Cucumber Segmentation and Occlusion Recovery for Autonomous Picking Robot

Sung Jay Kim
Dept. of Biosystems Engineering
Seoul National University
sj0408@snu.ac.kr

Sungjin Hong
Dept. of Mechanical Engineering
Seoul National University
hongluck@snu.ac.kr

Jae Won Chang
Dept. of Data Science
Seoul National University
jaewon415@snu.ac.kr

Abstract

Computer vision tasks such as image recognition, object detection, and semantic segmentation made many contributions to autonomous harvesting. However, these tasks are limited to visible parts of the object in the image. Still, they pose a challenge in detecting green vegetables with the subtle color differences in the neighborhood of targeted fruit. In this sense, the nature of cucumbers makes them hard to detect. To address this issue, we study reconstructing the occluded part of the cucumber to help autonomous robots detect and locate the picking point position. A dataset with cucumber images from three different farms located in South Korea is generated. The dataset is superimposed with synthetic leaf patches to simulate the occlusion effect. Using this dataset, we propose an amodal segmentation model with a U-Net reconstruction network. The model consists of a course mask segmentation module for bounding box regression and classification, a visible mask segmentation module for refining visible masks, and an amodal segmentation module for refining the amodal mask. Finally, the refined amodal mask and refined visible mask are concatenated and trained with U-Net. Our proposed model outperforms the previous shape prior embedding, at least in the cucumber dataset.

1. Introduction

Over the last few years, there has been a growing interest in the autonomous robot for harvesting agricultural products in the greenhouse. Because many field operations are cost-ineffective and labor-intensive – crops must be hand-picked at the right time to ensure good quality and high market value. Using technology in the predefined growing setting now permeates almost all facets of our lives. It is becoming the norm as to balance out the labor shortage in the places where farmers are more dependent on hand-harvesting.

Many computer vision tasks such as image recognition [9] [25], object detection [6] [22] [21], and semantic segmentation [24] [13] made a notable contribution. These may well have provided a foundation for the robot visual system in distinguishing the crops from leaves and other backgrounds [27] [10] [30]. But these tasks are limited to the visible part of the object in the image and still pose a challenge in detecting green vegetables with the subtle color differences in the neighborhood of targeted fruit. In that sense, the nature of the cucumber makes it hard to detect. The contrast between the cucumber and its leaf is not apparent, and the leaves can partially occlude cucumber. This is shown in Figure 1.

Intuitively, working with the standard cucumber color scheme has been explored. Zhang et al. [29] used blue and saturation color components to reduce the illumination effect and improve the pixel's semantic information. As an extension to this method, Mao et al. [14] proposed a cucumber detection pipeline based on I-RELIEF color component selection and machine learning techniques. The pipeline is as follows: (1) select three color components using the I-RELIEF module; (2) use the multi-path convolutional neural network (CNN) for feature extraction; (3) classify the cucumber fruit using a support vector machine (SVM) clas-

Figure 1: Cucumbers partially occluded by leaves or stems

sifier. Fernandez et al. [4] also proposed a way to quantify the intensity difference between the blue and green color component in the RGB and quantify the hue component in the HSV color space to make the SVM algorithm robust to illumination conditions. The proposals listed above adopted a way to utilize multiple color spaces to distinguish cucumber fruit from other parts of the image. However, these detection algorithms only work well under specific light conditions and do not suggest the picking point position considering that cucumbers are in various sizes and found in random orientations.

To tackle these problems, we study the problem of reconstructing the occluded part of the cucumber to help the autonomous robot detect and locate the picking point position.

2. Related Work

This section will first explore the instance segmentation task for dealing with occluded objects and then explore three different approaches to the occluded object reconstruction. These are: semantic segmentation, amodal segmentation, and generative model.

2.1. Instance Segmentation with Occlusion

Finding a target object from the mishmash stack is an ongoing area of research in computer vision. Because a pile of objects tends to come in different shapes and arrangements, it is hard to pick robots to aim for the desired object. Motivated by this challenge, Wada et al. [26] designed a system to recognise the visible and occluded region from the given image. The system consists of two parts: (1) The image synthesis of stacked objects generated with ground truth of visible and occluded region proposal of each instance; (2) The instance segmentation model that extends mask RCNN for multi-class segmentation. The instance mask predicted in the first stage of the Mask RCNN is converted to a density map and used to predict the instance masks in the second stage. Then these two stages are fused pixel-wise together to predict visible, occluded, and other in the final layer.

Chen et al. [2] also designed a framework to handle the occlusion problem. It starts by generating a category-independent segmentation proposal using multiscale combinatorial grouping. Then, an SDS-based architecture uses these proposals to extract features that are fed into class-specific classifiers as inputs to obtain a likelihood map and occluding regions. The output from the architecture also serves as inputs to the exemplar-based shape predictor to obtain a better shape estimation of an object. Finally, graph cut with occlusion handling to infer occluding regions, shape predictions, class-specific likelihood maps are formulated into an energy minimization problem to obtain the desired segmentation based on segmentation proposals with top classification scores. Similarly, Ke et al. [11] proposed a method for modeling the structure that decouples the boundaries of both occluding and occluded instances. This method consists of three modules: (1) Backbone with FPN for feature extraction from the image; (2) Fully convolutional one-stage object detection for each instance's proposal region prediction; (3) The occlusion-aware mask head, bilayer GCN structure, for decoupling overlapping relations from the object detection. The last module reformulates the traditional class agnostic segmentation to guide target object segmentation.

2.2. Semantic Segmentation

Traditional semantic segmentation labels each pixel in the image with corresponding semantic information. Purkait et al. [19] extended this idea and assigned a group of semantic labels at each pixel, indicating whether it is hidden or visible. They use U-Net [24] integrated model to identify the area of the occlusion. This model comprises encoder-decoder followed by instance normalization layers and ReLUs except for the last layer. Furthermore, at the last layer, the grouping strategy predicts semantic labels of the visible object along with the occluded portion and groups the semantic categories into one background.

2.3. Amodal Instance Segmentation

Amodal perception can interpolate the object's physical structure when parts of it are not visible. Recent studies applied this idea to segment an instance of the object with its occluded features. Li and Malik [12] proposed the earliest work on generating an amodal dataset. They randomly cropped an image with at least one foreground object instance and overlaid another random object instance on top of the cropped image. The dataset is then evaluated on the Faster RCNN model [22]. Inspired by the previous work, Zhu et al. [31] extended the Open Surfaces annotation tool [1] to generate COCO amodal annotations. They also proposed two deep networks - ExpandMask and AmodalMask - and compared them to DeepMask [17] and SharpMask [18] as the baselines. ExpandMask takes an image and a mask generated by SharpMask as an input and outputs an amodal mask, whereas AmodalMask takes an image and predicts an amodal mask.

Instead of focusing on augmenting the dataset from the existing dataset, several amodal segmentation models are developed to help advance amodal research. Qi et al. [20] proposed a generic amodal segmentation network that infers a missing shape of the instances in the image and outputs the complete shape of the object. It consists of an occlusion classification branch, determining whether there is an occlusion in the RoI or not, and multi-level coding, guiding mask prediction to complete the structure of the instances. Xiao et al. [28] proposed a framework to help algorithm mimic human's amodal perception. It extracts the features from the

image and predicts the coarse visible mask and the coarse amodal mask. Then these two masks each go into the visible mask segmentation module and amodal mask segmentation module. The former refines the visible mask using the amodal mask and the reclassification regularizer. Furthermore, the latter refines the amodal mask concentrating on the feature of the visible region and the object's shape to help alleviate the misleading occlusion features.

2.4. Generative Model

The generative adversarial network goes through a back-and-forth process, generating the training-like data and differentiating the data from the actual data until it comes to its satisfaction [7]. Ehsani et al. [3] proposed a GAN-based model called SeGAN that reconstructs the occluded portion of the object. They first create a large-scale occlusion dataset from the photo-realistic 3D scenes by changing the camera's location in the various scenes. Then use this generated data to segment occluded and non-occluded regions of the object and generate the appearance of the occluded region using cGAN [15].

3. Method

3.1. Data Acquisition

The dataset consists of cucumber images from three different farms, located in Gimje and Goheong, South Korea. The images are collected using four different types of camera - mobile phone camera, DLSR camera, and depth camera RealSense D435i under various lighting conditions. A detailed breakdown of the number of raw images in the dataset is shown in Table 1. Because ground truth annotations must represent the overall shape of the instances to recover the occluded part of the instances, some modifications to the dataset have been made. First, all incomplete masks of occluded instances are removed. Then synthetic leaf-shaped patches are generated to occlude each instance with 60% probability. Lastly, images with no masks are removed from the dataset. Table 2 shows the number of instances after modification.

3.2. Synthetic Patch

Some modifications to the dataset have been made to train the model to recover the occluded parts of the instances because ground truth annotations should represent the overall shape of the instances. First, all incomplete masks of occluded instances were removed. Then synthetic leaf-shaped patches were generated to occlude each instance's with 60% probability. The annotations of the synthetic patch were excluded from the baseline experiment as its structure did not have any heads for processing invisible mask. However, annotations of the synthetic patches are required for the amodal instance segmentation model in which the heads for processing invisible masks are included. Lastly, images with no masks or incomplete masks were removed from the dataset. Table 2 shows the number of instances after the modifications.

Because our amodal instance segmentation model include heads for processing invisible masks, the dataset is augmented to generate invisible masks. Generating synthetic patches similar to that of the masks is an efficient and time-saving process. That being said, the human annotator must extrapolate or interpolate the occluded part of the cucumber when annotating the raw images so that human-induced annotation errors are reduced even with a synthetic patch. As mentioned above, the incomplete masks with occluded regions were removed, and masks that contain perfect cucumber shapes were only preserved to label invisible masks and cucumber reasonably. Then, synthetic patches were superimposed on the annotated area of the original image. One hundred patches were made from our train dataset by manually cropping cucumber stem and leaf as shown in Figure 2 (a). The patches are randomly generated inside the annotated area with a 60% probability in respect to the annotation size. These patches are then synthesized into the area with gradient-domain image processing for spontaneous synthesis [16]. Image stitching results in discontinuous points in the image with unnatural features, as shown in Figure 2 (b).

So, we implemented a gradient domain image processing that reduces discontinuity and generates a more natural synthesis. This technique extracts image gradients and solves the Poisson equations in (1) and (2) while adjusting the patch pixels. The H represents an improved version of patch B that blends in better with source image A, N represents the number of valid neighbors inside the patch within the boundary, and Ω represents the selection area in the B and H while excluding the boundary with partial Ω.

Let $B_x = -B_{x-1,y} - B_{x+1,y}$

$B_y = -B_{x,y-1} - B_{x,y+1}$

Figure 2: Synthetic patch augmentation (a) Annotated Image (b) Simple Image stitching (c) Gradient Domain Image Processing

$$H_{(x,y)} = A_{(x,y)} \forall (x,y) \in \partial B \tag{1}$$

$$|\nabla B_{(x,y)}| = 4B_{(x,y)} + B_x + B_y \tag{2}$$

$$\begin{aligned} |N| H_{(x,y)} - \sum_{(dx,dy)+(x,y)\in\Omega} H_{(x+dx,y+dy)} - \sum_{(dx,dy)+(x,y)\in\partial\Omega} A_{(x+dx,y+dy)} \\ = \sum_{(dx,dy)+(x,y)\in(\Omega\cup\partial\Omega)} B_{(x+dx,y+dy)} - B_{(x,y)} \end{aligned} \tag{3}$$

As shown in Figure 2 (c) the leaf patches are naturally synthesized into the image.

3.3. Cucumber Segmentation

Xiao et al. [28]'s amodal segmentation model suggested a reconstruction network based on a pre-trained auto-encoder in which the latent space generated by the encoder works as a shape prior to each class. The use of shape prior refines the amodal mask such that the model targets the COCO dataset. Each class in the COCO dataset embodies shape features that each latent space of the auto-encoder can represent. However, in this study, the main goal is to recover the occlusion region in a cucumber class. Like other agricultural products, the cucumber comes in dynamic fruit size and shape from the aspects of morphology. In that, implementing a pre-trained shape prior for occlusion reconstruction is not a promising solution; we implemented U-Net as a reconstruction network to handle high shape variation characteristics in cucumber.

Semantic segmentation algorithm U-Net was first developed by Ronneberger et al. [23], built upon the concept of a fully convolutional network. It only contains convolutional layers and does not have any dense layer as a part of the architecture. In the medical imaging field, U-Net is widely used as a reconstruction model [5] because it can overcome bottleneck problems. The algorithm contains two paths. The first path is the contraction path that consists of repeated 3 x 3 unpadded convolutions followed by the ReLU activation function and 2 x 2 max-pooling operations. This path captures the context in the image. The second path is expanding path, consisting of upsampling of feature map followed by 2 x 2 convolution and 3 x 3 convolutions. This path uses context from the first path to improve the localization.

3.4. Overview of our approach

In this paper, we implement an amodal segmentation model with U-Net as a reconstruction network instead of an auto-encoder. The overall architecture is based on Xiao et al. [28]'s work except for the removal of the re-class network, which is unnecessary for our single class dataset. First, the coarse mask segmentation module consists of bounding box regression, classification, coarse visible mask, and coarse amodal mask. The ROI feature F goes through visible mask head (f_v) and amodal mask head (f_a) which consist of four convolution layers and one deconvolution layer. And then, in the visible mask segmentation module, coarse amodal mask and feature F refines the visible mask. Finally, in the amodal mask segmentation module, a refined visible mask and reconstruction network refine the amodal mask. The refined amodal mask and refined visible mask are concatenated together and go through the amodal mask head for the final amodal mask prediction, where U-Net is now used in the amodal mask segmentation module instead of the auto-encoder.

3.5. Loss Function

3.5.1 The coarse mask segmentation module

This module aims to extract the visual feature information and predict the coarse amodal mask M_a^c and the coarse visible mask M_v^c. Four loss terms are used in this module: classification loss L_{cls}, bounding box regression loss L_{reg}, a coarse amodal mask loss $L_{BCE}(M_a^c, M_a^g)$, and a coarse visible mask loss $L_{BCE}(M_v^c, M_v^g)$, where $L_{BCE}(\cdot,\cdot)$ is the binary cross entropy loss function. It corresponds to a grey section in the Figure 4.

3.5.2 The visible mask segmentation module

This module refines the visible mask using the amodal mask. This amodal mask uses visible region to distinguish occlusion in the image and alleviate the effect of the background features. The loss term of visible mask refinement is $L_v^r = \frac{1}{N}\sum_i^N L_{BCE}(f_v(\mathbf{F_i M^c_{a,i}}), \mathbf{M^g_{v,i}})$. It corresponds to a blue section in the Figure 4.

3.5.3 Feature Matching

The feature matching accelerates and compresses the network model. It reduces the gap between feature maps and

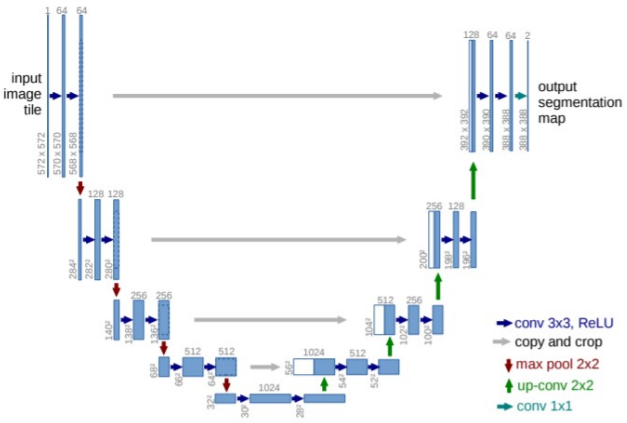

Figure 3: U-Net architecture Ronneberger et al. [23]

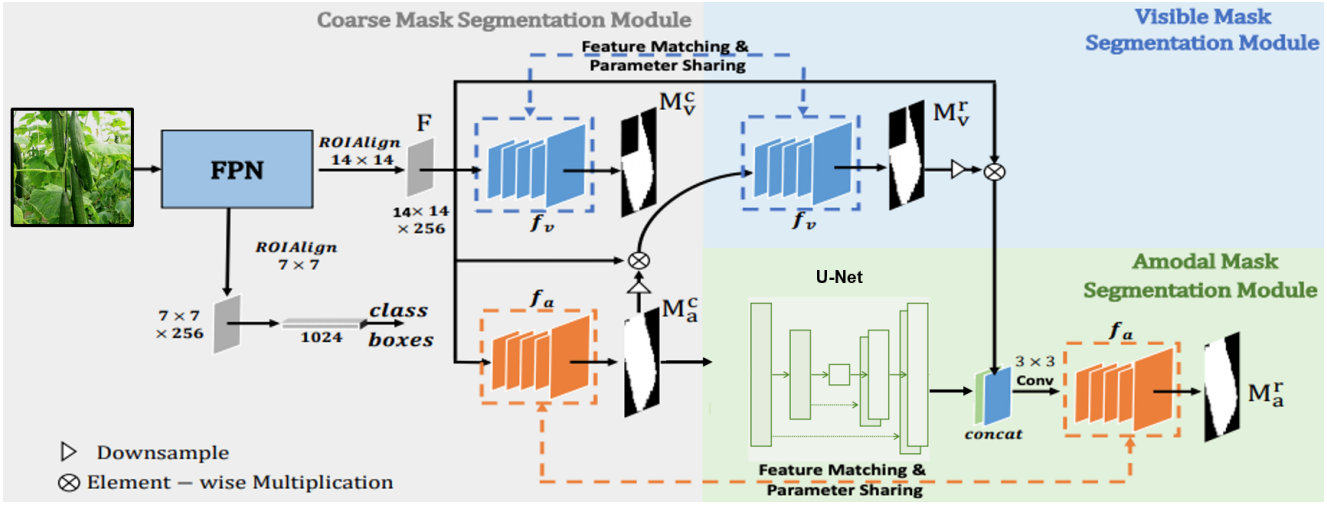

Figure 4: The overview of present study

the refined mask prediction. The loss between the coarse visible mask prediction and refined visible mask prediction focus on the visible region for visible mask segmentation. The loss of the visible mask head is $L_{vfm} = \frac{1}{N \cdot S} \sum_{i,j}^{N,S} \lambda_j L_S(f_v^{(j)}(\mathbf{F_i}), f_v^{(j)}(\mathbf{F_i} \cdot \mathbf{M^c_{a,i}}))$, where N, S, and L_S is the number of instances, the number of convolution layers of the visible mask, and cosine similarity respectively.

3.5.4 The amodal mask segmentation module

This module is defined to refine the coarse amodal mask by using the feature of the visible region and the shape prior. This helps the model alleviate the misleading effect of the occlusion feature. To imitate the perception of human that infers the amodal mask by concentrating on the appearance of the visible region and shape prior knowledge. This module uses $L_a^r = \frac{1}{N} \sum_i^N L_{CE}(f_a(cat(\mathbf{M^r_{v,i}}, \mathbf{M^k_{sp,i}})), \mathbf{M^g_{a,i}})$ loss function. And to enhance the capacity of focusing on the visible region when applying feature matching to the amodal mask head. This module uses $L_{afm} = \frac{1}{N \cdot S} \sum_{i,j}^{N,S} \lambda_j L_S(f_a^j(\mathbf{F_i}), f_a^{(j)}(\mathbf{F_i} \cdot \mathbf{M^r_{v,i}}))$ loss function. It corresponds to a green section in the Figure 4.

3.5.5 Final loss

The model adds up all these loss functions in the section 3.5.x together to compute the final loss.

4. Experiment

4.1. Dataset

The model is evaluated on the cucumber datasets. The training and test sets contain 1209 and 213 images, respectively, and we hold out 213 images as a validation set. Each dataset contains 2848, 328, and 434 cucumber instances. The annotations of synthetic patches are not required for the Mask-RCNN [8] model since its structure does not include any heads for processing invisible mask.However, annotations of the synthetic patches are required for the amodal instance segmentation model in which the heads for processing invisible masks are included. During the training process data augmentation random crop, flip, contrast, brightness was implemented in turn. For the data augmentation implementation, FAIR Detectron2's data augmentation module was used.

Camera	Images	Masked instances
Mobile phone	415	1391
Sony a6000	558	2442
RealSense D435i	447	1418
Total	1420	5251

Table 1: General information about the dataset

Camera	Images	Masked instances
Mobile phone	401	995
Sony a6000	506	1496
RealSense D435i	436	1119
Total	1327	3610

Table 2: Dataset after annotation modification

4.2. Metrics

The mean average precision(mAP) and mean average recall(mAR) are used to evaluate and quantify the model's

performance. Specifically, COCO dataset APIs are implemented in Detectron2. Specifically, in this paper, the predictions in regard to amodal masks are only considered.

4.3. Environment

Pytorch 1.4 deep learning framework was used, and Mask RCNN, Amodal Segmentation models were implemented with FAIR's Detecron2 API. The training is performed on a single NVIDIA QUADRO 6000 GPU, Intel Xeon Silver 4214R CPU, and 32GB RAM. During the implementation, the primary hyperparameter setting was as follows: the batch size of 128, the learning rate of 0.00025, and 20000 iterations. Training took approximately 2hrs long.

4.4. Result

Our model was first compared with the original Amodal Segmentation model on the COCOA class dataset. mAP of amodal segmentation model based on U-Net shows better performance by a small margin in mAP (Table 3). As shown in Table 4, our amodal segmentation model with U-Net reconstruction network and feature matching shows mAP of 49.22. It took about 2 hours and 12 minutes to train, and inference time was around 0.2762 seconds per image. While without feature matching, mAP is 47.95, training took about 2 hours and 18 minutes and around 0.1557 seconds per image to infer. We compare our U-Net reconstructive network model with the original amodal segmentation model on the cucumber dataset. Also, some ablation studies are included.

First, our baseline model, Mask RCNN, demonstrates relatively high mAP(46.84) and mAR(57.23) even without occlusion recovery task. In particular, this outperforms two amodal segmentation models by far. The proposed U-Net-based amodal segmentation model, however, performed higher mAP and mAR than that of baseline.

Second, the comparison is made with 2 cases of the original amodal segmentation model. For both cases, refinement based on shape prior is available, while feature matching is available for one case. As same as [28]'s experiment, the amodal segmentation model performed better With the feature matching method available. However, our model performed significantly better in all the cases when compared with our amodal segmentation model with U-net. The auto-encoder-based shape prior refinement performs well with extracting features among several classes because it stores them as prior shape knowledge. However, the U-Net seems to learn more variations in the single class dataset.

4.5. Limitation

The synthetic patch on the complete cucumber shape is necessary to retrieve morphological information of the occluded part of the cucumber in our dataset. So, we superimposed this patch to annotated part of the cucumber, and this turns out that the patch suffers the potential risk of an overfitting problem. Further investigation on the use of a synthetic patch to overcome this issue needs to be conducted.

Method	mAP	mAP50	mAP75	mAR
Amodal Segmentation [28]	35.41	56.03	38.67	37.11
Amodal Segmentation (UNet)	36.50	56.59	40.27	51.66

Table 3: The Comparison of models on COCOA cls dataset

5. Conclusion

This paper proposes a modified amodal segmentation model that enables autonomous picking robots to recognize and harvest cucumber. In particular, the results show that adding U-Net based reconstruction network outperforms the concept of shape prior embedding in the cucumber dataset. Though using shape prior may work well on recognizing the objects such as cars and books. Its performance on agricultural product dataset with large shape variation is still questionable. As such, we propose a U-Net based reconstruction network to enhance the capability of the model to predict with fewer constraints.

6. Future Work

The synthetic patches are collected from hundred leaf images from the training dataset, so there is a potential risk of overfitting. To prevent this problem, one can collect and add more patches to the dataset. In addition, as shown in Figure 6, our amodal segmentation model did not capture cucumber occluded by the other cucumber because synthetic patches were only made from the leaves. The patches taking these special cases into account can further be added.

Moreover, cucumber pictures containing incomplete masks were removed from the dataset. This removal process reduced the number of data available for training. The image of cucumbers with perfect shape will be collected for better performance and to refine the data quality. The cucumber dataset was taken from three greenhouses located in Korea. The dataset only contains a single cucumber species, so other cucumber species will be added to the dataset to enhance the model's generalizability.

7. Acknowledgement

This project could not have been possible without any support and assistance from the people who may not be enu-

Method	Shape Prior Refinement	Feature Matching	mAP	mAP50	mAP75	mAR	training time	inference time(sec/img)
Mask RCNN (ResNet50 FPN)	–	–	46.84	76.12	49.47	57.229	1 h58 m	0.1409
Amodal Segmentation (ResNet50 FPN C4)	O	O	39.96	63.35	44.08	60.58	2 h22 m	0.1588
Amodal Segmentation (ResNet50 FPN)	O	X	33.84	61.29	24.368	59.43	2 h21 m	0.1576
Amodal Segmentation (U-Net/ResNet50 FPN)	–	X	47.95	74.09	51.53	57.82	2 h18 m	0.1557
Amodal Segmentation (U-Net/ResNet50 FPN)	–	O	**49.22**	**75.11**	**55.61**	59.08	2 h12 m	0.2762

Table 4: Inference results and ablations

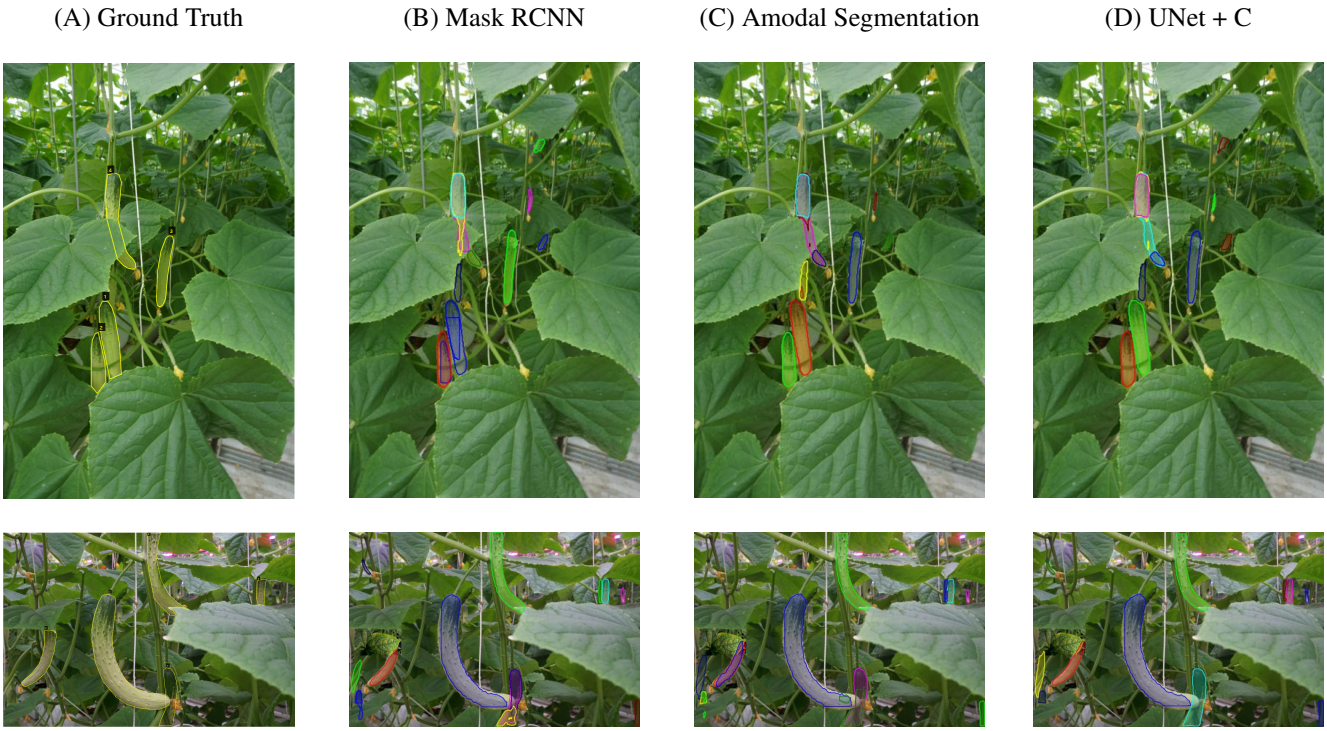

Figure 5: Some examples of instance segmentation results implemented by three different models: (B) Mask RCNN, (C) Amodal Segmentation, (D) UNet + C

merated. Their contributions to this group project are much appreciated. First, however, we would like to express a special thanks to Physical Properties and Process Engineering of Agriculture Product Laboratory (SNU PHEL) for providing us a cucumber dataset to conduct experimentation. In the process, we developed an open mindset towards agri-

Figure 6: Inference result of cucumber occluded to each other

cultural data science and hoped to maintain the learned process as a cornerstone throughout the remainder of education and beyond.

References

[1] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Opensurfaces: A richly annotated catalog of surface appearance. *ACM Trans. Graph.*, 32(4), July 2013. 2

[2] Yi-Ting Chen, Xiaokai Liu, and Ming-Hsuan Yang. Multi-instance object segmentation with occlusion handling. pages 3470–3478, 2015. 2

[3] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. 2018. 3

[4] Roemi Fernández, Héctor Montes, Jelena Surdilovic, Dragojlub Surdilovic, Pablo Gonzalez-De-Santos, and Manuel Armada. Automatic detection of field-grown cucumbers for robotic harvesting. *IEEE Access*, 6:35512–35527, 2018. 2

[5] Vahid Ghodrati, Jiaxin Shao, Mark Bydder, Ziwu Zhou, Wotao Yin, Kim-Lien Nguyen, Yingli Yang, and Peng Hu. Mr image reconstruction using deep learning: Evaluation of network structure and loss functions. *Quantitative Imaging in Medicine and Surgery*, 9:1516–1527, 09 2019. 4

[6] Ross Girshick. Fast r-cnn. page 1440–1448, 2015. 1

[7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. 2014. 3

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. 2018. 5

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 2016. 1

[10] Wei Ji, Dean Zhao, Fengyi Cheng, Bo Xu, Ying Zhang, and Jinjing Wang. Automatic recognition vision system guided for apple harvesting robot. *Computers Electrical Engineering*, 38(5):1186–1195, 2012. Special issue on Recent Advances in Security and Privacy in Distributed Communications and Image processing. 1

[11] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Deep occlusion-aware instance segmentation with overlapping bilayers. 2021. 2

[12] Ke Li and Jitendra Malik. Amodal instance segmentation. *CoRR*, abs/1604.08202, 2016. 2

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. pages 3431–3440, 2015. 1

[14] Shihan Mao, Y. Li, Y. Ma, Baohua Zhang, Jun Zhou, and K. Wang. Automatic cucumber recognition algorithm for harvesting robots in the natural environment using deep learning and multi-feature fusion. *Comput. Electron. Agric.*, 170:105254, 2020. 1

[15] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. 2014. 3

[16] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pages 313–318. 2003. 3

[17] Pedro H. O. Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. *CoRR*, abs/1506.06204, 2015. 2

[18] Pedro O. Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollàr. Learning to refine object segments, 2016. 2

[19] Pulak Purkait, Christopher Zach, and Ian Reid. Seeing behind things: Extending semantic segmentation to occluded regions, 2019. 2

[20] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. June 2019. 2

[21] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 1

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. 28, 2015. 1, 2

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 4

[24] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. 9351:234–241, 2015. (available on arXiv:1505.04597 [cs.CV]). 1, 2

[25] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. 2016. 1

[26] Kentaro Wada, Shingo Kitagawa, Kei Okada, and Masayuki Inaba. Instance segmentation of visible and occluded regions for finding and picking target from a pile of objects. 2020. 2

[27] Jingui Wu, Baohua Zhang, Jun Zhou, Yingjun Xiong, Baoxing Gu, and Xiaolong Yang. Automatic recognition of ripening tomatoes by combining multi-feature fusion with a bi-layer classification strategy for harvesting robots. *Sensors*, 19(3), 2019. 1

[28] Yuting Xiao, Yanyu Xu, Ziming Zhong, Weixin Luo, Jiawei Li, and Shenghua Gao. Amodal segmentation based on visible region segmentation and shape prior. 2020. 2, 4, 6

[29] Libin Zhang, Qinghua Yang, Yi Xun, Xiao Chen, Yongxin Ren, Ting Yuan, Yuzhi Tan, and Wei Li. Recognition of

greenhouse cucumber fruit using computer vision. *New Zealand Journal of Agricultural Research*, 50(5):1293–1298, 2007. 1

[30] Yuanshen Zhao, Liang Gong, Yixiang Huang, and Chengliang Liu. Robust tomato recognition for robotic harvesting using feature images fusion. *Sensors*, 16(2), 2016. 1

[31] Yan Zhu, Yuandong Tian, Dimitris Mexatas, and Piotr Dollár. Semantic amodal segmentation. 2016. 2