ICRA-2018 Workshop on Robotic Vision and Action in Agriculture,
Brisbane, Australia, May 2018

Deep Neural Network-based Recognition of Green Perilla Leaves for Robotic Harvest Support in Greenhouse Horticulture

Hiroaki Masuzawa and Jun Miura
Department of Computer Science and Engineering
Toyohashi University of Technology

***Abstract*— This paper describes an application of deep neural network to the recognition of green perilla leaves for harvest support in greenhouse horticulture. We are developing a robot which automates the selection and the bundling process. In order to manipulate the leaves exactly, the robot needs to recognize the leaves' parameters such as width, height, and orientation. Developing ordinary image processing algorithms is sometimes tedious due to a lot of parameters to tune and a variety of demands of farmers. We thus adopt deep neural network (DNN) techniques to this problem. We first developed an image processing algorithm for segmenting and annotating leaf images, followed by small manual corrections, to make an annotated dataset. We then supply the dataset to a DNN similar to U-Net to get recognition results. We also examine the processing time versus recognition accuracy trade-off by changing the number of convolutional layers.**

***Index Terms*— Image-based leaf recognition, deep neural network, greenhouse horticulture.**

I. Introduction

Labor shortage has been one of the most serious problems in Japanese agriculture as the portion of the elderly workers in agriculture is significantly increasing. One possible solution is to apply robotic technologies (RT) to automating and/or supporting agricultural works. We have been developing a harvest support robot for green perilla leaves[1] [1]. Fig. 1 shows the conceptual figure and the prototype robot. The target processing speed of the robot is about three seconds per leaf.

The robot does not automate the whole harvesting process from reaping to packing but does in the selection and the bundling process, which are most costly and time-consuming parts. The key technologies in this automation is soft object handling and visual recognition. This paper focuses on the visual inspection part and describes the development of a deep neural network-based recognition system. We also describe how to generate a dataset of training using image processing algorithms. The contribution of the paper lies in developing and testing a leaf segmentation method with front/back recognition method in a commercial-level harvest support robot.

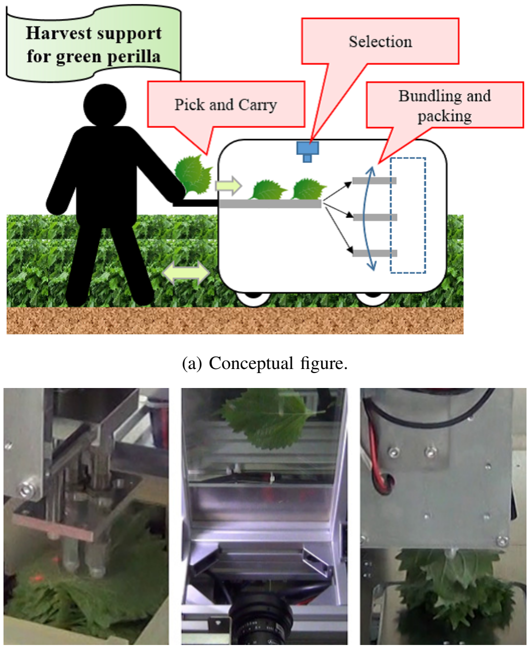
The rest of the paper is organized as follows. Sec. II describes related work. Sec. III defines the recognition task and dataset generation. Sec. IV explains the structure of our neural networks. Sec. V shows experimental results. Sec. VI concludes the paper.

II. Related work

A. Visual recognition of agricultural products

Visual recognition of agricultural products has been one of the important topics in agricultural robotics, especially for automating harvesting processes. In fruits detection, for example, various visual cues such as color, spectral reflectance, thermal response, texture, and shape have been adopted [2]. Fruits usually have distinctive appearance; however, detecting them in unstructured environments could sometimes be challenging. Leaf classification is also an active research topic. Hall et al. [3] evaluated various image features including DNN features. Wu et al. [4] developed a large leaf image dataset for leaf classification.

Detecting leaves is also difficult because all leaves look alike and often overlapped with each other. Xia et al. [5]



(a) Conceptual figure.

(b) Actual operations of the prototype robot (from left to right): picking up leaves, image-based recognition, piling up sorted leaves.

Fig. 1. Harvest support robot for green perilla leaves.

[1] Green perilla leaf is called "ooba" in Japanese. Its typical use in Japanese cuisine is to be an accompaniment to Sashimi (raw fish).

developed a method of leaf extraction by fitting an active contour model to leaf candidate regions. They deal with leaves with relatively simple contours. Kumar et al. [6] developed a software to identify plant species from a single leaf image using computer vision and machine learning techniques. It exhibits a nice performance but the leaf region extraction assumes a non-textured background.

B. *Deep neural networks for object recognition*

Deep neural networks (DNNs) are shown to exhibits very high performances in various recognition tasks [7], [8]. Sa et al. [9] applied DNN with RGB and NIR images to fruit detection to exhibit a high performance; the task of the network is to estimate the bounding boxes of crops.

DNNs for pixel-wise object classification (or semantic segmentation) have also been proposed [10], [11]. These show expressive performances once a reliable dataset with an enough amount of data is available.

III. Recognition Tasks and Dataset

A. *Recognition Tasks*

The robot takes a pile of leaves in a box, sorts them by size and quality, aligns a sorted set for bundling, and outputs the bundled sets. In these processes, the tasks of the recognition system are as follows:

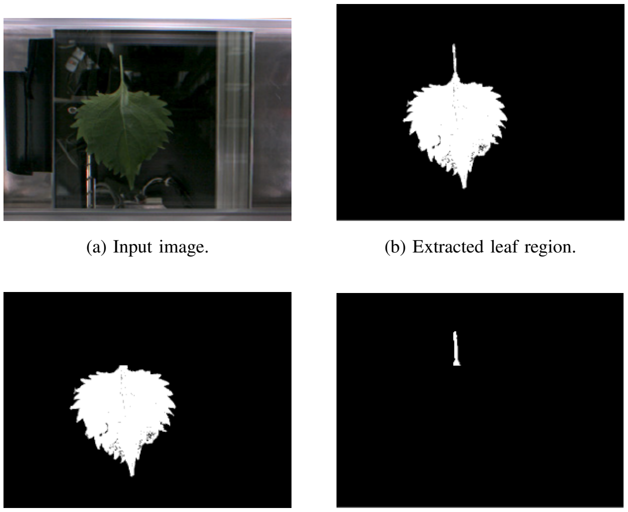
- Blade and petiole region extraction: classify pixels into blade, petiole, and background regions in order to calculate the size and the orientation of a leaf for sorting and alignment.
- Front/back recognition: remove leaves showing their backside for packing a consistent set of leaves.
- Defect detection: remove leaves with defects.

In this paper, we focus on the first and the second task.

B. *Generation of a dataset*

A certain amount and quality of data is necessary for an effective deep learning. In the segmentation tasks (the first one), a precise annotation of pixels is crucial in generating a dataset. For reliable and easy annotation, we developed an image processing algorithm for annotation proposals. The errors in the results are manually corrected; this error correction is basically a re-labeling of misclassified pixels. Corrected data are then augmented to cover possible defects. The detailed process of dataset generation is explained below.

1) Leaf/Background separation: We first construct a leaf appearance model using one leaf. We also construct a model for background in the robotic apparatus. The model for the leaves is a simple 3D Gaussian in the RGB color space, represented by mean vector and covariance matrix. The model for the background is composed of a set of Gaussians to cope with a variety of background appearances. They are calculated for respective clusters obtained by the k-means clustering method. The number of clusters is empirically set to ten.


(a) Input image.

(b) Extracted leaf region.

(c) Blade region.

(d) Petiole region.

Fig. 3. Example of algorithmic leaf recognition.

Let $\boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$ be the mean vector and the covariance matrix of the leaf regions and $\boldsymbol{\mu}_{b_i}$ and $\boldsymbol{\Sigma}_{b_i}$ be those of the ith cluster of the background. Then, the distance $d_l(i,j)$ in the color space of a pixel at (i,j) to the leaf region and the distance $d_b(i,j)$ to the background region are defined as the Mahalanobis distances as follows:

$$
\begin{aligned}
d_l(i,j) &= \left(\boldsymbol{c}(i,j)-\boldsymbol{\mu}_l\right)^T \boldsymbol{\Sigma}_l{}^{-1}\left(\boldsymbol{c}(i,j)-\boldsymbol{\mu}_l\right), && (1)\\
d_b(i,j) &= \min_i d_{b_i}(i,j), && (2)\\
d_{b_i}(i,j) &= \left(\boldsymbol{c}(i,j)-\boldsymbol{\mu}_{b_i}\right)^T \boldsymbol{\Sigma}_{b_i}{}^{-1}\left(\boldsymbol{c}(i,j)-\boldsymbol{\mu}_{b_i}\right), && (3)
\end{aligned}
$$

where $\boldsymbol{c}(i,j)$ indicates the RGB value of the pixel at (i,j).

We then use the following expression to determine the label (leaf or background):

$$
L(i,j) = \begin{cases} \text{leaf} & d_l(i,j) < d_b(i,j) \\ \text{background} & \text{otherwise} \end{cases}. \quad (4)
$$

We finally apply a labeling algorithm to the extracted connected leaf-labeled regions and select the largest such region as the true leaf region.

2) Petiole detection: The petiole part of a leaf region is detected using a geometric property of leaves that a blade has a circular shape while a petiole has a linear shape, extending radially from the center. Fig. 2 shows the flow of petiole detection. We first extract leaf pixels in a set of concentric mask regions and extract leaf pixels whose size is less than a threshold in each region. The extracted pixels are then merged to form petiole region candidates and the largest one is determined as a petiole region.

Fig. 3 shows an example process of blade and petiole detection using the designed algorithm.

3) Data augmentation: We now have a set of correctly annotated images of the leaves. Data augmentation such as flipping and random cropping is usually used for increasing the number and the variety of data. We do not such operations because the lighting condition is fairly stable for a camera inside the robot. Instead, we augmented the data in such a way that possible defects are added to original leaves. We consider cases of missing petioles and adding holes in the blade and additionally makes one image and nine images

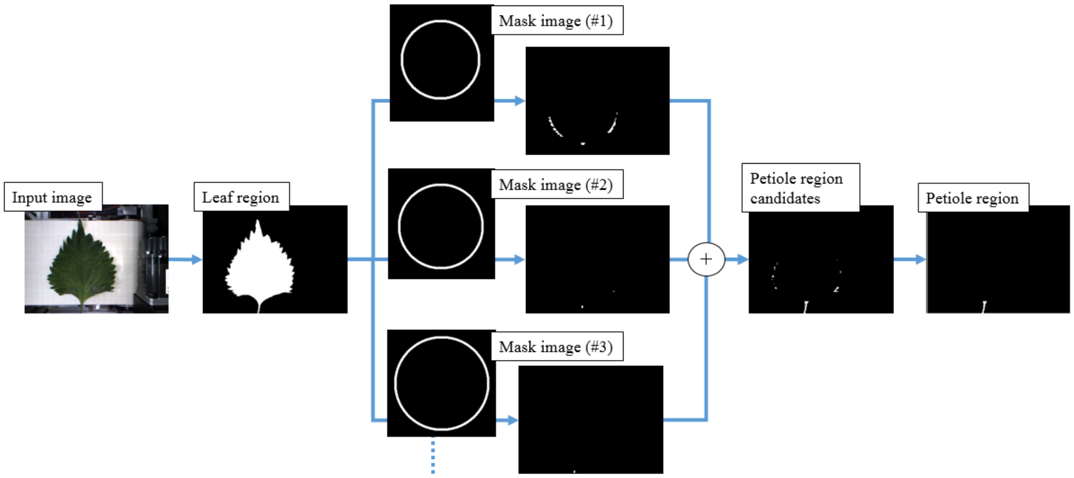
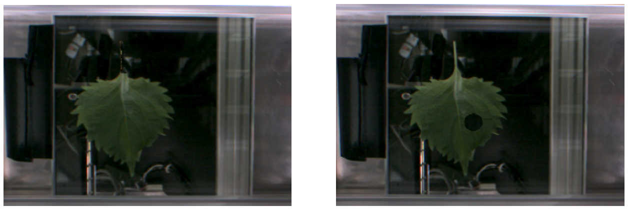


Fig. 2. Process of algorithmic petiole region detection.


(a) Missing petiole.

(b) Added hole.

Fig. 4. Data augmentation examples.

TABLE I

NUMBER OF RESPECTIVE LAYERS IN THE NETWORKS.

Network ID	Convolution	Max pooling	Deconvolution
1	15	3	3
2	19	4	4
3	23	5	5

for the first and the second case, respectively. Fig. 4 shows examples of both cases for the input image shown in Fig. 3(a).

IV. DEEP NEURAL NETWORK-BASED RECOGNITION

A. *Network for segmentation*

We use the U-Net [10] as a network structure. U-Net consists of convolution and deconvolution layers with direction connections between intermediate layers, which is to combine global and local features for a fine segmentation. Fig. 5 shows the network structure used in this research.

Increasing the layers usually improves the segmentation results at the cost of increased computation time. We compares the performance for several numbers of layers, as shown in Table I. The platform and the conditions for training are shown in Tables II and III, respectively.

B. *Adding front/back recognition*

TABLE II

DNN PLATFORM SPECIFICATIONS.

CPU	Intel Core i7-7700K
GPU	NVIDIA GeForce GTX 1080
OS	Ubuntu 16.04
Deep learning framework	TensorFlow 1.4.1

TABLE III

CONDITIONS FOR TRAINING THE NETWORK.

Loss function	softmax cross entropy
Optimizer	Adam
Batch size	25
Number of epochs	100

The network explained above is to segment an image into blade, petiole, and background regions, and the results are sufficient for estimating sizes and orientations for sorting and alignment. For bundling, however, all leaves in a set must be facing the same orientation (e.g., facing upwards). We therefore need to a front/back recognition from images.

This is not a pixel-wise but an image-wise two-class separation problem and can also be solved by neural networks. Supposing that the feature extraction layers of the above network are also effective for this classification, we simply add three consecutive fully-connected (FC) layers. As shown in Fig. 6, the output of the lowest-resolution layer is branched into the FC layers to output one of the two classes (front or back). We train this modified network as a whole. The loss function used is the sum of softmax cross entropy values for segmentation and front/back recognition.

V. EXPERIMENTAL RESULTS

A. *Training and test data*

We got two sets of leaves from two different glossary stores, and use one for training and the other for testing. The training set with 201 leaves, which is also used in dataset generation, is augmented to 2,211 images, while the test set with 206 leaves is augmented to 2,266 images. The time for training is about three hours.

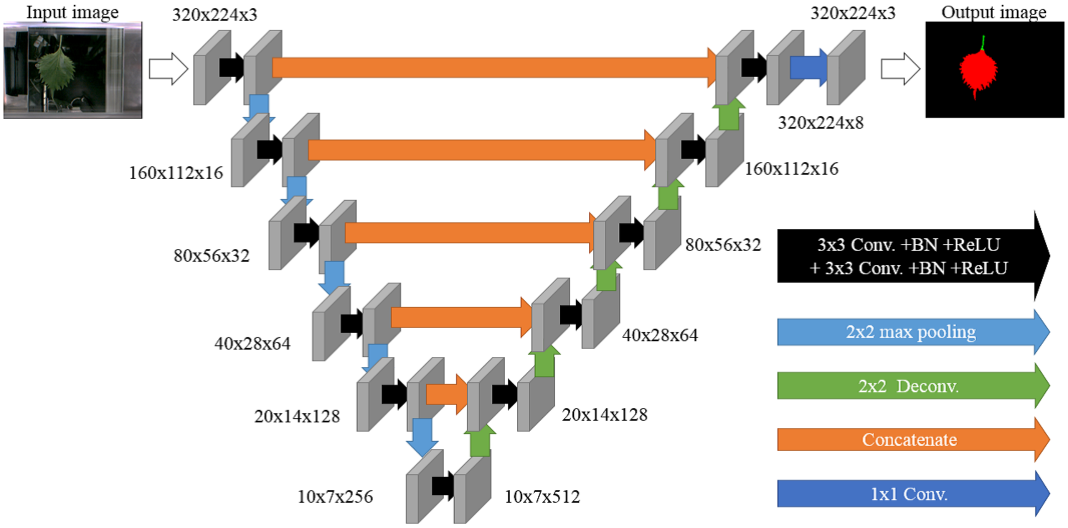


Fig. 5. U-Net network structure. The number of feature extraction layers is five in the figure.

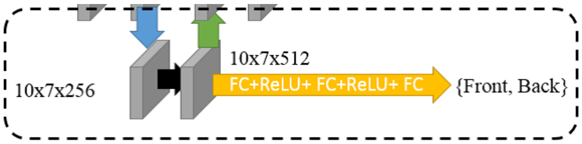
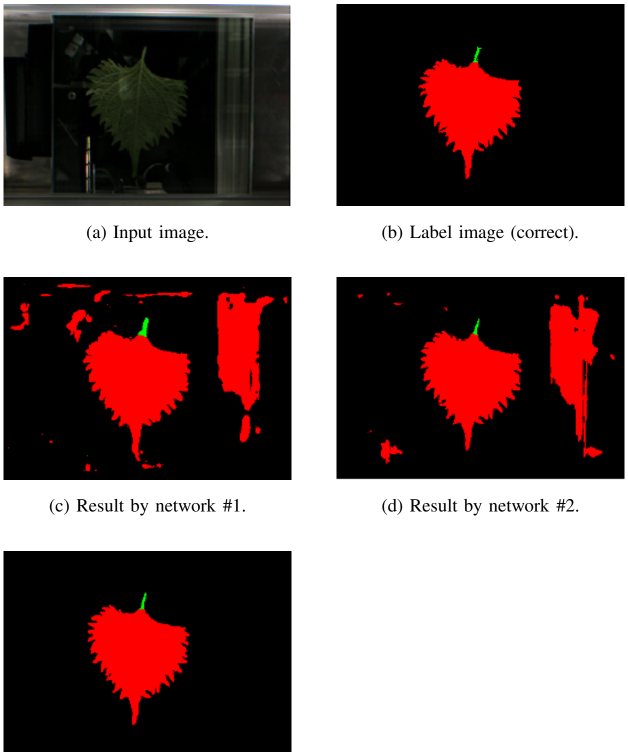


Fig. 6. Addition of front/back recognition sub-network.

TABLE IV

SEGMENTATION ACCURACY

Network ID	Training data	Test data
1	0.992	0.903
2	**0.998**	0.925
3	**0.998**	**0.997**



(a) Input image.

(b) Label image (correct).

(c) Result by network #1.

(d) Result by network #2.

(e) Result by network #3.

Fig. 7. Example segmentation results.

B. Segmentation accuracy

Fig. 7 shows example segmentation results for the three networks. Networks 1 and 2, which have less feature extraction layers, recognize parts of the background as blade region. Table IV summarizes the accuracy for the training and the test data. Tables V and VI summarize the precision and the recall values for each class for the training and the test data, respectively.

All networks exhibits acceptable performances for the training data, while only network 3 is effective for the test data. The most degraded measures for networks 1 and 2 are the precision of the blade and the petiole regions. This is mainly because those networks mis-recognize the background as leaf regions; it is conjectured that this mis-recognition is removed for network 3, by appropriately combining global and local features.

C. Front/back recognition accuracy

Table VII summarizes the accuracy of front/back recognition for the training and the test data. The performance is fairly good with a small extra cost to the segmentation system, as shown below.

D. Computation time

Table VIII(a) summarizes the averaged computation time for the segmentation of one input image. That for the image processing algorithm-based method described in Sec. III-B is 122[ms]. Therefore, the DNN-based method is faster even with only a CPU. Table VIII(b) shows the computation time

TABLE V

PRECISION AND RECALL FOR TRAINING DATA.

Network ID	Background		Blade		Petiole	
	Pre	Rec	Pre	Rec	Pre	Rec
1	**0.999**	0.991	0.921	0.999	0.459	0.999
2	**0.999**	**0.999**	**0.991**	0.989	0.905	**0.899**
3	**0.999**	0.991	**0.991**	**0.992**	**0.913**	0.865

Pre: precision, Rec: recall

TABLE VI

PRECISION AND RECALL FOR TEST DATA.

Network ID	Background		Blade		Petiole	
	Pre	Rec	Pre	Rec	Pre	Rec
1	**0.999**	0.891	0.517	**0.999**	0.480	**0.974**
2	0.998	0.918	0.590	0.989	0.845	0.821
3	0.998	**0.998**	**0.984**	0.988	**0.864**	0.801

Pre: precision, Rec: recall

TABLE VII

FRONT/BACK RECOGNITION ACCURACY

Network ID	Training data	Test data
1	**1.000**	0.995
2	**1.000**	0.999
3	**1.000**	**1.000**

when both the segmentation and the front/back recognition are performed. Although the time increases compared with (a), it is still acceptable for the target robot.

VI. CONCLUSIONS AND FUTURE WORK

This paper has described a method of recognizing green perilla leaves using a deep neural network (DNN) for a harvest support robot. The network is based on U-Net and a dataset is generated by an image processing algorithm followed by manual corrections. We compared the performances for networks with different numbers of feature extraction layers. We also showed that the proposed DNN-based method is faster than the algorithmic approach even without using GPUs.

We are now planning to extend the method to include defect detection. Designing a new network with quality judgement outputs is future work but the current network architecture could relatively easily extended as in the case of adding front/back recognition to the original network for segmentation. We also need to identify types of defects to which the DNN-based methods are effectively applied.

TABLE VIII

COMPARISON OF COMPUTATION TIME.

(a) Network for segmentation

Network ID	Using only CPU [ms]	with GPU [ms]
1	59.58	4.55
2	68.83	4.93
3	77.52	5.85

(b) Network for segmentation and front/back recognition

Network ID	Using only CPU [ms]	with GPU [ms]
1	69.48	4.99
2	77.57	5.15
3	83.29	6.18

Acknowledgment

The authors would like to thank Mr. Mitsuo Tsume and Mr. Kensuke Suzuki (Sinfonia Technology Co. Ltd.) and Mr. Kiyoaki Kakihara (KER Co. Ltd.) for designing and developing the robot. We also thank Dr. Shuji Oishi for fruitful discussion.

This work is supported by Knowledge Hub Aichi's Priority Research Project (2nd term).

REFERENCES

[1] J. Miura. Robotic Support for Regional Agriculture. *TUT Research: e-Newsletter from Toyohashi University of Technology*, No. 11, December 2017.

[2] K. Kapach, E. Barnea, R. Mairon, Y. Edan, and O. Ben-Shahar. Computer Vision for Fruit Harvesting Robots – State of the Art and Challenges Ahead. *Int. J. of Computational Vision and Robotics*, Vol. 3, No. 1/2, pp. 4–34, 2012.

[3] D. Hall, C. McCool, F. Dayoub, N. Sünderhauf, and B. Upcroft. Evaluation of Features for Leaf Classification in Challenging Conditions. In *Proceedings of 2015 IEEE Winter Conf. on Applications of Computer Vision*, pp. 797–804, 2015.

[4] S.G. Wu, F.S. Bao, E.Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang. A Leaf Recognition Algorithm for Plant Classification using Probabilistic Neural Network, 2007.

[5] C. Xia, J.-M. Lee, Y. Li, Y.-H. Song, B.-K. Chung, and T.-S. Chon. Plant Leaf Detection using Modified Active Shape Models. *Biosystem Engineering*, Vol. 116, pp. 23–35, 2013.

[6] N. Kumar, P.N. Belhumeur, A. Biswas, D.W. Jacobs, W.J. Kress, I. Lopez, and J.V.B. Soares. Leafsnap: A Computer Vision System for Automatic Plant Species Identification. In *Proceedings of 12th European Conf. on Computer Vision*, pp. 502–516, 2012.

[7] A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. 2012.

[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[9] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool. DeepFruits: A Fruit Detection System using Deep Neural Networks. *Sensors*, 2016.

[10] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv preprint arXiv:1505.04597*, 2015.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labeling. *arXiv preprint arXiv:1505.07293*, 2015.