International Journal of Advanced Smart Convergence Vol.10 No.4 110-116 (2021)
http://dx.doi.org/10.7236/IJASC.2021.10.4.110

IJASC 21-4-14

Real-Time Earlobe Detection System on the Web

Jaeseung Kim[1], Seyun Choi [2], Seunghyun Lee[3] and Soonchul Kwon[4*]

[1] M.S., Department of Plasma Bio Display, Kwangwoon University, South Korea
[2]M.S., Department of Smartsystem, Kwangwoon University, South Korea
[3]Professor, Ingenium College Liberal Arts, Kwangwoon University, South Korea
[4]Associate professor, Graduate School of Smart Convergence, Kwangwoon University, Seoul, Korea*
*{kjs1201, seyunchoi12, shlee, *ksc0226}@kw.ac.kr*

Abstract

This paper proposed a real-time earlobe detection system using deep learning on the web. Existing deep learning-based detection methods often find independent objects such as cars, mugs, cats, and people. We proposed a way to receive an image through the camera of the user device in a web environment and detect the earlobe on the server. First, we took a picture of the user's face with the user's device camera on the web so that the user's ears were visible. After that, we sent the photographed user's face to the server to find the earlobe. Based on the detected results, we printed an earring model on the user's earlobe on the web. We trained an existing YOLO v5 model using a dataset of about 200 that created a bounding box on the earlobe. We estimated the position of the earlobe through a trained deep learning model. Through this process, we proposed a real-time earlobe detection system on the web. The proposed method showed the performance of detecting earlobes in real-time and loading 3D models from the web in real-time.

Keywords: *Computer vision, Deep learning, Image processing, Object detection*

1. Introduction

Due to the recent COVID-19 outbreak, people are spending more time at home than outside. People who practice social distancing are consuming more content than ever before through online services [1]. People have increased the number of online shopping rather than visiting shopping malls. However, it is difficult to know whether jewelry such as necklaces and earrings are suitable for consumers online, and it is impossible to try them on. We thought of a way through augmented reality to allow people to try on the jewelry they want to buy at home without leaving the house to shop. Trying on items that others have tried on during store visits can lead to viral infections. The store also provides this function, so you can use a convenient try-on system. In addition, even when the COVID-19 situation is over, you can do it with a convenient try-on system at home or at a store. We considered a method to detect ear lobes using a deep learning server and output the ear lobes detected through the web in augmented reality. Deep learning technology is being used in many fields, and

Manuscript Received: October. 17, 2021 / Revised: October. 23, 2021 / Accepted: October. 25, 2021
Corresponding Author: ksc0226@kw.ac.kr
Tel: +82-2-940-8637, Fax: +82-50-4174-3258
Graduate School of Smart Convergence, Kwangwoon University, Korea

augmented reality is one of them. Object detection method for classification, real-time object tracking method and semantic segmentation are used when searching for an object [2,3,4]. We used the bounding box search method using YOLOv5. The test participants consisted of 100 males and 100 females, and the left and right earlobes were photographed respectively. We built a server to detect earlobes using a YOLOv5 model trained on our dataset. First, some parts of the user's face images are taken so that the ears could be seen from the user's device camera on the web. Afterwards, the user's face frame taken from the web is transmitted to the server to find the earlobe using deep learning model. Finally, using the earlobe coordinates found on the server, it goes through the process of printing earring models on the web.

2. Background Theory

2.1. YOLOv5

YOLO (You Only Look Once) is a deep learning model that predicts the type and location of an object just by looking at an image [5]. It is a model that integrates individual elements of object detection into a single neural network and utilizes the features of the entire image for prediction [6]. It can also detect multiple images in real time [7,8]. Because real-time detection is possible, it is available to quickly detect multiple objects on the screen while streaming video [9]. The fifth version of this YOLO is YOLOv5.

Figure 1 shows the YOLOv5 structure. Backbone of YOLOv5 was created by integrating CSPNet (Cross Stage Partial Network), which reduces heavy inference time calculations caused by redundant gradient problems, and Darknet, an object recognition open-source neural network [10]. YOLOv5's Neck uses PANet (Path Aggregation Network for Instance Segmentation) to improve object location accuracy. PANet is a model for checking whether an object exists for each pixel of an image based on Mask R-CNN. The head of YOLOv5 has 3 different sizes (18 x 18, 36 x 36, 72 x 72), and it is possible to predict large, medium, and small objects of different sizes using feature maps.

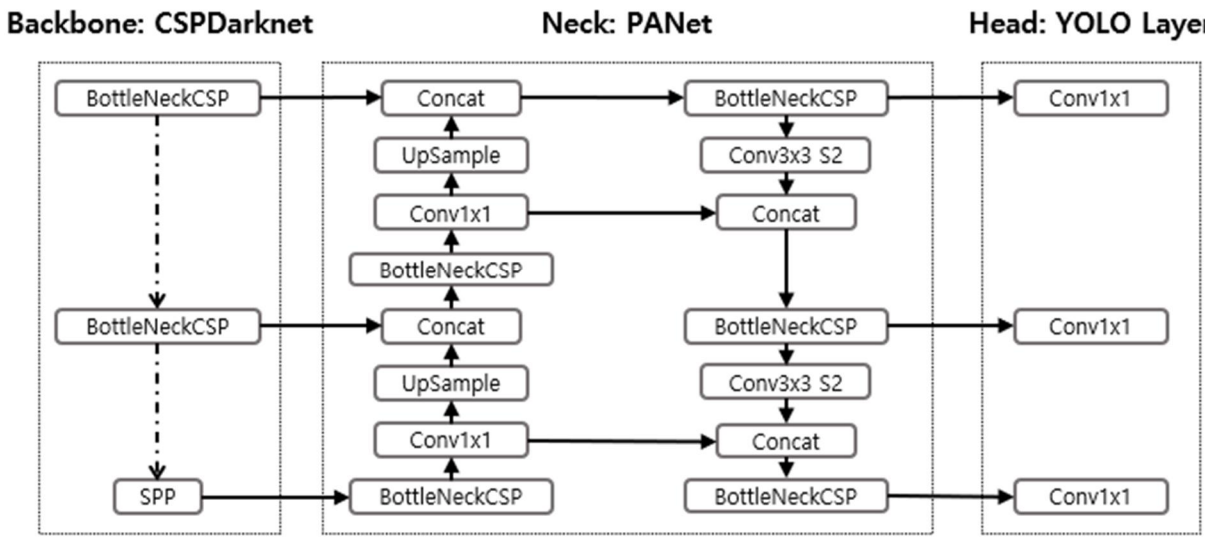

Figure 1. YOLOv5 architecture

Figure 2 shows the YOLOv5n model's structure that we used. YOLOv5-v6.0 contains completely new changes compared to previous versions. Changes in YOLOv5-v6.0 include YOLOv5n6 and YOLOv5n6 keeps the YOLOv5 depth multiplier at 0.33, but reduces the width multiplier for YOLOv5 from 0.50 to 0.25 and

reduces the parameters by 75%. Compared to previous models, YOLOv5n6 has reduced backbone network size and improved performance.

```
# Parameters
nc: 2  # number of classes
depth_multiple: 0.33  # model depth multiple
width_multiple: 0.25  # layer channel multiple
anchors:
  - [10,13, 16,30, 33,23]  # P3/8
  - [30,61, 62,45, 59,119]  # P4/16
  - [116,90, 156,198, 373,326]  # P5/32

# YOLOv5 v6.0 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [64, 6, 2, 2]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, C3, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 6, C3, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, C3, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32
   [-1, 3, C3, [1024]],
   [-1, 1, SPPF, [1024, 5]],  # 9
  ]
```

```
# YOLOv5 v6.0 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 6], 1, Concat, [1]],  # cat backbone P4
   [-1, 3, C3, [512, False]],  # 13

   [-1, 1, Conv, [256, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 4], 1, Concat, [1]],  # cat backbone P3
   [-1, 3, C3, [256, False]],  # 17 (P3/8-small)

   [-1, 1, Conv, [256, 3, 2]],
   [[-1, 14], 1, Concat, [1]],  # cat head P4
   [-1, 3, C3, [512, False]],  # 20 (P4/16-medium)

   [-1, 1, Conv, [512, 3, 2]],
   [[-1, 10], 1, Concat, [1]],  # cat head P5
   [-1, 3, C3, [1024, False]],  # 23 (P5/32-large)

   [[17, 20, 23], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5)
```

Figure 2. YOLOv5n6 structure

3. Proposed Model

Figure 3 shows the development workflow. First, images were obtained from the user's camera through the web. Each user device has a different screen resolution, and accordingly, the web image transmission time may vary for each user. Images are transmitted in real time, and images captured in the user device environment are resized to 224 x 224 in order to transmit 30 images per second. Resizing is performed on the user device, and real-time calculation is possible because the process does not use many resources. The server used the image frame received from the web as the input image of the deep learning model. The deep learning model generated the coordinates of the bounding box by detecting the earlobe in real time. When creating a bounding box by searching for an object, the center coordinates of the box were calculated. Afterwards, the AR object was displayed at the coordinates received from the web.

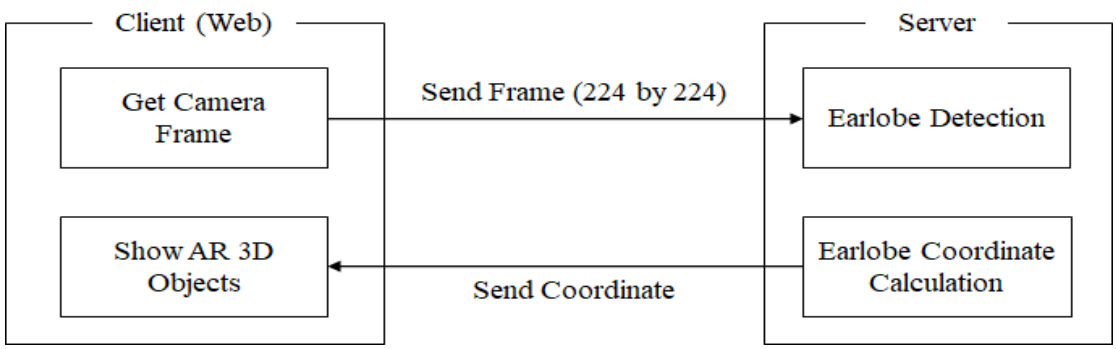

Figure 3. Development workflow

We implemented an online try-on method for earrings. Existing face detection methods are being studied in various ways. The existing method is learning by cropping the ear region from the face image, so the prediction accuracy for the ear from the entire face image is low [11]. Face detection methods do not make predictions about ears by searching only the contours of the face [12]. There is also a method to obtain the ear coordinates using the detected contour lines or coordinates of the face mesh. However, these methods have a low detection rate and cause a positional error when raising or turning the head. We built a dataset to obtain search results for earlobes from full-face photos or cameras. The dataset was constructed by creating an earlobe bounding box on about 200 face images found on the Internet and face photos of acquaintances. When proceeding to the entire ear region, it is difficult to learn because the shape of the pinna is different for each person. Learning only the earlobe region is advantageous for learning the contour features of the earlobe as a rule and shows a high detection rate. Our proposed method acquires the bounding box of the earlobe based on the constructed dataset and computes the center coordinates of the earlobe. We estimated the center point of the earlobe using the inner split point of the upper-left and lower-right coordinates of the bounding box.

4. Experiments and Results

As shown in Table 1 Intel's i7-10700k was used as hardware and Samsung's smart phone and tablet were used. The software used Python 3.7.11 language on Window 10, and OpenCV and PyTorch were used as the main libraries. The size of the input image is 224 pixels width and 224 pixels height.

Table 1. Experimental environments

Category	Enviroment
cpu	Intel i7-10700k
gpu	Nvidia Geforce RTX3090
os	Window 10
language	Python 3.7.11
library	Opencv 4.5.4.58, PyTorch 1.8.2
Device	(Samsung Galaxy) Note 20 Ultra, Tab S6 Lite

We trained the model using the YOLOv5n type in YOLOv5 to get advantages from mobile and CPU usage for fast real-time detection. As shown in Table 2, the model was trained by setting the epoch to 300, the batch size to 5, and the class to the left and right earlobes. We used the constructed dataset to train an earlobe search model and implemented earring try-on using the proposed method.

Table 2. Training hyper parameters

Hyper Parameter	Value
Epoch	**300**
Batch Size	**5**

learning rate	**0.01**
Optimizer	**SGD**

Figure 4 shows the result of creating an AR object using the method we proposed. Figures 4(a) and Figure 4(c) show the result of floating an AR object in the left ear. Figure 4(a) shows the upper part of the ear slightly hidden, and Figure 4(c) shows the face slightly lowered. Figure 4(b) and Figure 4(d) show the result of floating an AR object in the right ear. Figure 4(b) shows a state in which the image was taken at a normal angle and position, and Figure 4(d) shows a state in which the angle of the face is tilted. As shown in Figure 4, we were able to confirm that the AR object was successfully levitated in various states by the proposed method.

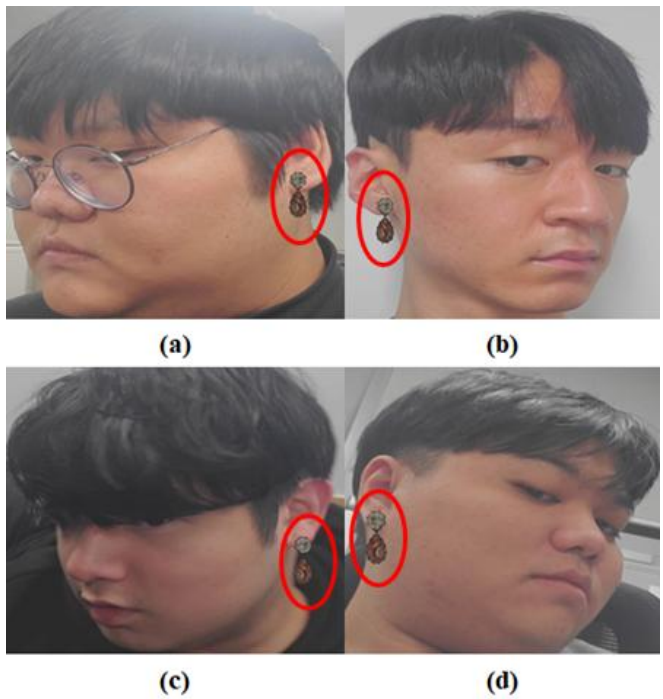

Figure 4. Example of wearing earrings through the proposed method
Figure 4(a): The upper part of the ear slightly hidden, Figure 4(b): Normal angle and position, Figure 4(c): The face slightly lowered, Figure 4(d): A state in which the angle of the face is tilted

As a result of the experiment, our proposed method showed a search result of more than 30 frames per second, showing that real-time earlobe detection is possible. Figure 6 shows the fps data measured to check the real-time. The proposed method showed that users can check in real time whether they fit their ears when they try on it through the web.

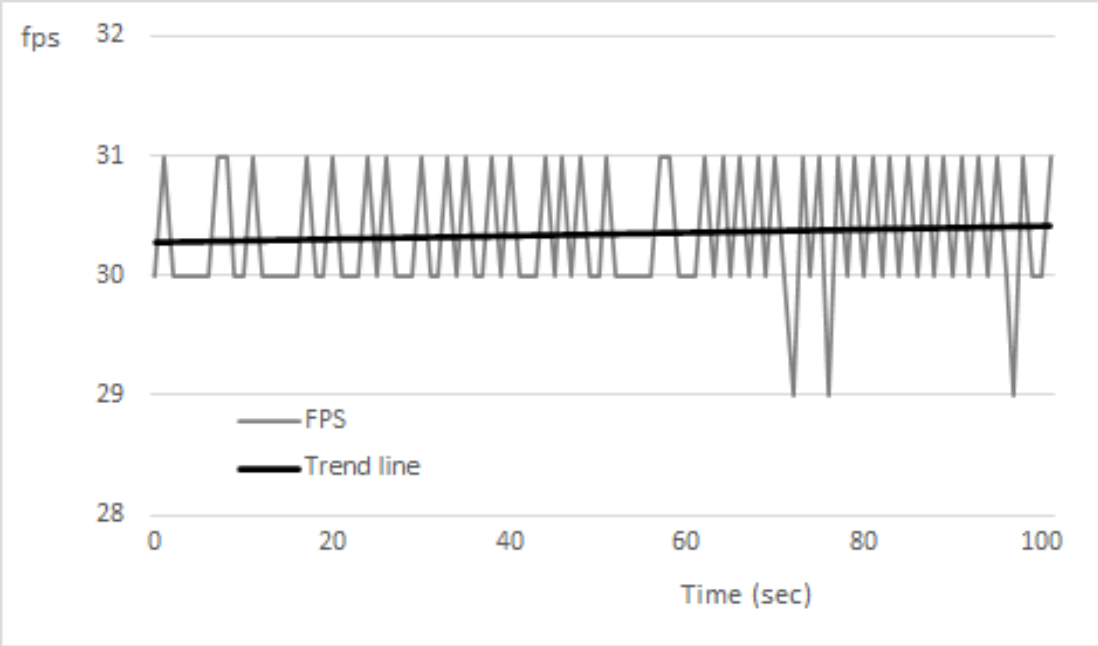

Figure 5. Real-time data images related to detection

5. Conclusion

In this paper, we proposed a real-time earlobe detection system on the web. COVID-19 has significantly activated a variety of virtual/augmented reality services such as virtual concerts and video conferences. With no idea when COVID-19 will end, people are expected to look for a wider variety of online services. The method we have proposed can serve as a virtual try-on service in online shopping. In the proposed method, we were possible to build a dataset for earlobes and detect earlobes in real time between the deep learning server and the client. We adjusted the size of the image sent to the server and receive only the searched coordinate values. Through this, we were reduced to minimize the delay of the amount of transmission between the server and the client. We were able to confirm that the proposed method was successfully detecting earlobes in various angles and positions. We proposed a real-time earlobe detection algorithm on the web using a server. Through our proposed method, it was possible to try on earrings by detecting the earlobe in real-time.

Acknowledgement

This research is supported by Ministry of Culture, Sports and Tourism and Korea Creative Content Agency(Project Number: R2021040083)

References

[1] K. Kim, “Virtual Livestreamed Performance and E-License,” International journal of advanced smart convergence, vol. 9, no. 3, pp. 78–84, Sep. 2020.
DOI: 10.7236/IJASC.2020.9.3.78

[2] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann, "Sub-millisecond neural face detection on mobile gpus," arXiv preprint arXiv:1907.05047, 2019.

[3] P. Chen, Y. Dang, R. Liang, W. Zhu and X. He, "Real-Time Object Tracking on a Drone With Multi-Inertial Sensing Data," in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 1, pp. 131-139, Jan 2018.

DOI: 10.1109/TITS.2017.2750091

[4] S. Xiong, S. Li, L. Kou, W. Guo, Z. Zhou, and Z. Zhao, "Td-VOS: Tracking-Driven Single-Object Video Object Segmentation," 2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC), pp. 102-107, 2020.
DOI: 10.1109/ICIVC50857.2020.9177471

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in IEEE conference, pp.779-788, June 2016.
DOI: 10.1109/CVPR.2016.91

[6] J. Redmon, A. Farhadi, "YOLO9000: better, faster, stronger," in IEEE conference, pp.6517-6525, July 2017.
DOI: 10.1109/CVPR.2017.690

[7] J. Redmon, A. Farhadi, "Yolov3: An incremental improvement," in arXiv, April 2018.

[8] A. Bochkovskiy, Wang Chien-Yao and Liao, and Hong-Yuan Mark, "Yolov4: Optimal speed and accuracy of object detection," in arXiv, April 2020.
DOI: 10.1109/CVPR.2016.91

[9] R. Xu, H. Lin, K. Lu, L. Cao, Y. Liu, "A Forest Fire Detection System Based on Ensemble Learning," in Forests, vol.12, no.2, pp.217, Feb 2021.
DOI: 10.3390/f12020217

[10] C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh , "CSPNet: A new backbone that can enhance learning capability of CNN," in IEEE/CVF conference, pp.1571-1580, June 2020.
DOI: 10.1109/CVPRW50498.2020.00203

[11] Y. Zhou, and S. Zaferiou, “Deformable models of ears in-the-wild for alignment and recognition. In 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition,” pp. 626-633, 2017.

[12] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, “Real-time facial surface geometry from monocular video on mobile GPUs,” arXiv preprint arXiv:1907.06724, 2019.