Journal of KIIT. Vol. 19, No. 12, pp. 105-113, Dec. 31, 2021. pISSN 1598-8619, eISSN 2093-7571 105
http://dx.doi.org/10.14801/jkiit.2021.19.12.105

Online Finger Circumference Measurement System using Semantic Segmentation with Transfer Learning

You-Eun Shin*, Woong-Jin Han**

This research was supported by the MSIT(Ministry of Science, ICT), Korea, under the High-Potential Individuals Global Training Program)(2021-0-01549) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)

Abstract

Previous methods on finger circumference measurement only have a single measurement feature provided in low accuracy. In this paper, we propose a new online finger circumference measurement system that improves both convenience and accurateness which previous methods lack. The measurement system is based on a mobile-optimized deep learning-based segmentation, DeepLabV3-MobileNetV2 pre-trained model with transfer learning, which allows us to get the finger circumference with the appropriate ring size by uploading a picture of one's hand. It is served in the form of a progressive web application that delivers a native app-like user experience on any mobile device on top of high performance and reliability. The experimental results validate the accuracy of our approach surpassing that of the existing method and four novel features provide great convenience to users.

요 약

기존의 손가락 둘레 측정 방식에는 둘레 측정 외에 유용한 기능이 부재하며 측정 방식의 정확도가 낮다. 본 논문에서는 딥러닝 모델을 사용하여 기존 방식의 정확도를 개선하고 부가 기능을 통해 사용자에게 향상된 편의성을 제공하는 새로운 온라인 손가락 둘레 측정 시스템을 구현하였다. 측정 시스템은 모바일 기기에 최적화된 딥러닝 기반의 의미론적 영상 분할 모델을 바탕으로 하며 사전 학습된 모델에 추가 데이터셋으로 전이 학습을 진행하여 손가락 윤곽 인식에 대한 정확도를 개선하였다. 해당 시스템은 웹과 네이티브 앱의 이점을 모두 갖는 프로그레시브 웹 앱으로 개발되어 손가락 둘레 측정 및 관련 기능을 빠른 속도와 높은 보안성, 그리고 모바일 앱의 사용자 경험과 동일한 수준으로 제공하는 것을 주된 특징으로 한다.

Keywords

semantic segmentation, dilated convolution, transfer learning, computer vision, online measurement system

* Dongguk University, Department of English Linguistics, Interpretation and Translation
- ORCID: https://orcid.org/0000-0003-0317-5802

** Dongguk University, Dongguk Institute of Convergence Education
- ORCID: https://orcid.org/0000-0003-1631-8200

· Received: Oct. 29, 2021, Revised: Dec. 10, 2021, Accepted: Dec. 13, 2021·

· Corresponding Author: Woong-Jin Han
Dongguk Institute of Convergence Education, Dongguk University, 30, Pildong-ro 1-gil, Jung-gu, Seoul, Korea
Tel.: +82-2-2290-1491, Email: wjhan@dongguk.edu

Ⅰ. Introduction

There have been various methods to measure the finger circumference for a proper ring size, but those have limitations whether they are about inaccuracy or inconvenience. In this paper, we propose a new method that fulfills both accuracy and convenience. Our system is served in a form of a progressive web application that shows high performance with the latest web technologies. Users can get the appropriate ring size by simply uploading an image of hand due to the semantic segmentation-based deep learning model that detects the hand and gets the outline of it.

We also focused on enhancing the convenience in terms of user experience when developing our web service. To enhance the convenience, we came up with four additional measurement-related features which previous methods lack. Users can get the ring size of any part of their fingers, save the data and check them later on, easily have access to our app via web and get the international size conversion chart of four countries by tapping the screen.

Ⅱ. Related Work

2.1 Online Finger Circumference Measurement System

Finger circumference measurement methods to get appropriate ring size can be broadly categorized into: (i) Traditional ways including paper strip methods and visiting offline stores which are inconvenient. While the latter ensures high accuracy, it is not easy to visit stores every time and it is almost impossible when purchasing rings at online shops; (ii) Applications to measure the size of the ring that one already has which is of no use when one does not have a ring; (iii) Measuring the circumference of a finger is another type of online measurement system[1].

Similar to the second type of online measurement system, we also compute the circumference of fingers. However, we use a semantic segmentation-based deep learning model to get the boundary of fingers, not leaving it users which only increases inaccuracy. In this paper, the comparison and improvements are directly made on a mobile application called PerfectFit by Radius Technologies that sizes the circumference of fingers.

2.2 Semantic Segmentation with Transfer Learning

Semantic segmentation, based on pixel-level image classification, is a fundamental approach in the field of computer vision for scene understanding. Compared to other techniques such as object detection in which no exact shape of the object is known, segmentation exhibits pixel-level classification output providing richer information, including the object’s shape and boundary[2].

Transfer learning is to recognize and apply knowledge and skills learned in previous tasks to novel tasks. In this regard, transfer learning aims to extract the knowledge from one or more source tasks and applies the knowledge to a target task[3]. It uses the knowledge gained while solving one problem (where we have access to a larger dataset) and applies it to a different but related problem[4]. With transfer learning, it is able to increase learning speed since there are fewer new things to learn and the algorithm is faster to generate high-quality output. On top of that, it reduces the amount of data required[5]. Therefore, the performance of one segmentation task can be effectively improved by transferring knowledge from one domain data to another.

Ⅲ. System Implementation

3.1 System Overview

Development environment as seen in Fig. 1, our system uses React for frontend, Nginx for the web server, Gunicorn as Web Server Gateway Interface (WSGI), Uvicorn as Asynchronous Server Gateway Interface (ASGI), FastAPI for the backend API server, and MongoDB for the database. We containerized the application using Docker, got SSL certificate, and deployed our app to Amazon Web Services (AWS).

The app is developed in the form of a progressive web application (Fig. 1(a)). Handling HTTP requests, sending responses and making use of resources are done with FARM Stack (Fig. 1(b)). TensorFlow and OpenCV are used to implement the deep learning model for finger circumference measurement (Fig. 1(c)).

3.2 FARM Stack

FARM stack stands for full web application development stack consisting of FastAPI, React, and MongoDB. Here, we utilized FastAPI as the python web framework paired with React frontend and MongoDB as a NoSQL database.

FastAPI is a modern web framework for the python backend, released in 2018. Because of its remarkably high performance on par with NodeJS and Go, Netflix, Uber, Microsoft amongst many other corporations are using FastAPI[6]. As the name itself has fast in it, it is faster in speed than other major python frameworks such as Flask and Django. This is because Flask is built over WSGI (Web Server Gateway Interface) whereby FastAPI uses Uvicorn as an ASGI (Asynchronous Server Gateway Interface) server[7].

ASGI is the spiritual successor to WSGI and able to achieve high throughput in IO-bound contexts which cannot be handled by WSGI. With the advancement of technology, a lightning-fast ASGI server without routing capabilities called Uvicorn was born. Then Starlette came along, which provides a complete ASGI toolkit on top of Uvicorn[8].

FastAPI is built on top of two libraries, Starlette for the web parts and Pydantic for data validation[9]. Starlette lets us serve a request and return a response asynchronously[10]. Hence, we were able to build a high-performant async IO microservice without the need for additional asynchronous task queues such as RabbitMQ and Celery. And thanks to the full compatibility with the Pydantic library, FastAPI can automatically validate the input data from the external callers at runtime and generates errors to the clients in JSON format when it receives invalid data[6].

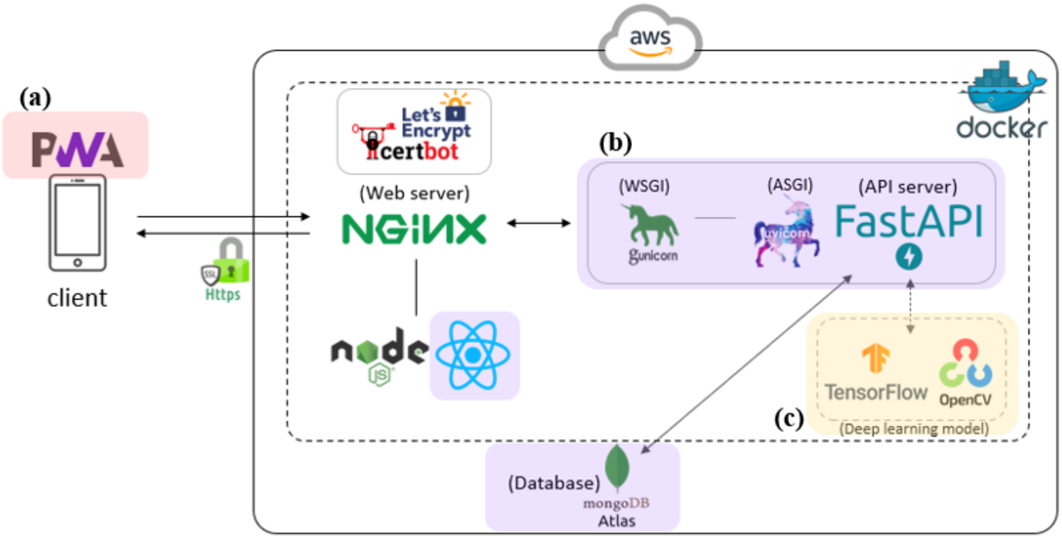


Fig. 1. Diagram of system architecture

As it uses Pydantic known as the fastest validation library (which Flask lacks and 12.3 times faster than Django-rest-framework), objects can be directly passed from the client to the database and the same applies to the other way around[11].

Besides, FastAPI generates data model documentation on the go coming with either Swagger UI or ReDoc. This not only increases the productivity during development working in a team, but also allows developers to test the API endpoints more conveniently[12].

Key features of FastAPI can be summarized as follows:

1. NodeJS-like high performance using Python programming language.
2. Native asynchronous support allowing concurrency and coroutines.
3. Data validation and serialization.
4. Automatically generated interactive API documentation with Swagger UI.

Amongst many other advantages complementing limitations of existing python frameworks, the native support for async IO operations itself was enough to adopt FastAPI as the backend server for our web application.

MongoDB is a document-oriented NoSQL database that stores BSON (Binary JSON) documents. Compared to Relational DataBase Management System (RDBMS), NoSQL (Not only SQL) database is schema-less which is a synonym for the flexible data model[13]. This means a single collection can hold multiple documents and these documents may consist of heterogeneous data[14]. Embedded documents and arrays reduce the need for expensive joins, unlike relational databases[15].

Indexing is of great importance for improving the performances of search queries. Since every field in the documents is indexed with primary and secondary indices, it is easier and takes less time to get or search data from MongoDB[16].

It also supports replication. If the primary server goes down during a transaction, then the secondary server handles the transaction without human interaction[17]. This saves our time for maintenance and makes operations smooth. Indexing and replication enable query responses to be faster, which eventually leads to high performance[18].

High availability and easy scalability should not be left out as these are the main reasons the MongoDB database was developed for. Considering the continuous updates for our service afterwards, easy and horizontal scalability was one of the core features to use MongoDB as a database[19].

Particularly, the fact that both MongoDB and FastAPI work natively with JSON, making a good pair themselves made us use MongoDB as a database backend[20]. We used PyMongo, the official python driver for MongoDB to create its client. As we deployed the web application to Amazon Web Services, we used their cloud hosting service, MongoDB Atlas instead of a local MongoDB database.

3.3 Deep Learning Model and Computer Vision

In this section, we describe the core idea of our measurement method to compute the width of the finger in an image. Deep learning model based on semantic segmentation enables the contour recognition of a hand with high accuracy and the width of the finger is computed in pixels via computer vision.

3.3.1 DeepLabV3-MobileNetV2 Model in TensorFlow

Among many semantic segmentation models including U-Net, FCN, and DeepLab, we decided to use the latest version of DeepLab model for our measurement system because of its high performance [21]. DeepLab is a state-of-art deep learning model for semantic image segmentation, where the goal is to assign semantic to every pixel in an input image[22].

DeepLabV3 adopts atrous convolution in order to overcome the challenges in applying Deep Convolutional Neural Networks (DCNNs). Atrous convolution, also known as dilated convolution, allows us to explicitly control the resolution of features computed by DCNNs without learning extra parameters and extract denser feature maps[23]. By removing the downsampling operations from the last few layers and upsampling the corresponding filter kernels, equivalent to inserting holes (‘trous’ in French) between filter weights, reduced feature resolution problem caused by consecutive pooling operations has been resolved[24].

Since our system is optimized for mobile devices, we implemented MobileNetV2 which is a neural network architecture specifically tailored for mobile and resource constrained environments. Due to the inverted residual structure with linear bottleneck, it significantly decreased the number of operations and memory needed while retaining the same accuracy. Here, the backbone, MobileNetV2 is used as feature extractors of DeepLabV3 model[25].

3.3.2 Width Computation with Computer Vision

To determine the size of an object in an image, a calibration has to be performed using a reference object. The reference object should have two important properties as follows:

1. Dimensions of the object (in terms of width or height) should be known in a measurable unit (such as millimeters, inches, etc.).
2. The reference object in an image should be uniquely identifiable, either based on the placement of the object or via appearances[26].

We used a coin as the reference object and located it in the left-most area in the image. Since the reference object being the left-most object is guaranteed, we sorted the object contours from left to right, grabbed the reference object and used it to define our metric, which we define as:

$$pixels\ per\ metric = \frac{width\ of\ the\ object}{known\ width} \tag{1}$$

3.4 Progressive Web App (PWA)

Progressive web app is a regular web application designed to look and function like a native mobile application[27]. PWA has features of web browsers with advanced enhancement strategies but allows users to have a native app-like experience on any device.

Another advantage about PWA is that it still works under network-disconnected conditions. IndexedDB is a way to persistently store data inside a user's browser. Since it allows us create web applications regardless of network availability, our application can work both online and offline[28]. When a web app makes an HTTP request for some data and the user doesn’t have internet connectivity, we can still serve this data from the local database cache[29].

Ⅳ. Experiments

4.1 Experiments with WebXR

WebXR Device API developed by W3C group implements the core of the WebXR feature set, renders the 3D scene to the chosen device at the appropriate frame rate, and manages motion vectors created using input controllers[30]. As shown in Fig. 2, it was impossible to get measurements in millimeters.

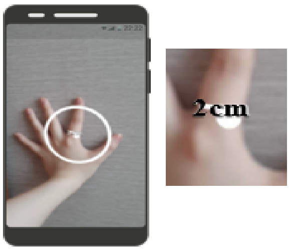


Fig. 2. Result of computing the length using WebXR

Since the ring size is differentiated by a millimeter, we decided not to implement WebXR on our system.

4.2 Semantic Segmentation with Transfer Learning

In this section, we describe our approach to measure the finger circumference by implementing a pre-trained semantic segmentation model called DeepLabV3-MobileNetV2 and train it with hand datasets through transfer learning. Our goal is to measure the width of a finger in an image, compute the actual width, and convert it into circumference.

4.2.1 Transfer Learning Process

Transfer learning is an effective way to customize the pre-trained model according to the purpose of use. For the transfer learning, we gathered the hand gesture recognition dataset from Kaggle beforehand. Then, we labeled the training dataset using OpenCV and converted it into TFRecord format to increase the learning rate and reduce overhead while reading the data. After the transfer learning with 1500 training datasets and 8000 epochs, there has been a remarkable improvement in segmentation mask containing the outline of the hand as shown in Fig. 3.

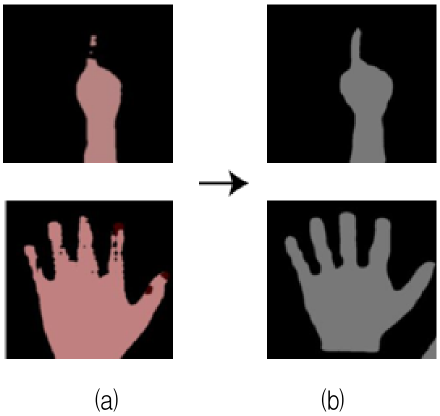

(a) (b)

Fig. 3. Result of segmentation mask via DeepLabV3-MobileNetV2 pre-trained model (a) and the model after transfer learning (b)

4.2.2 Finger Circumference Measurement

As seen in Fig. 4, a segmentation map of 513×513 matrix is the output of our model. The Segmentation map here tells which pixels are in the class of hand and which are not. Through the (x, y)-coordinates sent from the client side, the semantic image segmentation model gets the two endpoints of the finger to get measured. Then it computes the pixel-width of the finger in the image.

To convert the pixel-width into the actual width, we have adopted a 2.2×2.2cm size 100-won coin as the reference object. The formula called pixels per metric mentioned in Section 3.3 is applied at this point, where the numerator, the width of the object becomes the pixel-width and the denominator, the known width becomes 2.2cm.

The pixel-width between two endpoints of the finger is divided by the width of the 100-won coin, the actual length is computed as seen in Fig. 5. Lastly, the width of the finger is computed into the circumference based on the ring size chart.

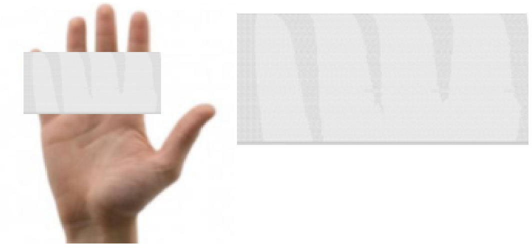


Fig. 4. Segmentation map as the output of the model

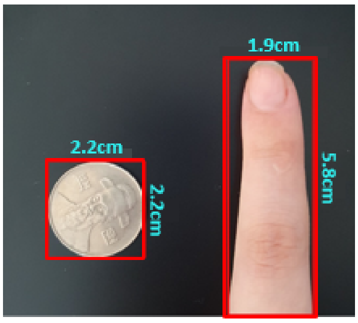


Fig. 5. Actual length computation using a 100-won coin as the reference object

4.3 Measurement Procedure and Main Features

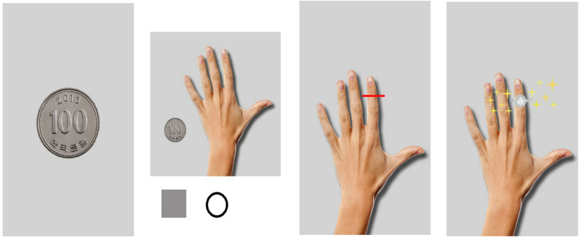
Users can easily access our web application by searching‘ringfit’in web browsers. Followings are written in the order of actual service flow.

As soon as users are on our website, they are informed of how to use the service in 4 steps as seen in Fig. 6. Users have to prepare a 100-won coin as the reference object for the first step. Next, they may upload a photo of a hand with the coin placed on the left side of their hand.

Once the image is successfully uploaded, users can choose which part of the finger to get measured by adjusting the red line on any part of their finger marked as the third step in Fig. 6. Soon, a proper ring size with finger circumference is provided.

At this point, users can easily find out the international ring size conversion chart retrieved from the database in the order of Korea, US, Italy, and the United Kingdom by tapping the screen as shown in Fig. 7.

Users can save the results in indexedDB with extra information consisting of which hand, which finger, and which part of the finger as shown in Fig. 8.



1 step 2 step 3 step 4 step

Fig. 6. Users’ guide in 4 steps

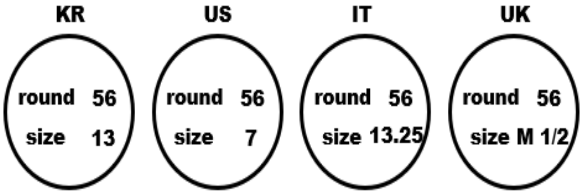


Fig. 7. International ring size conversion chart in the order of KR, US, IT, and the UK

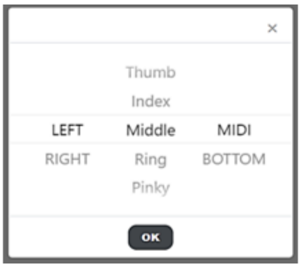
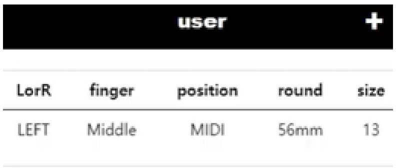


Fig. 8. Users can save the results in indexedDB



user +

LorR	finger	position	round	size
LEFT	Middle	MIDI	56mm	13

Fig. 9. Saved results can be checked even under network-disconnected conditions

Saved data can be checked anytime, anywhere, even when the devices are not connected to networks. Fig. 9 is an example of a user whose left middle finger fits size 13 with a circumference of 56mm.

4.4 Results

The comparison of measurement results between PerfectFit[1] and our proposed method is made in millimeters. We have made experiments on the cases of 51mm, 53mm and 55mm as seen in Table 1.

PerfectFit not only showed low accuracy but also computed certain values particularly around 45mm. Through semantic segmentation model and computer vision, we were able to come up with much more reliable results and reduce the error from 7.7mm to 2.3mm on average. Accuracy calculated in percentage also showed improvement from 85.6% to 97.5% on average.

Table 1. Comparison of the previous method and ours

Circumference (mm)	PerfectFit[1]	Proposed method
51	45	56
53	45	53
55	46	57
Error (mm)	7.7	2.3
Accuracy (%)	85.6	97.5

Ⅴ. Conclusion and Future Work

In this paper, we first proposed a novel online finger circumference measurement system based on a semantic segmentation model. Our method has improved the accuracy of the measurement system using semantic segmentation with transfer learning.

Various inconveniences that previous methods had also have been resolved by four additional features. Allowed data access even under network-disconnected conditions and measurement available on any part of fingers are some of them.

As explained in this paper, we have introduced the reference object for higher accuracy due to the limitations of current AR technology. In the future, improvements such as providing various options for reference objects or coming up with new methods without using reference objects can be made.

References

[1] Radius Technologies, https://radiustechnologiesinc.com/. [accessed: Aug. 26, 2021]

[2] S. Sharma, J. E. Ball, B. Tang, D. W. Carruth, M. Doude, and M. A. Islam, "Semantic Segmentation with Transfer Learning for Off-Road Autonomous Driving", Sensors, Vol. 19, No. 11, Jun, 2019. https://doi.org/10.3390/s19112577.

[3] S. J. Pan and Q. Yang, "A Survey on Transfer Learning", IEEE, Vol. 22, No. 10, pp. 1345-1359, Oct, 2010. https://doi.org/10.1109/TKDE.2009.191.

[4] M. Imad, O. Doukhi, and D. J. Lee, "Transfer Learning Based Semantic Segmentation for 3D Object Detection from Point Cloud", Sensors, Vol. 21, No. 12, Jun. 2021. https://doi.org/10.3390/s21123964.

[5] A. Goergen, https://levity.ai/blog/what-is-transfer-learning. [accessed: Sep. 19, 2021]

[6] F. Malik, https://towardsdatascience.com/build-and-host-fast-data-science-applications-using-fastapi-823be8a1d6a0. [accessed: Aug. 16, 2021]

[7] K. Gupta, https://www.analyticsvidhya.com/blog/2020/11/fastapi-the-right-replacement-for-flask/. [accessed: Aug. 16, 2021]

[8] N. Foong, https://betterprogramming.pub/migrate-from-flask-to-fastapi-smoothly-cc4c6c255397. [accessed: Jul. 17, 2021]

[9] FastAPI, https://fastapi.tiangolo.com/. [accessed: Jul. 14, 2021]

[10] Starlette, https://www.starlette.io/. [accessed: Aug. 13, 2021]

[11] Pydantic, https://pydantic-docs.helpmanual.io/benchmarks/. [accessed: Aug. 13, 2021]

[12] S. Lynn, https://towardsdatascience.com/understanding-flask-vs-fastapi-web-framework-fe12bb58ee75. [accessed: Aug. 16, 2021]

[13] A. Bassett, https://www.mongodb.com/developer/quickstart/python-quickstart-fastapi/. [accessed: Jul. 17, 2021]

[14] A. Saini, https://www.geeksforgeeks.org/what-is-mongodb-working-and-features/. [accessed: Aug. 10, 2021]

[15] P. Pedemkar, https://www.educba.com/what-is-mongodb/?source=leftnav. [accessed: Aug. 10, 2021]

[16] Data-flair.training, https://data-flair.training/blogs/mongodb-features/. [accessed: Aug. 10, 2021]

[17] P. Pedemkar, https://www.educba.com/advantages-of-mongodb/?source=leftnav. [accessed: Aug. 10, 2021]

[18] Data-flair.training, https://data-flair.training/blogs/advantages-of-mongodb/. [accessed: Aug. 10, 2021]

[19] MongoDB, https://docs.mongodb.com/manual/introduction/. [accessed: Jul. 17, 2021]

[20] E. Cerami, https://medium.com/fastapi-tutorials/integrating-fastapi-and-mongodb-8ef4f2ca68ad. [accessed: Jul. 17, 2021]

[21] I. Ahmed, M. Ahmad, F. A. Khan, and M. Asif, "Comparison of Deep-Learning-Based Segmentation Models: Using Top View Person Images", IEEE, Vol. 8, pp. 136361-136373, Jul. 2020. https://doi.org/10.1109/ACCESS.2020.3011406.

[22] H. Yu and C. Chen, https://github.com/tensorflow/models/blob/master/research/deeplab/README.md. [accessed: Aug. 17, 2021]

[23] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation", ECCV, Munich, Germany, pp. 833-851, Sep. 2018.

[24] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation", arXiV. 2017.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", The IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Salt Lake City, UT, USA, pp. 4510-4520, Jan. 2018. https://doi.org/10.1109/CVPR.2018.00474.

[26] A. Rosebrock, https://www.pyimagesearch.com/2016/03/28/measuring-size-of-objects-in-an-image-with-opencv/. [accessed: Aug. 17, 2021]

[27] N. McKenna, https://www.mckennaconsultants.com/progressive-web-apps-explained/. [accessed: Aug. 12, 2021]

[28] Mozilla, https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Using_IndexedDB. [accessed: Sep. 20, 2021]

[29] U. Hiwarale, https://medium.com/jspoint/indexeddb-your-second-step-towards-progressive-web-apps-pwa-dcbcd6cc2076. [accessed: Sep. 10, 2021]

[30] Mozilla, https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API. [accessed: Sep. 18, 2021]

Authors

You-Eun Shin

2018 ~ present : Undergraduate student in the Department of English Linguistics, Interpretation and Translation, Dongguk University

Research interests : Image Processing, Computer Vision, Natural Language Processing

Woong-Jin Han

1991 : B.S. degrees in Electrical Engineering at Yonsei University

1995 : M.S. degrees in Electrical Engineering at Yonsei University

2018 ~ present : Professor at Dongguk Institute of Convergence Education, Dongguk University

Research interests : artificial intelligence and image processing