Evaluation of Deep Learning Algorithms for Instance Segmentation on Cassava Root

Chanowat Tanasaksakul
Department of Computer Engineering, King Mongkut's University of Technology Thonburi
Bangkok, Thailand
chanowat.tan@gmail.com

Khajonpong Akkarajitsakul
Big Data Experience Center and Department of Computer Engineering, King Mongkut's University of Technology Thonburi
Bangkok, Thailand
khajonpong.akk@mail.kmutt.ac.th

Tobias Wojciechowski
Institute of Bio- and Geosciences – Plant Sciences (IBG-2), Forschungszentrum Jülich GmbH
Jülich, Germany
t.wojciechowski@fz-juelich.de

***Abstract*—Different algorithms were developed and evaluated for segmenting images of cassava roots. Cassava CSRs (CSR) have different shapes and sizes. Existing methods rely on image processing algorithms to segment CSR, but these often have problems with overlapping fibrous roots. In this study, deep learning architectures for instance segmentation HTC, SOLOv2, and Mask R-CNN were applied to CSR data to investigate whether these models can overcome the specific challenges of this domain. We created a custom dataset with 1,227 training, 100 validation, and 100 test images of CSRs with different root systems and high-quality segmentation masks. Then, we compared the result of the SOLOv2 model to the Mask R-CNN and HTC models using the same network architecture. The standard metric for comparing instance segmentation models in computer vision such as mean average precision, was used for the evaluation. The results showed that all tested models applied to the considered area showed good prediction accuracy. SOLOv2 showed the best results on the test set with the computation of the model less than that of HTC. Increasing the model complexity and size did not impact the prediction accuracy in the instance segmentation of CSR. HTC achieved almost the same result as that of the SOLOv2 model but required almost doubling the number of parameters. The visualization of the predictions showed quality differences in terms of the accuracy of the segmentation masks. The SOLOv2 model showed the best masks overall. However, each model had problems in assigning the segmentation masks due to the overlap of a CSR. Many root systems still need to be measured for gene discovery and subsequent breeding. The results demonstrated the potential of deep learning instance segmentation models for CSRs needs to be increased with further research.**

Keywords—Instance segmentation, Object detection, Convolutional neural network, Cassava roots.

I. Introduction

The cassava plant has two key types of roots: fibrous roots and storage roots. Initially, fibrous roots form, but only a fraction of them turn into larger CSRs. These CSRs (CSR) provide an important source of nutrition, especially a valuable supply of carbohydrates. Therefore, understanding the growth patterns of CSRs is important in phenotyping research. In recent years, image-processing tools have been developed for plant phenotyping [1]. Recent studies on image processing [2] proposed optimized hardware and software for high-throughput phenotyping of plant roots. They were validated on soybean and wheat and demonstrated excellent correlations and heritability, setting a benchmark for open platforms for plant phenotyping. The integration of the software allows image acquisition and analysis and efficient marker-assisted breeding and genetic mapping. Recent studies [3] introduced image processing-based phenotyping techniques for plant segmentation for rosette leaf counting. These studies employed a data-driven approach and used deep learning architectures. The proposed methods showed satisfactory performance and were compared with previous techniques. The framework achieved low absolute count differences and showed promising results in leaf segmentation and counting.

Recently, convolutional neural networks (CNNs) have revolutionized plant feature recognition by providing more robustness and accuracy. In Ref. [4], cassava root counting and measurement were conducted for crop yield and quality evaluation. To overcome this challenge, a direct image-to-number prediction model was proposed using synthetic root images generated with a conditional generative adversarial network (GAN). This model consisted of a generator network and a discriminator network that competed against each other to generate realistic synthetic data. In Ref. [5], the You Only Look Once (YOLO), a popular object detection algorithm was used for computer vision. The YOLO algorithm was modified with an inverted residuals module to accurately detect garlic bulbs. The results demonstrated the superiority of the proposed IRM-YOLOv2 model compared to other classical neural networks and provided valuable insights for the application of the YOLO algorithm in food research.

For computer vision, several techniques were developed to improve the accuracy of image analysis. However, these techniques often require a large amount of data for effectiveness based on deep neural networks, which was computationally intensive and inefficient on low-cost devices [16]. A special approach known as deformable convolution (DCN) overcomes the limitations of conventional convolutional methods [17]. While regular convolution uses a fixed rectangular grid to process input images or feature maps, this approach was suitable for detecting and classifying objects deformed or partially hidden in the images. DCN introduces a deformable grid where each grid point can be shifted using learnable offsets. The convolution operation is then performed on these shifted grid points, allowing the network to adapt to deformations and occlusions in the input data. As another technique used in object detection, the DCN improved accuracy in several studies in computer vision and provided a more flexible and effective way to handle complex and challenging visual scenarios.

DCNv2 [18], an improved version of DCN, further enhanced the deformable convolutional technique by introducing an additional modulation module. This new module modulates the amplitudes of input features based on their spatial locations or bins. In DCN, the deformable convolution module allows grid points to be shifted and adjusted for deformations in the input data. However, DCNv2 allows the network to dynamically adjust the importance or

significance of features from different spatial locations or bins. With this modulation module, DCNv2 suppresses certain features depending on their spatial context.

The objective of this study was to develop an automated recognition system based on image processing for accurate segmentation of CSRs. CSRs are often confused with fibrous roots and obstructed by other CSRs. It is difficult to distinguish these roots based on their similar color and texture as shown in Fig. 1. We reviewed previous studies that used deep CNNs, as well as studies on object detection and instance segmentation. By introducing the cassava root datasets derived from many cassava root videos used in the experiments, the process of data labeling, processing, and augmentation can be improved.

II. Related Work

In recent studies on CSRs, researchers used phenotyping methods [6,7]. Cassava root traits are usually evaluated during the yield season, taking into account various factors such as the number, size, and length of roots, structural characteristics of shoots, and biomass [7]. However, it remains uncertain which specific traits allow an accurate discrimination between different genotypes. The characteristics of CSRs are difficult to distinguish. Particularly, it is hard to classify CSRs from the fibrous root because the characteristics of the storage roots partially overlap the characteristics of fibrous roots.

In the literature, deep CNNs have been used for automatic counting of features such as leaves [3,8] and plants [9]. They have also been used for object detection in garlic root cutting [5] and disease identification on plant leaves [10]. However, there is limited research on the application of segmentation methods with deep CNNs in the context of cassava roots [4]. Segmentation-based approaches [3,9] typically use a CNN-based segmentation model to identify pixels belonging to a particular plant component, typically in RGB images.

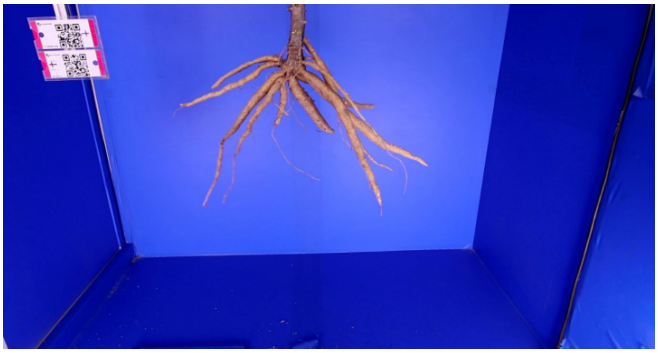


Fig. 1. Example image of casava roots.

In Ref. [9], the number of leaves was counted by summing the predictions of image patches generated by a deep CNN model. In Ref. [4], a CNN-based segmentation model and a conditional GAN were used to generate an image dataset for the classification of the age and number of cassava roots.

In the analysis, limitations were identified in the evaluation of instance segmentation models. In current research, the Common Objects in Context (COCO) data format is used widely for training and evaluation of instance segmentation models [11,12]. The COCO format provides a comprehensive and standardized framework for organizing and annotating object instances in images, enabling more reliable and consistent evaluation. The widespread adoption of the format allowed for better comparability and reproducibility of instance segmentation. COCO is widely used to evaluate the effectiveness of instance segmentation models and techniques such as Mask R-CNN. It serves as a standard reference for comparing different architectural approaches [13]. Intersection Over Union (IoU) is calculated by dividing the area of overlap between the two segmentations by the area of their union [14]. Another evaluation method called Mean Average Precision (mAP) for classification and detection tasks is usually used in instance segmentation tasks [15].

Reference [20] presented a framework called Hybrid Task Cascade (HTC). Instance segmentation of objects was identified and localized in an image. The framework was used for the interdependence of detection and segmentation by combining them in a joint multi-stage process. It included a fully convolutional branch to capture the spatial context and gradually learns more and more characteristic features while integrating complementary features at each stage. The proposed HTC framework outperforms the strong Cascade Mask R-CNN baseline on the dataset COCO. Reference [21] presented a simple and efficient approach to instance segmentation and showed impressive results. This approach was improved by an efficient instance mask representation scheme that dynamically segments each object in the image without relying on bounding box detection. The process of generating object masks is divided into two stages: Mask kernel prediction and mask feature learning. At these stages, convolution kernels or feature maps are generated in the convolution process. SOLOv2 effectively reduces the computational overhead during inference by using a matrix-based non-maximum suppression (NMS) technique. Roboflow software [19] is designed for image management and pre-processing, focusing on computer vision tasks. It provides a set of features and tools that facilitate work with images, such as data augmentation, annotation, and model deployment.

III. Dataset

The dataset used in this study was extracted from a video generously provided by Forschungszentrum Jülich, a respected member of the Helmholtz Association of German Research Centres, in Jülich, Germany. The dataset included a total of 1,427 high-resolution images of 1980 x 1080 pixels in size. The images from the dataset in Fig. 2 illustrate the complicated structure of the cassava root. These images show CSRs accompanied by several fibrous roots. The background of each image has shades of blue and dark blue, adding to the visual appeal. Capturing these diverse images provides opportunities for research and analysis in the field of cassava plant studies.

A. Data Labeling

In this study, Roboflow software was used to manually label bounding boxes, and the labeling files were saved in COCO format to create the dataset. The labeling in the format COCO contained points on the x- and y-coordinates and classes. Then, CSRs were manually labeled as polygons to contain the raw data of the image and points x, y in each row as shown in Fig. 2. These data were used to prepare the images for model input.

Fig. 2. Examples of raw images and their annotations.

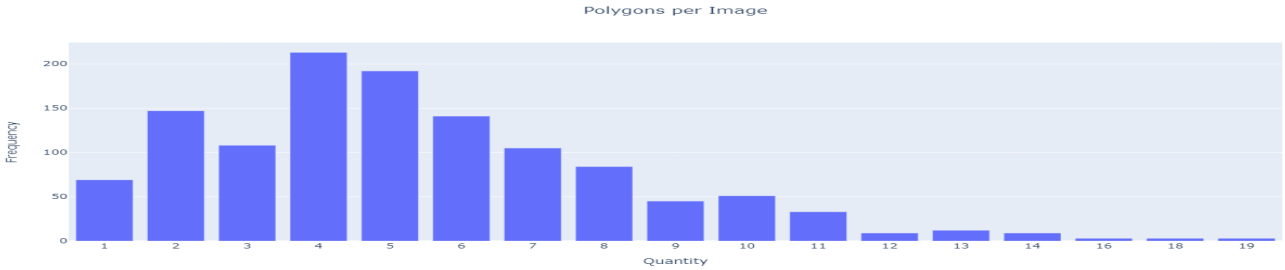
B. Data Augmentation

Rotating an image is a simple technique to improve model performance. Models learn the collections of pixels and the relationship between those in the image. However, machine learning models are brittle. They remember a particular arrangement of pixels that describes an object. When the same object was mirrored in the image, the proposed model had difficulty recognizing it. In this study, we applied the augmentation method to images by flipping the dataset horizontally and randomly increasing or decreasing the brightness and the noise in the training dataset as shown in Fig. 3.

Fig. 3. Example of image augmentation in the training dataset.

C. Statistics of Dataset

Figure 4 shows the descriptive statistics of the dataset. Each image in the dataset contained between 1 and 19 CSRs. The average number per image was 5.36. As for the annotations, each root was outlined by a mask of coordinate points. On average, there were about 41.72 coordinate points per mask, and several masks contained up to 172 points. To train and evaluate the model, the dataset was divided into three groups: Training, Validation, and Testing. The training dataset comprised 80% of the data, the validation dataset comprised 10%, and the testing and benchmarking dataset comprised the remaining 10%.



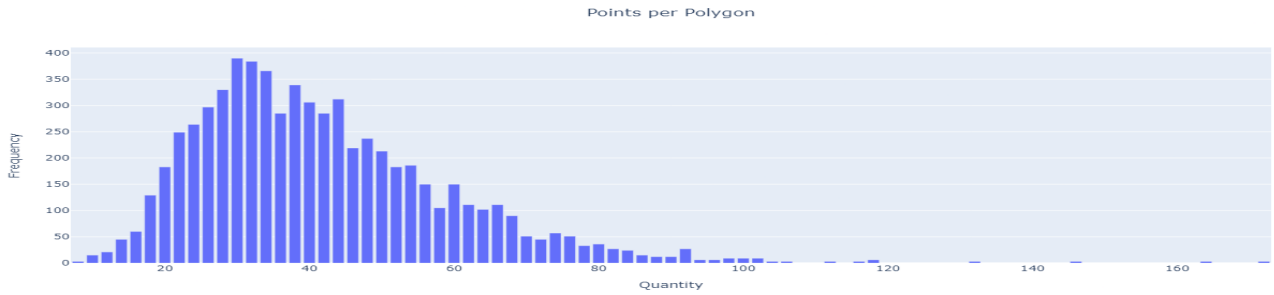


Fig. 4. Statistics of dataset

IV. Experiment

We evaluated the effectiveness of Mask R-CNN, HTC, and SOLOv2. We experimented for instance segmentation in the COCO format. Instance segmentation was used to identify and delineate individual objects in an image. In the experiment, the COCO format was used to evaluate the performance of the algorithms and compare it to that of existing methods. The results of the experiment were tested using the pre-trained backbone algorithms. In the ablation experiment, the most effective setting was found to be between single-scale training with fixed-size images and multiscale training with randomly selected images. Both methods trained the model without pre-trained weights. In the following experiment, a pre-trained weight model was used, incorporating the Backbone and the SOLOv2 weight from the 2017 COCO dataset. These experiments were conducted on the same machine.

A. Instance Segmentation

The instance segmentation was evaluated using a user-defined dataset. We measured the metric Average Precision (AP) and stochastic gradient descent optimization (SGD) to train the algorithms. In each training, a batch consisted of 8 images. The models were trained for 36 epochs with an initial learning rate of 0.01. Additionally, the learning rate was increased at the 27th and 33rd epochs. During training, we used a scaling technique in which the longer side of the image was fixed to a range of 1,333 pixels, while the other side was randomly sampled from the range of 640 and 800 pixels.

B. Test

Model training was performed on a Google Colab A100 GPU running at 40 GB VRAM, an Intel(R) Xeon(R) CPU @ 2.30GHz and 52 GB RAM. The MMDetection [22] framework was used to train and evaluate Mask-R CNN, HTC and SOLOv2. For testing each model, a Resnet 50 and 101 backbone was used in combination with FPN and DCNv2 and was added to each of the models to achieve higher accuracy so that all the models were trained and evaluated on the same basis.

C. Quantitative Evaluation

The performace of the models were evaluated based on their mAP, considering AP at different IoU thresholds for union overlap. The IoU threshold was determined using the degree of overlap required for a predicted object, which was a true positive. The mAP was the average of the AP scores computed at different IoU thresholds ranging from 0.5 to 0.95, with a step size of 0.05. Table I presents the performance and accuracy using SOLOv2. The model achieved a mAP value of 0.290, which was significantly better than that of HTC. SOLOv2 showed the best overall result for AP75 and on mAP, and AP50 without DCNv2. Mask R-CNN was the smallest model with a total of 64.9 million parameters. Overall, SOLOv2 and Mask R-CNN were not different in parameter size, but the result of SOLOv2 and HTC was different. The analysis result showed that SOLOv2 had the highest mAP among all tested models. In other words, it was the most accurate and precise segmentation method.

TABLE I. Mask AP (%) on Test Dataset

Model	**Backbone**	**Average Precision**			**#Parameters**	**FLOP**
		mAP	***AP50***	***AP75***		
Mask R-CNN	ResNet-50-FPN	0.204	0.496	0.151	43.9M	259G
Mask R-CNN	ResNet-101-FPN	0.210	0.481	0.163	62.9M	336G
Mask R-CNN	ResNet-101-FPN-DCNv2	0.217	0.507	0.171	64.9M	273G
HTC	ResNet-50-FPN	0.263	0.564	0.214	77.1M	1,700G
HTC	ResNet-101-FPN	0.267	0.579	0.208	96.1M	1,777G
HTC	ResNet-101-FPN-DCNv2	0.271	0.581	0.206	98.0M	1,714G
SOLOv2	ResNet-50-FPN	0.266	0.602	0.214	46.2M	245G
SOLOv2	ResNet-101-FPN	0.287	**0.609**	0.255	65.2M	321G
SOLOv2	ResNet-101-FPN-DCNv2	**0.290**	0.599	**0.269**	68.4M	**171G**

D. Qualitative Evaluation

The quality of the segmentation result is shown in Fig. 5. The Mask R-CNN model showed the worst performance in terms of AP. The segmentation masks generated by the Mask R-CNN model had significant deficiencies, especially in attempting to mask the roots. The masks were of low quality, had uncertainties, and were not precise. In contrast, the HTC and SOLOv2 models performed well in generating overall masks. Although uncertainties were observed in handling complex root structures, the masks generated by these models were smoother and finer than those of the R-CNN model. These results highlighted the superior mask generation capabilities of the HTC and SOLOv2 models, suggesting that they were effective in accurately delineating

CSRs. Despite the complexity of the overlapping root systems, these models showed satisfactory results.

E. Ablation Experiment

In the experiment with ResNet-101-FPN-DCNv2 on SOLOv2, we compared the aspect of single scale and multiscale at scale sets of 1,333 and 800 for the longer and shorter side, respectively. The longer side of the image was fixed at 1,333 pixels while the other side was randomly selected from the range of 640 and 800 pixels. Suitable models were chosen for the main experiment. The result showed that multiple scales outperformed a single scale in terms of accuracy (Table II).

TABLE II. SINGLE SCALE AND MULTISCALE

Scale	Average Precision		
	mAP	***AP50***	***AP75***
Single	0.269	0.572	0.248
Multi	**0.290**	**0.599**	**0.269**

F. Extension: Pre-trained model with 2017 COCO dataset

We compared the results of the model to those of other models for 36 epochs using the 2017 COCO dataset. The results showed a remarkable improvement in performance with the pre-trained model, with the mAP metric increasing from 0.290 to 0.328. Table III presents the benefits of pre-training in a large dataset to improve the effectiveness of the model. By using the knowledge and features learned from the dataset COCO, the pre-trained model showed a significant increase in performance for the CSR segmentation task. The result of the pre-trained model outperformed that of SOLOv2 without the pre-trained as shown in Fig. 5.

TABLE III. PRE-TRAINED MODEL

Pre-trained	Average Precision		
	mAP	***AP50***	***AP75***
No	0.290	0.599	0.269
Yes	**0.328**	**0.638**	**0.333**

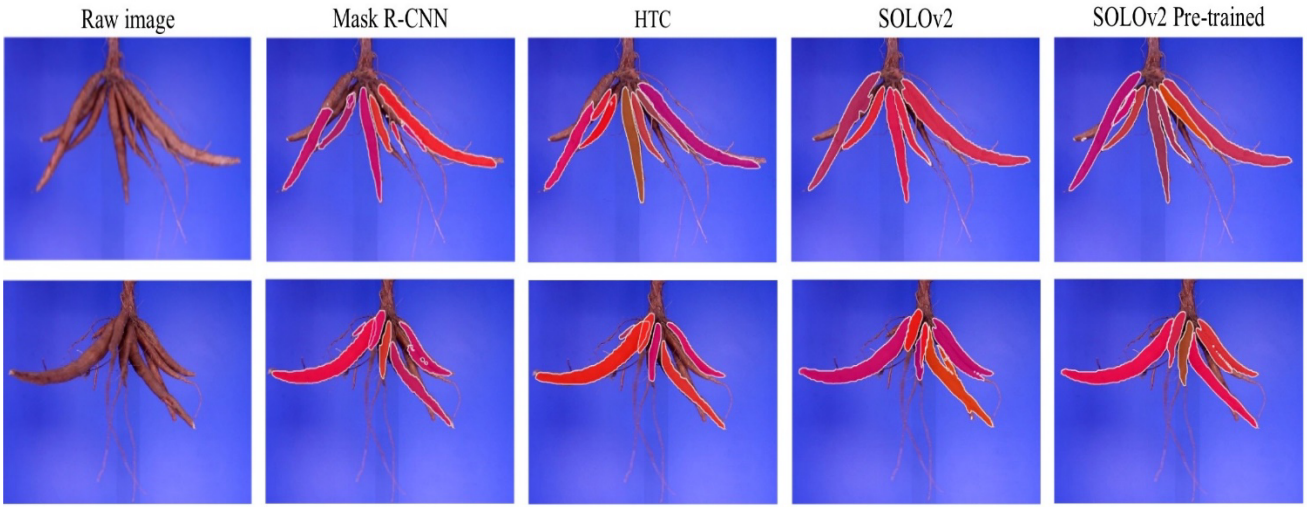


Fig. 5. Instance segmentation mask results

V. CONCLUSION

We compared Mask R-CNN, HTC, and SOLOv2 for instance segmentation in the COCO format. Multiscale training and the incorporation of pre-trained weights from the 2017 COCO dataset improved the performance. SOLOv2 showed the highest accuracy. HTC and SOLOv2 exhibited superior mask generation capabilities compared to Mask R-CNN. Pre-training and multiscale training have proven to be effective for accurate instance segmentation. All the tested models showed satisfactory accuracy in predicting the target range. Particularly, SOLOv2 outperformed other models with the highest overall mask quality while maintaining a smaller size. The complexity and size of the models did not significantly affect their ability to predict CSR segmentation. However, the overlap of CSRs within the root system hindered the accurate assignment of segmentation masks. The study result highlighted the potential of the instance segmentation of deep learning models for accurately segmenting CSRs. The result of this study can be a reference for future research and contribute to advancements in cassava breeding.

REFERENCES

[1] T. Galkovskyi et al., "GiA Roots: software for the high throughput analysis of plant root system architecture," BMC Plant Biology, vol. 12, no. 1, Jul. 2012.

[2] A. Seethepalli et al., "RhizoVision Crown: An Integrated Hardware and Software Platform for Root Crown Phenotyping," Plant Phenomics, vol.2020, pp.1–15, Feb. 2020.

[3] S. Aich and I. Stavness, "Leaf Counting with Deep Convolutional and Deconvolutional Networks," arXiv:1708.07570 [cs], Aug. 2017, Available: https://arxiv.org/abs/1708.07570

[4] J. Atanbori, M. E. Montoya-P, M. G. Selvaraj, A. P. French, and T. P. Pridmore, "Convolutional Neural Net-Based Cassava Storage Root Counting Using Real and Synthetic Images," Frontiers in Plant Science, vol. 10, Nov. 2019.

[5] K. Yang et al., "Convolutional Neural Network for Object Detection in Garlic Root Cutting Equipment," vol. 11, no. 15, pp. 2197–2197, Jul. 2022.

[6] A. Polthanee et al., "Root Yield and Nutrient Removal of Four Cassava Cultivars Planted in Early Rainy Season of Northeastern Thailand: Crop Experienced to Drought at Mid-Growth Stage," Asian Journal of Crop Science, vol.8, pp.24-30, 2016.

[7] M. O. Adu et al., "Characterising shoot and root system trait variability and contribution to genotypic variability in juvenile

cassava (Manihot esculenta Crantz) plants," Heliyon, vol. 4, no. 6, p. e00665, Jun. 2018.

[8] J. R. Ubbens and I. Stavness, "Deep Plant Phenomics: A Deep Learning Platform for Complex Plant Phenotyping Tasks," Frontiers in Plant Science, vol. 8, Jul. 2017

[9] S. Aich et al., "DeepWheat: Estimating Phenotypic Traits from Crop Images with Deep Learning," IEEE Workshop on Applications of Computer Vision (WACV), Mar. 2018.

[10] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep Learning for Image-Based Cassava Disease Detection," Frontiers in Plant Science, vol. 8, Oct. 2017.

[11] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," arXiv.org, 2014. https://arxiv.org/abs/1405.0312

[12] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit," Electronics, vol. 10, no. 3, p. 279, Jan. 2021.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," arXiv.org, 2017. https://arxiv.org/abs/1703.06870

[14] M. A. Rahman and Y. Wang, "Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation," Advances in Visual Computing, pp. 234–244, 2016.

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. 303–338, Sep. 2009.

[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," arXiv:1612.03144 [cs], Apr. 2017, Available: https://arxiv.org/abs/1612.03144

[17] J. Dai et al., "Deformable Convolutional Networks," arXiv:1703.06211 [cs], Jun. 2017, Available: https://arxiv.org/abs/1703.06211

[18] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More Deformable, Better Results," arXiv:1811.11168 [cs], Nov. 2018, Available: https://arxiv.org/abs/1811.11168

[19] B. Dwyer, J. Nelson, J. Solawetz, et al., "Roboflow (Version 1.0) [Software]," [Online]. Available: https://roboflow.com, [2022].

[20] K. Chen et al., "Hybrid Task Cascade for Instance Segmentation," arXiv:1901.07518 [cs], Apr. 2019, Available: https://arxiv.org/abs/1901.07518

[21] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "SOLOv2: Dynamic and Fast Instance Segmentation," arXiv.org, Oct. 23, 2020. https://arxiv.org/abs/2003.10152

[22] K. Chen et al., "MMDetection: Open MMLab Detection Toolbox and Benchmark," arXiv:1906.07155 [cs, eess], Jun. 2019, Available: https://arxiv.org/abs/1906.07155