International Journal of Data Science
Vol. 2, No. 2, December 2021, pp. 63-68

Mask RCNN Methods for Eyes Modelling

Rahmat Hidayat [a,1,*], Hendrick [b,2], Riandini [c,3], Zhi-Hao Wang [d,4], Horng Gwo-Jiun [e,5]

[a] Department of Information Technology, Politeknik Negeri Padang, West Sumatera, Indonesia
[b] Department of Electrical Engineering, Politeknik Negeri Padang, West Sumatera, Indonesia
[c] Department of Electrical Engineering, Politeknik Negeri Jakarta, DKI Jakarta, Indonesia
[d] Departement of Information Management, Southern Taiwan University of Science and Technology, Kaohsiung, Taiwan
[e] Departement of Computer Science and Information Technology, Southern Taiwan University of Science and Technology, Tainan, Taiwan
[1] rahmat@pnp.ac.id; [2]hendrick@pnp.ac.id; [3]riandini@elektro.pnj.ac.id; [4]zhwang0401@hotmail.com; [5]grojium@gmail.com
* corresponding author

ARTICLE INFO

Article history
Received August 21, 2021
Revised October 7, 2021
Accepted November 15, 2021

Keywords
eye
mask RCNN
object detection
webcam
OpenCV

ABSTRACT

Object detection is one of Deep Learning section in Computer Vision. The application of computer vision is divided into image classification and object detection. Object detection have target to find specific object from an image. The application of object detection for security are face recognition, and face detection. Face detection have been developed for medical application to identify emotion from faces. In this research, we proposed an eye modelling by using Mask RCNN. The eye model was applied in real time detection combined with OpenCV. The dataset was created from online dataset and image from webcam. The model was trained with 4 epochs and 131 iterations. The final model was successfully detected eye from real-time application.

1. Introduction

Eyes can provide some information's about brain activities by analyzing the pupil. For example, doctor tested brain activity by giving light to eyes manually. By using pupil reaction, some researcher find that it related with brain activities [rat analysis]. Some researcher also mentioned that the pupil also related with audio and visual stimulation [1]. The visual stimulation is also important to practice the focus. When the eyes are focused to an object, it will change the retina size [1]. The effect of the stimuli will change the retina size. The First stage to do is to create a model to detect eyes in real time using processing.

The method to detect specific object from images is called object detection. The common object detection is provided by OpenCV applying Haar Cascade method [2]. Nowadays, object detection is combined with Machine Learning and Deep Learning method. Both Machine Learning and Deep Learning method will produce a model to classify images [3], [4], [5], [6] and detect an object. One of the Deep Learning method is landmark [7]. Landmark detection will mark the face based on parts. The method provides some number for each part. There are two kind of landmark methods which have points 68 and 128 points. The others Deep Learning methods have been developed to detect object such as YOLO [8], SSD MobileNet [9], and Fast RCNN which based on Convolutional Neural Network (CNN). Object detection model is created by using several methods which are pre-trained, transfer learning, and training from scratch.

The proposed of this research is to present the eyes modelling by using RGB images. The eyes model was trained from scratch with our own dataset. To create the model, Mask RCNN method was selected to perform the eyes model for RGB image. The deep learning process have two steps which are training and deploying model in OpenCV. The Tensorflow-GPU is applied in this research to train dataset achieving the model. The final model will run on OpenCV by deploying the model which have type as hdf5 (h5). The training will run in 4 epochs and 131 iterations. Finally, the model will be tested by using image file and image from webcamera.

2. Methods

2.1. Data Collection

The dataset of this research was captured with webcam directly. The dataset image was built from 150 images. The image was captured by using webcam and saving in RGB format. The JPEG type was selected to save the images. Fig. 1. displays the example of dataset images. The image is not only for eyes images, but the image also contain all of face part.

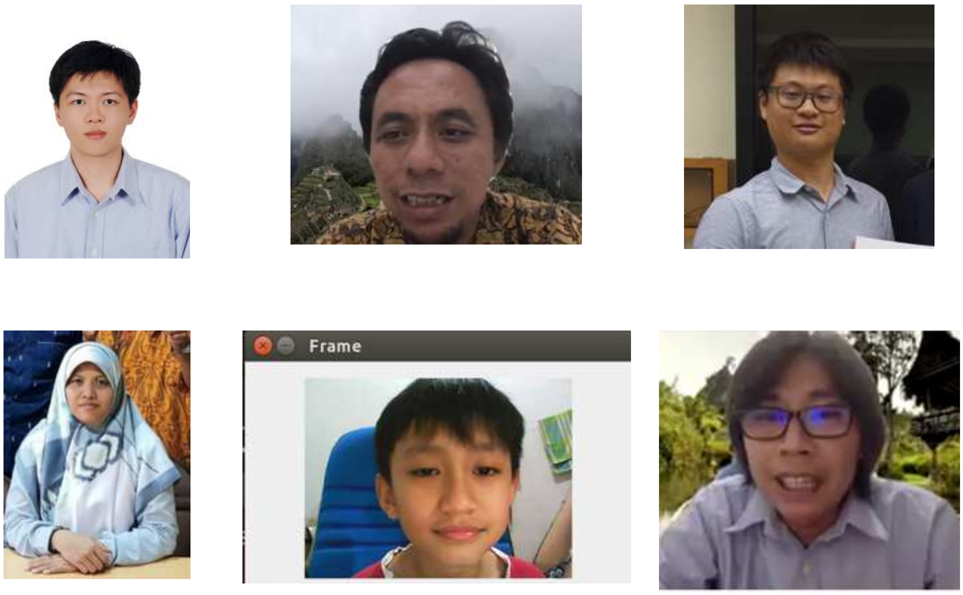


Fig. 1 Eyes sample image.

2.2. Labelling Object

The object detection dataset have to provide a label to locate the object location in image. The labelling process was done manually by using an open source software. Fig.2 depicts the labelling process by using LabelImage software. The labelling process produced the xml file which contain the object location. Fig.2.(a) depicts the image with specific objects which are eyes. Based on the image, there are two objects in one frame which are right eye and left eye. Fig.2.(b) shows the xml file which have object location and label. The object location have value which are xmin, xmax, ymin, and ymax.

(a)

(b)

Fig. 2 Object labelling and object location.

2.3. Mask RCNN

Figure 3 depicts the Mask RCNN architecture. Basically, Mask RCNN is almost the same with Fast RCNN, but the different is the instance segmentation. By applying instance segmentation, the Mask RCNN output will be segmented. Mask RCNN supports both object detection and object segmentation. Fig 4 shows the stage of Mask RCNN process.

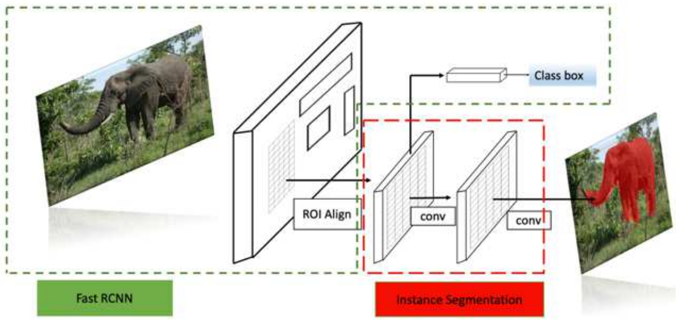


Fig. 3 Mask RCNN Architecture

2.4. Hardware and Software Preparation

The hardware was provided for the training process depicted in table 1. GPU was necessary in the training process to speed up the time process for eye modelling. Tensorflow-GPU version 1.12 was selected to handle the training process with Keras support. Beside the Tensorflow-GPU, OpenCV was also installed to run the model in real time application using webcam.

Tabel 1. Hardware Specification

Hardware	Specification
CPU	Core i7
GPU	GTX 750M (2G)
Memory	16G
Webcam	Logitech

3. Results and Discussion

The training process was done in 4 epochs and 131 iterations. The training time was approximately 7 hours. Based on training, the eyes model were saved into hdf5 files. All the models were tested to The test-image to find the best model performance . Fig. 4 depicts the tested images with 1st model. The red rectangle in image represented the predicted object from image. The first model have error when predicted the first image. The model predicted eyes as one object. The object should be predicted as two objects which were left eye and right eye. The others image were correctly predicted.

Figure 5 depicts the result of 2nd model prediction of images. The first image was predicted with wrong object. The model classified two eyes as one object. The other images were successfully predicted as eye for every image. Fig.6 shows the tested image with 3rd model. Some errors were found which were shown by 3 rectangles in image. It should have two rectangles from 1 image. Fig.7 shows the tested image with the 4th model of Mask RCNN. Every eye in images were successfully predicted. The 4th model was selected as final model in this research.

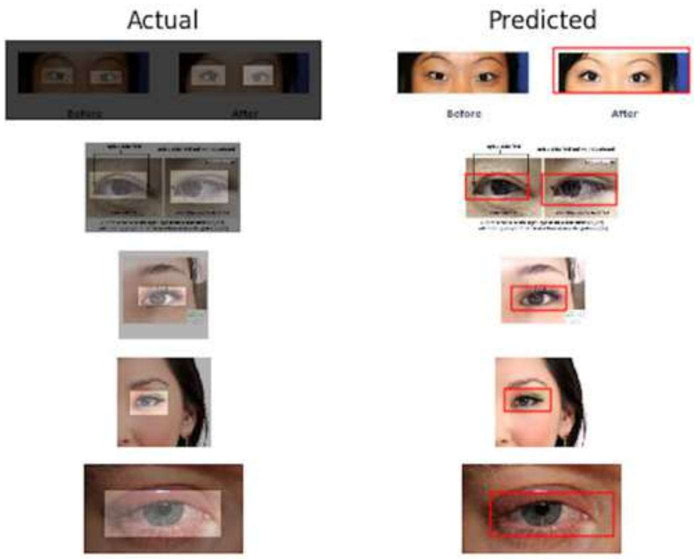


Fig. 4 Tested image with 1st model.

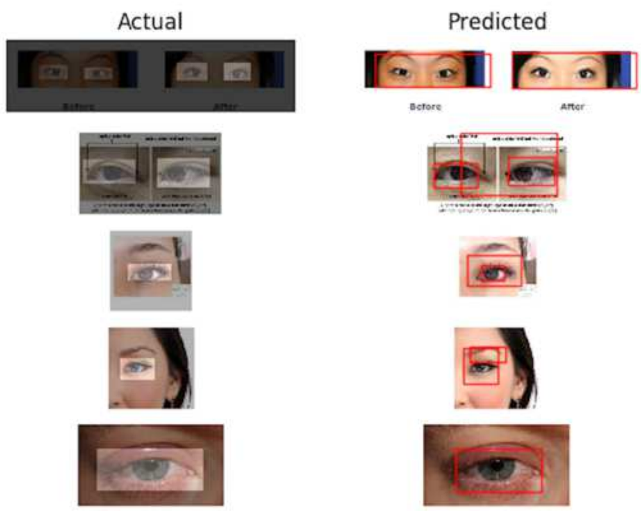


Fig. 5 Tested image with 2nd model

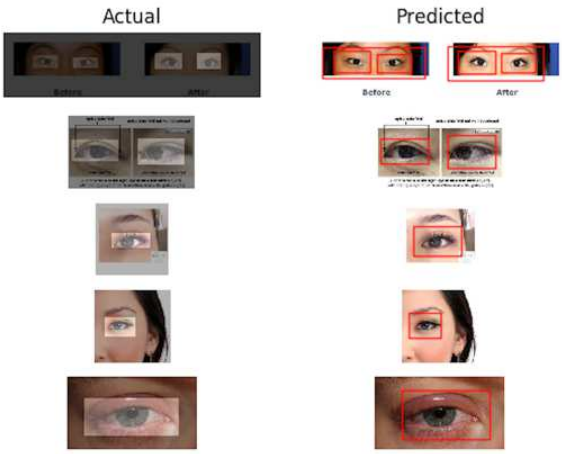


Fig. 6 Tested image with 3rd model

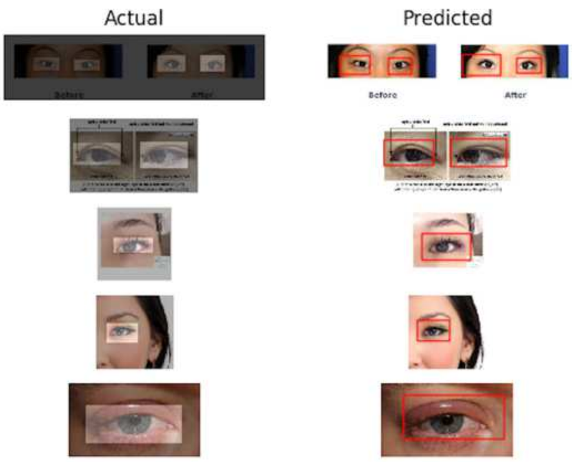
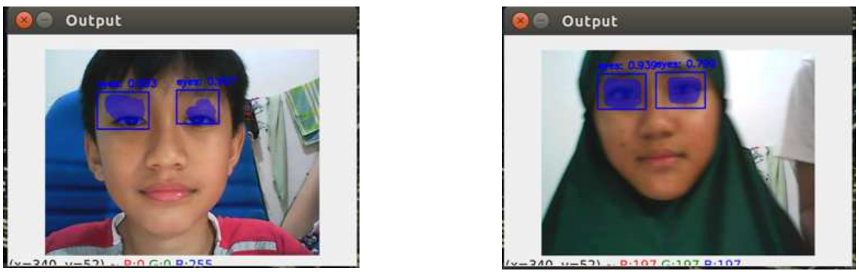


Fig. 7 Tested Image with 4th model

Figure 8 shows the real-time testing by using webcam. The program was combined with OpenCV to display object in real time app. The eyes were detected and localized by using Mask RCNN model. For example in Figure 5 (a), the eyes were detected with 0.893 predicted for left eye and 0.907 right eye predicted.



(a) (b)

Fig. 8 Mask RCNN eyes detection

4. Conclusion

The eye modelling in this research was trained by using Mask RCNN method. The dataset was created by capturing image directly using webcam. The dataset images were not only captured by webcam, but also downloaded from online dataset. The total image were 150 to perform the eye dataset. The training model was run with 4 epochs and 131 iteration. Finally, the 4th model was selected as the best model. Because it could predict all tested image correctly. The weakness of this research is the GPU capability lower than the minimum requirement of deep learning method.

For future works, this model will apply in high GPU capacity to increase the prediction speed. This model will develop to measure the retina size.

References

[1] N. Mohammad, Z. Omar, E. Supriyanto, A. Dietzel, and J. Haueisen, "Automated detection of microaneurysm for fundus images," IECBES 2016 - IEEE-EMBS Conf. Biomed. Eng. Sci., pp. 600–605, 2017, doi: 10.1109/IECBES.2016.7843520.

[2] R. Senthamizh Selvi, D. Sivakumar, J. S. Sandhya, S. Siva Sowmiya, S. Ramya, and S. Kanaga Suba Raja, "Face recognition using Haar - Cascade classifier for criminal identification," Int. J. Recent Technol. Eng., vol. 7, no. 6, pp. 1871–1876, 2019.

[3] H. Hendrick, "The Halal Logo Classification by Using NVIDIA DIGITS," in International Conference on Applied Information Technology and Innovation (ICAITI), 2018, pp. 162–165.

[4] H. Hendrick, W. Zhi-Hao, C. Hsien-I, C. Pei-Lun, and J. Gwo-Jia, "IOS mobile APP for tuberculosis detection based on chest X-ray image," Proc. ICAITI 2019 - 2nd Int. Conf. Appl. Inf. Technol. Innov. Explor. Futur. Technol. Appl. Inf. Technol. Innov., pp. 122–125, 2019, doi: 10.1109/ICAITI48442.2019.8982152.

[5] G.-J. Jong, Hendrick, Z.-H. Wang, K.-S. Hsieh, and G.-J. Horng, "A Novel Feature Extraction Method an Electronic Nose for Aroma Classification," IEEE Sens. J., vol. PP, no. c, pp. 1–1, 2019, doi: 10.1109/JSEN.2019.2929239.

[6] Hu Xiaoyan, Ji Zhenyu, You Fusheng, Long Keping, and Peng Yunfeng, "Study on Disease Screening and Monitoring System Based on Wireless Communication and IOT," 2012 Spring Congr. Eng. Technol., pp. 1–6, 2012, doi: 10.1109/SCET.2012.6341961.

[7] M. Pamplona Segundo, L. Silva, O. R. P. Bellon, and C. C. Queirolo, "Automatic face segmentation and facial landmark detection in range images," IEEE Trans. Syst. Man, Cybern. Part B Cybern., vol. 40, no. 5, pp. 1319–1330, 2010, doi: 10.1109/TSMCB.2009.2038233.

[8] G. Zheng, X. Wu, Y. Hu, and X. Liu, "Object detection for low-resolution infrared image in land battlefield based on deep learning," Chinese Control Conf. CCC, vol. 2019-July, pp. 8649–8652, 2019, doi: 10.23919/ChiCC.2019.8866344.

[9] J. Redmon and A. Farhadi, "YOLO v.3," Tech Rep., pp. 1–6, 2018, [Online]. Available: https://pjreddie.com/media/files/papers/YOLOv3.pdf.