frontiers in Artificial Intelligence

ORIGINAL RESEARCH
published: 25 March 2022
doi: 10.3389/frai.2022.830026

Behind the Leaves: Estimation of Occluded Grapevine Berries With Conditional Generative Adversarial Networks

Jana Kierdorf[1]*, *Immanuel Weber*[2], *Anna Kicherer*[3], *Laura Zabawa*[4], *Lukas Drees*[1] and *Ribana Roscher*[1]

[1] Remote Sensing Group, Institute of Geodesy and Geoinformation, University of Bonn, Bonn, Germany, [2] Application Center for Machine Learning and Sensor Technology, University of Applied Sciences Koblenz, Koblenz, Germany, [3] Julius Kühn-Institut (JKI), Federal Research Centre for Cultivated Plants, Institute for Grapevine Breeding Geilweilerhof, Siebeldingen, Germany, [4] Geodesy Group, Institute of Geodesy and Geoinformation, University of Bonn, Bonn, Germany

The need for accurate yield estimates for viticulture is becoming more important due to increasing competition in the wine market worldwide. One of the most promising methods to estimate the harvest is berry counting, as it can be approached non-destructively, and its process can be automated. In this article, we present a method that addresses the challenge of occluded berries with leaves to obtain a more accurate estimate of the number of berries that will enable a better estimate of the harvest. We use generative adversarial networks, a deep learning-based approach that generates a highly probable scenario behind the leaves exploiting learned patterns from images with non-occluded berries. Our experiments show that the estimate of the number of berries after applying our method is closer to the manually counted reference. In contrast to applying a factor to the berry count, our approach better adapts to local conditions by directly involving the appearance of the visible berries. Furthermore, we show that our approach can identify which areas in the image should be changed by adding new berries without explicitly requiring information about hidden areas.

Keywords: deep learning, machine learning, Generative Adversarial Networks, domain-transfer, grape generation, occlusions, yield counting

OPEN ACCESS

Edited by:
Ian Stavness,
University of Saskatchewan, Canada

Reviewed by:
Andrew P. French,
University of Nottingham, United Kingdom
Luigi Celona,
University of Milano-Bicocca, Italy

***Correspondence:**
Jana Kierdorf
jkierdorf@uni-bonn.de

Specialty section:
This article was submitted to AI in Food, Agriculture and Water, a section of the journal Frontiers in Artificial Intelligence

Received: *06 December 2021*
Accepted: *28 February 2022*
Published: *25 March 2022*

Citation:
Kierdorf J, Weber I, Kicherer A, Zabawa L, Drees L and Roscher R (2022) Behind the Leaves: Estimation of Occluded Grapevine Berries With Conditional Generative Adversarial Networks. Front. Artif. Intell. 5:830026. doi: 10.3389/frai.2022.830026

1. INTRODUCTION

With increasing competition on the wine market worldwide, the need for accurate yield estimations has been getting more and more important for viticulture. The variation of yield over the years is mainly caused by the berry number per vine (90%), while the remaining 10% are caused by the average berry weight (Clingeleffer et al., 2001), which is generally collected manually and averaged over many years. Traditionally, yield estimations in viticulture can be done at three phenological timepoints by (1) counting the number of bunches 4–6 weeks after budburst, (2) counting the number of berries after fruit set (May, 1972), or (3) destructively sampling vines or segments of vines close to harvest. Considering that yield estimation can be more accurately and reliably determined as harvest approaches, a berry count is a promising option that can be approached non-destructively and whose process can be automated.

Several papers show that machine learning-based methods for analyzing data from imaging sensors provide an objective and fast method for counting visible berries (Diago et al., 2012; Kicherer et al., 2014; Nuske et al., 2014; Roscher et al., 2014; Aquino et al., 2017; Coviello et al., 2020; Zabawa et al., 2020), and thus for automated yield predictions in the field. One of the main challenges in deriving berry counts from image data taken in the field is occlusions, which generally causes an underestimation of the number of berries and yield (Zabawa et al.[1]). First, occlusions of berries by other berries make it difficult to distinguish or count individual berries. Therefore, approaches that perform a segmentation of regions of berries and regions without berries is not sufficient, and more advanced methods that recognize individual instances of berries must be applied (Zabawa et al., 2020). Second, occlusions by leaves play a major role in underestimating the number of berries. Zabawa et al. (see text footnote 1, respectively) perform leaf occlusion experiments over two years and show that the yield estimation is highly dependent on the number of visible berries. With vines defoliated (i.e., with manually removed leaves) at pea size, they report an average error of total yield estimation of 27%, whereas Nuske et al. (2014) observed average errors between 3 and 11% using images of entirely defoliated fruit zones.

In order to overcome the challenge of leaf occlusions, defoliation can be performed in the grapefruit zone, but this is immensely time-consuming and labor-intensive. Partial defoliation is carried out in viticulture, for example, for ventilation and rapid drying of the grape zone to avoid fungal infections of the grapes or yield and quality regulation (Diago et al., 2009). However, complete defoliation is not feasible on a large scale or may lead to negative effects such as increased sunburn on the berries (Feng et al., 2015) or generally have an undesirable impact on yield results. Alternatively, machine-learning-based approaches can be used to obtain a more accurate estimation of the berry number. Numerous approaches rely on information where occlusions are present, which is generally provided as a manual input (Bertalmio et al., 2003; Barnes et al., 2009; Iizuka et al., 2017; Dekel et al., 2018; Liu et al., 2018). In contrast to this, two-step approaches first detect occlusions and then fill the corresponding regions with information according to the environment (Ostyakov et al., 2018; Yan et al., 2019).

This article addresses the challenge of occlusions caused by leaves by generating images that reveal a highly probable situation behind the leaves, exploiting learned patterns from a carefully designed dataset. The generated images can then be used to count berries in a post-processing step. Our approach generates potential berries behind leaves based on RGB information obtained by visible light imaging, as this is an efficient, cheap and non-harmful approach in contrast to data from material-penetrating sensors. In order to train our machine-learning method, we use aligned image pairs showing plants with leaves and the same plants after defoliation. In detail, we model this problem as a domain-transfer task and regard the aligned images containing occluded berries as one domain and images with revealed berries as a second domain. We resort to methods like Pix2Pix (Isola et al., 2017), that uses a conditional generative adversarial network (cGAN) (Mirza and Osindero, 2014) and can learn the described domain-transfer. In contrast to other works, we present a one-step approach that is end-to-end trainable, meaning the positions of the occlusions are identified, and patterns that need to be filled are learned simultaneously. Through the experience the model gains during training, it learns patterns such as grape instances with their appearing shapes, their environment, and where they occur in the image. This knowledge is exploited during the generation step, in which the learned domain-transfer model is applied to images of vines that have not been defoliated to obtain a high-probability and realistic impression of the scene behind the leaves. In order to obtain a berry count, the generated images are further processed with the berry counting algorithm of Zabawa et al. (2020). In this way, we provide a more accurate count of grape berries, since both visible berries and berries potentially occluded by leaves are taken into account.

A major challenge for training is that there is no large dataset of aligned natural images that includes both images with occluded berries and images with berries exposed by defoliation. In addition, in our case, the spatial alignment between the image pairs is not accurate enough since defoliation leads to a resulting movement of branches, grape bunches, and other objects in the non-occluded domain patches. As a result, the natural data is not sufficient to train a model that matches our requirements of a reliable model. Due to this, we propose the use of a synthetically generated dataset that contains paired data of both domains. Our main contributions of this article are:

- The true scenario behind the leaves without defoliation is unlikely to be identified. Therefore, our approach estimates a highly probable scenario behind the occlusions based on visible information in the image, especially of the surroundings of the occlusion, and learned patterns during the training process which include for example the berry shape and neighborhood of berries to obtain a distribution similar to the training data.
- We present a one-step approach, which can implicitly identify which image areas contain visible berries and which areas are occluded without supervision regarding occluded and non-occluded areas. This differs from approaches such as inpainting (Bertalmio et al., 2003; Barnes et al., 2009; Iizuka et al., 2017; Dekel et al., 2018; Liu et al., 2018), in which the occluded areas must be known *a priori*.
- In addition to the acquired images, we use so-called berry masks obtained by the approach presented in Zabawa et al. (2020), which uses semantic segmentation to indicate in the image which pixels belong to `berry`, `berry-edge`, and `background`. During training, this leads to a more stable and easier optimization process. During testing, the berry mask is only needed for the input image since our GAN-based method simultaneously generates the berry mask in which the berries are counted, in addition to the visually generated image.

Abbreviations: cGAN, conditional Generative Adversarial Network; SMPH, semi minimal pruned hedge; VSP, vertical shoot position.

[1]Zabawa, L., Kicherer, A., Klingbeil, L., Töpfer, R., Roscher, R., and Kuhlmann, H. (2021). Image-based analysis of yield parameters in viticulture. *Biosyst. Eng.* (under review).

- Since a direct comparison of the true scenario behind the leaves and our generated scenario is not appropriate using standard evaluation methods such as a pixel-by-pixel comparison, we perform a comprehensive evaluation using alternative evaluation metrics, such as generation maps and correlation, that assesses the performance of our approach.
- We show that the application of our approach minimizes the offset compared to the manual reference berry count and the variance, which is not achieved by applying a factor.
- We create various synthetic datasets and show that our approach trained on synthetic data also works on natural data.

The article is structured as follows: After surveying related works, we start by introducing our domain-transfer framework and describe the different components, such as the conditional generative adversarial network, that are used in our approach. We explain the data acquisition and post-processing of the natural and synthetic datasets we use in our work. We explain the evaluation metrics we use and then describe our experiments in which we analyze the generation quality of different synthetic input data, compare generated results with real results in the occluded as well as the non-occluded domain and analyze the berry counting based on the generated results. Finally, we investigate the application of the synthetically learned models to natural data. We end our article with the conclusion and future directions.

2. RELATED WORK

2.1. Yield Estimation and Counting

Since an accurate yield estimation is one of the major needs in viticulture, especially on a large scale, there is a strong demand for objective, fast, and non-destructive methods for yield forecasts in the field. For many plants, including grapevines, the derivation of phenotypic traits is essential for estimating future yields. Besides 3D-reconstruction (Schöler and Steinhage, 2015; Mack et al., 2017, 2018), 2D-image processing is also a widely used method (Hacking et al., 2019) for the derivation of such traits. For vine, one plant trait that strongly correlates with yield is the number of bearing fruits, that means the amount of berries. This correlation is underlined by the study of Clingeleffer et al. (2001), in which it is shown that the variation of grapevine yield over the years is mainly caused by the berry number per vine (90%).

The task of object counting can be divided into two main approaches: (1) regression (Lempitsky and Zisserman, 2010; Arteta et al., 2016; Paul Cohen et al., 2017; Xie et al., 2018) which directly quantifies the number of objects for a given input, and (2) detection and instance segmentation approaches which identify objects as an intermediate step for counting (Nuske et al., 2014; Nyarko et al., 2018). Detection approaches in viticulture are presented, for example, by Nuske et al. (2011), Roscher et al. (2014), and Nyarko et al. (2018), who define berries as geometric objects such as circles or convex surfaces and determine them by image analysis procedures such as Hough-transform. Recent state-of-the-art approaches, especially segmentation (He et al., 2017), are mostly based on neural networks. One of the earliest works combining grapevine data and neural network analysis was Aquino et al. (2017). They detect grapes using connected components and determine key features based on them, which are fed as annotations into a three-layer neural network to estimate yield. In another work, Aquino et al. (2018) deal with counting individual berries, which are first classified into berry candidates using pixel classification and morphological operators. Afterward, a neural network classifies the results again and filters out the false positives.

The two studies by Zabawa et al. (2019, 2020) serve as the basis for this article. Zabawa et al. (2019) use a neural network which performs a semantic segmentation with the classes `berry`, `berry-edge` and `background`, which enables the identification of single berry instances. The masks generated in that work serve as input for the proposed approach. The article by Zabawa et al. (2020) based on Zabawa et al. (2019) extends identification to counting berries by discarding the class edge and counting the berry components with a connected component algorithm. The counting procedure applied in that work is used for the analyses of the experiments.

2.2. Given Prior Information About Regions to Be Transferred

A significant problem in fruit yield estimation is the overlapping of the interesting fruit regions by other objects, like in the case of this work, the leaves. Several works are already addressing the issue of data with occluded objects or gaps within the data, where actual values are missing, which is typically indicated by special values like, e.g., not-a-number. The methodologies can be divided into two areas: (1) there is prior information available about where the covered positions are, and (2) there is no prior information. In actual data gaps, where the gap positions can be easily identified a priori, data imputation approaches can be used to complete data. This imputation is especially important in machine learning, since machine learning models generally require complete numerical data. The imputation can be performed using constant values like a fixed constant, mean, median, or k-nearest neighbor imputation (Batista and Monard, 2002) or calculated using a random number like the empirical distribution of the feature under consideration (Rubin, 1996, 2004; Enders, 2001; von Hippel and Bartlett, 2012). Also, possible are multivariate imputations, which additionally measures the uncertainty of the missing values (Van Buuren and Oudshoorn, 1999; Robins and Wang, 2000; Kim and Rao, 2009). Data imputation is also possible using deep learning. Lee et al. (2019), for example, introduce CollaGAN in which they convert the image imputation problem to a multi-domain image-to-image translation task.

In case there are no data gaps, but the image areas that are occluded or need to be changed are known, inpainting is a commonly used method. The main objective is to generate visually and semantically plausible appearances for the occluded regions to fit in the image. Conventional inpainting methods (Bertalmio et al., 2003; Barnes et al., 2009) work by filling occluded pixels with patches of the image based on low level features like SIFT descriptors (Lowe, 2004). The results of these methods do not look realistic if the areas to be filled are near

foreground objects or the structure is too complex. An alternative is deep learning methods that learn a direct end-to-end mapping from masked images to filled output images. Particularly realistic results can be generated using Generative Adversarial Networks (GANs) (Iizuka et al., 2017; Dekel et al., 2018; Liu et al., 2018). For example, Yu et al. (2018) deal with generative image inpainting using contextual attention. They stack generative networks to ensure further the color and texture consistence of generated regions with surroundings. Their approach is based on rectangular masks, which do not generalize well to free-form masks. This task is solved by Yu et al. (2019) one year later by using guidance with gated convolution to complete images with free-form masks. Further work introduces mask-specific inpainting that fills in pixel values at image locations defined by masks. Xiong et al. (2019) learn a mask of the partially masked object from the unmasked region. Based on the mask, they learn the edge of the object, which they subsequently use to generate the non-occluded image in combination with the occluded input image.

2.3. No Prior Information About Regions to Be Transferred

Methods that do not involve any prior knowledge about gaps and occluded areas can be divided into two-step and one-step approaches. Two-step approaches first determine the occluded areas, which then are used, for example, as a mask to inpaint the occluded areas. Examples are provided by Yan et al. (2019), which visualize the occluded parts by determining a binary mask of the visible object using a segmentation model and then creating a reconstructed mask using a generator. The resulting mask is fed into coupled discriminators together with a 3D-model pool in order to decide if the generated mask is real or generated compared to the masks in the model pool. Ostyakov et al. (2018) train an adversarial architecture called SEIGAN to first segment a mask of the interesting object, then paste the segmented region into a new image and lastly fill the masked part of the original image by inpainting. Similar to the proposed approach, SeGAN introduced by Ehsani et al. (2018) uses a combination of a convolutional neural network and a cGAN (Mirza and Osindero, 2014; Isola et al., 2017) to first predict a mask of the occluded region and, based on this, generate a non-occluded output.

3. MATERIALS AND METHODS

3.1. Framework

In our work, we regard the revealing of the occluded berries as a transfer between two image domains. We first detail this and show how we model this transfer for our data. Then we will lay out the cGAN and the framework we use for this task. Finally, we show how we train this network.

3.1.1. Domain-Transfer Framework

On a high level, the task of revealing the occluded berries can be described as generating a new impression of an existing image. We model this generative task as a transfer of an existing image from one domain, the source domain, to another domain, the target domain. In our work, we regard images where berries are occluded by various objects as the source domain and call it *occluded domain*. Accordingly, our target domain contains images of defoliated plants, and we call it *non-occluded domain*. Therefore, by performing this domain-transfer, we aim to reveal hidden berries. Samples of both domains are shown at the top of **Figure 1**.

This task can typically be learned by a cGAN, like Pix2Pix in our case. We train this network using aligned pairs of images from the occluded domain and the non-occluded domain and indicate them with $\boldsymbol{x}_{\text{occ}}$ and $\boldsymbol{x}_{\text{non}}$, respectively. The first ones are used as the network input and the latter ones, being the desired output, as the training target. Due to computational limitations, we use cropped patches from the original data and convert them to grayscale to develop an efficient approach that is independent of the berry color. In practice, we accompany the images of each domain with a corresponding semantic mask, that indicates per image pixel the content based on the classes `berry`, `berry-edge`, and `background`. This mask supports the discriminability of relevant information like the berries from the surrounding information in the image and the generation of separated berries, supporting the later counting step. After training, we use the cGAN to generate images, $\widetilde{\boldsymbol{x}}_{\text{non}}$, that we further process with a berry counting method.

Since we only have limited amounts of data available for training and testing, we resort to a dataset consisting of synthetic images for the occluded domain and natural images for the non-occluded domain that we describe in detail in section 3.2. In addition, we test our trained model on fully natural data to analyze the generalizability of the model. For the training set, the non-occluded domain contains natural images, whereas the images from the occluded domain are derived from the former domain, where berries are artificially occluded with leaf templates. To differentiate the different datasets of images, we further qualify the natural images with index N and the synthetic images with index S, which results in the two occluded domain groups: $\boldsymbol{x}^{\text{N}}_{\text{occ}}$ and $\boldsymbol{x}^{\text{S}}_{\text{occ}}$. The generated images are accordingly indicated by $\widetilde{\boldsymbol{x}}^{\text{N}}_{\text{non}}$ and $\widetilde{\boldsymbol{x}}^{\text{S}}_{\text{non}}$. We therefore train the model with input images $\boldsymbol{x}^{\text{S}}_{\text{occ}}$ and use $\boldsymbol{x}^{\text{N}}_{\text{non}}$ as target images. Finally, we apply the model on natural images $\boldsymbol{x}^{\text{N}}_{\text{occ}}$ and compute the berry counts of the generated output images, $\widetilde{\boldsymbol{x}}^{\text{N}}_{\text{non}}$.

3.1.2. Conditional Generative Adversarial Networks

The core of our framework is the cGAN that we use to generate images with berries being revealed. Specifically, we use the Pix2Pix (Isola et al., 2017) network and training method, which is illustrated in simplified form in **Figure 2**.

The model consists of two networks, the generator, and the discriminator. The generator network $\mathcal{G}$ takes images with occluded berries as an input and is intended to generate images with revealed berries $\mathcal{G}(\boldsymbol{x}_{\text{occ}}) = \widetilde{\boldsymbol{x}}_{\text{non}}$ that cannot be distinguished from real images $\boldsymbol{x}_{\text{non}}$ of the non-occluded domain. The adversarially trained discriminator network $\mathcal{D}$, on the other side, tries to discriminate between generated images $\widetilde{\boldsymbol{x}}_{\text{non}}$ and real images $\boldsymbol{x}_{\text{non}}$. The generator used in Pix2Pix is based on a U-Net (Ronneberger et al., 2015), the discriminator $\mathcal{D}$ on a PatchGAN.

As described by Goodfellow et al. (2014), both parts of GANs are trained simultaneously using a min-max approach. The goal

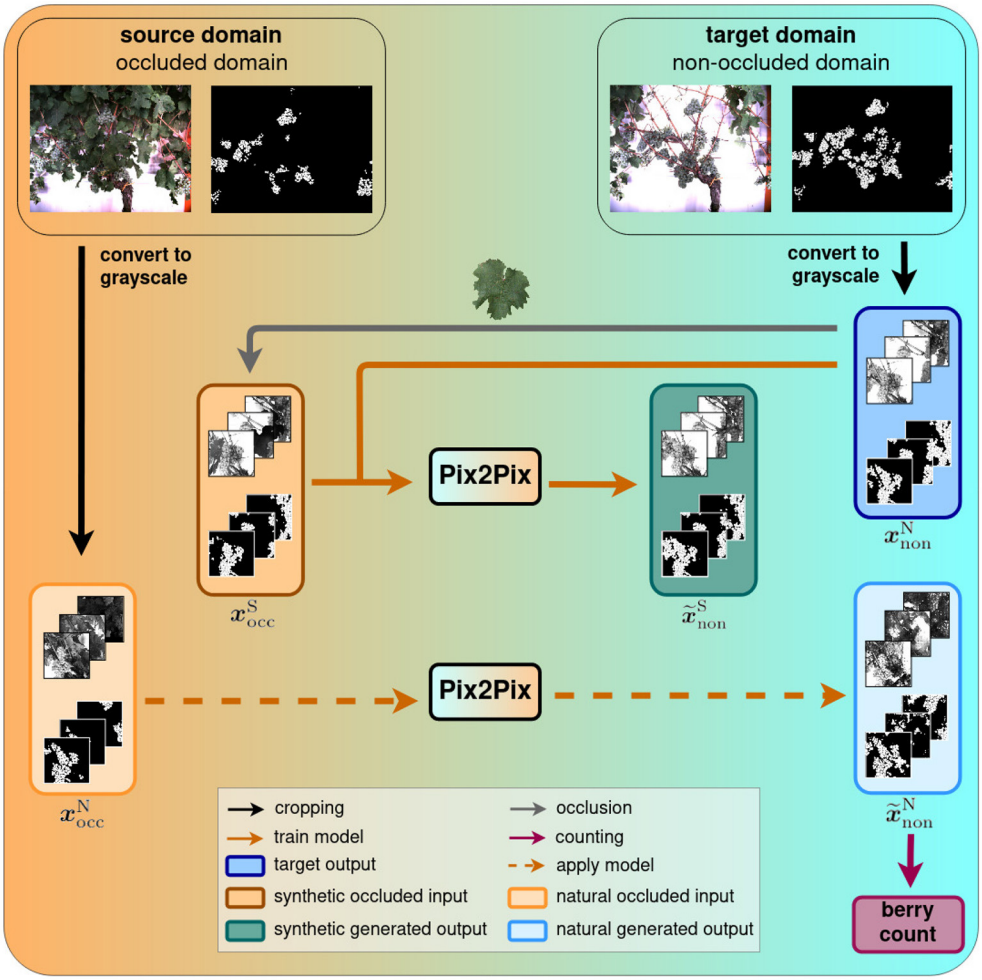


FIGURE 1 | Domain-transfer framework. We transfer images from the source domain with occluded berries to the target domain with revealed berries using the Pix2Pix cGAN. We train and test the model on synthetic data and subsequently apply it to natural data. Finally, a berry counting is performed on the generated outputs. Further evaluation steps will be performed in our experiments.

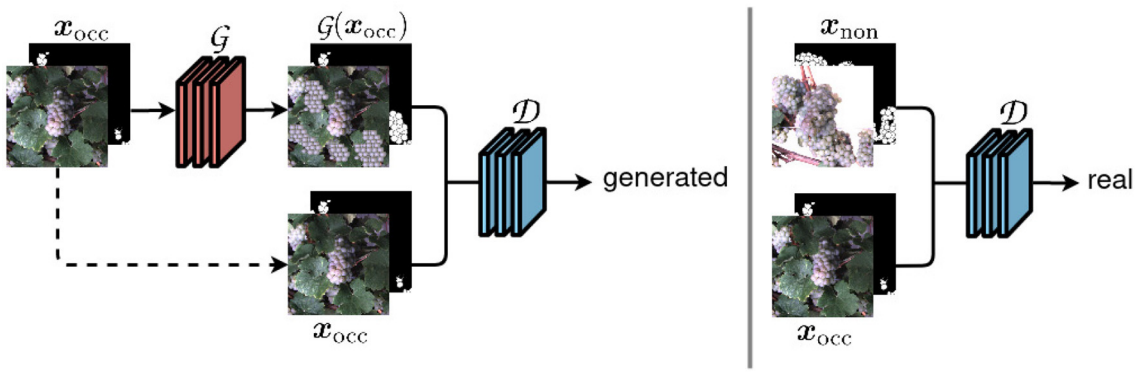


FIGURE 2 | Our network structure based on the Pix2Pix framework (Isola et al., 2017). An input image x_{occ} of the occluded domain is transferred to a non-occluded domain using a generator network $\mathcal{G}$. The discriminator network $\mathcal{D}$ distinguish whether the output of $\mathcal{G}$ looks real or generated.

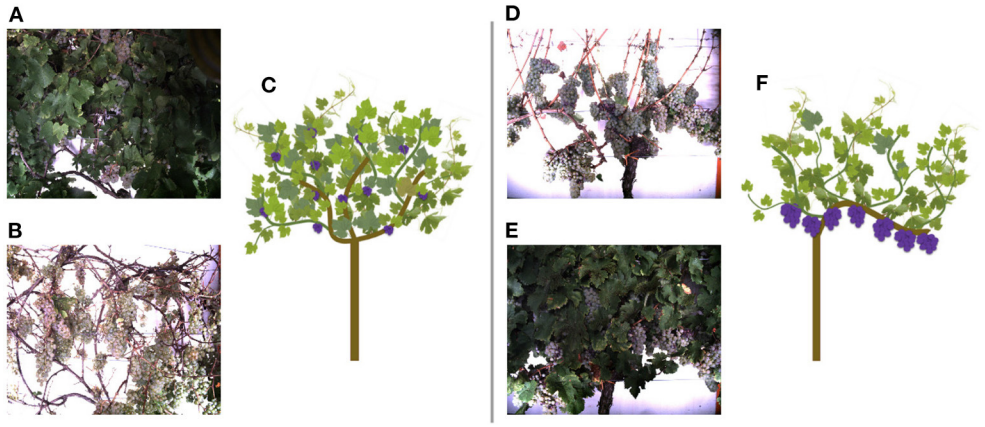


FIGURE 3 | Acquired images of the Phenoliner (Kicherer et al., 2017) for two different kind of cuttings: semi minimal pruned hedge SMPH **(A–C)** and vertical shoot positioned system VSP **(D–F)**. **(A,D)** Show example images before defoliation in the occluded domain. **(B,E)** Show example images after defoliation in the non-occluded domain.

of the discriminator during training is to be able to distinguish as good as possible between real and generated images. For this, the discriminator uses a mini-batch of input images $\boldsymbol{x}_{\text{non}}$ and computes the discriminator loss $\ell_{\mathcal{D}_{\text{real}}}$. Additionally, it uses generated images $\widetilde{\boldsymbol{x}}_{\text{non}}$ obtained from the generator $\mathcal{G}$ and computes the corresponding loss $\ell_{\mathcal{D}_{\text{gen}}}$. For both computations, the mean squared error (MSE) loss ℓ_{MSE} is used. The overall loss $\ell_{\mathcal{D}}$ of the discriminator is calculated as:

$$\ell_{\mathcal{D}} = \frac{1}{2} \cdot (\ell_{\mathcal{D}_{\text{fake}}} + \ell_{\mathcal{D}_{\text{real}}}) \tag{1}$$

The objective is to maximize this loss, as this means that the discriminator can distinguish between generated and real images with ease. The weights of the discriminator network are then updates with respect to this loss.

When generating new images, the generator tries to trick the discriminator at the same time, which is the adversarial part of the network. Compared to the maximization of the discriminator loss, the objective of the generator is to minimize the generator loss ℓ_G. This is calculated from a combination of MSE loss computed by $\mathcal{D}\left[\mathcal{G}\left(\boldsymbol{x}_{\text{occ}}\right)\right]$ referred to the reference label `generated` and a ℓ_1 loss, which avoids blurring. The ℓ_1 loss is computed using real and generated images, $\boldsymbol{x}_{\text{non}}$ and $\widetilde{\boldsymbol{x}}_{\text{non}}$, from the non-occluded domain. The generator loss $\ell_{\mathcal{G}}$ is then used to update the generator's weights.

$$\ell_{\mathcal{G}} = \ell_{\text{MSE}}(\mathcal{D}\left(\mathcal{G}\left(\boldsymbol{x}_{\text{occ}}\right)\right)) + \lambda \cdot \ell_1(\boldsymbol{x}_{\text{non}}, \widetilde{\boldsymbol{x}}_{\text{non}}) \tag{2}$$

The weighting factor λ adjusts the scale of the losses to each other and is, in our case, $\lambda = 100$.

The minimization of the generator loss $\ell_{\mathcal{G}}$ results in either a strong generator or a very weak discriminator. If the loss becomes maximal, the opposite possibilities can occur. The objective is to balance both adversarial goals at the end of the training in the best possible way by realizing both at the same time.

3.2. Data

3.2.1. Study Site

The data, we use in this work, were acquired at the experimental fields of JKI Geilweilerhof located in Siebeldingen, Germany. It was acquired using the Phenoliner (Kicherer et al., 2017), a reconstructed grape harvester that can be used as a phenotyping platform to acquire geo-referenced sensor data directly in the field. A description of the on-board camera setup can be found in Zabawa et al. (2020). The images were acquired in two different training systems of the cultivar Riesling (DEU098_VIVC10077_Riesling_Weiss_DEU098-2008-085): (1) Vertical shoot positioned (VSP) vines (**Figure 3C**) and (2) vines trained as semi minimal pruned hedges (SMPH) (**Figure 3F**) were chosen due to diverse difficulties in image analysis (Zabawa et al., 2020). The acquisition took place in September 2019 and 2020, before harvest at the plant growth stage BBCH89, and in each year the images were taken 1 day before (**Figures 3A,D**) and right after defoliation (**Figures 3B,E**). In 2019 50 cm and 2020, respectively, 100 cm of the grapevine canopy have been defoliated.

In our framework, we use three different types of inputs:

- **Natural data**: Images acquired in the vineyard before and after defoliation. For our studies, we use grayscale images. We denote this dataset with X^{N}.
- **Synthetic data**: Images acquired in the vineyard after defoliation. Images with occluded berries are synthetically generated. We denote this dataset with X^{S}.
- **Semantic segmentation masks (berry masks)**: So-called berry masks obtained by a semantic segmentation approach presented in Zabawa et al. (2019). Each pixel in these images is assigned to the class `berry`, `berry-edge`, or `background`. We denote this data as X_{B}.

The use of the mentioned grayscale images is indicated by the index G and with index B we denote the use of the berry masks. Moreover, we define X_{GB} as the input where the grayscale image

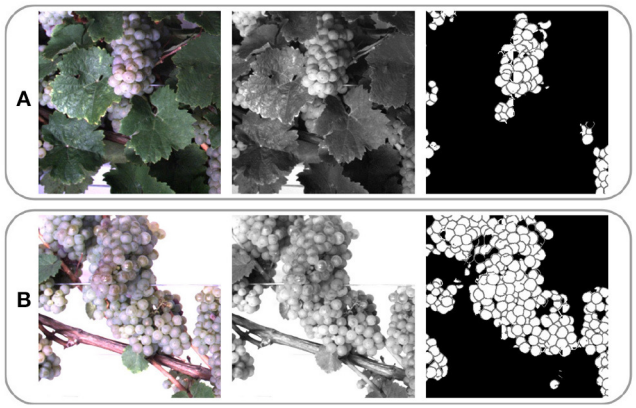


FIGURE 4 | Example patches extracted from images of **(A)** the occluded and **(B)** non-occluded domain. One row shows the same patch in RGB, grayscale, and berry mask format. One column represents a patch pair $\{\boldsymbol{x}_{\text{occ}}, \boldsymbol{x}_{\text{non}}\}$.

and the berry mask are stacked to form a multichannel 2D input. In the following, the used data is explained in more detail.

3.2.2. Natural Data

We convert the acquired RGB images into grayscale images in order to develop an efficient approach that is independent of the berry color. Covering the whole variability of possible berry colors is complex and not feasible in our case. For example, in the case of green berries, the color also does not serve to differentiate them from leaves.

Since the Phenoliner platform revisits the vine row for each data collection of the two domains, the images depicting the same scene are acquired at different times and from different positions, leading to differences in translation, rotation and scale. Moreover, the defoliation of vines causes a movement of the branches and grape bunches, and additional environmental changes between the two acquisition time points can result in different scenes in the aligned patches.

However, to obtain aligned image pairs for a qualitative evaluation, we manually align images from both domains. For this, we compute a four-parameter Helmert transformation (Helmert, 1880) between the two domains, where we manually define corresponding keypoints per image pair to calculate the parameters. We apply this transformation to images from the non-occluded domain to register them to the occluded domain.

Due to computational limitations, we use a sliding window of size 656 px × 656 px and stride 162 px to extract patches from the grayscale images. **Figure 4** illustrates one RGB patch, the grayscale patch, and the corresponding berry mask, which is explained in the following subsection, for both domains. We denote the aligned patch pair $\boldsymbol{x}^{\text{N}} = \{\boldsymbol{x}^{\text{N}}_{\text{occ}}, \boldsymbol{x}^{\text{N}}_{\text{non}}\}$, where $\boldsymbol{x}^{\text{N}} \in X^{\text{N}}$.

3.2.3. Semantic Segmentation Mask (Berry Mask)

Besides the acquired images, we use a berry mask, obtained with a semantic segmentation approach, presented by Zabawa et al. (2019). The identification of regions containing berries and the detection of single berry instances is performed with a convolutional neural network. The network uses a MobilenetV2 (Sandler et al., 2018) encoder and a DeepLabV3+ decoder (Chen et al., 2018). The network assigns each image pixel to one of the classes `background`, `berry-edge`, or `berry`, which corresponds to the grayscale values 0, 127, and 255. In contrast to a standard semantic segmentation without distinguishing between different instances, we use the additional class `berry-edge` to ensure the separation of single berries, which allows the counting of berries using a connected component approach.

For our task of generating a highly probable scenario behind leaves, the berry mask supports the discriminability of relevant information like the berries from the surrounding information in the image, and the generation of separated berries. In addition, since the berry masks contain a masking of existing berries, it provides further knowledge about which areas in the images do not show occlusions and should be preserved in the revelation process and where potentially occlusions might appear, which are areas that are unmasked.

Since we are interested in scenes in the image that depict berries, we only integrate patch pairs in training and testing, whose berry mask of the non-occluded domain contains more than 1/24 background pixels and mask of the occluded domain contains at least one pixel whose class differs from the background class.

3.2.4. Synthetic Data

One challenge for our application is that the amount of paired data from both domains containing both, occluded and non-occluded regions of berries, is limited for training a reliable model and for evaluation. We, therefore, resort to generate artificially modified images, where berries are artificially occluded, based on natural images of defoliated plants. This allows us to generate a large dataset to ease the described lack of natural images of both domains. We denote this synthetic dataset with X^{S}. The natural patches $\boldsymbol{x}^{\text{N}}_{\text{non}}$ of the non-occluded domain serve as a basis. We create paired patches $\{\boldsymbol{x}^{\text{S}}_{\text{non}}, \boldsymbol{x}^{\text{S}}_{\text{occ}}\}$ where $\boldsymbol{x}^{\text{S}}_{\text{non}} = \boldsymbol{x}^{\text{N}}_{\text{non}}$.

To generate $\boldsymbol{x}^{\text{S}}_{\text{occ}}$, we apply artificial data modification on both training and test data. We artificially occlude the patches using 24 different wine leaves (**Figure 5B**) with various shapes extracted from the natural dataset and use them as occluding objects in the patches. We use 18 leaves for augmenting the training set and six leaves to augment the test set. On the basis of one image patch $\boldsymbol{x}^{\text{S}}_{\text{non}}$, we create up to nine corresponding synthetically augmented versions of $\boldsymbol{x}^{\text{S}}_{\text{occ}}$ for the training set, resulting in nine aligned image patch pairs. During the procedure, a leaf is randomly selected from the set of leaves and rotated by a randomly chosen angle $\alpha \in \{-50, -30, -10, 0, 10, 30, 50, 70\}$. Converted to grayscale, it randomly overlays the grayscale patch and occludes parts of the visible berries. These steps are also performed for patches $\boldsymbol{x}^{\text{S}}_{\text{non}}$ of the test set. However, here only three new patch pairs are created. After applying artificial data modification, the proportion of test data amounts to ∼18–23% of the extracted patches depending on the type of defoliation (see **Figure 6B**). The split of the data into training and test data is illustrated visually and numerically in **Figure 6**.

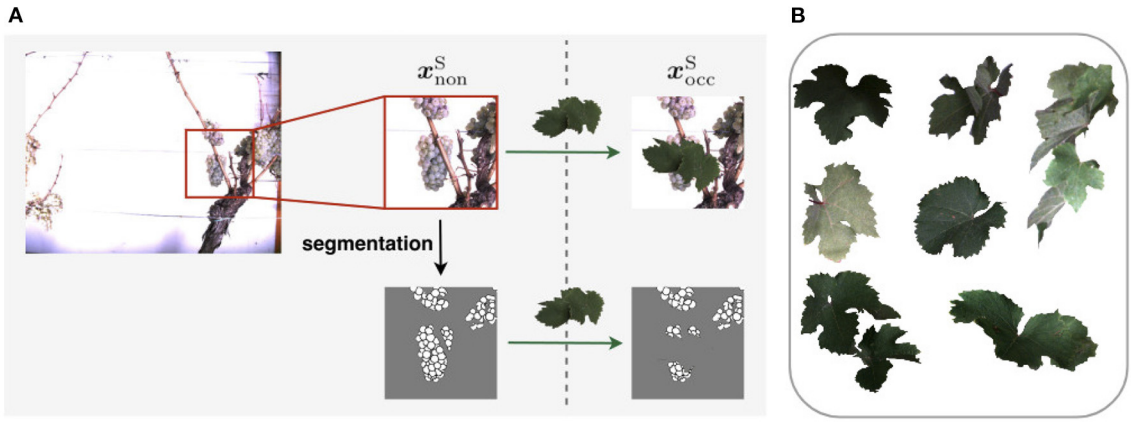
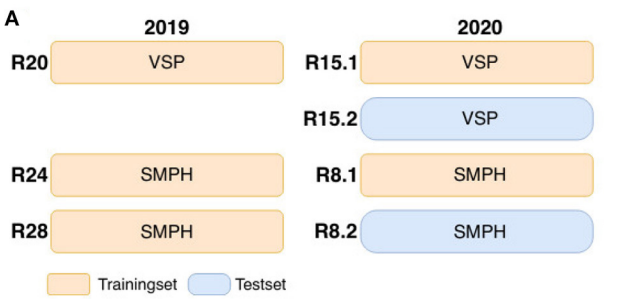


FIGURE 5 | (A) Visualization of synthetic data creation. The visualization indicates the use of an artificial leaf to calculate the corresponding mask of x^S_{occ} of the occluded domain instead of using the segmentation mask prediction based on the occluded RGB-image.Exemplary leaves used for data augmentation. **(B)** Shows exemplary leaves used for data augmentation.



B

	Training	Testing	Sum
SMPH	25.209 81.39%	5.763 18.61%	30.972 100%
VSP	15.996 77.82%	4.560 22.18%	20.556 100%

FIGURE 6 | Dataset composition. **(A)** Shows the visual division of training and test data for the two data sets semi minimal pruned hedge (SMPH) (see **Figure 3C**) and vertical shoot positioned system (VSP) (see **Figure 3F**) which were collected in the years 2019 and 2020. The data sets are taken from different rows R. Each row corresponds to one of the two sets. In 2020 the same rows were run twice. Once left (R15.1 and R8.1) and once right (R15.2 and R8.2) of the row. Orange marked areas are used for training, blue marked ones for testing. The table shown in **(B)** indicates the corresponding numbers of training and test data patches.

The test data is taken from the dataset collected in 2020 (see **Figure 6A**).

For each synthetic grayscale image, we calculate a corresponding berry mask. However, depending on the used procedure, the appearance of the berry mask differs. In our work, we create the masks for the two domains, as illustrated in **Figure 5A**. The mask of the non-occluded patch x^S_{non} is based on the segmentation step, described in Section 3.3.2, which needs RGB images as input. We compute the mask of x^S_{occ} by overlaying the pixels of the non-occluded mask of x^S_{non} that are covered with a leaf in the RGB, or respectively grayscale patch. These pixels in the berry mask are assigned to the class `background`. The leaf pixels adjacent to berry pixels are changed to `berry-edge` pixels. In this way, the overlapped berries have a closed contour. By adding these edges, the synthetic data thus has the same characteristics as berry masks derived from the natural data. With this step, we create two corresponding masks, x^S_{occ} and x^S_{non}, which match exactly in the non-occluded pixel.

Another way to define the occluded mask is a direct computation as for x^S_{non} using the segmentation step to create a predicted mask of the patch. Since the berry mask is an estimation, the class of individual non-occluded pixels may differ between x^S_{non} and x^S_{occ}. For a simplified analysis, we have chosen the first option.

Overall, for dataset ${}^{VSP}X^S$, we obtain 20.556 synthetic patch pairs, and for dataset ${}^{SMPH}X^S$, we obtain 30.972 synthetic patch pairs **Figure 6B**.

3.2.5. Challenges

Various challenges occur in the data, which influence our training and thus our results. Since our reference masks are not manually derived but are estimations, uncertainties can occur. For example, not all visible berries are entirely shown in the images of the non-occluded domain. Therefore, it can happen that either berries are missed or only partly detected in the mask. Additionally, the estimated contour in the berry mask may not be closed and parts of the berry region may be classified as

background. Thus, these errors in the reference could be learned in the model. Furthermore, there are images in the non-occluded domain, which contain leaves despite defoliation. In an ideal case, the model learns to ignore these faults in defoliation. Other challenges are the varying sharpness of the patches. This can be caused by resizing the data, shadows, or the varying distance of the berries to the camera. Furthermore, the illumination varies within the data, e.g., due to the coverage by surrounding objects like branches or leaves or the distance of the berries to the camera. Also, worth noting are the different growth stages of the grapes in 2019 and 2020, so the grapes have different sizes due to different berry sizes.

3.3. Model Evaluation

3.3.1. Data Post-processing

After the test phase, the generated masks do not only contain the values 0, 127, and 255. There are also mixed pixels that are not clearly assigned to one of the three classes. We use thresholding to ensure that only the values 0, 127, and 255 appear in the mask. We use the following class assignment.

- Pixel values in the interval [0, 50] are set to value 0 and assigned to class `background`.
- Pixel values in the interval [50, 180] are set to value 127 and assigned to class `berry-edge`.
- Pixel values in the interval [180, 255] are set to value 255 and assigned to class `berry`.

3.3.2. Evaluation Metrics

In the following, we describe several evaluation metrics used for our experiments. The first metric we use is the *area* F_c, that we define as the number of pixels within a mask that correspond to a class c with $c \in \{\texttt{background}, \texttt{berry-edge}, \texttt{berry}\}$. With area F_c and the generated area $\widetilde{F}_c$, which is based on the generated mask of the cGAN, we calculate the *intersection over union* IoU by dividing the area of overlap by the area of union.

$$\mathrm{IoU}_c = \frac{F_c \cap \widetilde{F}_c}{F_c \cup \widetilde{F}_c} \tag{3}$$

The IoU compares the similarity between two arbitrary shapes.

The second metric we use is the *pearson product-moment correlation coefficient*. It gives a measure of the degree of linear relationship between two variables. The correlation coefficient is obtained by the correlation coefficient matrix Q, which is calculated by means of the covariance matrix C,

$$\mathrm{Q}_{i,j} = \frac{\mathrm{C}_{i,j}}{\sqrt{\mathrm{C}_{i,i} \cdot \mathrm{C}_{i,j}}} \tag{4}$$

where i and j indicate the row and column index, respectively. The values of Q are between -1 and 1, inclusive. The correlation coefficient ρ between two variables can then be expressed by $\rho = \mathrm{Q}_{0,1}$. A correlation coefficient ρ equals 1 indicates that both input variables are equal. We use the correlation to compare the generated images $\widetilde{\boldsymbol{x}}_{\mathrm{non}}$ from the model with the input $\boldsymbol{x}_{\mathrm{occ}}$ as well as the target output $\boldsymbol{x}_{\mathrm{non}}$ on pixel level.

The *coefficient of determination*, also denoted by R^2, indicates the relationship between a predicted value with respect to a reference value. It provides a measure of how well-observed references are replicated by the model. In our case, we use the R^2 value for the comparison between the predicted number of berries generated by the model and the reference number from the berries manually counted in the non-occluded domain. Plots, as illustrated in **Figure 10**, represent the generated distribution of the model compared to the reference. Please note, that the gray line represents the reference values. The optimal generated samples are distributed along this line, reflected in a R^2 value equal to 1.

The *counting* is based on the procedure described in the work of Zabawa et al. (2020). The counting is performed based on the masks, which are predicted with the convolutional neural network presented in their work. The classes `background` and `berry-edge` are discarded, and the counting is solely performed with pixels of the class `berry`. Before counting the number of connected components of the berry mask, we introduce geometrical and qualitative filter stages to improve the count. Filtering follows the observations of Zabawa et al. (2020) (Table 3) which show that when the filter is applied, the misclassifications for VSP cutting decrease by 9% and for SMPH cutting by 11%. For the first step of filtering, we discard elements that are smaller than 25 pixels, since these artifacts are too small to represent berries. Secondly, we exploit the knowledge that berries are roughly round by removing objects with a minor-major-axis ratio below 0.3 and an insufficient area. The actual area of each component is compared to the expected area based on a radius, which is computed as the mean of the minor and major axis of the component. Lastly, we check how well each object is surrounded by an edge, since most high confidence predictions are well surrounded by an edge. For further details, we refer the reader to Zabawa et al. (2019).

Another metric we use for a visual comparison is the *generation map*. Generation mapping is used to visualize the differences between two masks. In our case the distances are calculated between (1) the input mask $\boldsymbol{x}_{\mathrm{occ}}$ and the generated mask $\widetilde{\boldsymbol{x}}_{\mathrm{occ}}$ (**Figure 7A**), (2) the target output mask of $\boldsymbol{x}_{\mathrm{non}}$ and the generated mask of $\widetilde{\boldsymbol{x}}_{\mathrm{non}}$ (**Figure 7B**), and lastly (3) the target output mask of $\boldsymbol{x}_{\mathrm{non}}$ and the generated mask of $\widetilde{\boldsymbol{x}}_{\mathrm{non}}$ including only two classes, where `berry` and `berry-edge` are considered as one class (**Figure 7C**). We denote this mask as binary mask.

The different colors allow us to make a statement about the area in which, for example, berries are generated where none are present in the reference. The colors can be analyzed as follows: For **Figures 7A,B**, at pixel positions with a medium orange and medium blue discoloration, either the class berry is predicted to be an edge or the class edge is predicted to be a berry. These two cases are acceptable for our task, since we do not want to map the reference, but generate images, which provide highly probable results with a distribution that matches the input. The other pixel values are to be avoided, since at these positions for a light and dark orange discoloration the classes `berry` and `berry-edge` are generated, where in the reference `background` occurs. At the positions with a light and dark blue discoloration the class

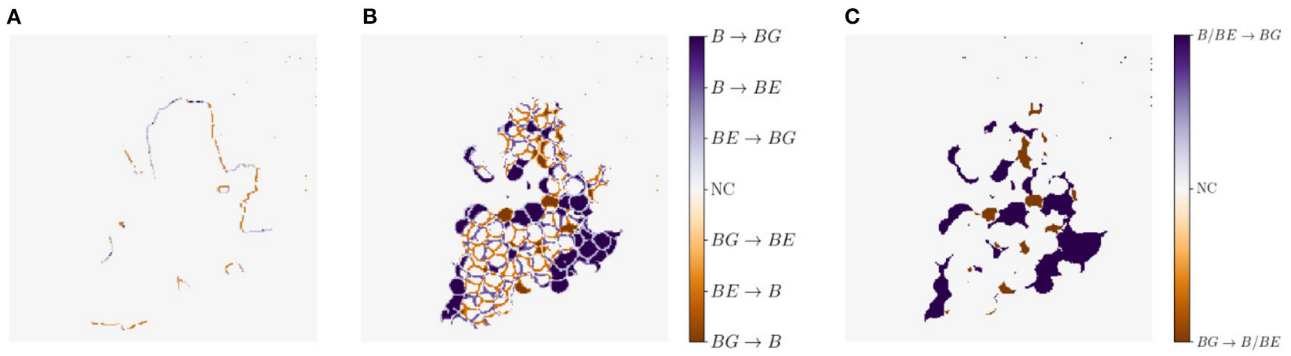


FIGURE 7 | Example representation of generation maps in **(A)** the occluded domain, and **(B,C)** the non-occluded domain for the berry mask **(B)** and a binary mask **(C)** where classes `berry` *B* and `berry-edge` *BE* are combined to one class. `Background` is indicated by *BG* and if there is no class change it is indicated by *NC* for no change.

`background` is generated, where in the reference the class `berry` or `berry-edge` is present. The generation map, where only two classes are included, highlights the non-acceptable pixel regions in the generated map.

4. RESULTS

4.1. Experimental Setup

Our experiments are designed to apply a domain-transfer using cGANs (section 3.1.2) to (1) learn a distribution by which we can generate a highly probable scenario of how occluded grapes could look like depending on the input, and (2) improve the counting of grapevine berries in images. To address the challenge of limited amount of natural data X^{N}, we perform four experiments based on a synthetic dataset X^{S}. In Experiment 5 (section 4.6), we show the applicability to natural data X^{N} based on the models and results learned in earlier experiments.

For our experiments, we define five different datasets, which are listed in **Table 1**. In addition to the natural data, described in section 3.3.1, we introduce a synthetic dataset in section 3.3.3. All five datasets, *Dataset 1-5*, will again be divided into the different types of defoliation SMPH and VSP. For our experiments, we also distinguish the set of input channels used. We claim that using a combination of grayscale image (G) and berry mask (B), denoted as GB, gives more accurate results both visually and in respect to berry counting than using the berry mask alone without grayscale information. We support this claim in Experiment 1. In the following experiments, the datasets are accordingly used with GB channels.

We resize all image patches to a uniform size of 286 × 286 px with nearest neighbor interpolation. During training, we follow the procedure of Isola et al. (2017) and add small variations to the data in each epoch by randomly cropping patches of size 256 × 256 px from the given patches. Additionally, patches are randomly flipped vertically, and the values within the patches are scaled and shifted to the range [−1, 1]. For testing, only scaling and shifting of the values to the range [-1,1] is carried out. The network output is scaled back to the value range [0, 255] for visualization.

TABLE 1 | Definitions of the used datasets.

Definition	*N*	S	SMPH	VSP	GB	B	Experiment				
							1	2	3	4	5
Dataset 1		✗		✗	✗		✗	✗	✗	✗	
Dataset 2		✗		✗		✗	✗				
Dataset 3		✗	✗		✗					✗	
Dataset 4	✗			✗	✗						✗
Dataset 5	✗		✗		✗						✗

The table shows which kind of data is used for which experiment.

To train the models, we use an Intel Core i7-6850 K 3.60 GHz processor and two GeForce GTX 1080Ti with 11 GB RAM. The models are trained over 600 epochs. We use the Adam optimizer, where the learning rate is constant at 0.0004 for the first 300 epochs and is reduced linearly toward 0 for the last 300 epochs.

4.2. Experiment 1—Comparison of Generation Quality Based on GB and B Data

With the first experiment, we analyze how the grayscale channel influences (i) the reproduction of hidden berries and (ii) the counting of berries per image. With the help of the grayscale channel G, it is possible to derive information about the presence of objects such as berries, leaves, and branches. Theoretically, this information helps to identify positions in the image where berries might be generated, for example, behind leaves or branches. In practice, however, in the non-occluded reference, a part of the berries is not present, since a proportion of berries is still occluded due to leaves or bigger branches not being cut away. This makes training more difficult, since it is generally learned that new berries should not be generated at the position of branches that have not been cut away. This implies, that we cannot expect to make new berries visible in the generated output $\tilde{\boldsymbol{x}}_{\mathrm{non}}$ while testing, that are never present in the reference data $\boldsymbol{x}_{\mathrm{non}}$ of the training set.

To get further insights into this, we analyze whether ignoring the G channel leads to a generation of berries in areas such as

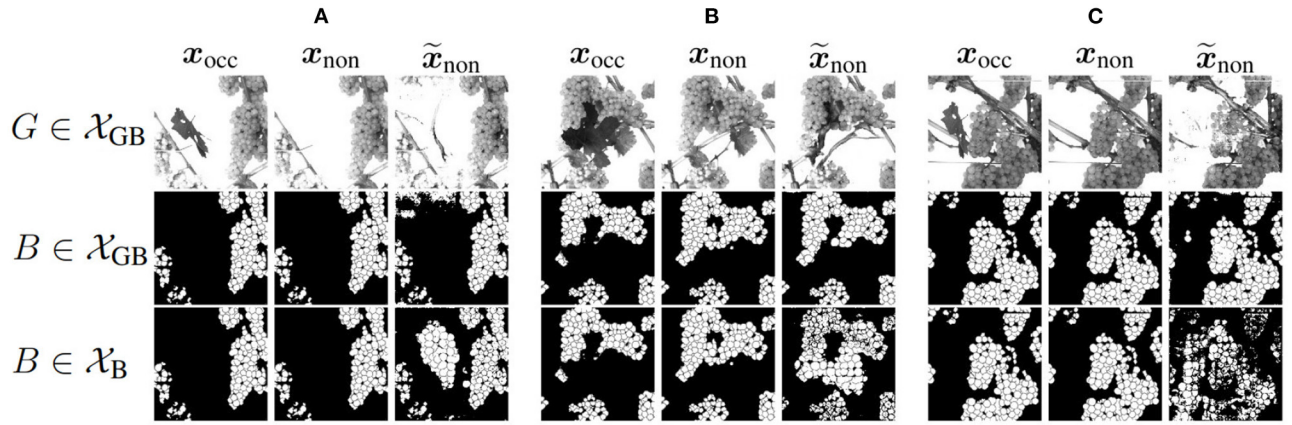


FIGURE 8 | Visual representation of generated test results $\widetilde{\boldsymbol{x}}_{\text{non}}$ in the non-occluded domain of *Dataset 1* including GB channel and *Dataset 2* including only B channel in comparison to reference target output $\boldsymbol{x}_{\text{non}}$ of the non-occluded domain and input $\boldsymbol{x}_{\text{occ}}$ of the non-occluded domain. Three example **(A–C)** are shown. The first row shows the G channel of $\mathcal{X}_{\text{GB}}$. The second row illustrates the mask B of $\mathcal{X}_{\text{GB}}$. The last row represents the corresponding result of $\mathcal{X}_{\text{B}}$.

branches. Moreover, we investigate if using channel B only is better suited on natural data, because information such as color, exposure, and lighting conditions have no influence. Thus, this experiment determines that the G channel adds value to the experiments and shows what this added value looks like.

4.2.1. Used Data, Model, and Evaluation Metrics

For this experiment, we train a cGAN model on each of the training sets of *Dataset 1* and *Dataset 2*. The evaluation is based on the corresponding test sets. Since we want to determine the value of the G channel with this experiment, we limit the used data exclusively to defoliation type VSP. SMPH type shows proportionally similar outcomes to the VSP results.

To compare the two datasets X_{B} and X_{GB}, we use the described metrics in Section 3.4.2. We compare the correlation and the IoU in the occluded domain between the input $\boldsymbol{x}_{\text{occ}}$ and the generated input $\widetilde{\boldsymbol{x}}_{\text{occ}}$, as well as in the non-occluded domain between the target output $\boldsymbol{x}_{\text{non}}$ and the generated output $\widetilde{\boldsymbol{x}}_{\text{non}}$ for both datasets. The generated input $\widetilde{\boldsymbol{x}}_{\text{occ}}$ is computed by taking the generated output $\widetilde{\boldsymbol{x}}_{\text{non}}$ and occlude the same pixels in the berry mask which are occluded in the input by a synthetic leaf.

4.2.2. Results

Figure 8 shows three example results to visually compare X_{B} and X_{GB}. The first two columns of an example show the reference of the two domains, where the third column represents the generated output $\widetilde{\boldsymbol{x}}_{\text{non}}$. The first row shows the grayscale channel of GB, the second row shows the mask channel of GB, and the bottom row shows the mask channel of B.

Using data without the G channel leads to higher generalizability regarding different varieties such as color, lighting conditions, and occlusions. Remarkable for the mask of B (row 3) is that for input patches containing many berries, proportionally too large and therefore too few berries are generated in $\widetilde{\boldsymbol{x}}_{\text{non}}$ of the test results. This applies to the entire dataset and is demonstrated by **Figures 8A,B**. Generated berries in $\widetilde{\boldsymbol{x}}_{\text{non}}$ of X_{GB} adapt better to existing berries in mask $\boldsymbol{x}_{\text{occ}}$ than $\widetilde{\boldsymbol{x}}_{\text{non}}$ of X_{B}. Furthermore, it turns out that the model trained on X_{B} has problems in generating patches with many berries. The berries are not only too big, but also in general berries are difficult to represent in their shape, as seen in **Figures 8B,C**.

Another positive aspect of X_{GB} is the already mentioned point that background information of the grayscale patch is included in the generation of new berries. The model learns to recognize where background is present in the patch and thus does not generate new berries in $\widetilde{\boldsymbol{x}}_{\text{non}}$ in contrast to the model trained on X_{B}. This is particularly obvious in Example 1 (see **Figure 8A**), where a whole grape bunch is generated in the center of the mask. In the reference input and output of G, it is visible that on this position, background occurs.

In the following, we will take a look at the objective metrics described above. If we compare them regarding the X_{GB} and X_{B} input, we notice that the results for correlation between $\boldsymbol{x}_{\text{occ}}$ and $\widetilde{\boldsymbol{x}}_{\text{occ}}$ are similar (see **Figure 9A**). For X_{B}, there are more generated patches with a correlation smaller than 0.8 and, therefore, less with a higher correlation. The correlation histogram between $\boldsymbol{x}_{\text{non}}$ and $\widetilde{\boldsymbol{x}}_{\text{non}}$, shown in **Figure 9B**, shows different distributions for the datasets. While the correlation histogram of BG, presented in orange, shows a left-skewed distribution, the amount of test patches of B increases on average with increasing correlation. At a correlation in the interval of [0.99, 1], represented by the right bar, the distribution shows a striking peak. However, there is a larger proportion of values below a correlation of 0.85. Even in the interval [0.85, 0.99], the percentages of patches for GB are higher than for B.

Figures 9B, **10C** present a counting comparison of the different models in the non-occluded domain using a R^2-Plot. Additionally, **Figure 9A** shows the counting results without domain-transfer, i.e. no additional generated berries. Counting applied to the target $\boldsymbol{x}_{\text{non}}$ in the non-occluded domain serves

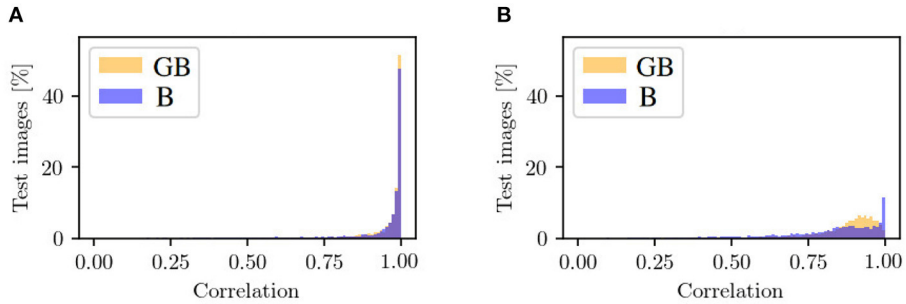


FIGURE 9 | Mask-based comparison of GB and B data with respect to **(A)** the correlation between input patch $\boldsymbol{x}_{\text{occ}}$ and generated input patch $\widetilde{\boldsymbol{x}}_{\text{occ}}$ in the occluded domain, and **(B)** the correlation between target output $\boldsymbol{x}_{\text{non}}$ and generated output $\widetilde{\boldsymbol{x}}_{\text{non}}$ in the non-occluded domain. Shown is the percentage of test images that are assigned to a specific range of correlation. One bar corresponds to the range of 0.01.

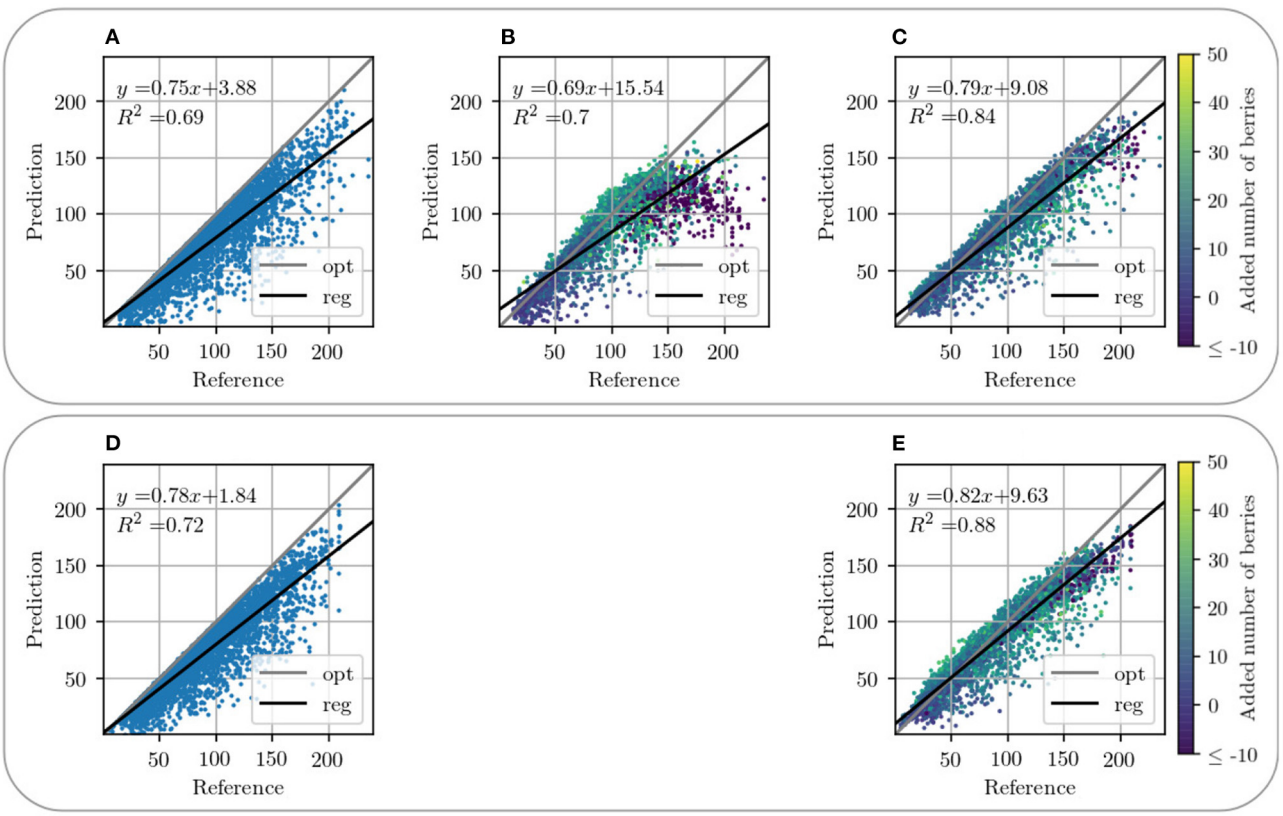


FIGURE 10 | Graphical visualization of the berry counting by a R^2-Plot. Row 1 illustrates the results for VSP defoliation while row 2 illustrates the results for SMPH defoliation. **(A,D)** Show the input of the occluded domain and **(B,C,E)** of generated berries in the non-occluded domain. Shown is the relation between the generated output $\widetilde{\boldsymbol{x}}_{\text{non}}$ in relation to reference $\boldsymbol{x}_{\text{non}}$ for data with input channel B **(B)** and GB **(C,E)**. The coloration of the data points in the plots **(B–D)** indicates the added number of berries compared to the non-occluded domain.

as the counting reference and is represented by the diagonal gray line. We observe that the results with input GB give the best matched results with respect to the reference. This is indicated visually as well as by the R^2 value of the different models, which is the highest for our approach in the non-occluded domain with input GB. As in the visual evaluation, the counting plot for input B in **Figure 10B** shows that the model indicates problems generating berries with a larger number of berries per patch. Also in the GB results, we observe that, especially with a reference counting number of more than 150 berries, the model does not reach the reference.

4.3. Experiment 2—Real vs. Generated Results in the Occluded Domain

In this experiment, we investigate whether the regions showing berries in the occluded domain stay unchanged in the transferred non-occluded domain. Furthermore, we verify that new berries are generated exclusively in the occluded area, and thus, the model detects where the appearance of berries is very likely.

4.3.1. Used Data, Model, and Evaluation Metrics

For this experiment, we use synthetic *Dataset 1* of the VSP defoliation. For evaluation, we use different masks: The first mask is the so-called generated input mask $\widetilde{\boldsymbol{x}}_{\text{occ}}$, for which we take the generated output $\widetilde{\boldsymbol{x}}_{\text{non}}$ of the test set and overlay it with the leaf used for data augmentation of the synthetic input $\boldsymbol{x}_{\text{occ}}$. The other mask is the so-called baseline mask $\boldsymbol{x}_{\text{non,leaf}}$ of this experiment. For this purpose, we use the target output $\boldsymbol{x}_{\text{non}}$ and overlay it likewise with the leaf used for data augmentation of the synthetic input $\boldsymbol{x}_{\text{occ}}$. Thus, only the non-occluded pixels of $\boldsymbol{x}_{\text{occ}}$ will remain visible in $\widetilde{\boldsymbol{x}}_{\text{occ}}$ and $\boldsymbol{x}_{\text{non,leaf}}$. The evaluation is then performed on the pairs $\{\boldsymbol{x}_{\text{occ}}, \boldsymbol{x}_{\text{occ,leaf}}\}$ and $\{\boldsymbol{x}_{\text{occ}}, \widetilde{\boldsymbol{x}}_{\text{occ}}\}$.

We use IoU and correlation as comparative metrics for this experiment. Additionally, we create generation maps which show the differences between the masks within each of the pairs $\{\boldsymbol{x}_{\text{occ}}, \boldsymbol{x}_{\text{occ,leaf}}\}$ and $\{\boldsymbol{x}_{\text{occ}}, \widetilde{\boldsymbol{x}}_{\text{occ}}\}$, as illustrated in **Figure 11**. For this experiment, the first three rows are of interest to us. The first row shows the respective grayscale patch of the generation maps. The second row shows the differences within the pair $\{\boldsymbol{x}_{\text{occ}}, \boldsymbol{x}_{\text{occ,leaf}}\}$. Row three shows the differences within the pair $\{\boldsymbol{x}_{\text{occ}}, \widetilde{\boldsymbol{x}}_{\text{occ}}\}$. The columns indicate different patch examples.

4.3.2. Results

The reference correlation within the mask pair $\{\boldsymbol{x}_{\text{occ}}, \boldsymbol{x}_{\text{occ,leaf}}\}$ is above 0.98 for all test patches. With our method, we manage to achieve a correlation of over 0.98 within the pair $\{\boldsymbol{x}_{\text{occ}}, \widetilde{\boldsymbol{x}}_{\text{occ}}\}$ for about 65% of the test images (see **Figure 9A**, orange). The remaining 35% are largely distributed over a correlation within the interval [0.75, 0.98]. The correlation strongly correlates with the IoU calculation of the `berry` area. The low correlations are either due to artifacts in the generated masks or to test images with a high number of berries. In this case, the model does not transfer all non-occluded pixels one to one into the non-occluded domain. The effect of the amount of berries in the patch is shown in the generation maps in **Figure 11** {column 1, row 3} and {column 3, row 3}.

For the patch examples in columns 2, 4, and 5, the generation maps of the pairs $\{\boldsymbol{x}_{\text{occ}}, \boldsymbol{x}_{\text{occ,leaf}}\}$ are almost identical to the generation maps of the pairs $\{\boldsymbol{x}_{\text{occ}}, \widetilde{\boldsymbol{x}}_{\text{occ}}\}$. Such maps correspond to correlation values close to 1. It is noticeable that in all five examples, the border of the leaf used for data augmentation is highlighted in the generation maps. The coloring occurs at transitions between the leaf and the adjacent `berry-edge` and `berry` pixel.

4.4. Experiment 3—Real vs. Generated Results in the Non-occluded Domain

In this experiment, we investigate the similarity of our generated output $\widetilde{\boldsymbol{x}}_{\text{non}}$ compared to the target output $\boldsymbol{x}_{\text{non}}$ in the non-occluded domain.

4.4.1. Used Data, Model, and Evaluation Metrics

In this experiment, *Dataset 1* is used to train the model. Since we are aiming only for a highly probable result rather than the exact position and shape of specific berries, for our evaluation, we additionally create a binary mask based on the berry mask, which includes only the classes `berry` and `background`. For this, we merge the classes `berry` and `berry-edge`. We compare the mask pair $\{\boldsymbol{x}_{\text{non}}, \widetilde{\boldsymbol{x}}_{\text{non}}\}$ of the non-occluded domain in respect to the berry and binary mask. We evaluate the correlation and IoU within this pair. Furthermore, we create generation maps that illustrate the difference between this pair. Exclusively for the berry mask, we calculate the area and diameter of all individual berries in the entire test data set.

4.4.2. Results

The correlation (**Figure 12A**) shows a similar left-skewed distribution for berry mask and binary mask. The majority of the test images show a correlation of above 0.8. Although our approach does not aim to generate the exact position and shape of berries, the results indicate that the similarity of the generated results and the reference are high. The IoU in **Figure 12B** also supports this finding. The IoU of the binary mask has on average higher values and is closer to the possible maximum than the berry mask. The generation maps from **Figure 11** also show this property in the fourth and fifth row. The fourth row shows example results for the berry mask, where two cases can be seen. *Case 1*: The medium orange and medium blue colors in the fourth row illustrate pixels where the classes `berry` and `berry-edge` are confused. This incorrect generation is acceptable due to the desired property of highly probable results instead of exactly matching results. *Case 2*: Dark and light blue, and dark and light orange are incorrectly generated classes that need to be avoided. In the fifth row, these pixel regions are highlighted by dark blue and dark orange. These regions either represent berries where there are no berries in the reference, or *vice versa*. Such incorrect generations shift the position and size of the grape bunches. In the example maps, however, it can be seen that *Case 1* occurs predominantly. It is obvious that berries are predicted in the right areas, but their shape and position do not correspond exactly to the reference.

At the transition from image areas with berries to `background` pixels, the second case occurs where too small or too large grape bunches are produced, because either too few or too many berries are generated. This is illustrated by the second and fourth column. The generation maps of the binary masks only highlight the areas that contradict the property of highly probable results.

To further check the similarity between generated and reference data, we consider the distributions for area and

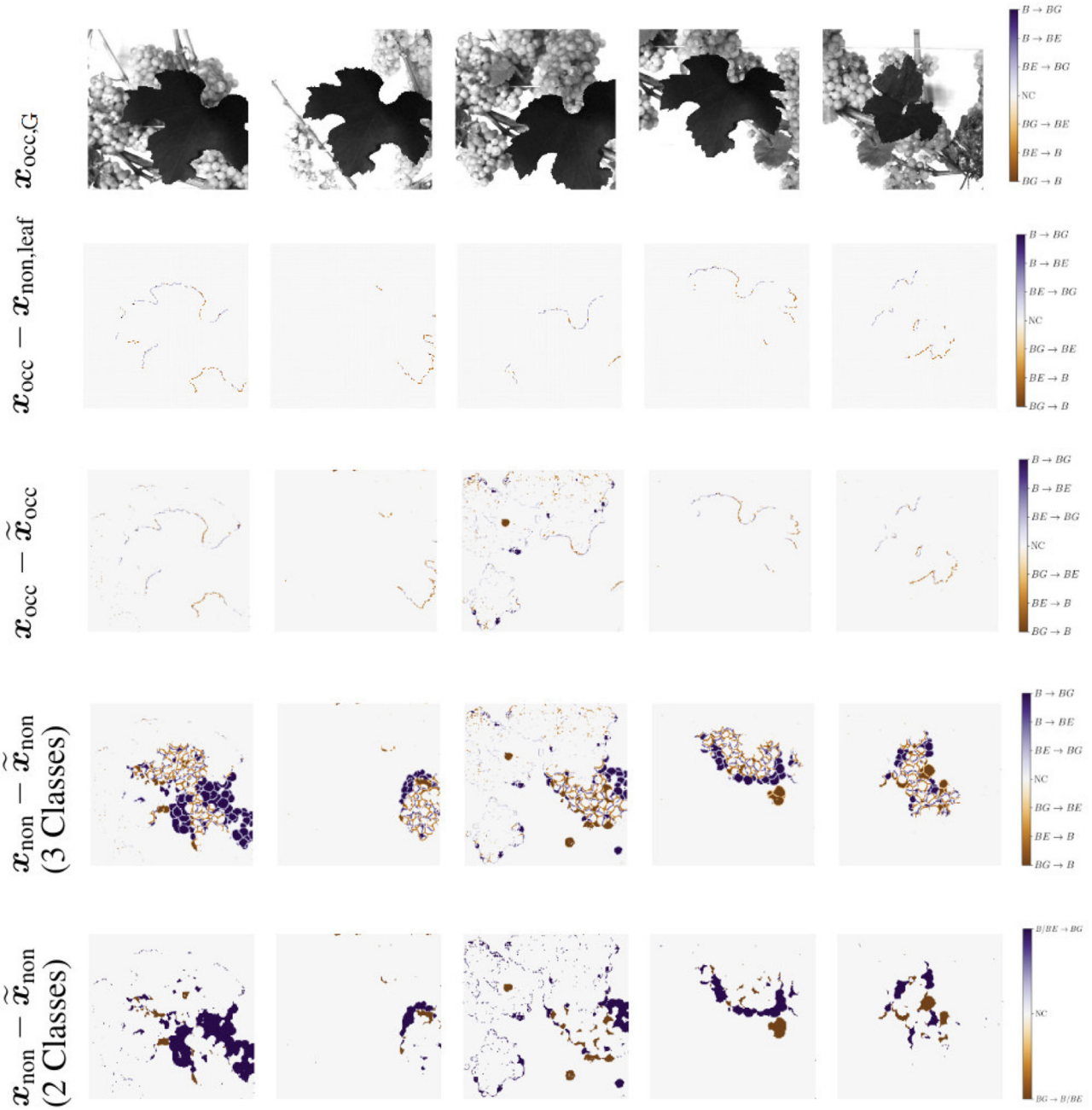


FIGURE 11 | Generation maps between target berry masks and generated output berry masks, as described in Section 3.4.2. The first row illustrates the respective grayscale input image. The second row shows the input $\boldsymbol{x}_{occ}$ compared to the target output occluded by the leaf used for creating the input $\boldsymbol{x}_{non,leaf}$. The third row shows the maps in the occluded domain. The input $\boldsymbol{x}_{occ}$ is compared to the generated output occluded by the leaf used for creating the input $\tilde{\boldsymbol{x}}_{occ}$. Second and third row occur in the occluded domain. The fourth and fifth row show the maps of the non-occluded domain. The target output $\boldsymbol{x}_{non}$ is compared to the generated output $\tilde{\boldsymbol{x}}_{non}$. In the fourth row all three classes are included. The last row illustrate the same, but including only two classes. Classes `berry-edge` *BE* and `berry` *B* are combined in one class. `Background` is indicated by *BG* and if there is no class change it is indicated by *NC* for no change.

diameter within the berry masks shown in **Figures 12C,D**. The distributions of the metrics are highly similar between generated result end reference. For both metrics, there is a slight tendency toward an increase in area and diameter for the generated berries.

4.5. Experiment 4—Counting in the Non-occluded Domain

Since the number of berries is of high importance for yield estimation, we investigate the estimation of this number in this experiment. We compare the counts based on the input

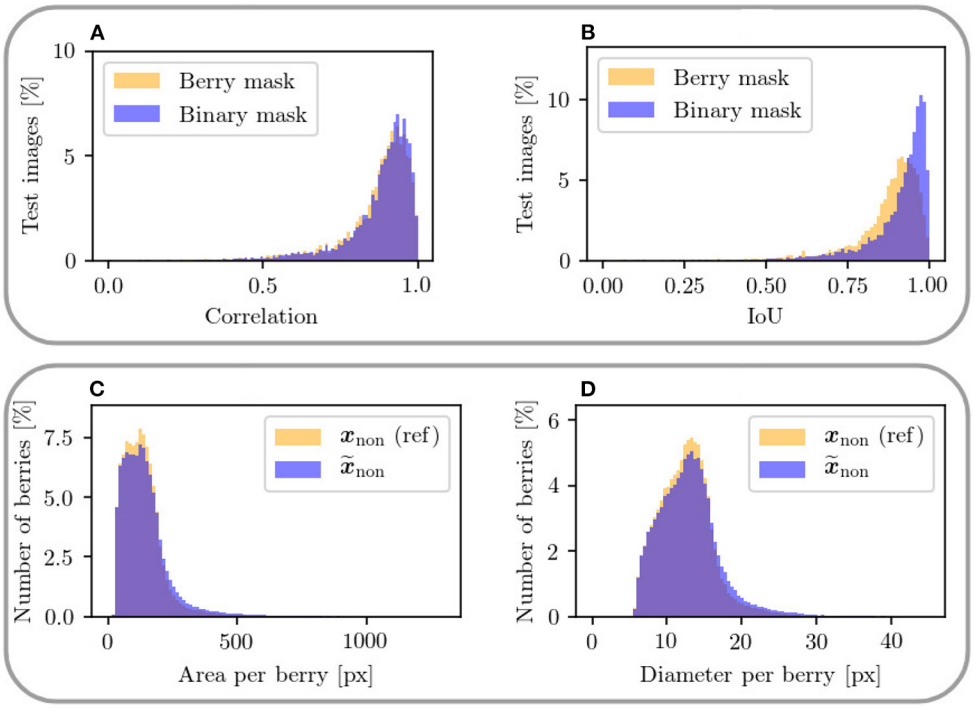


FIGURE 12 | The upper plots show a mask-based comparison within the non-occluded domain between berry mask and binary mask including only two classes for the metrics **(A)** correlation between x_{non} and $\tilde{x}_{non}$ and **(B)** IoU of the `berry` pixel in x_{non} and $\tilde{x}_{non}$. The lower plots **(C,D)** show a comparison of area and diameter per berry between target output x_{non} and generated output $\tilde{x}_{non}$ in the non-occluded domain. Only areas up to 1,300 px and diameters up to 45 px are plotted.

patches in the occluded domain and the target patches in the non-occluded domain with the generated results of our approach.

4.5.1. Used Data, Model, and Evaluation Metrics

For this experiment, we use the synthetic datasets *Dataset 1* and *Dataset 3* based on VSP and SMPH defoliation. Our model is trained on both training sets and evaluated on the corresponding test sets. During testing, we consider only the mask of the data patches. For the evaluation, we use the R^2-Plot to plot the absolute count of the input (**Figures 10A,D**) and the absolute count of the generated output of our method (**Figures 10C,E**) with the reference count from the target mask, respectively. Furthermore, we examine the distribution of the relative deviations from the reference (see **Figure 13**).

4.5.2. Results

Counting in the occluded domain, presented in **Figures 10A,D**, shows that there is an underestimation of the number of berries compared to the reference. Our model shows a shift of the number of berries toward the reference for both types of defoliation. In both cases, the R^2 value increases compared to the R^2 value of the occluded domain, which corresponds to a better approximation of the data compared to the reference. It is important to mention that not only the sample distribution shifts, but also compresses and concentrates along the reference line.

Figure 13 supports this observation. The plots show the relative difference of the counted berries in the occluded domain and our method in the non-occluded domain compared to the reference counting. Our method (blue) depicts a normal distribution with a mean near zero. If the values of the occluded distribution (orange) were increased by a factor, this would lead to a shift in the distribution, but it would still be more stretched than ours. The peaks at value 0 correspond mostly to synthetic images where the synthetic leaf does not cover any berries. This is the case, for example, with images that show few berries.

Both models exhibit problems in the generation of patches that depict more than 150 berries. This is the case for VSP (**Figure 10C**) and SMPH (**Figure 10E**). For both types of defoliation, a trend is nevertheless evident above the critical value of 150 berries. Even though an underestimation of berries tends to be counted above this value, the count fits the reference better than the count in the occluded domain.

In the occluded domain, there are data points that differ strongly from the reference. Our method reduces the amount of such points and also reduces the deviation of the highly deviating points.

4.6. Experiment 5—Application to Natural Data

One of the contributions of our work is to investigate the applicability of our approach to natural data. In detail, we evaluate whether our model generalizes to natural images when it is trained on synthetic data.

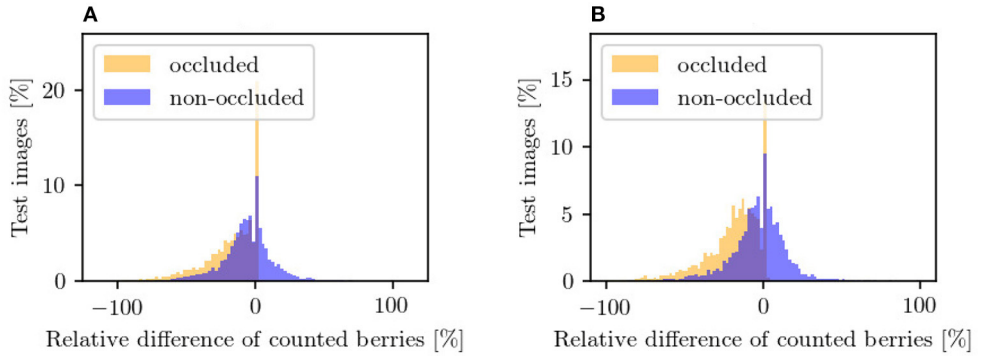


FIGURE 13 | Counting in the occluded domain (orange) and after applying our approach in the non-occluded domain (blue) relative to the reference counting in the non-occluded domain. The plots illustrate the results for **(A)** VSP defoliation and **(B)** SMPH defoliation. A negative value means that fewer berries are counted than in the reference and *vice versa*. Each bar corresponds to a width of 2%.

4.6.1. Used Data, Model, and Evaluation Metrics

We use the synthetic datasets *Dataset 1* and *Dataset 3* to train our model. For the test phase, we use the natural datasets *Dataset 4* and *Dataset 5*. One dataset each for VSP defoliation and one for SMPH defoliation.

The differences of the natural dataset to the synthetic dataset are the stronger coverage by a denser leaf canopy, the resulting deviating exposure ratios, and the lower contrast whereby the contours of the leaves are not easily distinguishable from berries. Other differences are found in the transformation applied to the natural dataset, since non-occluded areas are not identical in both domains, as already pointed out in the introduction. Depending on the patch position in the non-occluded domain in the original image, the transformation goes beyond the boundaries of the original image in the occluded domain. To achieve a patch size of 656 × 656 px which is equivalent to the cropped patch size of the dataset, the appropriate borders of the patch are filled with black spixels.

We perform our evaluation visually, which means we compare the input from the occluded domain with the generated output of our approach in the non-occluded domain. Due to the transformation issues, direct numerical comparison and evaluation between target and generated output are not useful for the majority of patches. However, we would like to give an impression of the results by means of the visual representation.

4.6.2. Results

In **Figure 14**, we provide example results of our approach applied to natural data. For each example, the first column shows the input $\boldsymbol{x}_{\text{occ}}$ of the occluded domain, the second column the reference $\boldsymbol{x}_{\text{non}}$ in the non-occluded domain, and the last column our generated output $\widetilde{\boldsymbol{x}}_{\text{non}}$ in the non-occluded domain. The first row visualizes the G channel of a patch and the second row the corresponding mask. The results show that the canopy is reduced and important areas in the patch are reproduced. Generally, the observations from the previously described experiments can be repeated. Using our generative approach, `berry` and `berry-edge` pixel regions in the input mask are also transferred to the generated output for the natural data. For input patches of the occluded domain being similar to the synthetic data (**Figures 14A–C**), the results show an expansion of the existing berry region. Our approach is also able to deal with transformation problems, as in **Figure 14A** where the transformation goes beyond the original image boundaries. There are examples, like seen in **Figure 14C**, that look similar to the target, or examples that look real compared to the input but do not reflect the target output (**Figure 14B**).

For the majority of natural data, exact transformations are not available, so this is challenging to evaluate. In examples like the one in **Figure 14D** transformation, rotation and scale fit, but due to defoliation, the orientation of the grape bunch is different in input and output target. In the input, the grape bunch is more horizontal. In the target, it is vertical. The example in **Figure 14E** shows that grape bunches are also completely different in translation due to the different weights attached to the branches. In this example, the grape bunch that is visible in the input is only partially visible at the top of the patch in the target output. The generated output adapts to the input and is also expanded, but is not comparable to the target.

Furthermore, we observe checkerboard artifacts that appear in the generated G patches (see **Figures 14A,C**). The artifacts occur more in patches that present a dense canopy.

5. DISCUSSION

5.1. Experiment 1

Our results confirm that our model trained on GB data learns where background is present. This is an important factor for realistic generated images. We found that the model trained only with berry mask B has more problems with images containing many berries than the model trained on GB data, both visually and in the counting results. The deficits in counting are explained by the fact that there are relatively few patches in the dataset with a number greater than 150 compared to the number of patches containing <150 berries. This is also true for the underestimation of the count with the GB dataset. However,

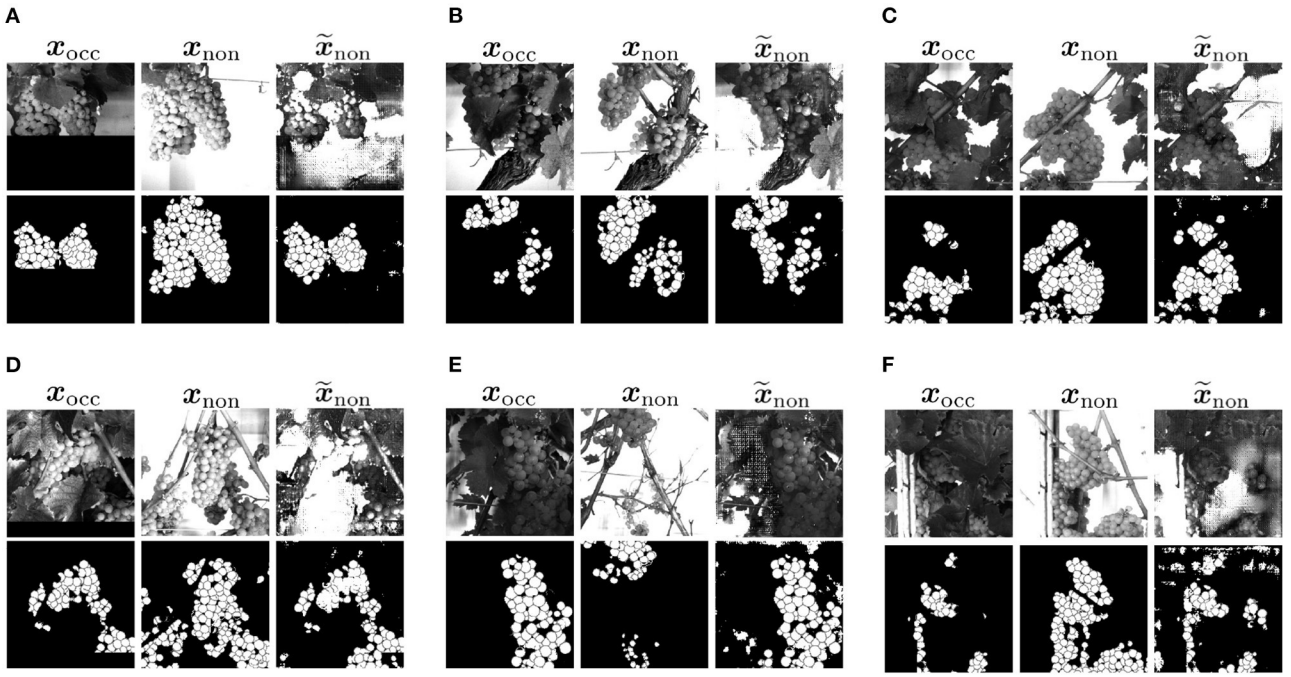


FIGURE 14 | Visual representation of generated result based on natural data input. **(A–C)** Examples with a good transformation between the patches. They are present as a minority in the natural dataset. **(D,E)** Show examples with a insufficient transformation between the patches. **(F)** Shows an example with artifacts in the generated mask.

by using the additional G channel, the result images can be generated more precisely. More detailed analyses of the berry counting can be found in Experiment 4. Taking into account the correlations and with the goal to generate highly probable results with a distribution that matches the input, rather than the exact image content of each image, Dataset 1 leads to better results on average as claimed in the beginning of the results section.

5.2. Experiment 2

We found that a high percentage of the results is correctly transferred from occluded to non-occluded domain. The occurring deviations between $\{\boldsymbol{x}_{\text{occ}}, \widetilde{\boldsymbol{x}}_{\text{occ}}\}$ can be traced back to the test results, which not only show the class values 0, 127, and 255 within the mask, but also pixels with values in between. This means that the model does not clearly assign the respective pixel to a class. At this point, we apply data post-processing to our generated data, as described in Section 3.4.1. Pixel in areas of class boundaries are particularly affected here, which is why the differences arise in these areas.

The deviations at the edges of the leaf are due to an additional edge with a width of about three pixels, which was added during the creation of the synthetic occluded input mask. The masks $\widetilde{\boldsymbol{x}}_{\text{non}}$ and $\boldsymbol{x}_{\text{non}}$, on which the masks $\widetilde{\boldsymbol{x}}_{\text{occ}}$ and $\boldsymbol{x}_{\text{occ,leaf}}$ used in this experiment are based on, show a continuation of the depicted grape branches exactly at these transitions. This results in variations between the paired masks at this location.

The key findings from this experiment are that despite individual deviations, the visible part of the mask of the occluded domain is safely transferred to the non-occluded domain and stays unchanged. We assume that the model will make no result-altering changes.

5.3. Experiment 3

Although our approach does not aim to generate the exact position and shape of berries, the results indicate that the similarity of the generated results and the references are high. The observed high IoU indicates a similar position of the grape bunches independent of the berry objects in the generated result compared to the reference. Berries are predicted in the right areas, but their shape and position do not correspond exactly to the reference. An increasing area and diameter suggest, that if the area of the total `berry` pixel per patch remains the same, there is a possibility that too few berries are predicted.

5.4. Experiment 4

In the berry counting, the underestimation of the amount of berries per patch is clearly evident in the concealed area, which can be explained by the occlusion covering part of the berries. The results indicate that we obtain better results with our approach than when we apply only a factor to the counting. We explain the deteriorating results above a berry number of 150 by the fact that the proportion of training images with a count above the critical value is relatively small in contrast to the number of images with an amount below the critical value. Our method reduces the number of outliers and additionally reduces the variance of the highly deviant points. We achieve a shift of the distribution

as well as a compression and concentration along the reference line, so that our results are more accurate than those in the occluded domain.

5.5. Experiment 5

Generally, our findings from the previously described experiments can be confirmed within this experiment. Although it is apparent that the model trained only on the synthetic data mentioned above is not yet strong enough to obtain similarly good results for the more complex natural data as for the synthetic data, we consider the results promising. We assume that mixing natural and synthetic data or using more complex synthetic training data can improve the results. The checkerboard artifacts that we observed could be reduced by improving the generator (Odena et al., 2016). This could also result in reduced artifacts, like they occur in the mask in **Figure 14F**. The artifacts occur more in patches that present a dense canopy.

5.6. Future Directions

To make the model more robust and generalizable to variations between natural and synthetic data, the synthetic data can be designed with more complex changes, for example, by increasing the synthetic occlusion through the use of more leaves per patch. In addition, brightness and contrast could be varied, for example, to reduce the dominant white background of the synthetic data and thus make it more difficult for the model to detect the occlusion. Interesting future work is the application of the model to other varieties and to see how it behaves. We assume that the model applied on varieties with a comparable or smaller grape bunch size and a similar data appearance will behave similarly to our presented results. With a larger grape bunch size and thus a larger number of berries, the model might have to be re-trained in order to achieve an accurate result for a large number of berries. Another promising future direction is to train the model from a combination of synthetic images and a limited amount of natural images. In this case, the transformation between the two required domains needs to be accurate enough and suitable data must be selected. Another possibility would involve extensive manual work on the transformation between the domains or more sophisticated techniques such as image warping. In the future, the checkerboard artifacts that occur in data could be reduced by replacing the transpose convolution layer of the decoder in the U-Net generator with bi-linear up-sampling operations, as described by Odena et al. (2016).

6. CONCLUSION

In this work, we have demonstrated the suitability of a conditional generative adversarial network like Pix2Pix to generate a scenario behind occlusions in grapevine images that is highly probable based on visible information in the images. Our experiments have shown that our approach has learned patterns that characterize typical berries and clusters without occlusions so that areas where berries are added and other areas where the image remains unchanged can be identified without having to provide prior knowledge about occlusions. Compared to counting with occluded areas, we show that our approach provides a count that is closer to the manual reference count. In contrast to applying a factor, our approach directly involves the appearance of the visible berries and thus better adapts to local conditions.

We have trained our conditional adversarial network-based model on synthetic data only in order to overcome the challenge of lacking aligned image pairs. We show that the model is also applicable to natural data, given that the canopy is not too dense and the variation between natural data and synthetic data is not too high.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

JK initiated, designed, and conducted the analyses. RR helped to initiate the work and co-designed the experiments. JK, AK, and LZ contributed to the data preparation. All authors contributed to the writing of this manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This project was funded by the European Agriculture Fund for Rural Development with contribution from North-Rhine Westphalia (17-02.12.01--10/16---EP-0004617925-19 -001). Furthermore, this work was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC 2070—390732324, and partially by the German Federal Ministry of Education and Research (BMBF, Bonn, Germany) in the framework of the project novisys (FKZ 031A349).

ACKNOWLEDGMENTS

The content of the manuscript have previously appeared in a preprint (Kierdorf et al., 2021).

REFERENCES

Aquino, A., Diago, M. P., Millán, B., and Tardáguila, J. (2017). A new methodology for estimating the grapevine-berry number per cluster using image analysis. *Biosyst. Eng.* 156, 80–95. doi: 10.1016/j.biosystemseng.2016.12.011

Aquino, A., Millan, B., Diago, M.-P., and Tardaguila, J. (2018). Automated early yield prediction in vineyards from on-the-go image acquisition.

Comput. Electron. Agric. 144, 26–36. doi: 10.1016/j.compag.2017.11.026

Arteta, C., Lempitsky, V., and Zisserman, A. (2016). "Counting in the wild," in *European Conference on Computer Vision* (Springer), 483–498. doi: 10.1007/978-3-319-46478-7_30

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* 28, 24. doi: 10.1145/1531326.1531330

Batista, G. E., and Monard, M. C. (2002). A study of K-nearest neighbour as an imputation method. *His* 87, 48. doi: 10.1109/METRIC.2004.1357895

Bertalmio, M., Vese, L., Sapiro, G., and Osher, S. (2003). Simultaneous structure and texture image inpainting. *IEEE Trans. Image Process.* 12, 882–889. doi: 10.1109/TIP.2003.815261

Chen, L., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR, abs/1802.02611*. doi: 10.1007/978-3-030-01234-2_49

Clingeleffer, P. R., Martin, S., Dunn, G., and Krstic, M. (2001). *Crop Development, Crop Estimation and Crop Control to Secure Quality and Production of Major Wine Grape Varieties: A National Approach*. Final Report. Grape and Wine Research & Development Corporation.

Coviello, L., Cristoforetti, M., Jurman, G., and Furlanello, C. (2020). GBCNet: in-field grape berries counting for yield estimation by dilated CNNs. *Appl. Sci.* 10, 4870. doi: 10.3390/app10144870

Dekel, T., Gan, C., Krishnan, D., Liu, C., and Freeman, W. T. (2018). "Sparse, smart contours to represent and edit images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 3511–3520. doi: 10.1109/CVPR.2018.00370

Diago, M., Martinez De Toda, F., Poni, S., and Tardaguila, J. (2009). "Early leaf removal for optimizing yield components, grape and wine composition in tempradillo (*Vitis vinifera* L.)," in *Proceedings of the 16th International GiESCO Symposium*, ed J. A. Wolpert, Davis, CA, 113–118.

Diago, M.-P., Correa, C., Millán, B., Barreiro, P., Valero, C., and Tardaguila, J. (2012). Grapevine yield and leaf area estimation using supervised classification methodology on rgb images taken under field conditions. *Sensors* 12, 16988–17006. doi: 10.3390/s121216988

Ehsani, K., Mottaghi, R., and Farhadi, A. (2018). "Segan: segmenting and generating the invisible," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 6144–6153. doi: 10.1109/CVPR.2018.00643

Enders, C. K. (2001). A primer on maximum likelihood algorithms available for use with missing data. *Struct. Equat. Model.* 8, 128–141. doi: 10.1207/S15328007SEM0801_7

Feng, H., Yuan, F., Skinkis, P. A., and Qian, M. C. (2015). Influence of cluster zone leaf removal on pinot noir grape chemical and volatile composition. *Food Chem.* 173, 414–423. doi: 10.1016/j.foodchem.2014.09.149

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.

Hacking, C., Poona, N., Manzan, N., and Poblete-Echeverría, C. (2019). Investigating 2-D and 3-D proximal remote sensing techniques for vineyard yield estimation. *Sensors* 19, 3652. doi: 10.3390/s19173652

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 2961–2969. doi: 10.1109/ICCV.2017.322

Helmert, F. (1880). *Die Mathematischen Physicalischen Theorieen der höheren Geodäsie*. B. G. Teubner. Available online at: https://books.google.de/books?id=g0vkwQEACAAJ

Iizuka, S., Simo-Serra, E., and Ishikawa, H. (2017). Globally and locally consistent image completion. *ACM Trans. Graph.* 36, 1–14. doi: 10.1145/3072959.3073659

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 1125–1134. doi: 10.1109/CVPR.2017.632

Kicherer, A., Herzog, K., Bendel, N., Klück, H.-C., Backhaus, A., Wieland, M., et al. (2017). Phenoliner: a new field phenotyping platform for grapevine research. *Sensors* 17, 1625. doi: 10.3390/s17071625

Kicherer, A., Roscher, R., Herzog, K., Förstner, W., and Töpfer, R. (2014). "Image based evaluation for the detection of cluster parameters in grapevine," in *XI International Conference on Grapevine Breeding and Genetics 1082*, Yanqing, 335–340. doi: 10.17660/ActaHortic.2015.1082.46

Kierdorf, J., Weber, I., Kicherer, A., Zabawa, L., Drees, L., and Roscher, R. (2021). Behind the leaves-estimation of occluded grapevine berries with conditional generative adversarial networks. *arXiv preprint arXiv:2105.10325*. doi: 10.48550/arXiv.2105.10325

Kim, J. K., and Rao, J. (2009). A unified approach to linearization variance estimation from survey data after imputation for item nonresponse. *Biometrika* 96, 917–932. doi: 10.1093/biomet/asp041

Lee, D., Kim, J., Moon, W.-J., and Ye, J. C. (2019). "Collagan: collaborative GAN for missing image data imputation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 2487–2496. doi: 10.1109/CVPR.2019.00259

Lempitsky, V., and Zisserman, A. (2010). Learning to count objects in images. *Adv. Neural Inform. Process. Syst.* 23, 1324–1332.

Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., and Catanzaro, B. (2018). "Image inpainting for irregular holes using partial convolutions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, 85–100. doi: 10.1007/978-3-030-01252-6_6

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60, 91–110. doi: 10.1023/B:VISI.0000029664.99615.94

Mack, J., Lenz, C., Teutrine, J., and Steinhage, V. (2017). High-precision 3d detection and reconstruction of grapes from laser range data for efficient phenotyping based on supervised learning. *Comput. Electron. Agric.* 135, 300–311. doi: 10.1016/j.compag.2017.02.017

Mack, J., Schindler, F., Rist, F., Herzog, K., Töpfer, R., and Steinhage, V. (2018). Semantic labeling and reconstruction of grape bunches from 3D range data using a new RGB-D feature descriptor. *Comput. Electron. Agric.* 155, 96–102. doi: 10.1016/j.compag.2018.10.011

May, P. (1972). Forecasting the grape crop. *Australien Wine, Brewing and Spirit Review*. 90, 46–48.

Mirza, M., and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*. doi: 10.48550/arXiv.1411.1784

Nuske, S., Achar, S., Bates, T., Narasimhan, S., and Singh, S. (2011). "Yield estimation in vineyards by visual grape detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, 2352–2358. doi: 10.1109/IROS.2011.6095069

Nuske, S., Wilshusen, K., Achar, S., Yoder, L., Narasimhan, S., and Singh, S. (2014). Automated visual yield estimation in vineyards. *J. Field Robot.* 31, 837–860. doi: 10.1002/rob.21541

Nyarko, E. K., Vidović, I., Radočaj, K., and Cupec, R. (2018). A nearest neighbor approach for fruit recognition in RGB-D images based on detection of convex surfaces. *Expert Syst. Appl.* 114, 454–466. doi: 10.1016/j.eswa.2018.07.048

Odena, A., Dumoulin, V., and Olah, C. (2016). Deconvolution and checkerboard artifacts. *Distill* 1, e3. doi: 10.23915/distill.00003

Ostyakov, P., Suvorov, R., Logacheva, E., Khomenko, O., and Nikolenko, S. I. (2018). Seigan: Towards compositional image generation by simultaneously learning to segment, enhance, and inpaint. *arXiv preprint arXiv:1811.07630*. doi: 10.48550/arXiv.1811.07630

Paul Cohen, J., Boucher, G., Glastonbury, C. A., Lo, H. Z., and Bengio, Y. (2017). "Count-ception: counting by fully convolutional redundant counting," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, Venice, 18–26. doi: 10.1109/ICCVW.2017.9

Robins, J. M., and Wang, N. (2000). Inference for imputation estimators. *Biometrika* 87, 113–124. doi: 10.1093/biomet/87.1.113

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Munich: Springer), 234–241. doi: 10.1007/978-3-319-24574-4_28

Roscher, R., Herzog, K., Kunkel, A., Kicherer, A., Töpfer, R., and Förstner, W. (2014). Automated image analysis framework for high-throughput determination of grapevine berry sizes using conditional random fields. *Comput. Electron. Agric.* 100, 148–158. doi: 10.1016/j.compag.2013.11.008

Rubin, D. B. (1996). Multiple imputation after 18+ years. *J. Am. Stat. Assoc.* 91, 473–489. doi: 10.1080/01621459.1996.10476908

Rubin, D. B. (2004). *Multiple Imputation for Nonresponse in Surveys, Vol. 81*. John Wiley & Sons.

Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., and Chen, L. (2018). Inverted residuals and linear bottlenecks: mobile networks for classification, detection and segmentation. *CoRR, abs/1801.04381*. doi: 10.1109/CVPR.2018.00474

Schöler, F., and Steinhage, V. (2015). Automated 3D reconstruction of grape cluster architecture from sensor data for efficient phenotyping. *Comput. Electron. Agric*. 114, 163–177. doi: 10.1016/j.compag.2015.04.001

Van Buuren, S., and Oudshoorn, K. (1999). *Flexible Multivariate Imputation by MICE*. Leiden: TNO.

von Hippel, P. T., and Bartlett, J. (2012). Maximum likelihood multiple imputation: Faster imputations and consistent standard errors without posterior draws. *Statistical Sci*. 36, 400–420. doi: 10.1214/20-STS793

Xie, W., Noble, J. A., and Zisserman, A. (2018). Microscopy cell counting and detection with fully convolutional regression networks. *Comput. Methods Biomech. Biomed. Eng*. 6, 283–292. doi: 10.1080/21681163.2016.1149104

Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., et al. (2019). "Foreground-aware image inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 5840–5848. doi: 10.1109/CVPR.2019.00599

Yan, X., Wang, F., Liu, W., Yu, Y., He, S., and Pan, J. (2019). "Visualizing the invisible: occluded vehicle segmentation and recovery," in *Proceedings of the IEEE International Conference on Computer Vision*, Salt Lake City, UT, 7618–7627. doi: 10.1109/ICCV.2019.00771

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2018). "Generative image inpainting with contextual attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 5505–5514. doi: 10.1109/CVPR.2018.00577

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2019). "Free-form image inpainting with gated convolution," in *Proceedings of the IEEE International Conference on Computer Vision*, Long Beach, CA, 4471–4480. doi: 10.1109/ICCV.2019.00457

Zabawa, L., Kicherer, A., Klingbeil, L., Milioto, A., Topfer, R., Kuhlmann, H., et al. (2019). "Detection of single grapevine berries in images using fully convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Long Beach, CA. doi: 10.1109/CVPRW.2019.00313

Zabawa, L., Kicherer, A., Klingbeil, L., Töpfer, R., Kuhlmann, H., and Roscher, R. (2020). Counting of grapevine berries in images via semantic segmentation using convolutional neural networks. *ISPRS J. Photogr. Remote Sens*. 164, 73–83. doi: 10.1016/j.isprsjprs.2020.04.002

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.