Mask R-CNN with data augmentation for food detection and recognition

Than D. Le[1,2]

***Abstract*— In this paper, we focus on simple data-driven approach to solve deep learning based on implementing the Mask R-CNN module by analyzing deeper manipulation of datasets. We firstly approach to affine transformation and projective representation to data augmentation analysis in order to increasing large-scale data manually based on the state-of-the-art in views of computer vision. Then we evaluate our method concretely by connection our datasets by visualization data and completely in testing to many methods to understand intelligent data analysis in object detection and segmentations by using more than 5000 image according to many similar objects. As far as, it illustrated efficiency of small applications such as food recognition, grasp and manipulation in robotics.**

***Index Terms*— Transfer Learning, Data Augmentation, Mask R-CNN, Object Detection, Segmentation.**

I. Introduction

The industry of fruit and food [3] is currently the potential applications in nowadays while people are pay attention on health problems, such as food delivery system, logistics, food health and safety, and agriculture Robotics [7] based on implementing the dexterous grasp and robust manipulation[8], [9]. Since, the food processing is recently concentrated by organizers or non-organizers based on marketing systems. By fast going deep learning based on computer vision given the robust solutions on image processing and segmentation, there are rapidly moving many research activities on food recognition by implementing the object detection and segmentation. All most the research here are unspecified really the food applications.

There are many food datasets existing as open source such as Food-101 Mining Discriminative Components with Random Forests [12], UEC FOOD 256 [13], and Dataset UPMC Food-101 [14]. But it is unpracticed for many engineer and researcher to implement. And there unfortunately is not easy to use it during training for food recognition [11]. The huge datasets will not ideal to apply specific domain. Hence, it is necessary to handle private data science.

Without taken many time for handle datasets, the transfer learning were are standardized for many applications using deep learning, which is essential to supervised to maximize the hierarchical representation to learn the targets.

One of the challenges on data science and machine learning was the convergence which is difficult to understand and interpretation on deep learning [10]. The visualization of neural network and deep learning which are challenge for all most the scientists, whom will be dealing the the convergence and divergence problems.

In this paper we are represented the data augmentation in implementing the deep learning approaches for visualizing the data science.

II. Related Work

In this section, we explore the concepts based on points, lines and conics. On the other side, we will also understand both the transformations and invariants in hybrid approach.

In order to removing projective distortion, we need to select the four points in a plane with known coordinates, and it can be represented by:

$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \tag{1}$$

Similarly, with y' can define

$$y' = \frac{x'_2}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \tag{2}$$

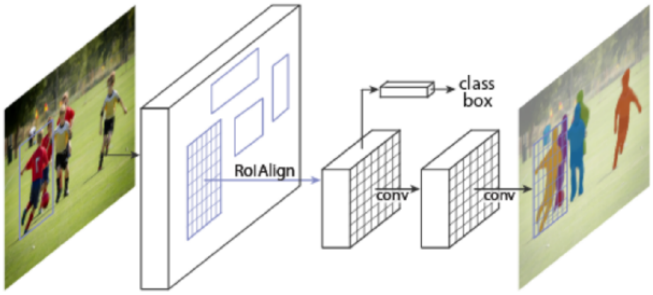

Fig. 1: [21] Mask R-CNN Architecture.

From equations 1 and 2, we can write equals:

$$x'(h_{31}x + h_{32}y + h_{33}) = h_{11}x + h_{12}y + h_{13} \tag{3a}$$
$$y'(h_{31}x + h_{32}y + h_{33}) = h_{11}x + h_{12}y + h_{13} \tag{3b}$$

A. Homogeneous Coordinates

Usually, the defining the datasets it is necessary to understand the principles of points, lines,and plane in geometry due to we support to all polygon for pre-processing of datasets. In this section, we address the fundamental concepts of geometry with 2D dimension. Home

Homogeneous Coordinates representation of lines will be formed by

$$ax + by + c = 0 \tag{4}$$

[1] is with University of Bordeaux, Bordeaux, France. than.ld@ieee.org

[2] is with Faculty of Information Technology, University Ton Duc Thang, Ho Chi Minh City, Vietnam.

or simply we can write a, b, c^T. There is a constrain that the point x must lies on the line, formalizes $x = (x, y)^T$

The homogeneous vectors

The most information thing is we would like to know how the intersection between two lines.

To define the conic we should have equal of power 2.

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{5}$$

According to parameters of this function, there exists five degree of freedom. And it can be simplify to basic formula by homogenizing

$$x^T C x = 0 \tag{6}$$

or convert it to

$$a_1^2 + bx_1x_2 + cx_1^2 \tag{7}$$

where C equals

$$C = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix} \tag{8}$$

To define the a conic, there is must given by five points as constrains to $ax_i{}^2 + bx_iy_i + cy_i{}^2 + dx_i + ey_i + f = 0$ or shortly $(x_i{}^2, x_iy_i, y_i{}^2, x_i, y_i, f)c = 0$, where c can be written by $c = a, b, c, d, e, f)^T$

$$\begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ x_3^2 & x_3y_3 & y_3^2 & x_3 & y_3 & 1 \\ x_4^2 & x_4y_4 & y_4^2 & x_4 & y_4 & 1 \\ x_5^2 & x_5y_5 & y_5^2 & x_5 & y_5 & 1 \end{bmatrix} c = 0 \tag{9}$$

III. The Hierarchy of Projective Geometry and Transformations of 2D

The hierarchical transformations is as the projective linear group with modifying the affine group. For instance, the last row of matrix will be followed to $[0, 0, 1]$

The isometries, the matrices are defined by following:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \varepsilon cos\phi & -sin\phi & t_x \\ \varepsilon sin\phi & cos\phi & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

Where ε equals to ± 1, it is the orientation preserving when $\varepsilon = 1$, otherwise is the orientation reversing.

From equation 10, it can be simplifying shortly

$$x' = H_E x = \begin{bmatrix} R & t \\ 0^t & 1 \end{bmatrix} x \tag{11}$$

In this case, the matrix rotation R or $R^T R$equals$I$ In specifically, there are 3 degree of freedom, involving one rotation and two translations. In special cases, it will only contain the pure rotation and translation. There is constraints of properties in length, angle, and area that it required to invariants.

The second transformation is the similarities, where it allows to isometry and scale respectively. We can formalize like this:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} scos\phi & -ssin\phi & t_x \\ ssin\phi & scos\phi & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{12}$$

or equal generally 13.

$$x' = H_s x = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} \tag{13}$$

It is probably four DOF by adding one more parameter comparing with the Isometries, namely one scale is plus. other name is known as shape preserving. Importantly, the is acquired the fixed parameterization such as ratios of length, angle, ratios of areas, parallel lines based invariants.

A. Projective Transformations

A projectivity is an invertible mapping h from $(P2)$ to it self such that three points x_1, x_2, x_3

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = Hx \tag{14}$$

B. Representations of Affine Transformations

Next, we will explore the affine transformations that make the shape can be able to rotation and deformation. Therefor, it required to six-DOF by representing the two rotations, two translation, and two scales.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & a_{21} \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{15}$$

Similarly, it can be written by following:

$$x' = H_A x = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} x \tag{16}$$

Which A chain of matrices products, and defined:

$$A = R(\mathrm{o})R(-\phi)DR(\phi) \tag{17}$$

where

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \tag{18}$$

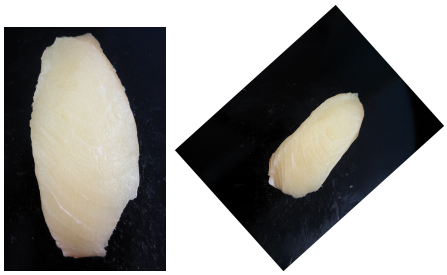

Fig. 2: Example of Projective Geometry and Transformation.

C. Projective Transformation

In some case, your pre-processing of training phases, it's also essential to define the infinity, where it is out of indexed point of images.

The projective transformations will have with 8-DOF by describing two more parameters. That is two line at infinity. It is called the action non-homogeneous over plane.

$$x' = H_P x = \begin{bmatrix} A & t \\ v^T & v \end{bmatrix} x \tag{19}$$

in where, the parameter v is declared:

$$v = (v_1, v_2)^T \tag{20}$$

The invariants will be contained in the cross-ratio of your points on a line.

$$\begin{bmatrix} A & t \\ 0^T & v \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} = \begin{bmatrix} A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ 0 \end{bmatrix} \tag{21}$$

Line at infinity stays at infinity, but points move along line.

$$\begin{bmatrix} A & t \\ v^T & v \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} = \begin{bmatrix} A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ v_1 x_1 + v_2 x_2 \end{bmatrix} \tag{22}$$

It is easily to calculate Line at infinity becomes finite, that allows to observe vanishing points, horizon.

we can decompose the projective transformations

$$H = H_S H_A H_P = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ v^T & v \end{bmatrix} = \begin{bmatrix} A & t \\ v^T & v \end{bmatrix} \tag{23}$$

Where A can be generalized by:

$$A = sRK + tv^T \tag{24}$$

in case, K is an upper-triangular, and $detK = 1$

D. Infinity Geometry Transformation

There is also necessary to define the line at infinite, where it is useful in practices where somehow existing the exceptions of error out of index during the preprocessing image.

In previous subsection, $(x_1, x_2{}^T)$, there is three degrees of freedom by following matrix:

$$\overline{x'} = H_{2x2} \overline{x} \tag{25}$$

Where $x_2 = 0$, therefore, the cross ratio will define by:

$$Cross(\overline{x}_1, \overline{x}_2, \overline{x}_3, \overline{x}_4) = \frac{|\overline{x}_1, \overline{x}_2||\overline{x}_3, \overline{x}_4|}{|\overline{x}_1, \overline{x}_3||\overline{x}_2, \overline{x}_4|} \tag{26}$$

From definition l_∞ as fixed line under the projective H transformation, we can represent by:

$$l'_\infty = H_A^{-T} l_\infty = \begin{bmatrix} A^{-T} & 0 \\ -At & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = l_\infty \tag{27}$$

And the
Projective metric is used by:

$$\begin{bmatrix} l'_1 & l'_2 & l'_3 \end{bmatrix} \begin{bmatrix} KK^T & K^T v \\ v^T K & v^T v \end{bmatrix} \begin{bmatrix} m'_1 & m'_2 & m'_3 \end{bmatrix} = 0 \tag{28}$$

IV. Mask R-CNN

Mask R-CNN was generalized from Faster RCNN

- Network Architecture (Convolutional Backbone): It is used the network depth feature based on standardized convolutional neural network, namely: ResNet50 and ResNet101. More detail is described at [5]. It's used to provide the feature extraction. At the beginner, the purpose is to detect the low level feature such as corners, edges, etc, after that it will be target on detect all kinds for food as the high level features.
  The figure 5 shown the resnet architecture. Let consider a is the activation function with respect to the l is a linear function of next output.

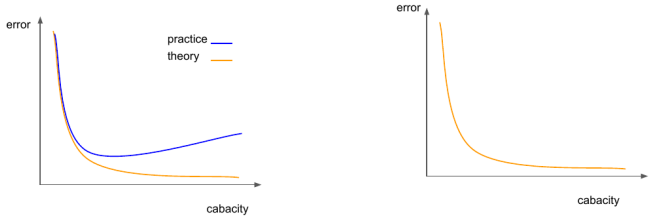

Fig. 3: Left: Plain Network; Right: Residual Network

Deep network is said to be more capable of learning complex feature than shallow networks. However, adding more hidden layers to a sufficiently deep network may degrade the model accuracy due to vanishing

gradient problem. This is a well-known issue in training a neural network where weights of the first layers can not be updated correctly through backpropagation of the error gradient. ResNet can avoid this issue by preserving the gradient during the backpropagation process. The basic idea behind gradient preservation is to backpropagate the error through the identity function such that the gradient would simply be multiplied by 1 (i.e., preserved). In detail, the transformation $y = \mathcal{F}(x) + x$ is manipulated instead of $y = \mathcal{F}(x)$ where x and y are input and output of stacked layers of which $\mathcal{F}$ is a nonlinear activation function, also called residual function. Figure 4 shows structure of a residual block.

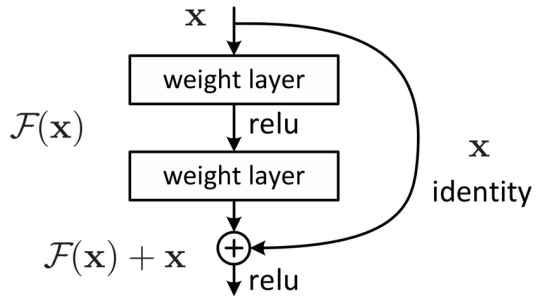

Fig. 4: Residual block.

The purpose of Residual Network (ResNet) is to build the deep network which is enable to train the very deep layers. For instance, we can use to build more than 100 layers, such as ResNet-101, ResNet-152.

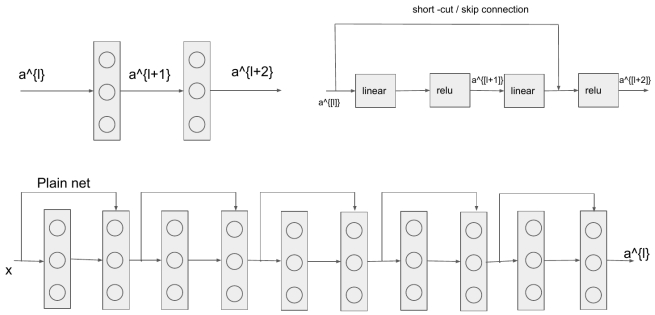

Fig. 5: ResNet backbone architecture: Top-Left: Residual Block; Top-Right: Designing the short-cut or skip connection; Bottom: Types of Plain Network and based on Residual Network.

Figure5, the Top-Left describe the block with beginning with the activation layer a^l as input, and then through out the linear operation

$$z^{[l+1]} = W^{[l+1]}a^{[l]} + b^{l+1} \tag{29}$$

Where b is adding as the bias vector, and the Weight matrix $W^{[l+1]}$ is used at $l + 1$.
And then we apply the nonlinear layer, called ReLU function to govern the output.

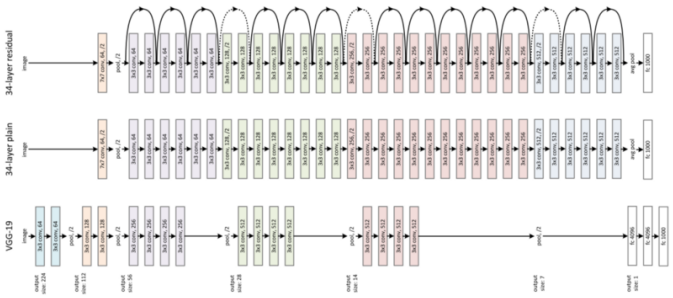

Fig. 6: ResNet backbone architecture

$$a^{[l+1]} = g(z^{[l+1]}) \tag{30}$$

For $z^{[l+1]}$, it is updated:

$$z^{[l+2]} = W^{[l+2]}a^{[l+1]} + b^{l+2} \tag{31}$$

Finally, we add another nonlinear function ReLU

$$a^{[l+2]} = g(z^{[l+2]}) \tag{32}$$

According to Figure:5 in Top-Right, the short-cut function in ResNet will be rewritten equation30 after applying the short-cut:

$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]}) \tag{33}$$

To access for both lower and higher level features, the Feature Pyramids Network is extended to improve standard feature extraction.

configuration	Parameters
ROI POSITIVE RATIO	0.33
RPN ANCHOR RATIOS	[0.5 1 2]
RPN ANCHOR SCALES	[32 64 128 256 512]
RPN ANCHOR STRIDE	1
RPN BBOX STD DEV	[0.1 0.1 0.2 0.2]
RPN NMS THRESHOLD	0.7
RPN TRAIN ARCHORS PER IMAGE	256
USE RPN ROIS	TRUE

- Region proposal networks (RPN) [20] is to share computation. The purpose is to take it as input for feature mapping, which is fed into fully connected layers, namely a box regression layer and box classification layer.
- Feature Mapping: it is defined by sliding the
- RoIAllign:
- Fixed size feature mapping:
- Mask branch:
- Fully connected layer:
- Box regression
- Classification:

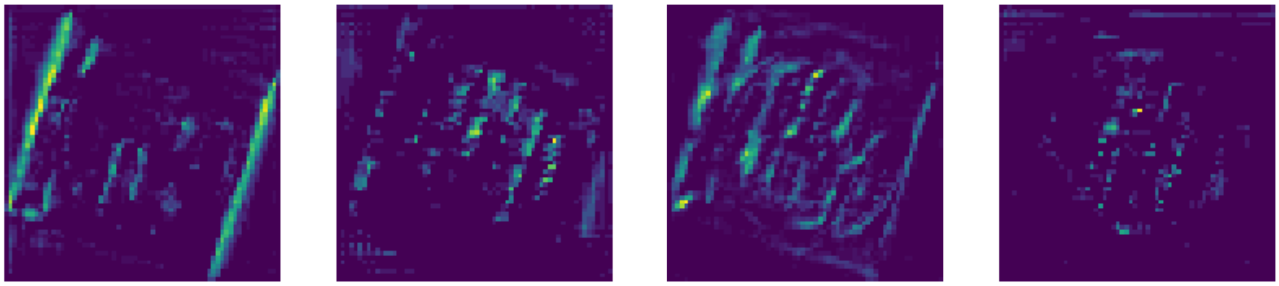

Fig. 7: Top: it described the layer of activation.

configuration	Parameters
Backbone	resnet101/resnet50
Backbone Strides	[4 8 16 32 64]
Batch Size	1
BBOX_STD_DEV	[0.1 0.1 0.2 0.2]
DETECTION MAX INSTANCES	100
DETECTION MIN CONFIDENCE	0.6
DETECTION NMS THRESHOLD	0.5
GPU count	1
GRADIENT CLIP NORM	5.0
IMAGE PER GPU	1
IMAGE MAX DIM	1024
IMAGE META SIZE	30
IMAGE MIN DIM	256
IMAGE MIN SCALE	0
IMAGE RESIZE MODDE	square
IMAGE SHAPE	[1024 1024 3]
Learning momentum	0.9
Learning rate	0.001
Loss weights	{ 'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0, 'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss' : 1.0 }
MASK POOL SIZE	14
MASK SHAPE	[28 28]
MASK GT INSTANCE	100
MEAN PIXEL	[123.7 116.8 103.9]
Number of classes	18
Pool Size	7
Post NMS ROIS INFERENCE	1000
Post NMS ROIS Training	2000
STEPS PER EPOCH	10
TRAIN BN	FALSE
TRAIN ROIS PER IMAGE	20
USE MIN MASK	TRUE
VALIDATION STEP	50
WEIGHT DECAY	0.0001

V. Experiment and Evaluation

A. Data Configurations

The configuration of training based on food datasets. Figure 8 provides the sampling for data training.

B. Training Mask R-CNN

1) Data Augmentation: Our method proposal is to support in order to handle of point, line, circle, rectangle and polygon in implementation.1

C. Results

1) Mask R-CNN for detection and segmentation: Currently, we use the Precision-Recall [2] to evaluate the object detection precisely. The mean average precision (mAP)

The histogram distribution can be able to visualize the data distribution. It is not only represented the comparison

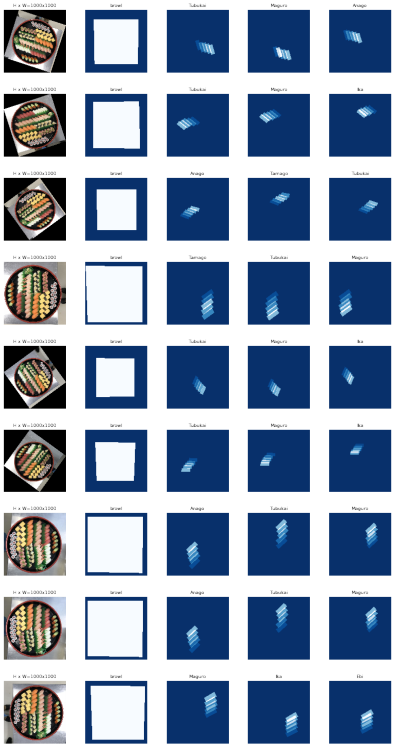

Fig. 8: Example of Display Samples.

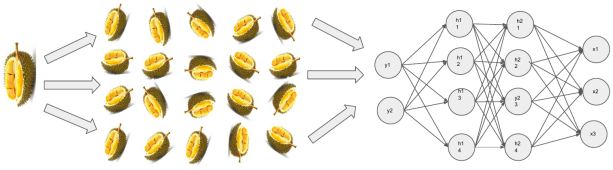

Fig. 9: Example of Data Augmentation, where x_i is the inputs, y_i is the output, and h_i is the hidden layer, and $i = 1, 2, \cdots, n$.

Name	tub.	mag.	tek.	ana.	sho.	ika	ebi	sam.	han.	neg.	bro.	ham.	iku.	eng.	sim.	bin.
Orig.	2500	2500	6000	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500
Q.1	35000	35000	84000	35000	7000	35000	35000	35000	35000	35000	35000	35000	35000	35000	35000	3500
AMax	93.6	92.2	99.5	94.4	0	93.5	92.7	98.6	98.4	98.4	98.3	99.8	93.6	98.4	95.0	95.4
Amin	74.8	66.8	99.2	87.4	0	69.9	78.8	77.8	77.8	92.9	96.9	90.7	74.8	96.9	90.7	89.2

TABLE I: First row shown the number of objects trained before augmentation data. The second row also provides the number of object labeled. And it is excitingly representing the last two rows respectively the maximum and minimum of detection accuracy.

Number	First Time	Second Time	n Time
Time of Performance	75	34	34

between data over time and range, but also illustrated the distribution of patterns and range based on datasets.

Another useful is to see of overview based on giving the probability distribution. In this case, we shown the distribution of four parameters, namely Figure:15 the first two values is illustrated for

There is a small problem on training

In addition, there

$$T_{TimeOfPerformance} = t_0 + 34 \times n_i \tag{34}$$

Where $i = 1, 2, 3, \cdots$, and there are two cases in this equation

- If $t = 0$ is an initial state, the $T_{TimeOfPerformance} = 75$
- If $t = i$, the result will be the products of $ntimes$ according the constant is equal to 34.

In our experiences, the most important things on training datasets by using data augmentation and transfer learning is

One of most weak-points by using the data-driven approach is range of variance of data sets.

VI. Conclusion

In this paper, we focus on representing the visualization and detection analysis based on deep learning approaches in order to understanding deeper on food data recognitions.

In the future works, we will investigate the increasing variety of datasets. Secondly, it is necessary to improve the current time execution

References

[1] Hartley, R. I. and Zisserman, A., "Multiple View Geometry in Computer Vision", Cambridge University Press, ISBN: 0521540518, Second Editor, 2004.

[2] Henderson P., Ferrari V. "End-to-End Training of Object Class Detectors for Mean Average Precision." In: Lai SH., Lepetit V., Nishino K., Sato Y. (eds) Computer Vision ACCV 2016. ACCV 2016. Lecture Notes in Computer Science, vol 10115. Springer, 2017.

[3] "Food market structures: Overview". Economic Research Service (USDA).

[4] Yuzhen Lu, "Food Image Recognition by Using Convolutional Neural Networks (CNNs)." arXiv, 2018.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun "Deep Residual Learning for Image Recognition." Computer Vision and Pattern Recognition (CVPR), 2016.

[6] Tsung-Yi Lin, Piotr Dollr, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie, "Feature Pyramid Networks for Object Detection." arXiv, arXiv:1612.03144, 2017.

[7] Tom Duckett, Simon Pearson, Simon Blackmore, Bruce Grieve, Wen-Hua Chen, Grzegorz Cielniak, Jason Cleaversmith, Jian Dai, Steve Davis, Charles Fox, Pl From, Ioannis Georgilas, Richie Gill, Iain Gould, Marc Hanheide, Alan Hunter, Fumiya Iida, Lyudmila Mihalyova, Samia Nefti-Meziani, Gerhard Neumann, Paolo Paoletti, Tony Pridmore, Dave Ross, Melvyn Smith, Martin Stoelen, Mark Swainson, Sam Wane, Peter Wilson, Isobel Wright, Guang-Zhong Yang, "Agricultural Robotics: The Future of Robotic Agriculture". arXiv, arXiv:1806.06762 [cs.RO], 2017.

[8] Hoi V. Nguyen, Dung D. Huynh, Than D. Le, and Peter Nauth, Forward Kinematics of a Human-Arm System and Inverse Kinematics using Vector Calculus The International Conference on Control, Automation, Robotics, and Vision (ICARCV), 2016.

[9] A. Bicchi, "Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity," IEEE Transactions on Robotics and Automation, 2000.

[10] Matthew D Zeiler, Rob Fergus, "Visualizing and Understanding Convolutional Networks," arXiv:1311.2901 [cs.CV], 2013.

[11] Gianluigi Ciocca, Paolo Napoletano, Raimondo Schettini, "Food Recognition: A New Dataset, Experiments, and Results," IEEE Journal of Biomedical and Health Informatics, 2017.

[12] Kawano, Y. and Yanai, K.", "Automatic Expansion of a Food Image Dataset Leveraging Existing Categories with Domain Adaptation", Proc. of ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV), 2014

[13] Bossard, Lukas and Guillaumin, Matthieu and Van Gool, Luc, "Food-101 – Mining Discriminative Components with Random Forests," European Conference on Computer Vision, 2014.

[14] Dataset UPMC-G20 (gaze for UPMC Food-101), http://visiir.lip6.fr/.

[15] Kaiming He ; Xiangyu Zhang ; Shaoqing Ren ; Jian Sun, "Deep Residual Learning for Image Recognition." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[16] R. Ranjan, V.M. Patel, R. Chellappa, "HyperFace, A Deep Multi-task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition.", The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2016.

[17] S, Yang, P. Luo, C. Loy, X. Tang, "From Facial Parts Responses to Face Detection: A Deep Learning Approach", The IEEE International Conference on Computer Vision (ICCV), 2015.

[18] F. Schroff, D. Kalenichenko, J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering", Computer Vision and Pattern Recognition (CVPR), 2015.

[19] R. Girshick, "Fast R-CNN", The IEEE International Conference on Computer Vision (ICCV), 2015.

[20] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", Conference on Neural Information Processing Systems (NIPS), 2015

[21] K. He, G. Gkioxari, P. Dollár, R. Girshick, "Mask R-CNN", The IEEE International Conference on Computer Vision (ICCV), 2017.

[22] "VGG Image Annotator", http://www.robots.ox.ac.uk/ vgg/, Oxford.

[23] Luis Perez, Jason Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning", arXiv, 2017.

[24] Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson, "How transferable are features in deep neural networks?" Advances in Neural Information Processing Systems, 2014.

[25] CNN Features off-the-shelf: an Astounding Baseline for Recognition Ali S. Razavian, Hossein Azizpour, Josephine Sullivan Stefan Carlsson, "CNN Features off-the-shelf: an Astounding Baseline for Recognition", arXiv, 2017.

[26] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, Trevor Darrell, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", The International Conference on International Conference on Machine Learning, 2017.

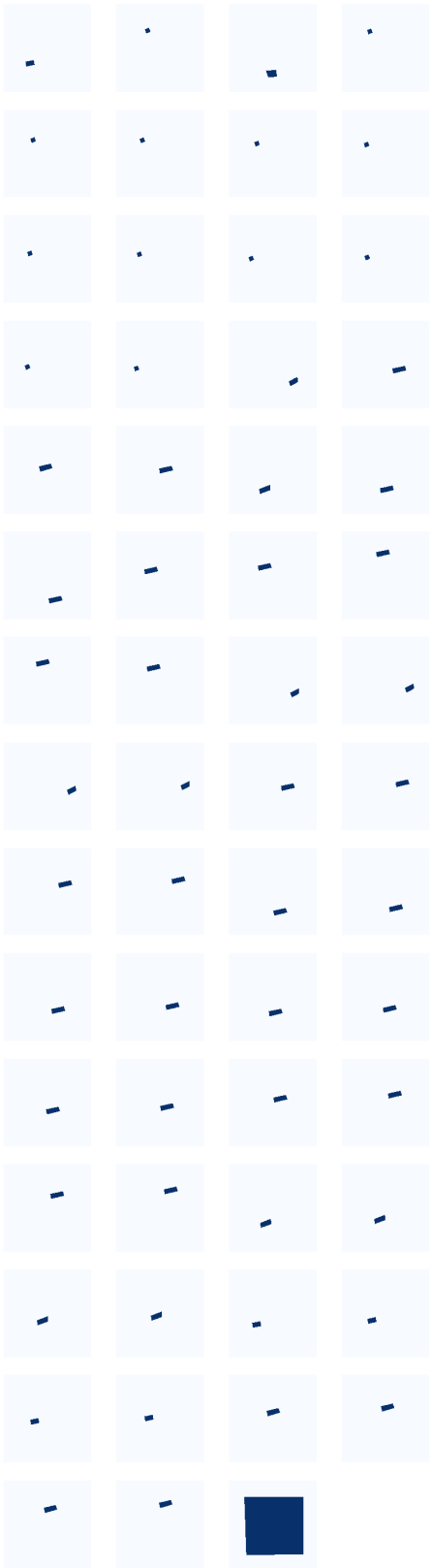

Fig. 10: Visualization of Generalized Mask.

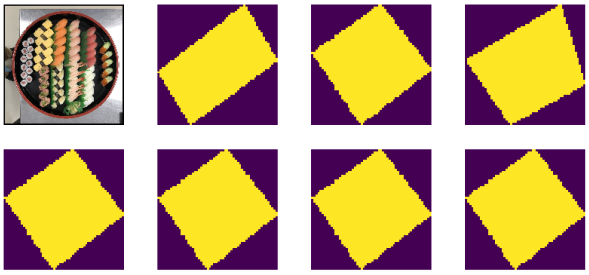

Fig. 11: Visualization of Augmentation.

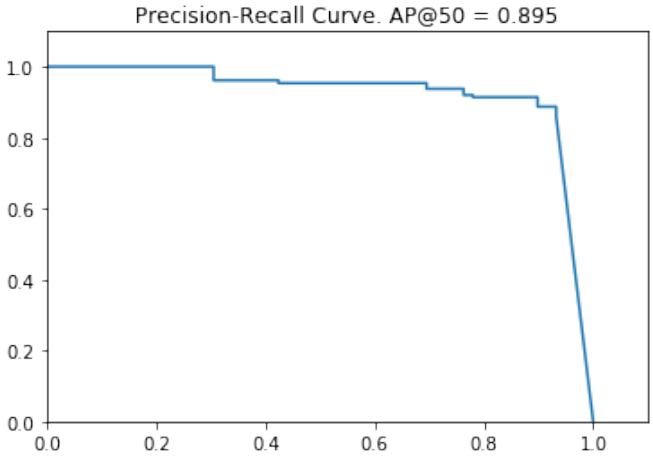

Fig. 12: Precision-Recall visualization with 5000 sushi images.

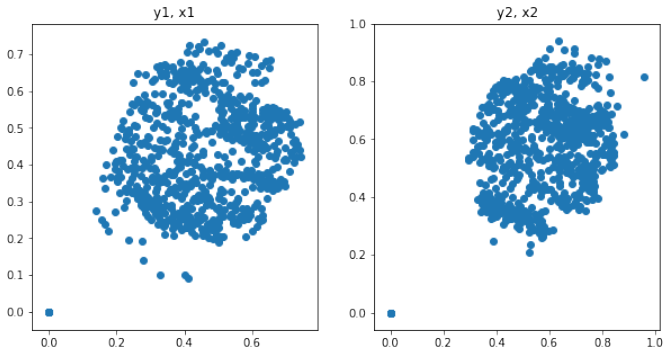

Fig. 13: Distribution of normalized coordinates (y_1, x_1) and width-height (y_2, x_2) of data samples. Top is 5000 images images.

[27] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei, "ImageNet: A large-scale hierarchical image database", IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[28] Joseph Redmon, Ali Farhadi, "YOLO9000: Better, Faster, Stronger", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[29] Mask R-CNN implementation, https://github.com/matterport/Mask_RCNN

[30] Ian Goodfellow and Yoshua Bengio and Aaron Courville, "Deep Learning (Adaptive Computation and Machine Learning)", MIT Press, 2016.

[31] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, " Rich feature hierarchies for accurate object detection and semantic segmentation", Techical report, 2014.

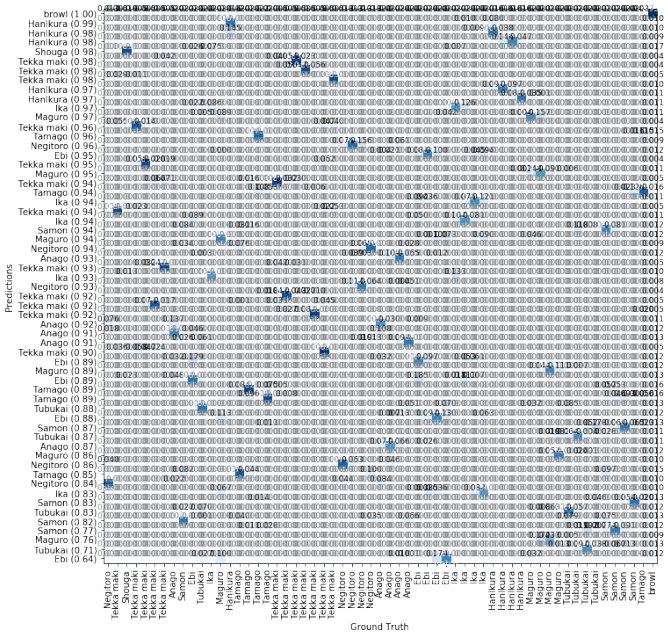

Fig. 14: .

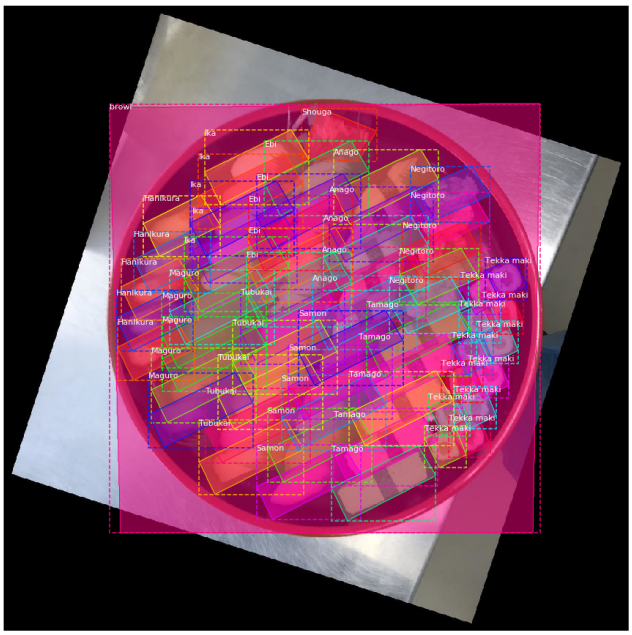

Fig. 17: .

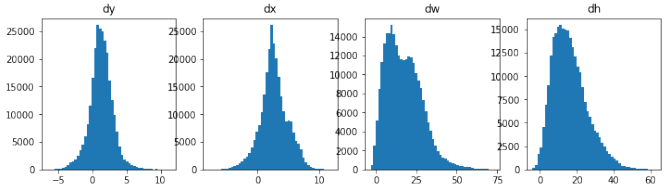

Fig. 15: Representation of example of the histogram distribution based on both training and modelling datasets. Firstly, dx and dy are represented the coordinate system of x axis and y axis.

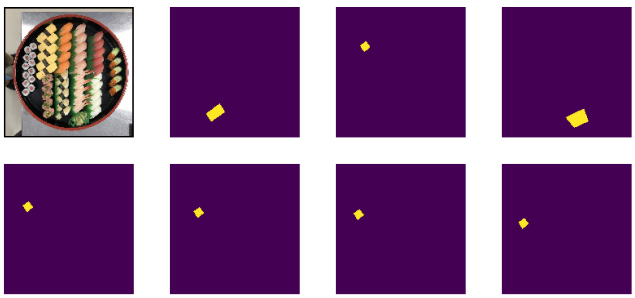

Fig. 16: C.

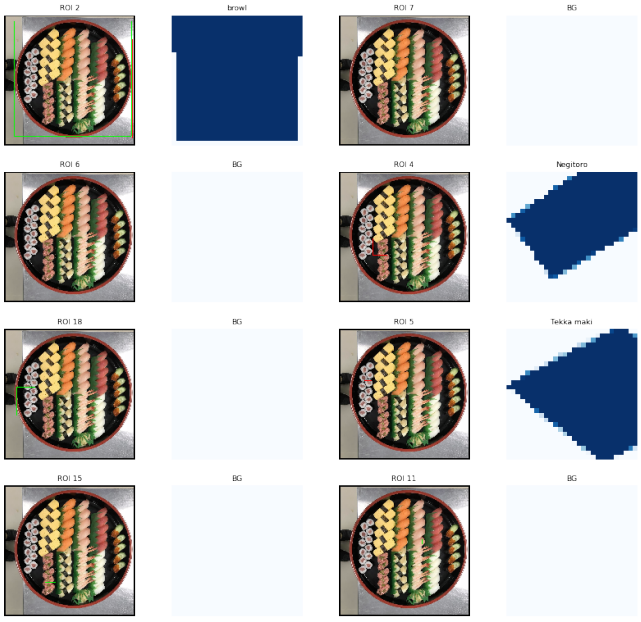

Fig. 18: C.