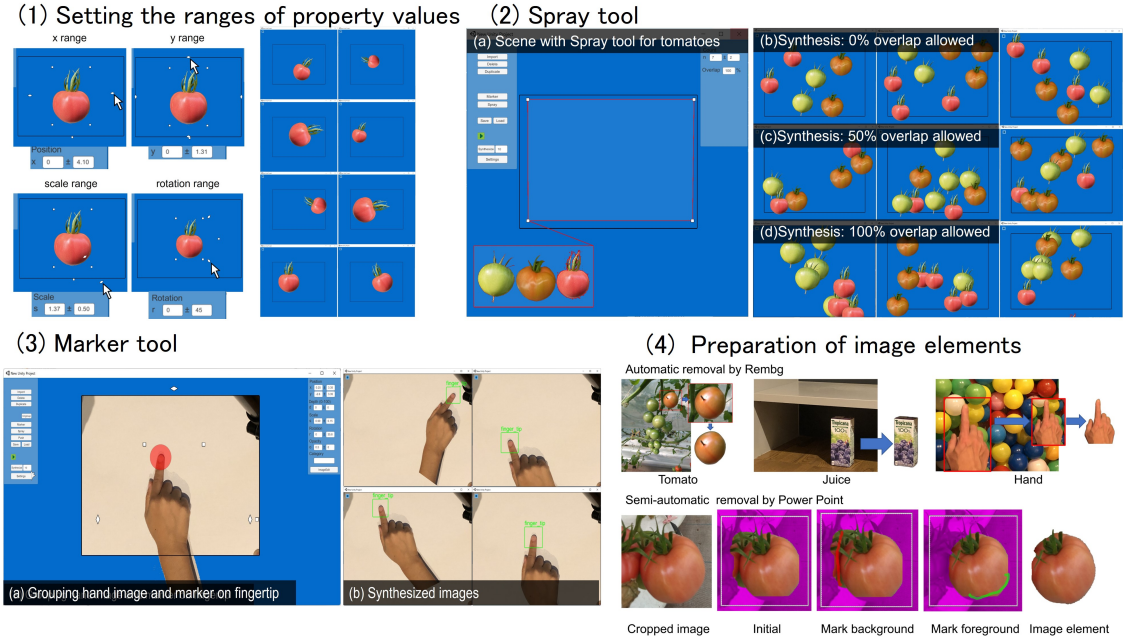


**Figure 2: (1) Setting the ranges of property values. Left: direct manipulation of ranges. Right: Synthesis results. The black frame represents the outer border of the composite image. (2) Spray tool. Left: The user specifies the source elements and target region. Right: the system generates random copies. The black frame represents the outer border of the composite image. (3) Marker tool. The red circle in the left image shows the position of the marker that is grouped with the hand image. The right image shows generated images and annotations that the annotations appear in the positions of fingers. (4) Preparation of image elements by automatic and semi-automatic processes.**

therefore includes a "range" parameter in addition to a "value". For example, if the value of a property is 2.0 and its range is 0.5, then the system assigns random values between 1.5 and 2.5 to the property when synthesizing images. The user can adjust those ranges either using mouse manipulations or by typing specific values in the property panel (Figure 2 (1)). "Depth" parameters are also available for all image elements to control their "layer order" (whether they appear in front of or behind other elements).

Another important feature of VisAugment is the spray tool (Figure 2 (2)) to generate a random number of instances and place them at random locations. A spray operation is characterized by the image element to position and three parameters: 1) the number of instances to synthesize, 2) the spray area, and 3) the maximum overlap between synthesized elements within that area. If the maximum overlap is set to 0%, no overlap is allowed, whereas 100% means that complete overlap is permitted. When synthesizing images, the system randomly generates the specified number of instances of the source elements inside the target area, honouring the overlap settings.

VisAugment supports adding labels to a part or keypoint of an image object for regression tasks (e.g., to locate a fingertip in a hand image). This is done by adding a labeled marker to the target image, as shown in Figure 2 (3). This marker is only used as visual feedback for the user in the editor and does not appear in the generated images.

After composition operations have been defined, a "preview" button can be pressed to show a short preview sequence of images synthesized with the chosen augmentations. If the user is satisfied with the results, they can press the "synthesize" button to actually generate the augmented images with their labels (saved as JSON files in the COCO data annotation format). The images and labels can then be used as input data to train a neural network.

VisAugment is implemented as a Unity application running on Windows 10.

## 3.2 Preparation of Image Elements

We used automatic and semi-automatic processes to obtain foreground image elements to compose images with VisAugment, as shown in Figure 2 (4). In the automatic process, we used Rembg to extract image elements and remove their background [8]. Rembg uses $U^2$ Net [23] as model, which requires less than 5 seconds for inference on an Intel i5-10300H CPU. We confirmed that the automatic process works well in most cases. We also used the Remove Background function of PowerPoint for complex cases of image elements, as shown at the bottom of Figure 2 (4). The interactive function allows the user to assign foreground and background regions manually and updates the results of removal similar to Lazy Snapping [15]. With this function, we can quickly extract image material from complex backgrounds in less than 1 minute.

# 4 PERFORMANCE EVALUATION

We evaluated the capabilities of VisAugment to synthesize useful training data using a public dataset. We are interested in the accuracy of models trained with synthetic datasets made from a limited number of image elements using our system compared to models trained on 100% natural images. In this evaluation, the authors prepared data and scene, trained models and analysed the results on validation data to assess performance in an idealized setting.

## 4.1 Dataset and Procedure

We chose the Laboro Tomato Big [7, 14] dataset, a small-scale public dataset that contains manual bounding box annotations for object detection tasks [1]. We used this dataset as a single class detection task that aims to detect different kinds of tomatoes, including green, half-ripened, and fully-ripened tomatoes. Laboro Tomato Big has only 353 training and 89 test images. We split the training data into 247 training and 106 validation images, where the average number of labels (i.e., tomatoes) per an image is 6.67.

In this evaluation, we focused on the potential performance of object detection models trained with a dataset synthesized using VisAugment. We use 106 validation images to improve the synthesized training data. The test data was not used in that process. We first examined the training data and cut out detection targets based on the associated annotations using PowerPoints and Rembg [8]. Backgrounds and elements that can appear in front of the tomotoes (e.g. stems and leaves) were also cut out from the training data and used for synthesis.

We used VisAugment and cut-out image elements to create a synthetic dataset. We also used the validation data as a reference to edit the scenes. After training models using the augmented dataset, we measured their accuracy with validation data and visualized the detection results. Based on this review, we modified augmentation operations in VisAugment in order to try to improve the quality of the generated dataset. We repeated this process 5 times to observe potential performance increases. We further examined the effects of the number of cut-out image elements and synthesized training images.

We used YOLO v3 [24] with a pre-trained model (DarkNet 53) for learning detection models for all conditions. We followed the common pipeline in model learning with default data augmentations (enlargement, random crop, random flip, colour distortion, and padding) for both synthesized and original datasets. Those default augmentations always produce different images, even with the same input. We trained models until loss converged for all datasets (mean: 273 epochs). We used mean Average Precision (mAP) as evaluation metric for this object detection task. For all conditions, we trained five times with 200 synthesized images and then computed the means for mAP.

We created five scenes that contain 6, 9, 12, 18, 24 tomato image elements. Each of the scenes contains an equal number (2, 3, 4, 6, 8 tomatoes, respectively) of images of the three different tomato states (i.e., green, half-ripened, and fully-ripened tomatoes). We used the Spray tool for tomato elements and obstacles. Each scene had the same total number of background images and obstacle image elements as the number of tomato image elements. For example, a scene with 6 tomatoes contains 2 backgrounds, 2 leaves, and 2 stems. Since tomato element images contained tomato hulls, we also used the Marker tool for several tomatoes to match the annotated region with the fruit region of the tomato image, following the labels of the training data. We also trained 9 detection models from natural images (original training data) that contain 1, 2, 4, 8, 16, 32, 64, 128, 247 images respectively. For all conditions, we used different images and trained five times and calculated the mean value of the mAPs.

[1] https://github.com/laboroai/LaboroTomato (CC BY-NC-SA 4.0) (Jan. 17, 2023 Accessed)

## 4.2 Results

Figure 3 (1) shows the results of different numbers of element images. We confirmed that the detection performance improves as the number of tomato image samples increases up to 18 images.

We confirmed that the performance of a model trained with the synthesized images made from a limited number of tomato images using VisAugment was comparable to that of a model trained with a larger number of tomato images. Specifically, the mAP score of the synthesized images obtained from 18 ripe tomato image elements was comparable to that of 32 natural images containing roughly 200 ripe tomato image elements. This is a significant reduction in the number of ripe tomato image elements necessary to achieve comparable performance. Those results support findings in previous studies showing that artificial training data contributes to improving model performance [4, 9], even though synthetic data may look artificial. We thus confirmed that VisAugment could be used effectively for machine learning tasks when only a small amount of training data is available.

# 5 USER EVALUATION

We conducted a user study with six machine-learning (ML) experts to determine how VisAugment could assist them in creating and improving augmented datasets to train better-performing models. Furthermore, we were interested in the feedback of ML practitioners using a visual and interactive tool to define augmentation operations instead of using code.

## 5.1 Scenario

We make two important assumptions for the scenario of the user study. First, we assume that the user only has a limited number of source images to train an ML classifier (only a few images containing objects to detect are available). In our scenario, the images are split into validation data and sample images containing the objects to detect. Second, we consider the situation where the experts iteratively improve the quality of the synthesized data using VisAugment through a trial-and-error process: They start by defining or adjusting the augmentation operations using the interface, then they train a model with the generated data and finally they verify its performance with the validation set. This process is ideally repeated until the results are satisfactory, but for the purpose of keeping the experiments feasible within reasonable time constraints, we limited the study to two iterations, one per day (since time is required in between to train the models).

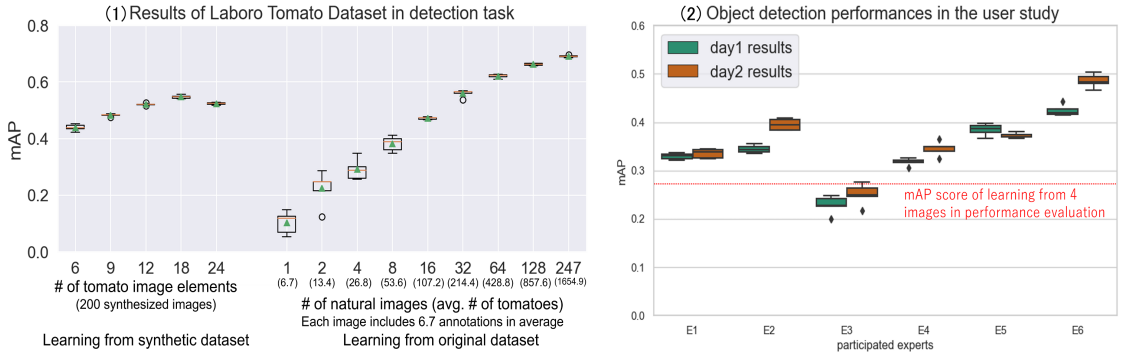


**Figure 3: (1) Results of the performance evaluation showing the influence of the number of tomato image samples. (2) Results of the user evaluation with mAP scores of synthesized datasets for the first and second days. The results on 5 of 6 experts with image elements from 4 training images outperform the mean of mAP when learning from 4 real images. The red line shows the mean of mAP scores when learning from 4 real images as shown in (1).**

## 5.2 Setting and Procedure

We recruited experts with a background in machine learning from our institution to participate in our study. This specific requirement for expertise in ML resulted in a limited participant pool of six individuals. Five of the recruited experts had regular experience with image data augmentation. The user study sessions were conducted remotely via online chat. Each participant was assigned two tasks, which they completed using VisAugment over the course of two days. The tasks took approximately 30 to 50 minutes for each participant to complete per day.

We chose again the Laboro Tomato Big Dataset for the task. We prepared image elements of detection targets (tomatoes), backgrounds, and occluding elements (e.g., leaves and stems). We provided the participants with 12 tomato images (4 fully-, 4 half-ripened, and 4 green tomatoes), 8 occluding elements (4 leaves and 4 stems) extracted from 4 training images and 8 background images. Finally, we selected 8 images to be used as the validation set. Each participant was provided with the same set of images to ensure consistency.

We first explained to the participants how to use VisAugment and asked them to familiarize themselves with the different tools of the application using several practice tasks, which included importing, placing image elements and backgrounds and defining augmentation operations. A document summarizing the functions of the application was provided for quick reference. The experimenter could also be directly consulted for clarifications when desired. After participants had sufficiently practised, we explained the main task, which consisted in creating synthesized datasets to train models for the detection of tomatoes. We asked participants to try to complete each day's task within 30 minutes, but this was not a hard limit.

For the Day 1 task, participants were asked to use VisAugment to synthesize images from the initial dataset so that a model trained with those images would achieve the best possible detection performance. Participants could refer to the 8 validation images at anytime for help. After defining augmentation operations in the application, 200 synthesized images were created and fed to a neural network for training. We followed the same training pipeline as described in Section 4.1. After the model completed training, we verified its performance on the validation set. We marked the detected tomatoes with bounding boxes and their labels in the images as reference for the second-day task.

For the Day 2 task, participants reviewed the results of their first model's performance and were instructed to modify and fine-tune the augmentations they had defined previously so that performance would increase. Specifically, the objective was to try to improve mAP scores by reducing false detections. After completing the task, participants reported on their experience in an interview.

## 5.3 Results

Figure 3 (2) shows the performance results (mAP) of for the two days. For both days, average mAP is 0.338 (SD: 0.061) and 0.364 (SD: 0.071), respectively. We saw performance improvements from the first day to the second day for E2, E3, E4, and E6, whereas E1 and E5 showed no improvement. The average editing times for the first and second days are 35.5 minutes (SD: 7.99) and 25.17 minutes (SD: 10.45), respectively.

In the performance evaluation, we iteratively improved model performance by repeatedly setting augmentation operations with VisAugment, training model and confirming detection accuracy with the validation dataset. A similar process was used for the expert-based evaluation, but only with two iterations. In that experiment, 5 of the 6 experts could successfully train better-performing models by augmenting the 4 training images with VisAugment compared to using a standard data augmentation procedure using code only. Those improvements were also achieved with a small amount of validation data (8 images). Furthermore, 4 of 6 experts were able to improve model performance from the first day to the second day. One behaviour that we observed in particular, is that participants reacted to low detection accuracy for certain types of tomatoes by increasing the number and coverage of samples generated for those cases. Similar strategies were used for backgrounds and images of occluding elements. Some participants noticed that

small tomatoes[2] were not detected because such tomatoes did not appear in the synthesized dataset and so they compensated for that by adding small tomatoes to the synthesized dataset. Those results show that VisAugment can be successfully used in a trial-and-error process to iteratively improve performance.

## 6 DISCUSSION

In the user study, the experts felt interactive preview and spray functions in VisAugment were useful for editing synthetic datasets. Interactive preview allows users to review synthetic images and their annotations quickly. This is helpful for trial-and-error processes of editing synthesized scenes. The Spray tool enables random placements of detection targets or obstacles in a designated area. All experts used the Spray tool to place tomatoes, leaves, and stem in the task. We thus conclude that these functionalities are essential for interactively creating synthesized training data.

In contrast, the participants mentioned that other functions could be introduced to VisAugment. For example, experts requested the interactive generation of color variations such as brightness and contrast. An expert mentioned that setting variations of image quality and noise might be effective in increasing the complexity of the training data. Some experts strongly encouraged developing a 3D version of VisAugment that randomly places 3D models of detection targets. We left the development of these functionalities and the 3D version VisAugment as our future work.

Since our tool is intended to improve performance through iteration, we think that users will learn effective uses of the tool. Five experts reported that interface operations were not easy for first-time users. Although all experts successfully created a synthetic dataset on the first day, the usability of VisAugment leaves more space for improvements.

When comparing detection performance in the Performance Evaluation and the Expert Study, we observe a gap, which can be attributed to the differences in amount of 1) image elements (i.e., tomatoes and backgrounds), validation data, and iterations. For the amount of image elements and validation data, we considered a scenario in which only very limited training and validation data were available. The evaluation shows that performance can be improved by increasing the number of image elements. With regard to the number of iterations, we saw that most experts could increase model performance with just two rounds. Given more time, we believe further improvements could be achieved with more iterations.

## 7 CONCLUSION

We presented VisAugment, a visual interactive editor to help machine-learning developers define composition operations for data augmentation and the creation of synthetic datasets to train detection models. In two evaluations we confirmed that models trained with images synthesized using VisAugment have higher performance compared to models trained using standard data augmentation pipelines. Our tool is particularly suited for scenarios, in which only a small amount of source data is available.

[2]Laboro Tomato Big Dataset contains only normal tomatoes, but several tomatoes appear small on images.

## REFERENCES


[1] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. 2021. The MVTec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision* 129, 4 (2021), 1038–1059.

[2] Marcus D Bloice, Peter M Roth, and Andreas Holzinger. 2019. Biomedical image augmentation using Augmentor. *Bioinformatics* 35, 21 (04 2019), 4522–4524. https://doi.org/10.1093/bioinformatics/btz259 arXiv:https://academic.oup.com/bioinformatics/article-pdf/35/21/4522/30330763/btz259.pdf

[3] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. 2020. Albumentations: Fast and Flexible Image Augmentations. *Information* 11, 2 (2020). https://doi.org/10.3390/info11020125

[4] Paola Cascante-Bonilla, Arshdeep Sekhon, Yanjun Qi, and Vicente Ordonez. 2021. Evolving Image Compositions for Feature Representation Learning. In *British Machine Vision Conference (BMVC)*.

[5] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. 2019. AutoAugment: Learning Augmentation Strategies From Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. 2020. Randaugment: Practical Automated Data Augmentation With a Reduced Search Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[7] Umme Fawzia Rahim and Hiroshi Mineno. 2022. Highly Accurate Tomato Maturity Recognition: Combining Deep Instance Segmentation, Data Synthesis and Color Analysis *(AICCC '21)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3508259.3508262

[8] Daniel Gatis. 2020. Rembg. https://github.com/danielgatis/rembg.

[9] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. 2021. 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*. 2918–2928.

[10] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. 2020. Faster AutoAugment: Learning Augmentation Strategies Using Backpropagation. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 1–16.

[11] Hironori Hattori, Vishnu Naresh Boddeti, Kris M Kitani, and Takeo Kanade. 2015. Learning scene-specific pedestrian detectors without real data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3819–3827.

[12] Hiroshi Inoue. 2018. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929* (2018).

[13] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. 2020. imgaug. https://github.com/aleju/imgaug.

[14] Laboro.AI. 2020. Laboro Tomato Dataset. https://github.com/laboroai/LaboroTomato.

[15] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. 2004. Lazy snapping. *ACM Transactions on Graphics (ToG)* 23, 3 (2004), 303–308.

[16] Daojun Liang, Feng Yang, Tian Zhang, and Peter Yang. 2018. Understanding mixup training methods. *IEEE Access* 6 (2018), 58774–58783.

[17] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. 2019. Fast AutoAugment. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/6add07cf50424b14fdf649da87843d01-Paper.pdf

[18] Feng Lu, Yusuke Sugano, Takahiro Okabe, and Yoichi Sato. 2012. Head pose-free appearance-based gaze sensing via eye image synthesis. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 1008–1011.

[19] Agnieszka Mikołajczyk and Michał Grochowski. 2018. Data augmentation for improving deep learning in image classification problem. In *2018 International Interdisciplinary PhD Workshop (IIPhDW)*. 117–122. https://doi.org/10.1109/IIPHDW.2018.8388338

[20] Francisco J. Moreno-Barea, Fiammetta Strazzera, José M. Jerez, Daniel Urda, and Leonardo Franco. 2018. Forward Noise Adjustment Scheme for Data Augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. 728–734. https://doi.org/10.1109/SSCI.2018.8628917

[21] Jiteng Mu, Weichao Qiu, Gregory D. Hager, and Alan L. Yuille. 2020. Learning From Synthetic Animals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[22] Zoe Papakipos and Joanna Bitton. 2022. AugLy: Data Augmentations for Robustness. arXiv:2201.06494 [cs.AI]

[23] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. 2020. U2-Net: Going deeper with nested U-structure for

salient object detection. *Pattern recognition* 106 (2020), 107404.

[24] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[25] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of big data* 6, 1 (2019), 1–48.

[26] Cecilia Summers and Michael J Dinneen. 2019. Improved mixed-example data augmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1262–1270.

[27] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. 2019. Data augmentation using random image cropping and patching for deep CNNs. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 9 (2019), 2917–2931.

[28] Yosuke Toda, Fumio Okura, Jun Ito, Satoshi Okada, Toshinori Kinoshita, Hiroyuki Tsuji, and Daisuke Saisho. 2019. Learning from Synthetic Dataset for Crop Seed Instance Segmentation. *bioRxiv* (2019). https://doi.org/10.1101/866921 arXiv:https://www.biorxiv.org/content/early/2019/12/07/866921.full.pdf

[29] Shashank Tripathi, Siddhartha Chandra, Amit Agrawal, Ambrish Tyagi, James M. Rehg, and Visesh Chari. 2019. Learning to Generate Synthetic Data via Compositing. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 461–470. https://doi.org/10.1109/CVPR.2019.00055

[30] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. 2017. Learning From Synthetic Humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[31] Eric Wu, Kevin Wu, David Cox, and William Lotter. 2018. Conditional infilling GANs for data augmentation in mammogram classification. In *Image analysis for moving organ, breast, and thoracic images*. Springer, 98–106.

[32] Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, and Gang Sun. 2015. Deep image: Scaling up image recognition. *arXiv preprint arXiv:1501.02876* 7, 8 (2015).