Comparison of Kidney Segmentation Under Attention U-Net Architectures

Marcia Hon, Vasileios Alevizos,
Ryerson University, University of Aegean

***Abstract*—One of the most prominent machine learning advantages in the medical industry is the early detection of disease. Automatic kidney detection is of great importance for rapid diagnosis and treatment, where related diseases occupy over 73,750 new cases in the US in 2020 [1]. Today, the performance of diagnosis has been by highly trained radiologists. However, the complex structures contribute to speckle noise and inhomogeneous intensity profiles. Thus, there is a necessity to automate segmentation on kidney ultrasounds using U-Net Deep Learning architectures - an innovative solution for Medical Imaging Analysis. In this research, our focus is on the comparison of Attention U-Net in the context of different backbones such as VGG19, ResNet152V2, and EfficientNetB7. By providing this comparison, we will accomplish a survey for future researchers to more effectively decide on which Attention U-Net architecture to utilize for their segmentation projects.**

***Index Terms*—kidney, segmentation, U-Net, CNN.**

I. INTRODUCTION

In recent years, intensive research on medical imaging and pattern recognition with performance equal to human-handed inspection or even better has seen exponential growth – albeit not free of criticism and controversy. However, medical applications are under pressure of high accuracy in the detection of convoluted geometrical shapes. Traditionally, architectures were either non-standard or very complex to use to highlight these shapes. Accordingly, in 2015, U-Net was introduced to accomplish the function of automated image segmentation with regard to medical imaging. It is a system with a specific Deep Learning architecture that resembles a "U" - encoding followed by decoding with skip connections.

The goal of this research is to use kidney detection as a proof of concept to provide an analogy of the performance of different U-Net models. The hope is to facilitate future researchers when deciding on the best U-Net Segmentation algorithm to use. To the best of our knowledge, there does not appear to be any paper comparing Attention U-Net with regards to backbones - VGG19, ResNet152V2, and EfficientNetB7 and exclusively within the context of kidney segmentation. We provide recommendations on what architectures are the best to use.

II. MOTIVATION

This academic contribution aims to demonstrate a segmentation system based on U-Net, to address the elusive challenges that hinder a complex deep network process for medical diagnosis. Comparison of different backbone algorithms was also scarce based on classifications with recent encoders and backbones. Another question that sparked curiosity was tuning a U-Net architecture with the latest backbone algorithms, hitherto without any previous related comparison. The main characteristics of backbones aim to solve efficiency problems by reducing unnecessary computations.

A. Previous work

Related research has been conducted on kidney datasets using U-Net architectures, namely 3D U-Net [2]. Nevertheless, none of them explore the potential advantages of backbones such as VGG19, ResNet152V2, and EfficientNetB7. These backbones are CNN (Convolutional Neural Network) architectures that have won the ImageNet competitions whereby millions of images have been categorized into around 1000 categories like dogs and cats.

Moreover, Seum et. al. [3] suggested incorporating segmentation as the first step for the COVID-19 diagnosis pipeline. The reason for this was the enhancement of tuning what is being sent to the CNN for classification. For instance, kidney segmentation could be utilized to determine kidney location prior to sending to a CNN. This information would indicate if there is a disease such as tumors or stones within the kidney annotated region. Thus, improving the performance of CNN.

Z. Wang et. al. [4] on the other hand, proposed a brand new U-Net called "RAR-U-Net" which stands for "Residual encoder to Attention decoder by Residual connections framework for medical image segmentation under noisy labels". Our investigation deals with the ordinary specimen of healthy kidney ultrasounds that are of relatively good quality, thus, a "noisy label" is not of concern to us. If we are to expand our project to accommodate more complex images, RAR-U-Net would be implemented.

Li. et. al. [5], introduce "ANU-Net". A creation that was attempted for a new "U-Net" that is more robust and able to more correctly annotate the medical images under attention mechanism. In the course of this investigation, we are only considering U-Nets that have already been formally designed and tested - in our case by being part of the Keras library. This article is an excellent springboard for generating our custom U-Net if we wish to pursue further understanding of how U-Nets work and how to improve them.

Overall, we investigate exclusively kidney detection with regards to a comparison of Attention U-Net with various

corresponding backbones VGG19, ResNet152V2, and EfficientNetB7. We believe that such a comparison has not been done, especially within the context of kidney detection.

B. *Dataset*

The dataset was obtained freely from Kaggle under the title "CT2USforKidneySeg - A Dataset synthesized US images from CT data with labels"[1]. In total, the number of samples was 200, with separate segmented masks, rounded on 256 x 256 scale. The slices consist of kidney ultrasounds whereas the masks contain the outline of the kidneys. Furthermore, the repository was randomly shuffled and splitted into a training-set (90%), and a validation-set (10%) to evaluate the experimenting models. Below is an example of an ultrasound followed by the corresponding mask that annotates the kidney.

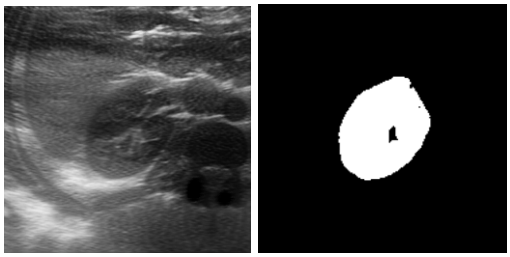

Fig. 1. Sample figure next to masked implementation.

III. Algorithms

Algorithms were completed in Python using Keras[2] and TensorFlow[3]. Specifically, the Keras U-Net Collection library was leveraged. It provides 10 different U-net architectures. Our comparison is of the Attention U-Net model. In addition to these models, we also have transfer learning from the following backbones: VGG19, ResNet152V2, and EfficientNetB7 with and without ImageNet weights. We seek to compare the performance of all these Attention U-Net and backbone combinations in order to provide a benchmark to facilitate future research.

TABLE I
U-Net table of selected backbones.

Segmentation Models	**CNN (Convolutional Neural Network) Backbones**
Attention U-Net	No Backbone
	VGG19 - with/without ImageNet
	ResNet152V2 - with/without ImageNet
	EfficientNetB7 - with/without ImageNet

A. *Segmentation models*

Segmentation is the ability to separate an image into its semantic components - regions that describe a specific object. Our example is to highlight the borders of a kidney in an ultrasound with a simple binary mask. The following describes U-Net and its successor - Attention U-Net.

a. *U-Net*

The U-Net architecture belongs to the FCN (Fully Convolutional Networks) family, differentiating from conventional CNN by having an extra layer that enables for complex calculations of various sample sizes. U-Net was the first Deep Learning architecture built for performing biomedical purposes in 2015. Essentially it is a "U"-like autoencoder architecture whereby the first half encodes (dimensionality reduction) and the second half decodes (dimensionality increase).

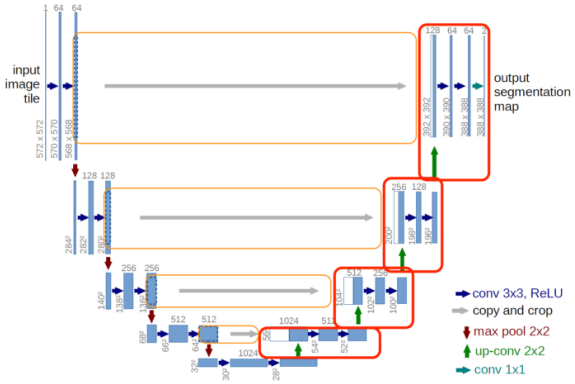

Fig. 2. U-net architecture [6].

The encoding is performed by a CNN-like structure that uses kernels and pooling in order to preserve important information while compressing it into a smaller context. The decoding is the opposite whereby up-convolutions and also up-pooling essentially mean that it increases the sizes based on the categories obtained in the encoder half. Additionally, there are skip connections that connect components of the encoder with its corresponding components in the decoder of the same layer. Training using U-Net is accomplished by providing, as input, both the original images and the corresponding masks. Essentially, the main idea behind U-Net is that the original image and mask are condensed into its semantic parts. Then, it is uncompressed based on the expansion of these semantic parts. Thus, the image is separated into the semantic parts.

[1] https://www.kaggle.com/siatsyx/ct2usforkidneyseg/version/1

[2] Keras. (2021). Keras (v2.6.0) . Keras team. https://keras.io

[3] TensorFlow Developers. (2021). TensorFlow (v2.4.3). Zenodo. https://doi.org/10.5281/zenodo.5189249

b. Attention U-Net

In 2018, Attention U-Net was created as an enhancement to the classical U-Net. Essentially, it highlights target structures while mitigating irrelevant regions, thus, an "attention mechanism".

c. U-Net Loss Function (Binary Cross Entropy)

For all U-Nets, the loss function is the Binary Cross Entropy. It is given below.

$$Log\ loss\ = \frac{1}{N}\sum_{i=1}^{N} - (y_i\ *\ log(p_i)\ +\ (1-y_i)\ *\ log(1-p_i)) \quad (1)$$

B. CNN (Convolutional Neural Network) backbones

In addition to the U-Net architecture, we also train models that include a CNN backbone. These CNN backbones are ImageNet competition winners - a yearly competition to find the algorithm that is the best in classifying millions of images into thousands of categories. CNNs have been the Deep Learning structures that have been able to accomplish such a massive task successfully. It is these fine-trained CNN architectures that we seek to use in order to give better results in our project. Generally, CNNs are structured like the following:

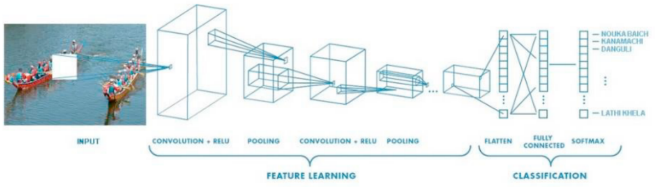

Fig. 3. Generic Convolutional Neural Network [7].

The different CNNs investigated are: VGG19, ResNet152, and EfficientNetB7. These models have pre-trained weights trained on the ImageNet dataset. We investigated using these different backbones with the Attention U-Net model. Essentially, this is transfer learning applied to segmentation tasks.

a. VGG19

VGG was created by the Visual Geometry Group at Oxford in 2015. Essentially, it is a variant of the VGG model group, and it consists of 19 layers (16 convolutional, 3 fully connected, 5 MaxPool, and 1 SoftMax). It has the ability of over 19.6 billion FLOPS (Floating point operations per second) [8].

b. ResNet152V2

It is a Residual Network that won the ImageNet competition in 2015. It consists of 152 layers. The breakthrough with ResNet is the ability to train very deep networks. Additionally, it is the architecture that introduced "skip connections" to CNN whereby different layers are connected directly [9].

c. EfficientNetB7

The first EfficientNet was introduced in 2019. It was considered one of the most "efficient" models. Overall, it reaches "state-of-the-art" accuracy on ImageNet and also on transfer learning tasks [10].

d. Loss Function for CNN

Generally, for all CNNs the Loss function is as follows with s_1 the probability and t_1 the target. It is called the Cross Entropy or Binary Cross Entropy if only two classes are used.

$$Cross\ Entropy\ =\ -\sum_{i}^{C} t_i log(s_i) \quad (2)$$

$$Binary\ CE\ =\ -\sum_{i=1}^{C'=2} t_i log(s_i)\ = \\ -t_1 log(s_1)\ -\ (1-t_1) log(1-s_1) \quad (3)$$

IV. Analysis

This is a Deep Learning segmentation project, accordingly, there are several metrics universally recognized to be used to measure its abilities. These include Confusion Matrix, Precision and Recall, Accuracy, Jaccard index / IoU (Intersection over Union), DICE, and Loss.

A. Confusion Matrix

This is a table that compares the true results with the predicted results. It is important because it gives a representation of how the algorithm is performing. Most specifically, it is very bad to have positives shown as negatives (False Negative - FN). This result is especially worrisome within the medical field with drastic consequences for patients. Essentially, they have a disease but the algorithm fails to detect this.

TABLE II
Confusion matrix.

Predicted Class / True Class	Positive	Negative
Positive	True Positive (TP): Predicted positive and actual positive	False Positive (FP): Predicted positive and actual negative
Negative	False Negative (FN): Predicted negative and actual positive	True Negative (TN): Predicted negative and actual negative

B. Precision and Recall

Precision is known as the "positive predictive value". It is the ratio of correct positive predictions to the total predicted positives.

$$P = \frac{TP}{TP + FP} \tag{4}$$

Recall is also known as "Sensitivity / Probability of Detection / True Positive Rate". It is the ratio of correct positive predictions to the total positive examples.

$$R = \frac{TP}{TP + FN} \tag{5}$$

C. Accuracy

This is the percentage of correct predictions divided by all predictions. In our case, the pixel accuracy might not represent a strong metric in our analysis, because semantic segmentation increases the correlation between an object and the background, thus, causing a high accuracy score attributed to overfitting. Accordingly, we ignore accuracy in our project.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

D. Jaccard index/IoU (Intersection over Union)

The Jaccard is also known as the IoU (Intersection over Union). It is basically a measure of overlap between images divided by the union of the images. A value of zero means no overlap whereas a one means complete overlap. The goal is to reach close to one meaning that the images are very similar.

$$IoU = \frac{TP}{TP\,FP + FN} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{7}$$

E. DICE Loss (Sørensen–Dice coefficient)

This metric was developed in the 1940s to measure the similarity between two samples just like Jaccard. Values fluctuate between zero and one, where zero means no spatial overlap and one indicates complete overlap. DICE is calculated by two times the area of overlap divided by the total number of images in both images.

$$DICE\ Loss = \frac{2 \times \left|X \cap Y\right|}{\left|X \cup Y\right|} \tag{8}$$

F. Loss Function - Binary Cross Entropy

This is the calculation that is used to lower the differences between the output produced and the desired output of the segmentation engine. The same Loss Function, Binary Cross Entropy, has been applied to both Attention U-Net and the corresponding CNN backbones.

V. Results

The overall training of all models was as follows. Firstly, the U-Net took over 300hrs to finish due to our processing limitations. We were forced to limit ourselves to adopt feasible abilities given our processing characteristics. The following demonstrates what our algorithm accomplished. As can be observed, the mask prediction, below, performed very well.Additionally, we have included a table that summarizes our results. We facilitate experiments with different weights that could deal with the negative background class. Overall, our results have interesting interpretations which we will discuss in section VI DISCUSSIONS and VII CONCLUSIONS.

Fig. 4. Illustration of mask prediction.

TABLE III
Comparison of Attention U-Net across IoU, DICE, Precision, and Recall.

Method	IoU	DICE	Precision	Recall
Attention U-Net [5]	85.77	92.34	0.9097	0.9376
Attention U-Net [no backbone, no weights]	96.39	94.96	0.9746	0.9413
ImageNet and No Freeze				
Attention U-Net VGG19	90.94	93.16	0.9644	0.9217
Attention U-Net ResNet152V2	166.71	95.635	0.9831	0.9346
ImageNet and Freeze				
Attention U-Net VGG19	103.06	86.89	0.9355	0.8169
Attention U-Net ResNet152V2	100.05	74.07	0.8930	0.6202
No ImageNet and No Freeze				
Attention U-Net VGG19	89.23	86.58	0.9181	0.8868

Attention U-Net ResNet152V2	166.71	85.181	0.8477	0.9016
No ImageNet and Freeze				
Attention U-Net VGG19	89.73	85.57	0.8797	0.8899
Attention U-Net ResNet152V2	89.23	66.93	0.7856	0.6814

VI. DISCUSSION

In this section we discuss the Limitations, Expectations, Interpretations, and Recommendations.

A. Limitations

Our major limitation was processing speed by using Google Collab. Accordingly, our results are based on a relatively small number of epochs, sixty, and the number of participants, two-hundred. Overall, it took an average of two hours to train one single model. Nevertheless, our experience is important because we can translate our results to the practical world given that most researchers are similarly limited [13]. Also, because of the nature of binary segmentation of the imbalanced classes, different weights should be calibrated to fit model needs.

B. Expectations

This academic examination produced some unanticipated results. Originally, we had assumed that utilizing the latest CNN as the backbone with ImageNet weights and freezing of the backbone would have produced the overall clear best results. This has not occurred. Instead, we have learned that it is a much more nuanced task to select an appropriate and effective Attention U-Net deep learning architecture. We had assumed that the latest ImageNet winner, EfficientNetB7 would produce the best results. Instead, it tended to produce the worst results which are most likely attributed to its huge architecture and, thus, the larger training requirements - it consists of over 800 layers and over 60 million parameters. Additionally, as we learned later, EfficientNet on Tensorflow accepts only raw images and does not work with masked samples. Thus, we ignore EfficientNet.

C. Interpretations

Highest Overall Scores:
- IoU - Attention U-Net (No backbone) (96.39%)
- DICE - ResNet152V2 (ImageNet/Nofreeze) (95.64%)
- Precision - ResNet152V2 (ImageNet/no freeze) (98.31%)
- Recall - Attention U-Net (No backbone) (94.13%)

TABLE IV
AVERAGE BEST BACKBONE IOU/DICE/PRECISION/RECALL

Model (order of best)	**IoU**	**DICE**	**Precision**	**Recall**
(A) ImageNet and No Freeze	128.82 %	94.3975 %	0.97375 %	0.92815 %
(B) No ImageNet and No Freeze	127.97 %	85.8805 %	0.8829 %	0.8942 %
(C) ImageNet and Freeze	101.53 %	80.48 %	0.89925 %	0.71855 %
(D) No ImageNet and Freeze	89.48 %	76.25 %	0.83265 %	0.78565 %

TABLE V
DICE EPOCH PERFORMANCE

Model (order of best)	**DICE Average number of Epochs**
(A) ImageNet and No Freeze	22.5
(B) ImageNet and Freeze	25
(C) No ImageNet and No Freeze	37.5
(D) No ImageNet and Freeze	60

From our results, we took into consideration the performance measurements (IoU %, DICE %, Precision %, and Recall %) as seen in Table IV. They are the standard used to compare the performance of segmentations. Firstly, we learned that “no backbone” produces the best results, however, this may be a result of our rather simple and small dataset and classification. It does not give the opportunity to exploit the benefits of a well-trained, award-winning, Deep CNN.

In general, VGG19 outperformed for more instances better than ResNet152V2 in terms of accuracy across the 60 epochs. The reasons for these successes are a result of both limited dataset and limited training time. ResNet152V2 is newer than VGG19, however, it consists of more layers and more parameters, thus, to function correctly and effectively it needs more processing time.

If we ignore the “no backbone”, the order of the best average is detailed in Table IV. When we initialize with (A), we are given an architecture that just needs to be tweaked at the classification layer. Whereas, with (D), essentially, with “No ImageNet” (B & D) this means that our models have weights that are arbitrary and meaningless. Moreover, to freeze this means that we retain the insignificant weights. Now with regards to (B), this is the equivalent of training the whole architecture, however, the convergence takes longer due to the large architectures of the CNN. Finally, with (C), it is no good because we are not changing the weights at all to suit our

project needs.

Additionally, in regards to training speed, we have determined that the order of performance from best to worst, regarding DICE epoch convergence is displayed in Table V. This is logical considering that having a baseline with ImageNet accelerates convergence due to it being pre-trained. When Freezing is concerned, it is not as important as using ImageNet weights. However, when Freezing is combined with No ImageNet it gives worse overall performance. The reason why is because we are stuck with weights not trained for our current purposes - or any other purposes.

Thus, we summarize that for complex and larger datasets and enough processing powers, a recent ImageNet architecture should be leveraged with Attention U-Net. This ImageNet backbone should be initialized with its corresponding ImageNet weights and trained without freezing of these weights. Additionally, if there are limitations to training time, pick the best architecture based on the DICE epoch convergence.

D. Recommendations

Arguably, every dataset requires specific hyperparameter tweaking based on complexity and robustness. Due to our resource limitations, we used the standard of a batch size of 8, 60 epochs, loss function as Binary Cross-Entropy, optimizer as Adam, a learning rate of 1e-3, number of participants of 200, and a train-test (split of 90%-10%). From our results, the best overall model to select is Attention U-Net without any backbones. However, the reason why the other backbones did not achieve the best results is that they were built for complex data and complex classification. In this project, we leveraged a small and simple dataset with binary colors and binary masks, thus, we were not able to fully appreciate the advantages of these CNN backbones (VGG19 and ResNet152V2).

If we consider the backbones alone, without the no backbone, we determine that the best backbones have the following characteristics as per Table IV. This behavior is logically based on the complexity of ImageNet and Freezing.

Furthermore, we found that VGG19 performed better than ResNet152V2. This is because of the larger training time required for ResNet152V2 given its more sophisticated and deeper architecture. However, with regards to a faster convergence in DICE score, ResNet152V2 consistently outperformed VGG19.

Training speed is also affected by how the CNN is initialized and if it is frozen. Overall, if it is initialized with ImageNet, then you will have a faster convergence of DICE. Freezing affects it, especially giving the worst results if it is the combination of “No ImageNet and Freeze”. Thus, we recommend “ImageNet and No Freeze” because of its faster processing time.

We recommend studying the backbones closely before application. Therefore, when deciding on what Attention U-Net to pursue, you need to consider the dataset size, the complexity, the number of classes, and one’s processing abilities. Ideally, we recommend ResNet152V2 with “ImageNet and No Freeze” and be trained with more epochs, more participants, and more processing power.

VII. Conclusion

This research has generated many questions and exciting future areas of research to be answered by Attention U-Nets in conjunction with other Deep Learning architectures. We have determined that for complex datasets with enough processing power, Attention U-Net works best under ImageNet and No Freeze. One avenue of further research includes working with 3D models to diagnose kidney disease [11]. As well, although this project focused on the detection of kidneys in ultrasounds, it could easily be extended to find tumors and other abnormalities by providing the appropriate annotations and using the pipeline of first Attention U-Net for detection and then CNN transfer learning for diagnosis. Additionally, we have learned that finding the best model to work with is not a trivial task and involves careful considerations of dataset size, complexity, the number of categories and processing abilities. Overall, online available medical data is difficult to attain, thus, a future avenue would be to generate synthetic data with the help of GAN (Generative Adversarial Network) [12]. Finally, the most important goal which has yet to be answered by researchers in any medical field: is it possible to predict when a disease will develop and how it progresses with the use of Deep Learning.

Acknowledgment

The authors would like to thank the organizers of McMedHacks 2021, who sparked the idea of this contribution.

References

[1] R. L. Siegel, K. D. Miller, and A. Jemal, “Cancer statistics, 2020,” CA: A Cancer Journal for Clinicians, vol. 70, no. 1, pp. 7–30, Jan. 2020, doi: 10.3322/caac.21590.

[2] F. Isensee and K. H. Maier-Hein, “An attempt at beating the 3D U-Net,” 2019, Accessed: Aug. 29, 2021. [Online]. Available: http://dx.doi.org/10.24926/548719.001.

[3] Seum, A., Raj, A., Sakib, S. and Hossain, T., 2021. A Comparative Study of CNN Transfer Learning Classification Algorithms with Segmentation for COVID-19 Detection from CT Scan Images. [online] Search.bvsalud.org. Available at: <https://search.bvsalud.org/global-literature-on-novel-coronavirus-2019-ncov/resource/en/covidwho-1331685> [Accessed 29 August 2021].

[4] Z. Wang, Z. Zhang, and I. Voiculescu, “RAR-U-NET: A Residual Encoder to Attention Decoder by Residual Connections Framework for Spine Segmentation Under Noisy Labels,” Sep. 2021, Accessed: Aug. 29, 2021. [Online]. Available: http://dx.doi.org/10.1109/icip42928.2021.9506085.

[5] Li, C., Tan, Y., Chen, W., Luo, X., He, Y., Gao, Y. and Li, F., 2021. ANU-Net: Attention-based nested U-Net to exploit full resolution features for medical image segmentation.

[6] Ronneberger, O., Fischer, P. and Brox, T., 2021. U-Net: Convolutional Networks for Biomedical Image Segmentation.

[7] Islam, M., Foysal, F., Neehal, N., Karim, E. and Hossain, S., 2018. InceptB: A CNN Based Classification Approach for Recognizing

Traditional Bengali Games. Procedia Computer Science, 143, pp.595-602.
[8] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” CoRR, vol. abs/1409.1556, 2014.
[9] Gizem,K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” 2015.
[10] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” 2019.
[11] M. Denninger and R. Triebel, “3D Scene Reconstruction from a Single Viewport,” in Computer Vision – ECCV 2020, Cham: Springer International Publishing, 2020, pp. 51–67.
[12] “Ian Goodfellow’s Generative Adversarial Networks: AI Learns to Imagine,” in The Artist in the Machine, The MIT Press, 2019.
[13] Du, G., Cao, X., Liang, J., Chen, X., & Zhan, Y. (2020). Medical image segmentation based on u-net: A review. Journal of Imaging Science and Technology, 64(2), 20508-1.