

# Line Drawing Mobile App using Hybrid Approach

Yujung Gil\*, Kiho Kim\*\*, and Kwangil Kim\*\*\*

---

“This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW supervised by the IITP(Institute of Information & communications Technology Planning & evaluation)”  
(2016-0-00017)

---

## Abstract

In recent years, image-to-image translation(I2I) has drawn much attention and shown noticeable success in many areas such as image synthesis, segmentation, style transfer, restoration, and pose estimation. Among these applications, the line drawing model, which is a subfield of style transfer, transforms any portrait picture into a line drawing. In these days, line drawing apps are very popular among young people. The most common feature of the current model is to delicately display all lines as if sketched with a pencil. Because of this, not only are the results not natural, but they are also less aesthetically interesting. In this paper, we design and implement a light and fast hybrid line drawing model which combines deep neural networks and traditional edge detection filter. We show that proposed model is efficient and outputs more aesthetically pleasing results.

## 요약

최근 수년간, 이미지-이미지 변환(I2I) 기법이 많은 주목을 받았으며 이미지 합성, 이미지 분할, 스타일 전이, 복원 및 자세 추정 등 많은 분야에서 괄목할 만한 성공을 거두고 있다. 이러한 응용 분야 중에서, 인물사진을 라인 드로잉(Line drawing)으로 변환해주는 스타일 전이 모델의 한 분야인 라인 드로잉 모바일 앱이 젊은이들 사이에 인기가 높다. 현존 모델들의 가장 일반적인 특징은 모든 선들을 마치 연필로 스케치한 것처럼 섬세하게 모두 표현하는 것이다. 이 요인으로 인해 결과물이 자연스럽지 못하고 미적인 흥미를 떨어트린다. 이 논문에서는 심층신경망과 전통적인 OpenCV의 이미지 처리 함수들을 조합한 가볍고 빠른 하이브리드한 라인 드로잉 모델을 제안한다. 제안된 모델이 효율적이며 미적인 결과를 산출한다는 것을 보여준다.

## Keywords

image-to-image translation, style transfer, line drawing, deep neural networks, image segmentation, edge detection

---

\* Department of Computer Science and Engineering,  
Dongguk University, Seoul

- ORCID: <https://orcid.org/0000-0002-6139-9831>

\*\* Department of Computer Engineering, Korea  
Polytechnic University, Korea

- ORCID: <https://orcid.org/0000-0001-6512-642X>

\*\*\* Dongguk Institute of Convergence Education, Dongguk  
University, Seoul

- ORCID: <https://orcid.org/0000-0002-9371-7873>

• Received: Oct. 28, 2021, Revised: Nov. 17, 2021, Accepted: Nov. 20, 2021

• Corresponding Author: Kwangil Kim

Dongguk Institute of Convergence Education, Dongguk University

Tel.: +82-2-2290-1404, Email: [kikim6114@dongguk.edu](mailto:kikim6114@dongguk.edu)

## I. Introduction

In recent years, image-to-image translation(I2I) has shown noticeable success in many areas such as image synthesis, segmentation, style transfer, restoration, pose estimation, etc.[1]. Among these, line drawing art is popular among people in their teens and 20s. In addition to the school community and commercial sites such as "idus.com"[2], you can often check the demand for line drawing images. Line drawing images are simple illustrations that draw the outline and color of the main objects in the photographic image. However, the line drawing usually needs expensive devices such as iPads and some proficiency in image manipulation software such as Photoshop. Hence the quality of a line drawing image heavily depends on the level of craftsmanship of the worker and the working environment.

As the need for the line drawing mobile app grows, several image-to-sketch apps have been introduced. Image-to-sketch is different from line drawing in that it is focused on the capability to represent all the lines in as much in detail as if sketched with a pencil. As there is a saying that less is more, moderation is also needed in paintings like line drawing. In one word, though current sketch-to-image apps are producing fine pencil sketch images, they lack naturalness and aesthetical interest.

To make the app more attractive, we need a line drawing model which can omit details and draw lines somewhat roughly and naturally. In addition, all processes must be processed automatically and instantly without additional manual works.

In this paper, we propose a lightweight and fast line drawing app (See Figure 1) which is capable of automatically producing more natural and concise line drawing by using deep neural networks and a conventional edge detection filter. The proposed app has the following main features:

(1) Detailed lines are expressed concisely in a way

that boldly omits or leaves only the outer outline. In other words, the result becomes a fairly simplified picture.

(2) The characters and the background can be completely separated and the background can be removed or left as the original picture.

First, for these goals, we need the pre-trained image segmentation model that is precise and is particularly smooth at the image boundary. To this end, our model uses U2-Net[3]. Second, we need an algorithm that detects edges from the separated portrait image and can be fine-tuned for adjusting the omitted range of the detailed lines. For this purpose, a bilateral edge detection filter[4] is used in this paper.

This paper is organized as follows: Section 2 examines the various techniques used in other apps. Section 3 describes the procedure we've taken for selecting models and techniques. Section 4 describes the implementation process and systems environment. Section 5 describes the conclusion of this study and future research direction.

## II. Related Works

In computer vision, edge detection is very useful for improving various tasks such as semantic segmentation, object recognition, stereo, and object proposal generation. While classical edge detection filters such as Sobel detector[5], Canny detector[6], and Bilateral filter[4] are still used in many fields, deep learning based edge detection methods have been actively studied and shows remarkable advancement.

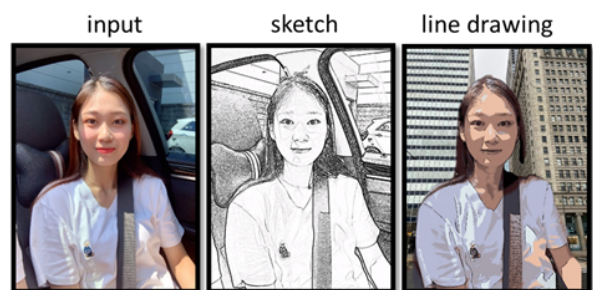


Fig. 1. Output of sketch app and line drawing app

Image-to-Image translation(I2I) is the task of taking images from one domain and transforming them so they have the style (or characteristics) of images from another domain. In recent years, I2I has drawn much attention and shown noticeable success in many areas such as image synthesis, segmentation, style transfer, restoration, and pose estimation.

With the popularity of convolutional neural networks in computer vision, many CNN-based I2I and edge detection models were introduced. Some examples are DeepEdge[7], HED[8], RCF[9], and CASENet[10].

Generative Adversarial Network(GAN)[11] is widely used as an unsupervised learning method for implementing I2I. GAN consists of two networks. One is the generator and the other is the discriminator. They learn in co-evolutional way. More specifically, the generator tries to fool the discriminator by generating real-looking images while the discriminator tries to distinguish between real and fake images. After training, the generator can produce images that are very close to the real images in the dataset.

Pix2pix[12] is the first GAN-based I2I model which uses conditional GAN (cGAN) to solve wide range of supervised I2I tasks. Pix2pix inspires many I2I works and is used as a strong baseline model for I2I. However, Pix2pix has limitations that it needs paired images for translation. And the result of pix2pix is not good when an input is unusual or sparse.

CycleGAN[13] tries to learn a mapping for I2I translation without requiring paired input-output images, using cycle-consistent adversarial networks. For this purpose, CycleGAN uses loss, which only transforms the style of the image to the extent it can be restored to the original image. Also, one major difference between the pix2pix and the CycleGAN is that unlike the pix2pix which consists of only two networks (Discriminator and Generator), the CycleGAN consists of four networks (two Discriminators and two Generators).

DiscoGAN[14] is much similar to CycleGAN. The

main difference between DiscoGAN and CycleGAN is that the former uses two reconstruction loss, one for both the domain whereas the latter uses single cycle-consistency loss.

Although there are many models that can be used for segmentation and edge detection, the most of them typically require large datasets or are too heavy to run on mobile phone.

In addition to the methods reviewed above, there are many models for segmentation and edge detection, but most models have disadvantages such as requiring large-scale training datasets, taking a lot of time for training, or being too heavy to run on mobile devices. Therefore, it has been concluded that it is reasonable to utilize a pre-trained light model and traditional computer vision tool together so that users instantly view and enjoy the results on mobile devices.

### III. Model

When a human performs manual line-drawing on an image, he/she usually keep a mental map of both segmentation and delineation. Inspired by this idea, we design a new model which combines these two approaches.

For the segmentation task, we use pre-trained U2-Net[3] which is inspired by U-Net[15]. Original U-Net is convolutional network architecture for fast and precise segmentation of biomedical images. In U-Net, CNN is followed by a symmetric deconvolutional network. While CNN part does the usual convolution to discover the important features, the deconvolutional part does an up-sampling. U-Net shows excellent performance in image segmentation even when we have small amount of examples by utilizing the data augmentation. U2-Net has a two-level nested U-structure (See Figure 2). This design has the following advantages: (1) due to Residual U-blocks(RSU), it is able to capture more contextual information from different scales, (2) it increases the depth of the whole architecture without significantly increasing the

computational cost because of the pooling operations used in RSU blocks.

Architecture of RSU (See Figure 3) is similar to that of U-Net. RSU is a building block of U2-Net, and consists of three main components:

- (1) an input convolution layer
- (2) U-Net like symmetric encoder-decoder
- (3) a residual connection

RSU block is in itself a deep neural network which resembles U-Net with different residual connection.

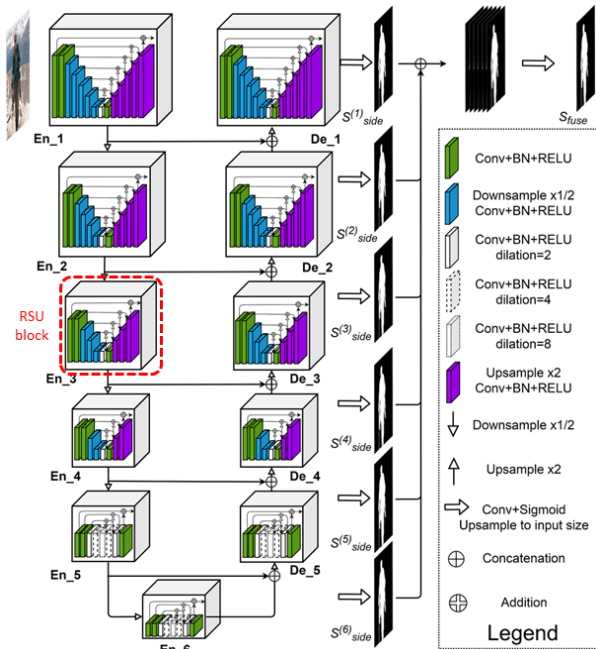


Fig. 2. Overview of U2-Net Architecture (source: [3])

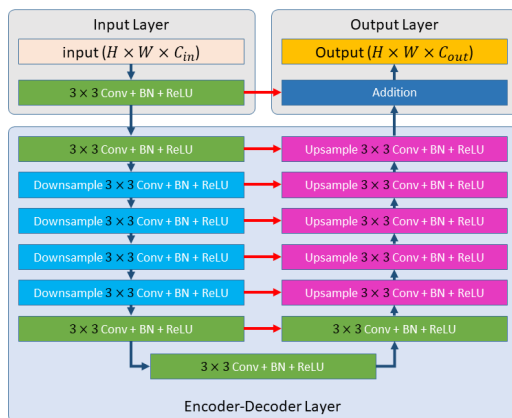


Fig. 3. Architecture of RSU block (source: [3])

This structure makes RSU block extract intra-stage multi-scale features without degrading the feature map resolution. In fact, the image segmentation output of U2-Net has more smooth and noiseless boundaries than U-Net (See Figure 4).

For the delineation task, a number of CNN-based model can be considered[7]-[10]. However, since the common goal of these models is to predict contours in detail, it is not suitable for line drawing (See Figure 5).

Furthermore, if we looking into the ground truth image of the dataset[16]-[18], we can easily find the fact that they were made to learn outline prediction in supervised way.



Fig. 4. Background-removed outputs from U-Net [15] and U2-Net [3]. Outputs from U2-Net show smoother and noiseless boundaries

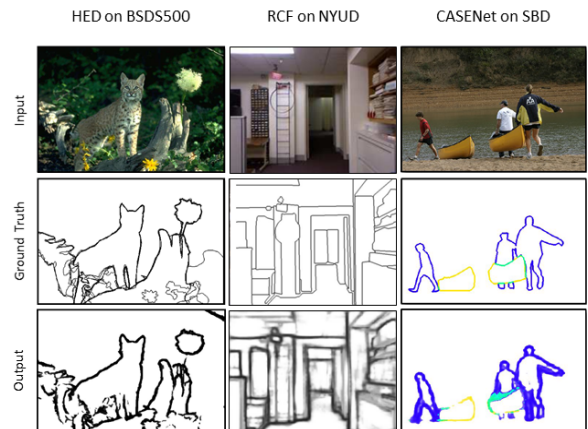


Fig. 5. Outputs from CNN-based edge detection models

In order to verify whether the output of general I2I models are suitable to be used line drawing application, we examined a number of CNN-based models listed in Section 2. Among those models, we show a sample output of a CNN-based style transfer model[19] in Figure 6. As we can see in this example, when we use a style transfer model, it is hard to choose a suitable line drawing image as the reference image. This is because the color of the person's clothes in the line drawing output is severely affected by the reference image.

As an alternative, U-GAT-IT[20], a GAN-based model is considered. U-GAT-IT model enables image-to-image translation using the new attention module and normalization function, which allows the model to identify which parts will be strongly transformed while distinguishing the difference between source image and target image. The result of experiment indicates that the look and feel of the output image is not the one that we want. Furthermore, because U-GAT-IT requires quite a huge amount of line drawing data, it is not suitable for our model.

Due to above reasons, we get to the conclusion that if we combine the traditional computer vision techniques for edge detection and image filtering with deep neural network model, it can be one of a good trade-off. Using OpenCV[21] functions, images are processed as the following steps:

- (1) Convert the image to grayscale.
- (2) To emphasize the contour lines, extract boundary values by using adaptiveThreshold() function.
- (3) To make colors uniform, apply the bilateral filter bilateralFilter() to the image. Here we set both values of sigmaColor and sigmaSpace to 250 which make the noise of the image increase. Note that we typically use bilateralFilter to decrease the noise.
- (4) Superimpose the image to the original image.

The resulting inputs and outputs of our model are shown in Figure 7.



Fig. 6. Result from a CNN-based style transfer model [19]

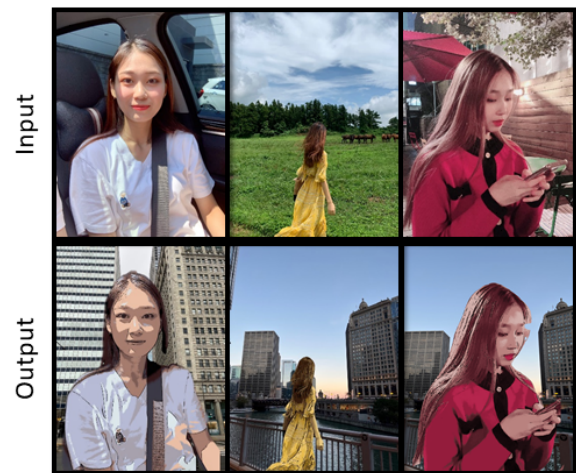


Fig. 7. Input images and output line drawings

#### IV. Implementation

We use React Native[22] to unify the development environment so that the mobile applications can be used on both iOS and Android (See Figure 8). First, the React Native app on the client side sends the original image through the REST API of axios. Images are delivered to Flask[23] via Gunicorn[24].

Then, we send the original image to the AI model that we put in the Flask to remove the background, extract the boundary value with the threshold technique, and convert the image to the hand painting style with OpenCV bilateral filter. This stores the line drawing image obtained by the style conversion to Google Cloud Storage (GCS). The overall system flow proceeds by storing the link of the image stored in GCS and the user information from the client side to Flask in MySQL DB. The user interfaces and screen shots of our app in shown in Figure 9.

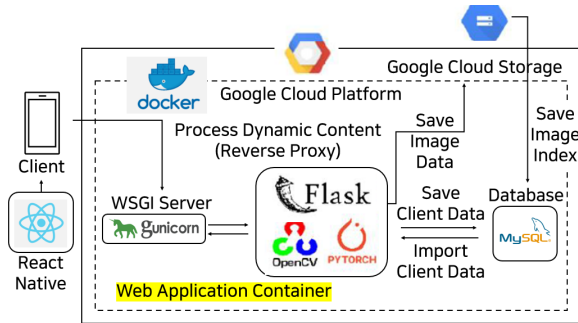


Fig. 8. This is the overall system architecture in which the mobile application operates. The database used MySQL, the web framework used flask, and the generic WSGI server used. google cloud storage was placed to store image data, and docker was used to match the development environment

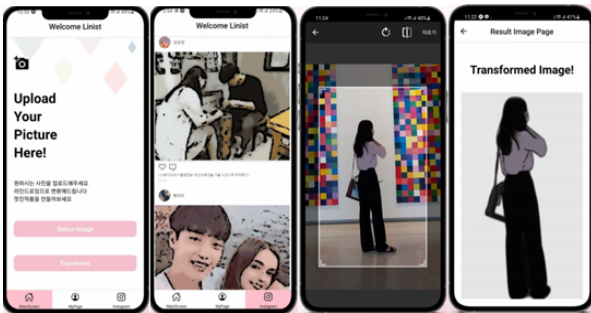


Fig. 9. User interfaces and outputs of our app

## V. Conclusions

We cannot easily get line drawings simply by applying image segmentation and edge detection to target images. Image-to-sketch models focus on expressing delicate pencil touches well. Therefore, it is difficult to use the output of the image-to-sketch model as an input of the line drawing model. CNN-based or GAN-based style transfer models can perform segmentation and edge detection at the same time, but due to the nature of deep neural networks, it is difficult to control the omission and naturalness required by artistic line drawing. In this study, a hybrid method combining deep neural networks and traditional computer vision techniques is used.

For the future study, developing CNN-based edge detection algorithms that can control the properties of lines with only small-scale line drawing training data,

or develop GAN-based style transfer algorithms that can learn the style of lines rather than colors may be a new challenge.

## References

- [1] Y. Yingxue, J. Lin, T. Qin, and Z. Chen, "Image-to-Image Translation: Methods and Applications", arXiv preprint arXiv:2101.08629, Jul. 2021.
- [2] <https://www.idus.com/w/main/category/e3ba45b3-c165-11e3-8788-06fd000000c2> [accessed: Oct. 27, 2021]
- [3] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-Net: Going Deeper with Nested U-structure for Salient Object Detection", *Pattern Recognition*, Vol. 106, pp. 107-404, Aug. 2020. <https://doi.org/10.1016/j.patcog.2020.107404>.
- [4] S. Sylvain, P. Kornprobst, J. Tumblin, and F. Durand, "Bilateral filtering: Theory and applications. Now Publishers Inc", *Foundations and Trends® in Computer Graphics and Vision*, Vol. 4, No. 1, pp. 1-73, Aug. 2009.
- [5] K. Josef, "On the Accuracy of the Sobel Edge Detector", *Image and Vision Computing*, Vol. 1, No. 1, pp. 37-42, Feb. 1983. [https://doi.org/10.1016/0262-8856\(83\)90006-9](https://doi.org/10.1016/0262-8856(83)90006-9).
- [6] J. Canny, "A Computational Approach to Edge Detection", *Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, pp. 679-698, Nov. 1986. <https://doi.org/10.1109/TPAMI.1986.4767851>.
- [7] G. Bertasius, J. Shi, and L. Torresani, "Deepedge: A multi-scale bifurcated deep network for top-down contour detection", *Proceedings of the IEEE conference on computer vision and pattern recognition*, Vol. 3, pp. 4380-4389, Apr. 2015. <https://doi.org/10.1109/CVPR.2015.7299067>.
- [8] S. Xie and Z. Tu, "Holistically-nested Edge Detection", *Proceedings of the IEEE conference on computer vision and pattern recognition*, Santiago,

- Chile, Vol. 2, pp. 1395-1403, Oct. 2015. <https://doi.org/10.1109/ICCV.2015.164>.
- [9] Y. Liu, M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer Convolutional Features for Edge Detection", Proceedings of the IEEE conference on computer vision and pattern recognition, Vol. 3, pp. 3000-3009, Apr. 2017.
- [10] Z. Yu, C. Feng, M. Liu, and S. Ramalingam, "CASENet: Deep Category-Aware Semantic Edge Detection", Proceeding of the IEEE conference on computer vision and pattern recognition, Vol. 1, pp. 5964-5973, May. 2017.
- [11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks", arXiv preprint arXiv:1406.2661, Jun. 2014.
- [12] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks", Proceeding of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, Vol. 3, pp. 1125-1134, Jul. 2017. <http://dx.doi.org/10.1109/CVPR.2017.632>.
- [13] J. Zhu, T. Park, P. Isola, and A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", Proceedings of the IEEE international conference on computer vision, Venice, Italia Vol. 1, pp. 2223-2232, Mar. 2017.
- [14] T. Kim, M. Cha, H. Kim, J. Lee, and J. Kim, "Learning to Discover Cross-Domain Relations with Generative Adversarial Networks", International Conference on Machine Learning, International Convention Centre Sydney, Vol. 2, pp. 1857-1865, May 2017.
- [15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional Networks for Biomedical Image Segmentation", International Conference on Medical image computing and computer-assisted intervention, Munich Germany, Vol. 1, pp. 234-241, Nov. 2015.
- [16] D. Martin, C. Fowlkes, and J. Malik, "Learning to Detect Natural Image Boundaries using Local Brightness, Color, and Texture Cues", IEEE transactions on pattern analysis and machine intelligence, Vol. 26, No. 5, pp. 530-549, May 2004. <https://doi.org/10.1109/TPAMI.2004.1273918>.
- [17] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual Organization and Recognition of Indoor Scenes from RGB-D Images", 2013 IEEE Conference on Computer Vision and Pattern Recognition, Vol. 26, pp. 564-571, Oct. 2013.
- [18] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic Contours from Inverse Detectors", 2011 International Conference on Computer Vision, pp. 991-998, Nov. 2011.
- [19] L. Gatys, A. Ecker, and M. Bethge, "Image Style Transfer using Convolutional Neural Networks", 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2414-2422, Dec. 2016.
- [20] J. Kim, M. Kim, H. Kang, and K. Lee, "U-GAT-IT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization for Image-to-Image Translation", arXiv preprint arXiv:1907.10830, Vol. 4, Apr. 2020.
- [21] OpenCV, <https://opencv.org/> [accesssed: Jul. 15, 2021]
- [22] React Native, (<https://reactnative.dev/>) [accesssed: Jul. 15, 2021]
- [23] Flask, <https://flask.palletsprojects.com/en/2.0.x/> [accesssed: Jul. 15, 2021].
- [24] Unicorn, <https://unicorn.org/#docs> [accesssed: Jul. 15, 2021]

Authors

Yujung Gil



2019 ~ present : Undergraduate student in the Department of Computer Science and Engineering, Dongguk University

Research interests : Research Interests: Image Processing, Big

Data, Database

Kiho Kim



2018 ~ present : Undergraduate student in the Department of Computer Engineering, Korea Polytechnic University

Research interests : Deep Learning, Data Processing, Big Data

Kwangil Kim



1981 : B.S degree in Department of Industrial Engineering, Hanyang Univerisity  
1983 : M.S degree in Department of Management Science, KAIST  
2017 ~ present : Professor in Department of Convergence

Education, Dongguk Univerisity

Research interests: Artificial Intelligence, Data Science