Faster R-CNN with Inception V2 for Fingertip Detection in Homogenous Background Image

IOP Conf. Series: Journal of Physics: Conf. Series **1196** (2019) 012017 doi:10.1088/1742-6596/1196/1/012017

Faster R-CNN with Inception V2 for Fingertip Detection in Homogenous Background Image

Derry Alamsyah[1], Muhammad Fachrurrozi[2]

[1] STIMIK MDP Palembang, Indonesia

[2] Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya, Indonesia

mfachrz@unsri.ac.id

Abstract. Fingertip Detection is a versatile research field in computer vision, since it has multiple purpose such as natural user interface, robotic, 3D simulation etc. It is a challenging research field in computer vision. There are some hand segmentation methods, pre-processing phase, was conducted to provide area of fingertip detection in image. However, in this research, fingertip detection can be done by directly find the fingertip itself. This approach cut off pre-processing phase by using Faster R-CNN method and inception V2 architechture directly to find the fingertip in image. With a homogenous background as a simple input image, this approach showed a good accuracy in its performance. It has 90% and 91% accuracy in way to detect fingertip for both male and female hand datasets. More over, exchanging male and female model toward to male and female dataset gave 94% and 92% accuracy that showed the different pattern between both.

1. Introduction

Fingertip detection is part of computer vision field that can be utilized for multiple purpose. One of them is used in Natural User Interface (NUI) system. This system can be operated by some natural movements such as eye, head or hand and even fingertip. Others, fingertip detection can be combined with 3D or robotic system which can be used as 3D visual/simulation or hand robot respectively.

Fingertip detection can be done by several methods, such as by Geometric Structure, Geometric feature, Depth Information, etc. The first method was used some geometric structure like convex hull [1] or gradient information [2]. Those method are aimed to find the line that tend to be a curved. In [1] is used a preprocessing method like edge detection. Second method on the other hand is aimed to find certain feature, such as circle [3], curvature points [4], or even basic feature like color [3]. The last method much modern, it utilized the depth information [5] [6]. Depth information can be obtained by such tool like Kinect [1] or using stochastics method as time series information from each pixel [7].

As a trend method in artificial intelligence or machine learning, deep learning significantly spread out in many researches. It has better accuracy on their duty. First, common method is convolution neural network (CNN). It works like a neural network but utilize such feature to make a deep learn. CNN was improved by [8], they use such module as a wider layer called Inception. The module reduced CNN computational time.

More over CNN was modified or improved instead to gain better performance, Recurrent CNN was introduced. R-CNN than is modified to be fast R-CNN and faster R-CNN. The last method was

introduced by [9] it used Region Proposal Network (RPN) to reduce computational time and make a good accuracy as their predecessor method. Faster R-CNN than spread out in many researches in object detection, they are [10] detect book, [11] detect pedestrian, etc.

From the previous research, there is an area of research remains that is directly using deep learning method to detect the fingertip than use some method or device to segment hand region. By using the better performance of Inception V2 in CNN and recognition ability of Faster R-CNN, this research conduct both methods to detect each of fingertip in one hand image.

2. Faster R-CNN

Faster R-CNN introduced the Region Proposal Network (RPN) in their architecture. It means that it handles slow search selective algorithm with fast neural network. RPN is placed after last convolution layer of CNN. Proposal from RPN are fed to ROI pooling layer followed by Classifier and Bbox Regressor. [9] The architecture of Faster R-CNN is showed by figure 1.

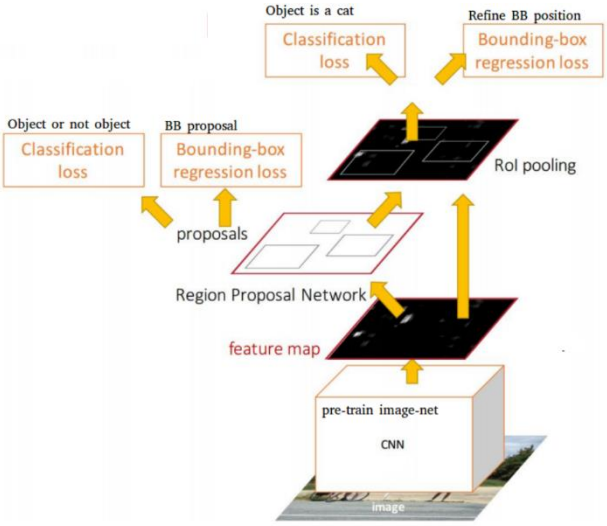


Figure 1. Faster R-CNN architecture.

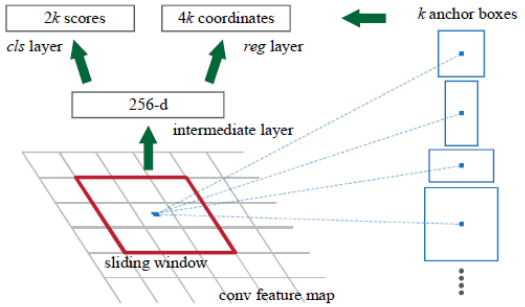


Figure 2. Region Proposal Network (RPN).

RPN maps out the last layer of CNN (sliding window) to a lower dimension (256-d) into feature map, showed by figure 2. It generates multiple possible region based on k fixed-ratio anchor box for each sliding window [9]. The RPN architecture is showed in figure 2. The given region proposal consists of object score and $4k$ coordinates (showed by equation (1-8)) representing the bounding box region. The object score represents the soft-max probability and it is determined whether the box pass or not as region proposal. The decision is influenced by certain threshold. For $4k$ coordinates is denoted by the box center (x, y), width (w) and height (h). The variables x, x_a and x_a^* (and the other) are for predicted box, anchor box, and ground truth respectively.

$$t_x = \frac{(x - x_a)}{w_a} \tag{1}$$

$$t_x^* = \frac{(x^* - x_a)}{w_a} \tag{2}$$

$$t_y = \frac{(y - y_a)}{h_a} \tag{3}$$

$$t_y^* = \frac{(y^* - y_a)}{h_a} \tag{4}$$

$$t_w = \log\left(\frac{w}{w_a}\right) \tag{5}$$

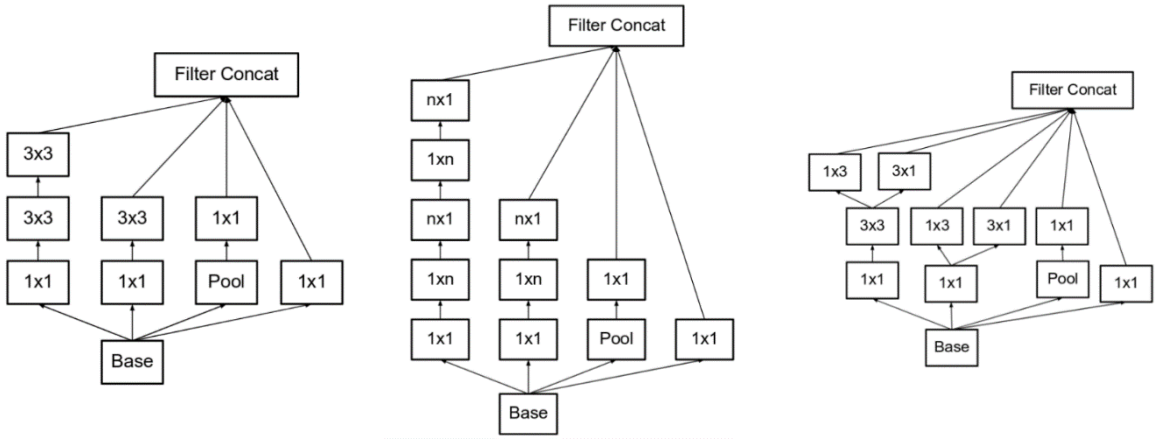
$$t_w^* = \log\left(\frac{w^*}{w_a}\right) \tag{6}$$

$$t_h = \log\left(\frac{h}{h_a}\right) \quad (7)$$

$$t_h^* = \log\left(\frac{h^*}{h_a}\right) \quad (8)$$

The learning of region proposal used (9) equation as a Loss Function that add both of Classification Loss (L_{cls}) and Regression Loss(L_{reg}). It is used predicted probability (p_i) and 4k coordinates (t_i) for each i-anchor in mini batch. On the other side, p_i^* and t_i^* are the ground-truth label and ground-truth box respectively. The two terms are normalized with N_{cls}, N_{reg} and with a balancing weight λ [9].

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, pi^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i * L_{reg}(t_i, t_i^*) \quad (9)$$



(a) Module A. **(b)** Module B. **(c)** Module C.

Figure 3. Module Inception V2

3. Inception V2

Inception V2 was a module that designed to reduce the complexity of convolution network. This module makes convolution network going to be wider than deeper. Inception V2 has three different type modules said A, B, C. They are showed by figure 3. The fist module (A), showed by Figure 3.a, replaced 5x5 convolution to be 3x3 convolution. This follow the principle said spatial aggregation can be done over lower dimensional embedding without much or any loss in representational power. By conducted the 3x3 convolution, convolution performance was boosted [8].

By factorized convolutions filter size *n* x *n* into *1* x *n* and *n* x *1* convolutions, [8] found their method 33% cheaper than the single 3x3 convolution. This was showed by Figure 3.b as module B. Moreover, the filter was expanded that followed the principle of higher dimensional representations are easier to process locally within a network. The expanded module was showed by figure 3.c.

4. Experiment and Result

This research used public hand dataset. The hand dataset was taken from IT Department Mutah University as Webcam Hand Images Database (WEHI). There is 40 hand dataset sample was randomly taken and group by two kind data (Training and Testing). Here, training hand dataset was showed by Figure 4 and 5 in the first row. On the other hand, number of test data are 10 for each men and women.

This research has four models from given hand image dataset. First model used 10 hand image datasets consist of 5 hand images for male and female. Both female and male image was used simultaneously in this model. With the same process, second model used 20 hand image datasets where

the dataset consists of 10 hand images for female and male. The performance of models is showed in Table 1 and 2 respectively.

Figure 4. Female Hand Dataset

Figure 5. Male Hand Dataset.

Table 1. Fingertip Detection using 10 Train Hand Dataset (Model 1) Performance

Hand Image	Detected Fingertips	Undetected Fingertips	Processing Time (Second)
Image 1 (F)	4	1	12
Image 2 (F)	5	0	09
Image 3 (F)	5	0	10
Image 4 (F)	2	3	16
Image 5 (F)	4	1	10
Image 6 (F)	4	1	14
Image 7 (F)	5	0	11
Image 8 (F)	5	0	14
Image 9 (F)	5	0	10
Image 10 (F)	5	0	10
Image 11 (M)	5	0	18
Image 12 (M)	5	0	13
Image 13 (M)	3	2	16
Image 14 (M)	5	0	14
Image 15 (M)	4	1	17
Image 16 (M)	5	0	16
Image 17 (M)	4	1	14
Image 18 (M)	5	0	15
Image 19 (M)	5	0	20
Image 20 (M)	5	0	14

Table 2. Fingertip Detection using 20 Train Hand Dataset (Model 2) Performance

Hand Image	Detected Fingertips	Undetected Fingertips	Processing Time (Second)
Image 1 (F)	5	0	11
Image 2 (F)	5	0	09
Image 3 (F)	5	0	09
Image 4 (F)	1	4	08
Image 5 (F)	4	1	09
Image 6 (F)	4	1	09
Image 7 (F)	5	0	08
Image 8 (F)	5	0	09
Image 9 (F)	5	0	08
Image 10 (F)	5	0	08
Image 11 (M)	5	0	08
Image 12 (M)	5	0	09
Image 13 (M)	4	1	08
Image 14 (M)	5	0	08
Image 15 (M)	4	1	08
Image 16 (M)	5	0	09
Image 17 (M)	4	1	08
Image 18 (M)	5	0	09
Image 19 (M)	5	0	08
Image 20 (M)	5	0	08

**F: Female and M: Male*

The classification loss function is showed by figure 8 and 9 for each train data sample group, which is 10 and 20 hand datasets respectively. All Loss function was stable below 0.05. The train step

performance shows good result for fingertip detection. The result of fingertip detection was showed by figure 6,7.

For next two model, this research tries to find out the different of hand effect for male or female hand, as shows in figure 8. The third model used only female hand dataset, 10 images. Then, by the given female model, male hand dataset is detected. The result of using third model (female fingertip detection model) to male hand dataset showed in Table 3. The rest, female hand dataset was detected by male fingertip detection model showed in Table 4.

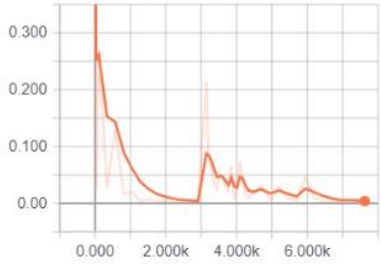


Figure 6. 10 train data sample

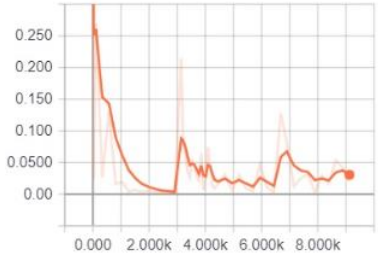


Figure 7. 20 train data sample

Table 3. Female Fingertip Detection (Model 3) Performance toward Male Hand Image

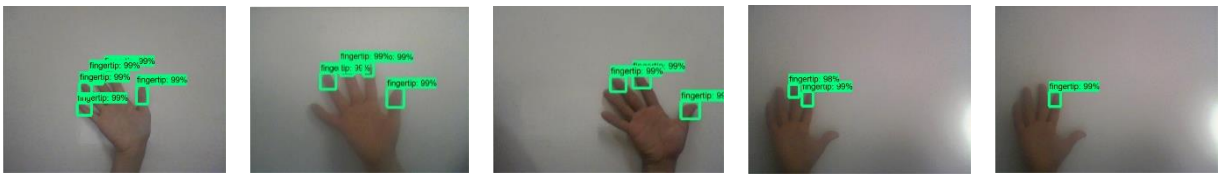
Male Hand Image	Correct Fingertips	Incorrect Fingertips	Processing Time (Second)
Image 1	5	0	11
Image 2	5	0	8
Image 3	4	1	9
Image 4	5	0	9
Image 5	4	1	10
Image 6	5	0	9
Image 7	4	1	10
Image 8	5	0	9
Image 9	5	0	10
Image 10	5	0	10

Table 4. Male Fingertip Detection (Model 4) Performance toward Male Hand Image

Female Hand Image	Detected Fingertips	Undetected Fingertips	Processing Time (Second)
Image 1	5	0	10
Image 2	5	0	10
Image 3	5	0	10
Image 4	3	2	10
Image 5	4	1	10
Image 6	4	1	10
Image 7	5	0	09
Image 8	5	0	09
Image 9	5	0	09
Image 10	5	0	10

In summary, given four models showed each of their accuracy in Table 5. The first two model showed that even the dataset is added, the accuracy is not increasing significantly. However, processing time is reduced. From the first two model performance give information that after added some hand dataset the model can figure out more specific rule/threshold. With the more specific rule the time process was reduced.

For the rest model, Table 5 show that there is a different pattern in male and female hand dataset. It showed by the different processing time and even the accuracy. Overall, each of model has good accuracy, all more than equal to 90%.



(a) (b) (c) (d) (e)

Figure 8. Fingertip Detection: (a) all correctness, (b) 4 correctness, (c) 3 correctness, (d) 2 correctness, (e) 1 correctness

Table 5. Summary of Model Performance

Model	Acc.	Processing Time (Average)
Model 1	90 %	14 s
Model 2	91 %	9 s
Model 3	94 %	9 s
Model 4	92 %	10 s

5. Conclusion

The utility of inception V2 as a module in CNN shows almost perfect results. The classification by using faster R-CNN model for fingertip detection produce accuracy rates 90%, 91%, 94% and 92% from different training phase. Faster R-CNN are potentially able to be a good method in detecting and tracking fingertips.

References

[1] D. Alamsyah and M. I. Fanany. 2013. *Particle filter for 3D fingertips tracking from color and depth images with occlusion handling*. IEEE conference on International Conference on Advanced Computer Science and Information Systems (ICACSIS). pp. 445-449.

[2] A. Wang, H. Lu, and H. Lu. 2015. *An effective real-time fingertip positioning system based on gradient information extraction from frame image sequences*. 8th International Congress on Image and Signal Processing (CISP). pp. 330-334.

[3] Md. J. Alam and M. Chowdhury. 2013. *Detection of fingertips based on the combination of color information and circle detection*. IEEE 8th International Conference on Industrial and Information Systems. pp. 572 - 576.

[4] G. Wu and W. Kang. 2017. *Vision-Based Fingertip Tracking Utilizing Curvature Points Clustering and Hash Model Representation*. IEEE Transactions on Multimedia. pp. 1730-1741. Vol. 19.

[5] Y. Bak, M. Li, L. Sun, and Q. Huo. 2017. *Fingertip detection based on protuberant saliency from depth image*. IEEE International Conference on Image Processing (ICIP). pp. 3380-3384.

[6] C. Liang, Y. Song, and Y. Zhang. 2015. *Real-time fingertip detection based on depth data*. 3rd IAPR Asian Conference on Pattern Recognition (ACPR).

[7] M. Rachmadi and D. Alamsyah. 2017. *Estimasi Citra Kedalaman Dengan Conditional Random Field (CRF) dan Structured Support Vector Machine (SSVM)*. Jurnal Rekayasa Sistem dan Teknologi Informasi (RESTI). pp. 198-203. Vol. 1. No. 3.

[8] C. Szegedy, V. Vanhoucke, S. Ioffe, and J Shlens. 2016. *Rethinking the Inception Architecture for Computer Vision*. IEEE conference on Computer Vision and Pattern Recognition (CVPR). pp. 2818-2826.

[9] S. Ren, K. He, R. Girshick, and J. Sun. 2016. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. IEEE transactions on Pattern Analysis and Machine Intelegence. pp. 1137-1149. Vol. 39.

[10] B. Zhu, X. Wu, L. Yang, Y. Shen, and L. Wu. 2016. *Automatic detection of books based on Faster R-CNN*. Third International Conference on Digital Information Processing, Data Mning, and Wireless Communication (DIPDMWC). pp. 8-12.

[11] H. Zhang, Y. Du, S. Ning, Y. Zhang, S. Yang and C. Du. 2017. *Pedestrian Detection Method Based on Faster R-CNN*. 13th International Conference on Computational Inteligence and Security (CIS). pp. 427-430.