# A Deep Learning Approach for on-site Plant Leaf Detection

Huu Quan Cap, Katsumasa Suwa, Erika Fujita
Applied Informatics, Graduate School of Science and Engineering, Hosei University, Tokyo, Japan

Satoshi Kagiwada
Clinical Plant Science, Faculty of Bioscience and Applied Chemistry, Hosei University, Tokyo, Japan
kagiwada@hosei.ac.jp

Hiroyuki Uga
Saitama Agricultural Technology Research Center, Saitama, Japan
uga.hiroyuki@pref.saitama.lg.jp

Hitoshi Iyatomi
Applied Informatics, Graduate School of Science and Engineering, Hosei University, Tokyo, Japan
iyatomi@hosei.ac.jp

***Abstract*—Plant diseases are the major problem in the worldwide agriculture sector. Therefore, the early detection is essential for reducing economic losses and mitigating the seriousness of the global food problem. Some fast and accurate computer-based methods have been applied to detect plant diseases. However, as far as our best knowledge, all those methodologies only accept a narrow range image, typically one or limited number of target(s) are in the image frame as their input. Thus, they are time-consuming and difficult to be applied for on-site wide range images (e.g. images or videos from stationary surveillance camera). In this paper, we propose leaf localization method from on-site wide-angle images with a deep learning approach. Our method achieves a detection performance of 78.0% in F1-measure at 2.0 fps.**

***Index Terms*—deep learning; plant disease; plant diagnosis; convolutional neural networks; object detection.**

## I. INTRODUCTION

Plants have been faced with many dangerous diseases which cause a serious reduction in quality and quantity of agriculture products. Therefore, detecting and preventing plant diseases promptly is essential to resolve this issue. In general, plant diagnosis is performed with visual inspection by experts and biological examination is second choice if needed. They are usually expensive and time-consuming. Several computer-based methodologies have been applied to detect plant diseases based on their leaf images [1-6]. Mohanty et al. [1] analyzed 14 kinds of plants from PlantVillage dataset [7] with convolutional neural networks (CNN) and attained over 99% of classification accuracy on images in the research environment (i.e. each leaf is manually cropped and put it on uniform background). Wang et al. [2] applied transfer learning technique on the same PlantVillage dataset and shows an accuracy of 90.4%. Fujita et al. [3] used their own in-field cucumber leaf dataset (seven types of diseases and healthy) and analyzed them with CNN. They showed an average of 82.3% accuracy under various background and photographic conditions. The authors in [4] used a CNN-based system to identify 13 types of diseases in five crops using images downloaded from the Internet and got the overall 96.3% of accuracy.

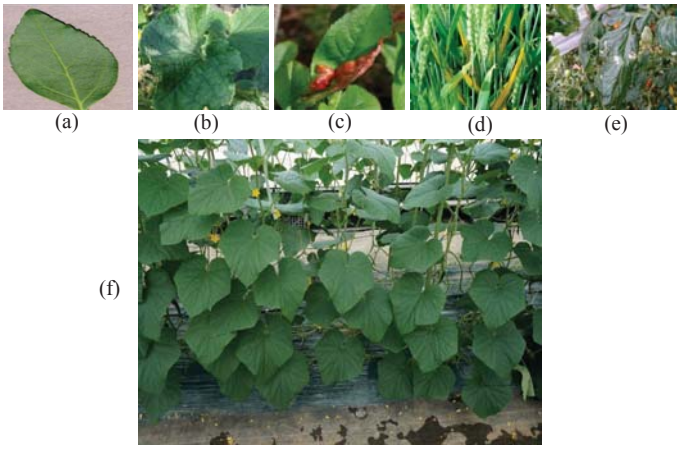


Fig. 1. The comparison between the proposed system input images (a, b, c, d, e) and wide-angle images (f).

Some methods investigate not only detecting the diseases of plant, but also localizing their involved areas. In machine learning and image processing field, object detection and localization recently have much attracted attention and many promising methods have proposed [8-11]. Most of the state-of-the-art methodologies are designed jointly worked with or implemented on CNNs and demonstrated brilliant performance. As application research on plant diagnosis, Fuentes et al. [5] used these CNN-based systems (e.g. pre-trained network combined with Faster R-CNN [8]), which performed object localization and diagnosis processes simultaneously. They used their own annotated tomato leaf images and their system attained 86.0% of mean average precision. Lu et al. [6] applied fully

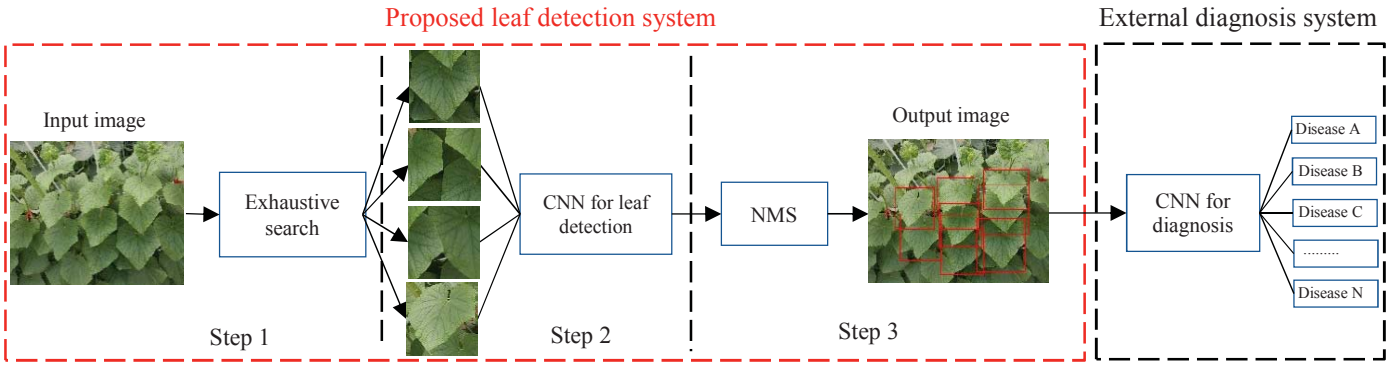


Fig. 2. The schematics of the practical leaf diagnosis system included our proposed leaf detection system (surrounded by red dotted line), and the external diagnosis system (surrounded by black dotted line).

convolutional networks for wheat disease database 2017 (WDD2017) and their system achieved 98.0% of mean recognition accuracy.

These studies have achieved excellent performance. However, they are only effective for private or small-scale facilities, while we believe they still have a room need to be addressed for practical situations. Fig. 1 shows examples of typical images used in literatures; clear background (a) in [1, 2], and in-field images (b)-(e) in [3-6], respectively. Recent sophisticated systems accept wide variety of background, but all of them are narrow range image, i.e. the regions of interests (ROIs) are located in the center of the input. Therefore, diagnosing on-site images (e.g. the images are taken with real condition by surveillance cameras, (see Fig. 1(f)) requires steady work and thus is time consuming. When assuming a development of practical plant disease detection system, solving this issue is essential.

Further, diseases caused by molds or other visible symptoms shown in literatures [5, 6] tend to have clear boundary and thus are relatively easy to identify. Abovementioned simultaneous localization and identification works well for these tasks. But as we experienced, on the other hand, plant symptoms are highly diverse, especially when they are infected with virus which are critical and urgent treatment (removal) is necessary. Symptoms of viral disease appear all over the plant surface but they often hardly discernible. Similar discussion also applies to the detection of initial symptoms. Therefore, simultaneity process has difficulty to do so especially target image is wide-angle in practical situation. To the best of our knowledge, there is no literature providing systematic solutions on this. In such problem setting, the ROIs detection should be developed and performed independently as the front of the classification part followed by.

We also experienced that the detection of ROIs of plant (e.g. leaf detection) from wide-angled image is more difficult than commonly seen object detection tasks such as face detection, pedestrian detection, etc. This is because, in leaf detection task, the object to be detected and its background is the same as its heart and, in addition that they have often heavily overlapped each other.

In such backgrounds, we propose an easy and practical method to localize whole cucumber leaves in wide-angle images based on CNN with sliding windows. Our strategy focuses only on the detection of ROIs (i.e. fully leaf) to be diagnosed with following diagnosis stage separately developed. In practical, the diagnosis part not frequently but need to be updated as occasion arises, and this is also part of the basis of our motivation.

## II. Proposed Leaf Detection System

The objective of this system is to localize the "fully leaf" part from the input image. For clarity, the definition of "fully leaf", "not fully leaf", and "none leaf" is as follows: "Fully leaf" indicates the region contains almost whole leaf (qualitatively more than 80%). "Not fully leaf" and "none leaf" are the regions contains part of a leaf and none leaf object (i.e. background), respectively.

Fig. 2 shows the whole schematics of the practical plant diagnosis system included our proposed leaf detection system (surrounded by red dotted line). It is designed to be combined with the external diagnosis system behind (surrounded by black dotted line). Leaf detection consists of 3 steps. Firstly, given a wide-angle image, our system extracts a numerous candidate boxes that may contain fully leaf regions. Secondly, specially trained CNN classifier analyzes those boxes to find location of fully leaf (i.e. select boxes identified as fully leaf). Finally, the non-maximum suppression (NMS) is used to remove the overlapping bounding boxes. The detected fully leaf regions (red boxes) will be diagnosed by the following external diagnosis systems.

### *A. Exhaustive search for leaf candidates*

To detect fully leaf regions to be diagnosed from wide-angle image, the exhaustive search will be performed on it with multiple window sizes. Based on our preliminary experiments, we found that it does not need to perform this process on real resolution input image (2976×2232 in this experiment, detailed in later). Thus, the reduced resolution domain (200×150) is used to be applied with different window sizes. Concretely, eight types of searching window size $S \times S$ ($S$=20, 25, 30, 35, 40, 45, 50, 55) were used to search on the resized image with stride size 20% of their edge. They correspond to between the range of roughly 1/10 and 1/4 of major axis of the image and their size in original resolution satisfies the required input size of widely

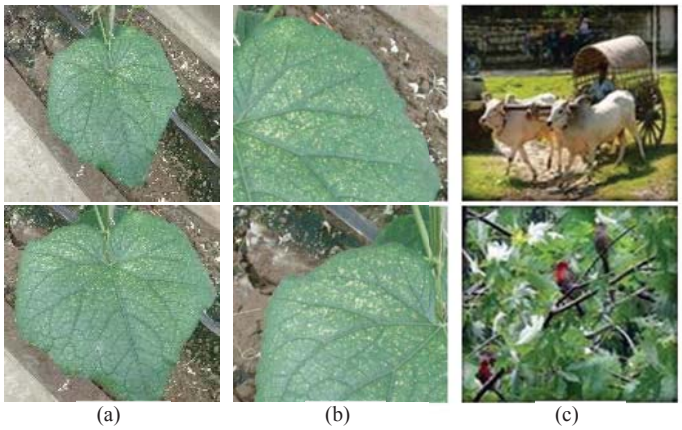


Fig. 3. The dataset for training the CNN model. From left to right, (a) original and cropped images of "fully leaf", (b) "not fully leaf", and (c) "none leaf" from ImageNet dataset.

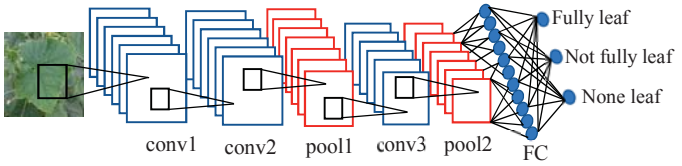


Fig. 4. The proposed CNN architecture consists of 7 layers (6 hidden layers and 1 output layer).

available pre-trained CNN models (e.g. Alex-net [15], VGG [16], Res-net [17], etc.).

Since the reduced domain is small, exhaustive search combining with our light weight CNN model makes the processing time fast enough; nearly real-time on GPU. Note that we've already investigated the optimization for sliding windows to eliminate not small redundant convolution process (i.e. calculate the convolution on the whole input image). It significantly reduces the amount of total calculation but gain of processing time is trivial in our task due to slow memory translation on GPU. We therefore choose simple strategy here. The detail of our CNN model and processing time will be in the following sections.

### *B. The proposed CNN for leaf detection*

Our CNN for leaf detection in Fig. 2 is three classes classifier, namely it discriminates input image patch cropped by exhaustive search as either of "fully leaf", "not fully leaf" or "none leaf". The objective of this study is to determine the boundary box of "fully leaf", which is acceptable for the following diagnosis step. Note that both locations of "not fully leaf" and "none leaf" should not be detected. Since the appearance and image property of them are completely different, we left two different classes even if they should be rejected.

#### *1) Dataset for training the CNN*

A total of 1.44 million image patches, consisting of 480,000 for each class were created for training the CNN. For "fully leaf" class, the dataset of 60,000 images was given by Saitama Agricultural Technology Research Center, Japan. Each image is square image that contains a single cucumber leaf roughly in the center and surrounded with various backgrounds (see Fig. 3(a)). This dataset was augmented by cropping center and clockwise rotation. For detail, firstly, we made a copied version of the original dataset, then crop its image center. This cropping process removes a part of background which lies in the border of image in order to highlight the ROIs (cucumber leaf). Secondly, for the cropped dataset, each image now contains its center leaf with the width and the height now reduced to 75% and 87.5% compared to the original image, respectively. Finally, each of those original and cropped images are rotated clockwise with the incremental step size of 90 degrees. The total number of "fully leaf" images now becomes eight times larger with 480,000 images (60,000×2×4).

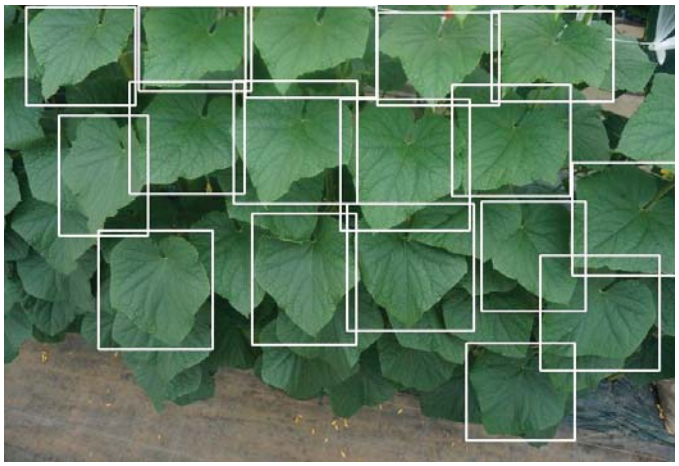

Fig. 5. The example of ground-truth image in "wide-angle" images dataset.

TABLE I. THE PROPOSED CNN ARCHITECTURE

| **Layer name and output size [width×height×depth]** | **Detail (filter size, num. of filters, padding, stride)** |
|---|---|
| input – [16×16×3] | N/A |
| conv1 – [16×16×32] | [3×3], 32, 1, 1 |
| conv2 – [16×16×64] | [3×3], 64, 1, 1 |
| pool1 – [8×8×64] | [2×2], 64, 0, 2 |
| conv3 – [8×8×64] | [3×3], 64, 1, 1 |
| pool2 – [4×4×64] | [2×2], 64, 0, 2 |
| FC [3×1] | [1×1], 30, 0, 1 |

For "not fully leaf" class, each of abovementioned "fully leaf" images (original and cropped one) was divided into quarters, i.e. total 120,000×4=480,000 images. Here, the same leaf is included both in "fully leaf" and "not fully leaf" datasets. This strategy is expected to help boosting the discrimination performance between those classes rather than using completely different dataset since under our situation, those leaves have overlapped each other. For "none leaf" class, 480,000 images were collected randomly from ImageNet dataset [12].

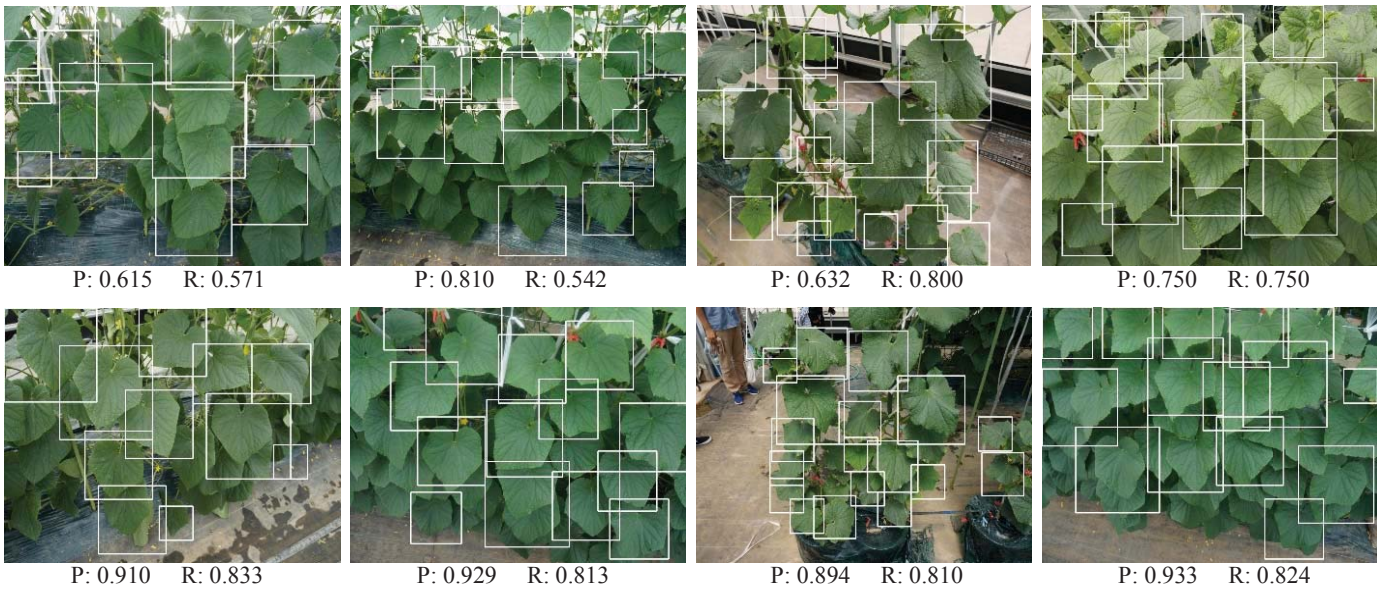


Fig. 6. The detection result of leaf locations. Precision (P) and recall (R) are shown on each result.

This dataset is used for training and testing the CNN. The training examples for "fully leaf", "not fully leaf", and "none leaf" are shown in Fig. 3 (a), (b), and (c), respectively.

*2) The proposed CNN architecture*

In our preliminary experiments, the size of CNN input image patch was firstly explored. The CNN was trained with some input sizes namely: 56×56, 32×32, 28×28, and 16×16; where we found that the CNN with 16×16 input was fast, and accurate enough for our system.

Architecture of the proposed CNN is shown in Fig. 4 and Table I. It accepts a color image with 16×16 pixels as the input and consists of six hidden layers and one output layer with three units. The hidden layers have three convolution layers as *conv1*, *conv2*, and *conv3*, two Max-pooling layers (*pool1*, *pool2*), and fully connected (*FC*) layer which has 30 units. The dropout technique [13] with the ratio of 50% was used to weights between *pool2* and *FC*. Note that batch normalization [14] is applied after each convolutional layer and the rectified linear unit (ReLU) is used as the activation function.

## III. Experiments and Result

### A. Training the CNN

Totally, 1.44 million images were divided into 60% of training set (864,000 images) and 40% of testing set (576,000 images). Each dataset has the same amount from three classes. The batch normalization with the batch size of 300 was used to train the classifier within 20 epochs. Training of our CNN model is fast and required only 10 minutes on our environment with Core i7-3770K CPU, 16GB RAM, and GTX 1080Ti GPU. The accuracy of our CNN on test dataset was 96.1%, whereas 93.1% for training dataset. This result confirmed low training gap between them.

### B. Testing the whole system

For testing the whole system, 100 wide-angle on-site images were collected, each of which has 2976×2232 pixels in the size and contains multiple leaves. They were taken by Sony DSC-RX100 camera in daytime with various conditions. We refer this as "wide-angle" images dataset for clarity. To evaluate our system, total of 2,571 fully leaves on these images were carefully annotated with bounding boxes and were used as ground-truth. Fig. 5 shows an example of one ground-truth image from "wide-angle" images dataset.

For each image, our exhaustive search extracts 4,283 boxes consisting of various window sizes. Then, our CNN is used to classify all extracted boxes. After that, NMS is applied to remove the overlapped boxes. Concretely, given all scored boxes after classification, the NMS will reject a box if it has lower score than a selected box and the intersection-over-union (IoU) between them is larger than 0.2.

F1-score criteria is used to measure the performance of our system. Given the result bounding boxes and ground-truth, the F1-score is calculated by the following equations:

$$\mathrm{F1} = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}},$$

where:

$$\mathrm{precision} = \frac{\text{Number of correctly detected boxes}}{\text{Number of detected boxes}}$$

$$\mathrm{recall} = \frac{\text{Number of correctly detected boxes}}{\text{Number of ground-truth boxes}}$$

A detected box is considered as a correctly detected where the IoU of that box and corresponding ground-truth box is equal or larger than 0.5 - commonly used in the evaluation of object detection studies. Fig. 6 shows our result on several "wide-angle" images. We achieved the average 80.8% of precision, 75.3% of recall, and 78.0% of F1-score. In additional, our

system takes around 0.5 seconds to do all the detection process per image.

## IV. Discussion and Conclusion

This paper presents a simple and accurate leaf regions detection system with high affinity with other existing disease diagnosis systems. We confirmed that the performance of 78.0% in F1-score is sufficiently acceptable for this task from visual assessment.

Precision and recall are trade-off criteria. Considering the practical application of whole plant diagnosis schema in Fig. 2, it is not necessary to detect exactly the whole fully leaf from the images. In the fact that we need to detect some of, or at least one infected leaf per disease in the image. This is because nearby leaves might have the same disease and the prime objective of our system is to help early detection of disease, and this helps one can take detailed examination. From this point, we can tolerate some false negative. Conversely, we should not pass completely wrong area to the classifier followed by especially when the following classifier is not so robust. That is, we need a certain level of precision. Therefore, appropriate control of balance between false positive and false negative is required. Considering these facts, we think the current balance (precision=80.8%, recall=75.3%) is considered reasonable.

In addition, using small input CNN model (i.e. 16×16) with reduced searching domain is fast (2.0 fps) and therefore, the processing time does not affect much on the processing performance of the diagnosis systems behind.

We achieved a promising detection performance on practical on-site images. On the other hand, however, almost all of leaves in wide-angle on-site images used in this study are healthy ones. In near future, we will investigate and evaluate our methodologies in other practical environments with many infected leaves and build an end-to-end practical plant diagnosis system.

## Acknowledgment

This research was partially supported by the Ministry of Education, Culture, Science and Technology of Japan (Grant in Aid for Fundamental research program (C), 17K8033, 2017-2020).

## References

[1] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, 2016.

[2] G. Wang, Y. Sun, and J. Wang, "Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning," *Computational Intelligence and Neuroscience*, vol. 2017, pp. 1-8, 2017.

[3] E. Fujita, Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, "Basic Investigation on a Robust and Practical Plant Diagnostic System," *IEEE Proc. on Machine Learning and Applications*, pp. 989-992, 2016.

[4] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," *Computational Intelligence and Neuroscience*, vol. 2016, pp. 1-11, 2016.

[5] A. Fuentes, S. Yoon, S. Kim, and D. Park, "A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition," *Sensors*, vol. 17, no. 9, 2017.

[6] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, "An in-field automatic wheat disease diagnosis system," *Computers and Electronics in Agriculture*, vol. 142, pp. 369-379, 2017.

[7] D. P Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *CoRR*, abs/1511.08060, 2015.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," *ECCV 2016 Lecture Notes in Computer Science*, pp. 21-37, 2016.

[10] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," *Advances in Neural Information Processing Systems*, vol. 29, pp. 379-387, 2016.

[11] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *CoRR*, abs/1703.06211, 2017.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," *IEEE Proc. on Computer Vision and Pattern Recognition*, pp. 248-255, 2009.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proc. of the 32nd International Conference on Machine Learning*, vol. 37, pp. 448-456, 2015.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097-1105, 2012.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, abs/1409.1556, 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Proc. on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.