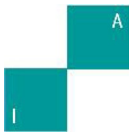
Inteligencia Artificial 25(70) (2022), 64-76
doi: 10.4114/intartif.vol25iss70pp64-76



INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Defective sewing stitch semantic segmentation using DeepLabV3+ and EfficientNet

Nguyen Quoc Toan [1,*]

[1]Department of Electronic and Electrical Engineering, Hongik University, Wausan-ro 94, Mapo-gu, Seoul, South Korea

*quoctoann3@gmail.com

Abstract Defective stitch inspection is an essential part of garment manufacturing quality assurance. Traditional mechanical defect detection systems are effective, but they are usually customized with handcrafted features that must be operated by a human. Deep learning approaches have recently demonstrated exceptional performance in a wide range of computer vision applications. The requirement for precise detail evaluation, combined with the small size of the patterns, undoubtedly increases the difficulty of identification. Therefore, image segmentation (semantic segmentation) was employed for this task. It is identified as a vital research topic in the field of computer vision, being indispensable in a wide range of real-world applications. Semantic segmentation is a method of labeling each pixel in an image. This is in direct contrast to classification, which assigns a single label to the entire image. And multiple objects of the same class are defined as a single entity. DeepLabV3+ architecture, with encoder-decoder architecture, is the applied technique. EfficientNet models (B0-B2) were applied as encoders for experimental processes. The encoder is utilized to encode feature maps from the input image. The encoder's significant information is used by the decoder for upsampling and reconstruction of output. Finally, the best model is DeepLabV3+ with EfficientNetB1 which can classify segmented defective sewing stitches with performance (*MeanIoU*: 94.14%).

Keywords: Computer Vision, Semantic Segmentation, Convolutional Neural Networks, DeepLabV3+, EfficientNet

1 Introduction

The current growth of fast fashion has forced large retailers and their preferred suppliers to respond to rapid changes in market demand in the garment industry. Preferred suppliers must now deal with unforeseen orders from retailers in addition to their large normal basis. International garment manufacturers have relocated their manufacturing facilities to and found new suppliers in, developing countries with abundant cheap labor. Garment development continues to be one of the most labor-intensive industries. However, due to global increases in material and labor costs, as well as a plummeting average garment price, outsourcing to lower-wage suppliers will be exhausted soon. As a result, garment manufacturers must adopt new technologies to overcome existential issues. A sewing defect is a flaw on the garment surface caused by the manufacturing process. As a result, defect detection is a critical step in textile quality assurance. Most defects are visually inspected during manufacturing by skilled human inspectors who quantify raw fabric materials and semi-finished or finished garments. Unfortunately, the inspection process is time-consuming and laborious. Furthermore, stress and fatigue frequently result in inconsistent, inaccurate, and biased human errors. This phenomenon can cause the entire production line to be reworked. Furthermore, additional inspection costs raise the overall cost of the garment. Several automated methods to replace manual inspection have been developed to tackle these issues while improving the testing accuracy of sewing stitches. Among the most comprehensive methods are image processing and computer vision. Although traditional defect detection applications have demonstrated good performance, they are typically configured with handcrafted features designed by human operators to account for variations in real-world manufacturing conditions. Human error is a paramount factor to consider. As a result, handcrafted features are required before deployment for each inspection task. As a result, developing a more general discriminative defect detection method is challenging.

In this paper, DeepLabV3+ architecture [8] has been applied, which achieves good performance on the VOC dataset and MS-COCO dataset, to the defect segmentation of sewing stitches, and proposes a method of combining encoders with backbones (EfficientNetB0, EfficientNetB1 EfficientNetB2), resulting in the suitable solution for the task. EfficientNet (B0-B2) was prioritized in this paper due to limitations of hardware configuration as well. However, the model with the best evaluation metrics can be used in real-world solutions for intelligent sewing systems.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

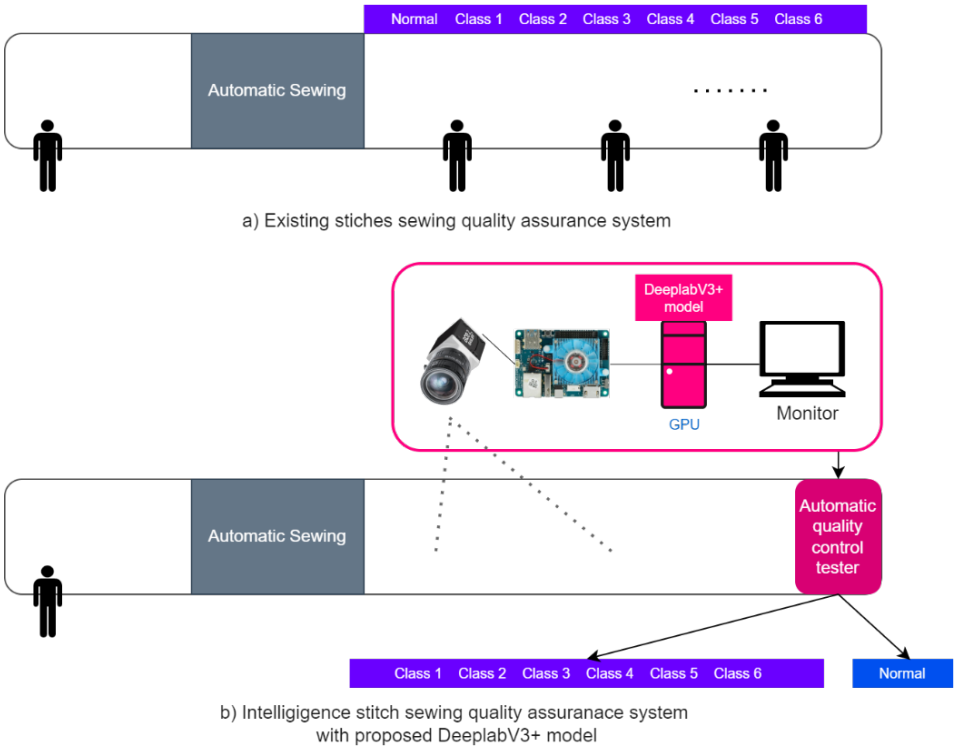


Figure 1. Illustration of the existing system and intelligent system with the proposed model.

2 Related work

This section provides a brief overview of defect detection in garment manufacturing. Fabric cutting, sewing, and product packaging are typical aspects of garment manufacturing [12]. Most defect detection methods for these processes are primarily concerned with defects in fabric textures as raw materials [21] and [22] recently summarized automated fabric defect detection methods into five categories: structural, statistical, spectral, model-based, and learning. The structural (i.e., feature) approach infers the placement rule using a primitive pattern. This method detects simple defects on regular textures but is incapable of dealing with complex features and defect variations. Histograms, co-occurrence matrices, autocorrelations, and mathematical morphologies are used in the statistical approach to provide statistical characteristics for spatial relationships within the target images. Notwithstanding, the target images' reference features must first be defined. Furthermore, this method is sensitive to defect size and shape and necessitates a large computational capacity for large images. The spectral (filter-based) approach analyses patterns in new domains by transforming target images with Fourier, wavelet, and Gabor transforms. This method is resistant to spatial changes such as scaling and rotation, but it requires knowledge of transformation methods as well as difficult-to-judge choices. The model-based approach detects flaws by analyzing target image patterns with stochastic modeling methods such as the autoregressive model and Markov random fields. These methods are light-sensitive and have difficulty detecting minor flaws. Finally, the learning method involves artificial and deep neural networks to learn complex nonlinear relationships to distinguish between normal and defective images. This approach achieves well in terms of feature extraction and classification in general. Obtaining a large dataset to train, validate, and respond to various features and variations, on either hand, is difficult. On top of that, CNN feature map and image-processing techniques with VGG 16 backbone were employed in [23], defects precisely, achieving 92.3% accuracy, 100% recall, and 87.5% precision.

Most garment defect detection studies, as summarized above, have primarily focused on the fabric surface, because the value of a manufactured garment decreases significantly if the fabric contains defects. Furthermore, most traditional defect detection methods require skilled human operators to extract or design reference features from or for target images. Even so, there has been no segmentation for defective sewing stitches research, and only a simple amount of research has been conducted, which are simpler constructs than fabric textures but can also affect garment quality. Because stitch defects are thin surface defects, they are tough to detect using methods developed for fabric defect detection. [24] proposed a back-propagation-based wavelet-based image extraction and defect, classification model. Furthermore, [25] extracted the sewing-stitch outline by the Canny edge detection algorithm to detect defective stitches. Even though these models demonstrated the ability to detect such defects,

they are limited in their application to similar inspection tasks since they require several equations and handcrafted threshold values provided by a skilled human researcher. Before deployment, these configurations must be trailed and tuned for each inspection task.

3 Dataset

Deep learning methods are heavily reliant on data. The supervised learning algorithm is conducted for end-to-end learning in deep learning-based semantic segmentation models. Supervised learning algorithms are built to learn by doing. These examples are known as training data, and they will include inputs paired with the correct output. Deep learning models will learn the patterns that exist between the input and the target output throughout training. Once the model has learned those connections, it will be evaluated on previously unknown data sources and will predict the output. This training process necessitates an enormous amount of data for the model to grasp patterns more efficiently.

The limited availability of data on defective sewing stitches imagery provided a challenge in training deep learning models. To tackle this issue, a dataset was gathered at Hongik University Artificial Intelligence Laboratory (HAIL, Seoul, Republic of Korea) using an industrial ELP Aptina camera (1.3MP, lens 8mm) and supportive led light to ensure brightness conditions. Figure [2] is the setup for collecting images for the proposed dataset. The final dataset includes 900 images (1280x960) labeled into six semantic classes, ‘B’ stands for Blue, and ‘W’ stands for White stitches: BDiagonal, BHorizontal, BStraight, WDiagonal, WHorizontal, WStraight.

This labeling was done by assigning one of the six semantic classes in numeric format to each pixel of the captured image. The dataset is divided into training, validation, and testing sets for training, evaluation, and performance comparison as shown in Table [1]. The dataset split was chosen based on the balance of classes, ensuring that all classes have the same quantity during the phases. During the training process, the training set is used to teach the model the relations between the input and the target output, whereas the validation set is used to test the learned model's accuracy during the training process. The test set is a different set of inputs that will be used to evaluate the performance of the final model. Figure [3] displays an example labeled image from each of the six classes.

Table 1. Summary of the proposed dataset

Split	Training	Validation	Testing	Total
No.of images	720	90	90	900

Figure 2. The setup for collecting images for the proposed dataset.

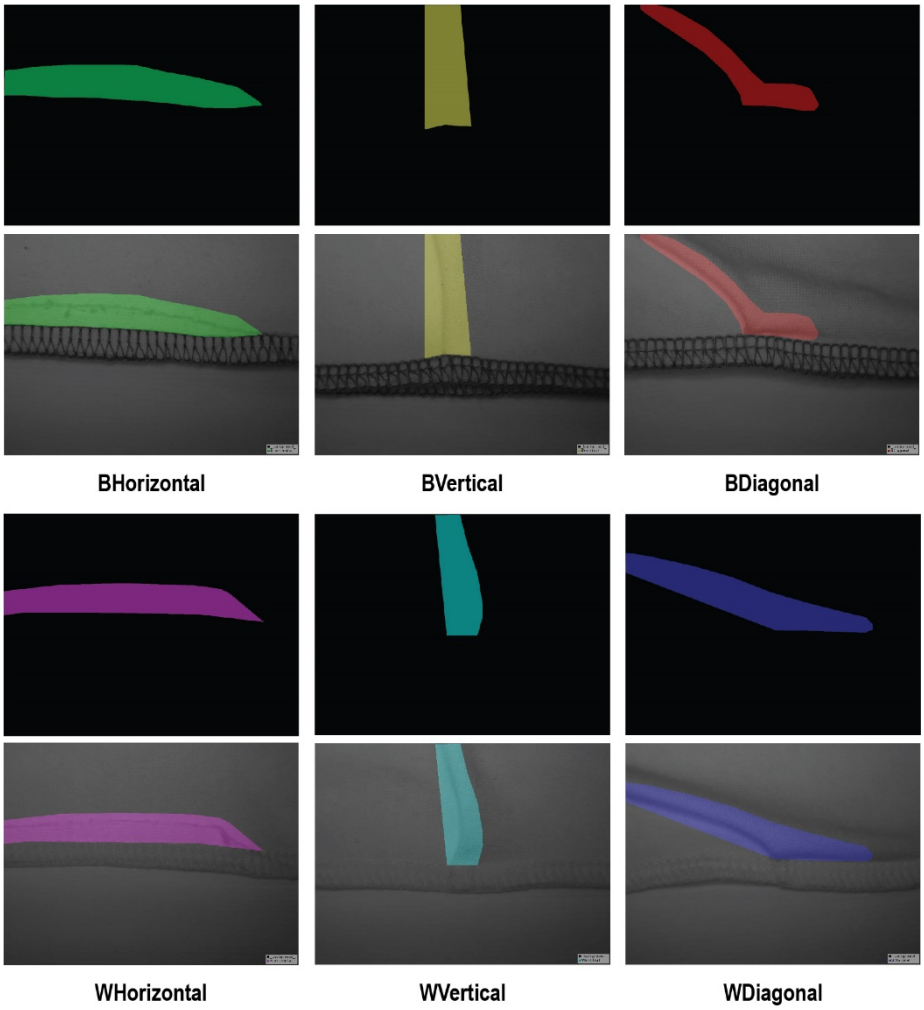


Figure 3. Example of 6 labeled classes from the proposed dataset.

4 Methodologies

4.1 DeepLabV3+

4.1.1 Overview

DeepLabV3+ [5] is a semantic segmentation architecture that enhances DeepLabV3 [8] by including a simple but effective decoder module to achieve segmentation results. The most used structures for semantic segmentation tasks are dilated convolution and encoder-decoder structures. The DeepLabv3+ structure, on the other hand, combines two approaches to addressing the issue of inaccurate segmentation due to spatial information loss, resulting in segmentation maps with more detailed boundary information. Therefore, I use DeepLabV3+ as the network framework for the segmentation stage. Figure [7] depicts the structure of the proposed network.

4.1.2 Fully convolutional network

[26] was the first to propose a deep CNN [27] for semantic segmentation in the field of fully supervised training. In the same year, an FCN for semantic segmentation was introduced [34]. The network weights are adjusted using feedforward inference and feedback learning, as shown in Figure [4], and the fully connected layers used for classification are discarded. To realize the prediction for each pixel, the entire network employs convolution operations, obtains depth information by downsampling, and restores the original size by upsampling.

With the establishment of FCNs came a slew of semantic segmentation algorithms based on them. The DeepLab framework was used in this study's experiment to achieve excellent achievements in the semantic segmentation field using the multipath fusion approach.

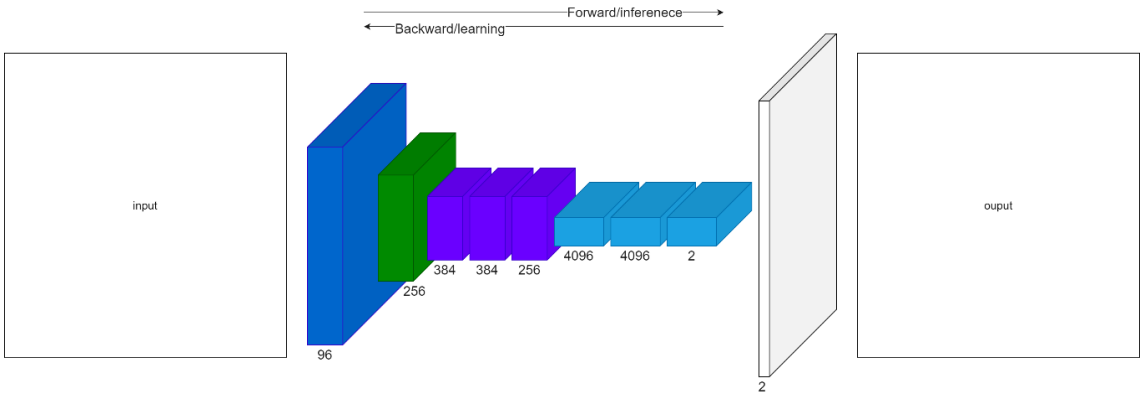


Figure 4. Fully convolutional network (FCN).

4.1.2 Atrous spatial pyramid pooling

To cope with the information loss caused by pooling, the DeepLabV3+ model employs atrous spatial pyramid pooling (ASPP), which can extract features at various resolutions and feature layers for semantic segmentation. As shown in Figure [5], when the receptive field remains constant, the number of weight parameters is reduced, and the location information loss caused by mean pooling is solved.

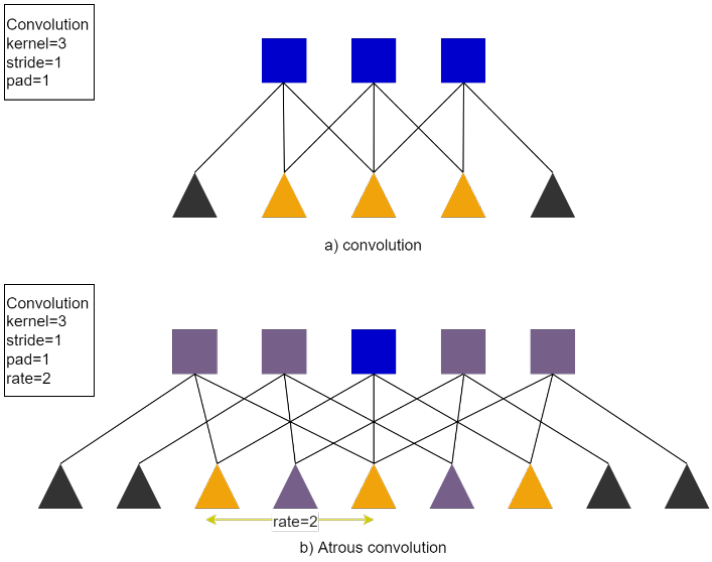


Figure 5. View of atrous convolution in one dimension.

The receptive field can be expanded using atrous convolution without increasing the volume or parameters. Rather than mean pooling, atrous convolution can effectively obtain the details after the convolution. Formula (1) denotes a filter with length k and an input signal of x.

$$[\mathrm{i}] = \sum_{k=1}^{K} x[i + r \bullet k]w[k] \tag{1}$$

The above equation is downsampling of step 2; the image's resolution is reduced, and then a convolution operation with a convolution kernel size of 7x7 is performed to obtain a feature map, which is then double upsampled to restore the original resolution. As a 7x7-size hole convolution, the convolution kernel is used. After a direct convolution, the feature map is obtained. According to the comparison results, the hole convolution map is more detailed. Although the hole convolution increases and the nonzero filter value is considered in the calculation, the actual parameters remain unchanged and the operation cost is reduced, as illustrated in [28].

4.1.4 Codec model structure

DeepLabV3+ employs a codec structure that includes shallow and deep upsampling features. As shown in Figure [6], the input features are fed into a deep CNN to produce a high-resolution abstract feature map with a lower resolution, [28], [5], and different volume convolutions are used to perform the convolution. The obtained high-level feature map is merged with upsampling four times and the shallow features to grasp the decoded output in deep feature sampling.

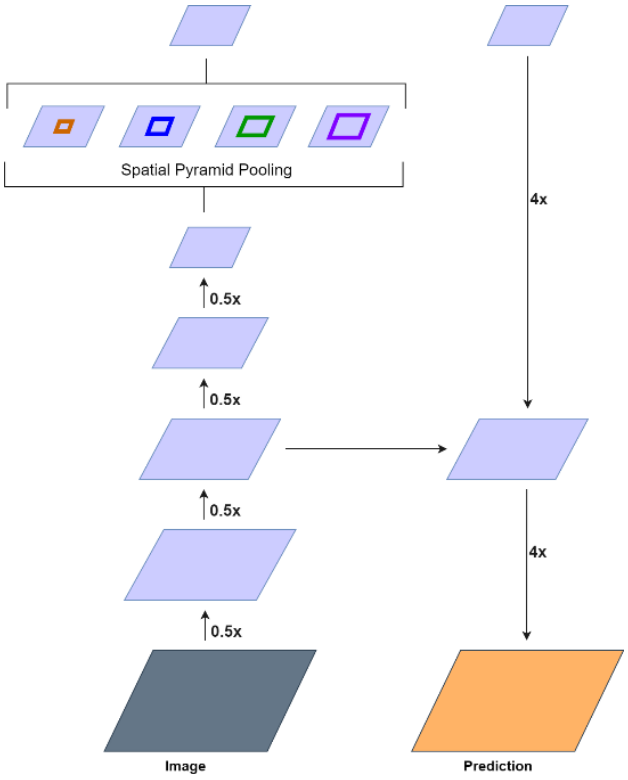


Figure 6. Structure of the atrous convolutional codec.

The DeepLabV3+ model is divided into two parts: encoding and decoding to maintain the high-level abstract information large enough to facilitate pixel location prediction, and the coding section eliminates the deep pool of the feature extraction network. By replacing the deep pooling layer with ASPP, more details are preserved under the same receptive field conditions without increasing the training parameters, thereby bettering model prediction performance. The target samples are obtained with widely differing amounts of information using multiscale information sampling, which improves the model's robustness. The use of a 1x1 size convolution after a multi-scale hole convolution increases the coding structure's nonlinearity. The shallow features are first received by the decoding part, which then uses the 1x1 size convolution to reduce the number of channels in the feature map so that the feature map obtained by upsampling four times after encoding is nearly identical to the number of channels in the feature map, which is beneficial to the model's learning. Convolutional shallow features are combined with upsampled deep features, and convolution is used to refine feature details. The final prediction result is obtained after four times of upsampling at the same resolution as the original image. Figure [7] depicts the DeepLabV3+ model's structure.

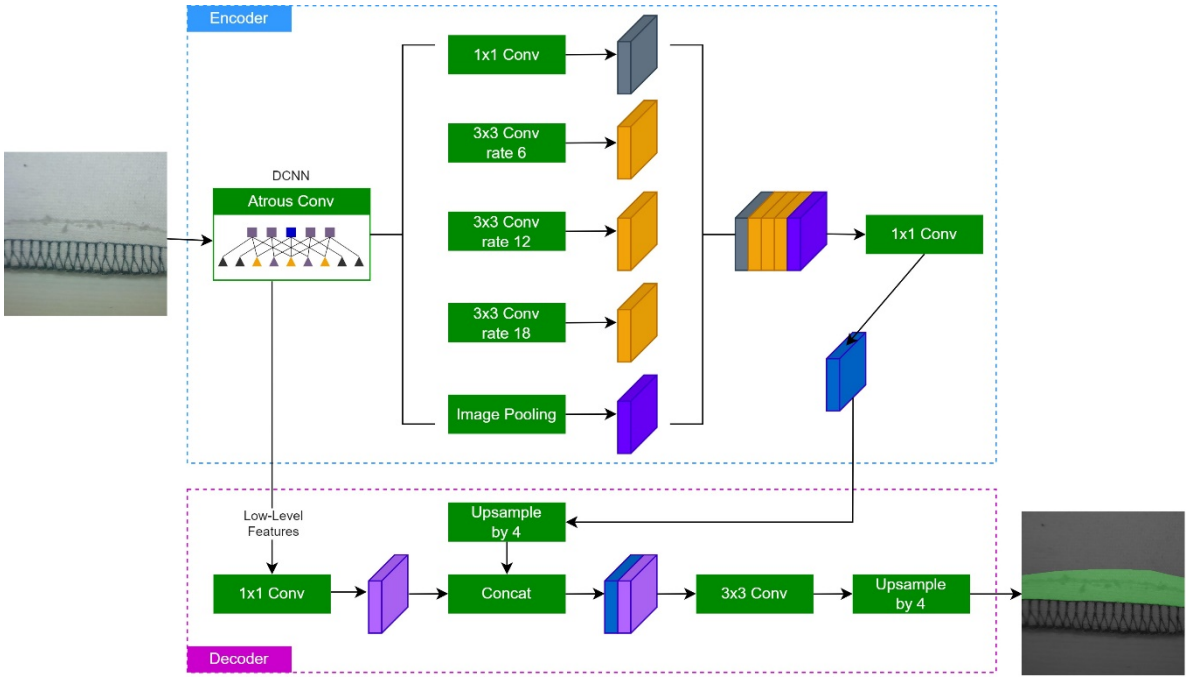


Figure 7. Structure of DeepLabV3+ model.

4.1.4 Algorithm flow

The DeepLabV3+ segmentation model workflow consists of five steps:

- Step 1: Before feeding the input image into the model, the size and resolution are fixed and unified.
- Step 2: Extract image features using the backbone networks (EfficientNetB0-B2) and retain image detail information using hole convolution.
- Step 3: High-level features are placed in the ASPP structure, while low-level features are placed in the Decoder structure, preserving the image feature information's integrity.
- Step 4: The high-level feature map in the Encoder structure is up-sampling using bilinear interpolation to link the size of the feature map after feature refinement in the Decoder. Following sampling and refinement, the results are feature-fused to produce a feature-rich image.
- Step 5: After feature fusion, upsampling the image to obtain segmented images that are consistent with the input image parameters, and the segmentation process has been completed.

4.1.5 EfficientNet

EfficientNet, presented by Google AI research, is a collection of CNN models that outperforms its predecessors with minor alterations [4]. It comes in 8 variations, numbered B0 to B7, with each subsequent model number referring to variants with more parameters and higher accuracy. EfficientNet functions in three stages:

- **Depthwise + Pointwise Convolution:** Depthwise convolution operates independently on each input channel. This is an instance of spatial convolution. The channel output from depthwise convolution is projected onto a new channel space by pointwise convolution. This is a convolution of 1x1.
- **Inverse Res:** ResNet blocks are made up of two layers: one that squeezes the channels and one that extends them. It connects skip connections to rich channel layers in this manner [15].
- **Linear bottleneck:** In the final layer of each block, linear activation is used to prevent information loss from ReLU [30].

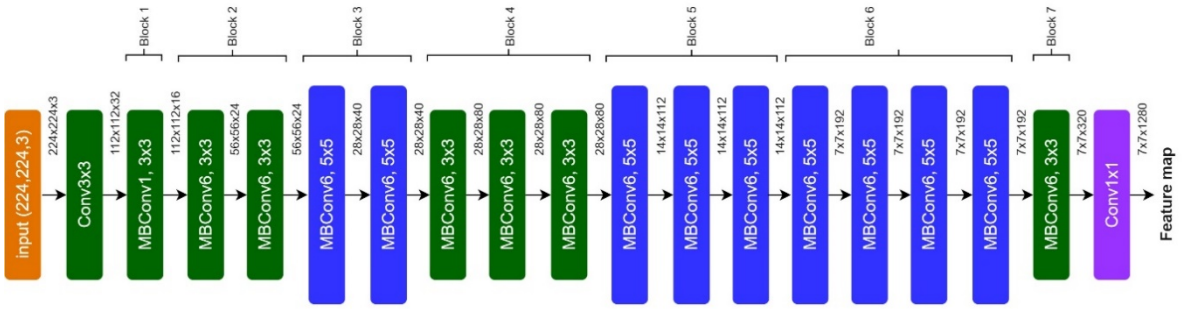


Figure 8. EfficientNet's Architecture with MBConv as fundamental building blocks.

As aforementioned, EfficientNet has eight varieties, B0 - B7, of which the first three have been conducted in this paper. The remaining models were ignored as the complexity increased because they produced underappreciated results with poor performance while consuming valuable runtime. Each model's layers (B0 - B7) can be structured using the five standard modules shown in Figure [9].

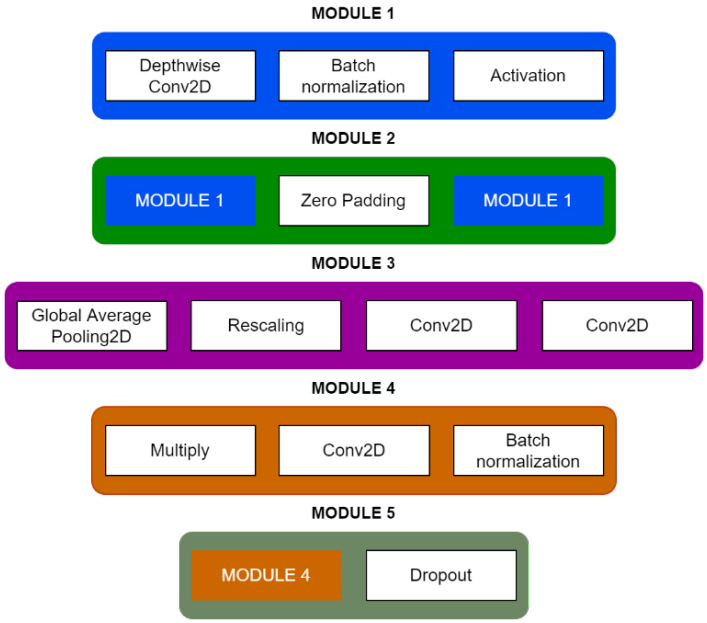


Figure 9. Modules that were used to implement layers in all 8 EfficientNet models.

- **Module 1** sets the stage for the sub-blocks
- **Module 2** works as the initialization point for the first sub-block of each of the 7 main blocks except the first one
- **Module 3** acts as a skip connection block for all sub-blocks
- **Module 4** combines the skip connections produced in the initial sub-blocks
- **Module 5** combines each sub-block that is linked to the sub-block before it in a skip connection.

Individual modules are then combined to form sub-blocks in a variety of ways, as illustrated in Figure [10]. The differences between the models are easily discernible, with a gradual increase in the number of sub-blocks. The MBConv layer, an inverted residual block first used in MobileNetV2 [31], serves as the foundation for EfficientNet. Figure [8] depicts the basic building blocks of EfficientNet models concerning MBConv layers. The EfficientNet models (B0 - B7) share common blocks with subtle complexities in their architectures. EfficientNet is identified as a scaled-up neural network architecture in which all dimensions are scaled with a compound coefficient, a method known as Compound Scaling [32]. In this context, scaling up is defined as a standardized, principled scaling of three components: depth, width, and resolution.

- **Width scale** extends more feature maps to each layer.
- **Depth scale** puts more layers on the network.
- **Resolution scale** enhances the resolution of the input images.

Every architecture is like previous versions. The only difference is that the number of parameters is extended by using different feature maps. Except for the multiplied block (x2), which expands and covers more blocks, all the models have the same architecture as the earlier one. This model is robust because it provides many parameters for use in calculations. The differences between all the models are clear, and they gradually increased the number of sub-blocks [4]. Starting with EfficientNet-B0, the compound scaling method was used to scale up in two steps:

- **Step 1**, the coefficient was set to 1, assuming twice as many resources are available, and it pushes a small grid search for the network depth, width, and resolution constants.
- **Step 2**, the constants are then fixed, and the baseline network is scaled up with different coefficients to produce the consecutive variations from B1 to B7.

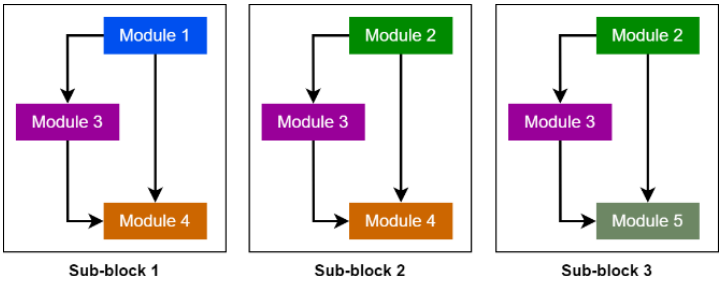


Figure 10. Sub-blocks using individual modules are presented in Figure 9.

4.2 Evaluation standard

The experiment used multiple levels of control parameter variables for the evaluation to measure the model's performance and learning cost, as well as to evaluate the model more effectively. The primary evaluation indicators were the model's training time, model prediction accuracy, memory occupancy, and model parameter size. In a comparison experiment, a controlled hardware configuration and fixed parameters were used.

Image segmentation performance can be evaluated using a variety of standards. However, mean Intersection-Over-Union (*MeanIoU*) is the most representative and accurate evaluation index in general. It denotes the point at which the model's predicted values meet the true values of the sample labels. The union ratio is calculated by adding the average of the intersections of each class. It has the following mathematical expression:

$$MeanIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{3}$$

where k represents the number of categories, up to a maximum of k+1 classes (including a background class), and p_{ii} defines the pixel quantity predicted to be accurate. The number of pixels predicted to be in the background but are positive labels is p_{ij}, and p_{ji} is the quantity of the pixel predicted to be in the foreground but are negative labels.

5 Experimental

5.1 Overview of graphics

The experiment made use of Pytorch, a deep learning framework. On the hardware, the following configurations and software were installed:

- CPU: AMD Ryzen Threadripper 2950X @ 4.40 GHz (16 threads x 32 cores)
- RAM: 64GB DDR4 2666MHZ
- Graphics card: NVIDIA GeForce GTX 2080 Ti 12GB x 2
- Operating system: Linux Ubuntu 20.04.4 LTS
- Language: Python 3.9.12

- Pytorch version: 1.10.1

5.2 Experimental Algorithm

The algorithm below explains how the experimental processes are carried out.

Algorithm 1: Training models

Input: DeepLabV3+, MANet, UNet. EfficientNet, ResNet50, ResNet101, MobileNETV2 encoder (Installed necessary libraries and dependencies)
Output: Defective stitch segmentation weights
Data: Dataset (Train/Val/Test)

```
method=(EfficientNetB0-DeepLabV3+,
EfficientNetB1-DeepLabV3+,
EfficientNetB2-DeepLabV3+,
EfficientNetB0-MANet,
EfficientNetB1-MANet,
EfficientNetB2-MANet,
EfficientNetB0-UNet,
EfficientNetB1-UNet,
EfficientNetB2-UNet,
ResNet50-DeepLabV3+,
ResNet101-DeepLabV3+,
MobileNETV2-DeepLabV3+ )
modelWeights=[]
for model in method do
    Train Train Dataset
    (–imgSize 1056x704 –batch 4 –epochs 500 –device 0,1)
    Evaluate The model
    Compute MeanIoU, Loss
    Display Model performance on Val dataset
    Append modelWeights
for modelWeight in modelWeights do
    Test Test dataset
    Evaluate The model
    Compute MeanIoU Display Model performance on Test dataset
end
```

Figure 11. Experimental training models algorithm.

5.3 Experimental Results

Table [2] represents the results from the proposed method compared to other experimental methods.

Table 2. Comparison between encoders and decoders on the test dataset

Encoder	Decoder	Training Time (hour)	Input	Param (M)	Model Size (MB)	*MeanIoU* (%)
EfficientNetB0	***DeepLabV3+***	17.6	1056x704	4.5	19.0	**92.95**
	MANet [34]	15.5	1056x704	8.7	28.7	92.01
	UNet [34]	15.3	1056x704	5.8	33.4	91.78
EfficientNetB1	***DeepLabV3+***	22.2	1056x704	7.0	35.0	**94.14 (*)**
	MANet [34]	17.7	1056x704	11.2	44.7	92.75
	UNet [34]	17.5	1056x704	8.3	51.7	92.10
EfficientNetB2	***DeepLabV3+***	23.7	1056x704	8.1	24.1	**93.25**
	MANet [34]	18.0	1056x704	12.9	33.8	92.23
	UNet [34]	17.9	1056x704	9.5	38.8	92.15
ResNet-50 [15]	***DeepLabV3+***	28.0	1056x704	26.7	107.1	92.07
ResNet-101 [15]	***DeepLabV3+***	31.9	1056x704	46.7	183.4	92.93
MobileNetV2 [31]	***DeepLabV3+***	13.2	1056x704	4.4	17.8	88.96

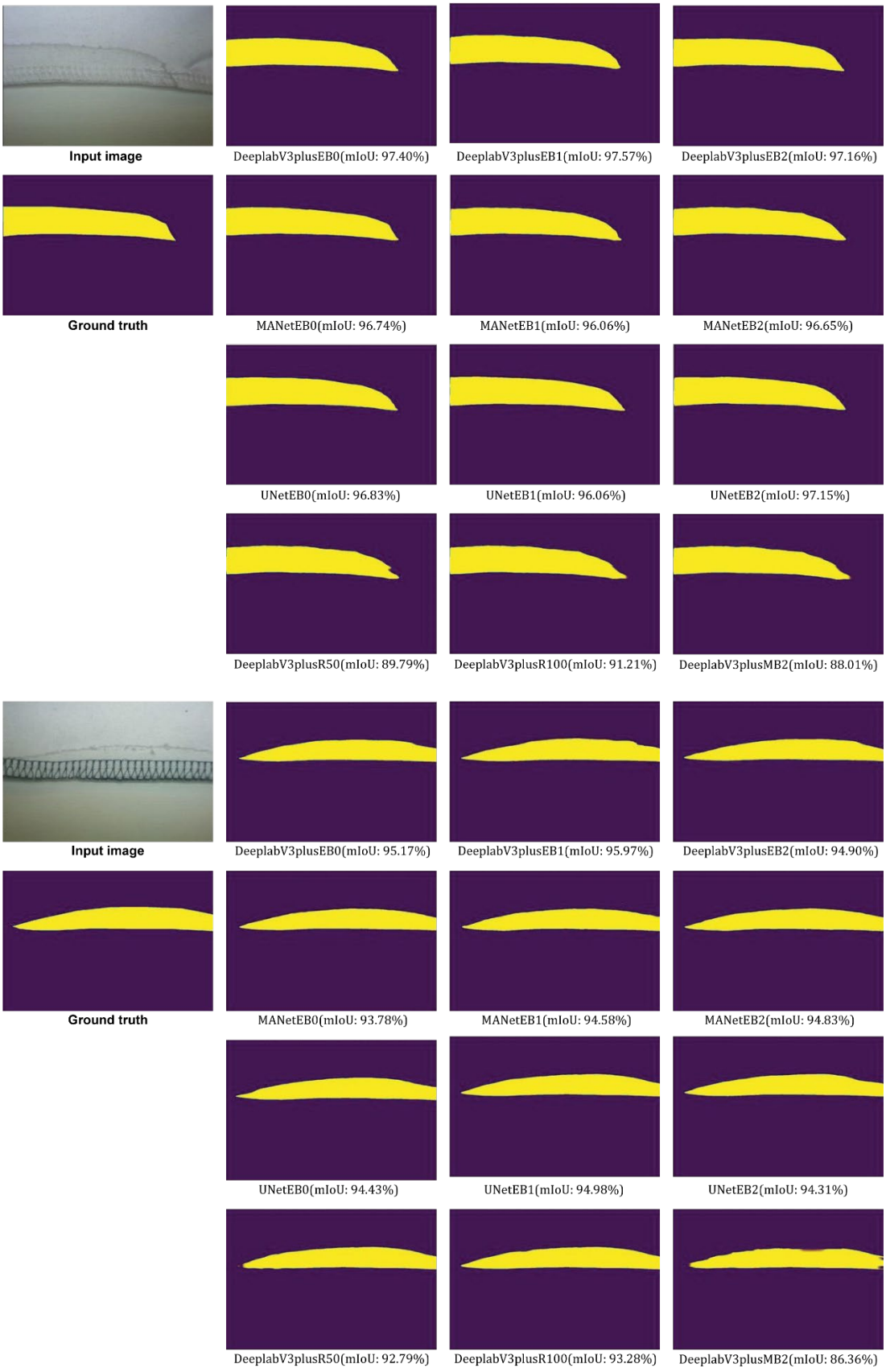


Figure 12. Sample predictions on experimental models.

6 Conclusion and Future work

In this research, an image semantic segmentation technology has been used to classify defective sewing stitches at the pixel level, achieving the desired goal and separation effect. The DeepLabV3+ semantic segmentation architecture was chosen during the experiment due to its superior performance, and its segmentation principle and advantages are systematically explained in this article. The semantic segmentation of defective sewing stitches was achieved using the DeepLabV3+ semantic segmentation model in conjunction with the EfficientNet family (B0-B2) and Resnet101 as a feature extraction network. The best model is DeepLabV3+ and EfficientNetB1 acquired *MeanIoU:* 94.14% achieving the maximum separation effect within the hardware environment's allowable range. My future research will enhance the DeepLabV3+ model's performance, functionality for sewing inspection in factories, and semantic segmentation to detect various defective categories for other types of clothing or accessories. Bags and wallets are examples.

Acknowledgments

I would like to express my appreciation to Professor Seongwon Cho, and HAIL (Hongik University Artificial Intelligence Laboratory, Seoul, Republic of Korea) supervised by Professor Seongwon Cho for supporting me to conduct this research.

References

[1] V. Badrinarayanan, A. Kendall, R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 39, no 12, 2481-2495, 2017. https://doi.org/10.1109/TPAMI.2016.2644615

[2] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia. Pyramid scene parsing network. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* volume 2017, 6230-6239, 2017. https://doi.org/10.48550/arXiv.1612.01105

[3] S. Bargoti, J. P. Underwood. Image segmentation for fruit detection and yield estimation in apple orchards. *J. Field Robot.*, volume 34, no. 6, 1039-1060, 2017. https://doi.org/10.1002/rob.21699

[4] M. Tan, Q. V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97, 6105-6114, 2019. https://doi.org/10.48550/arXiv.1905.11946

[5] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*, 801-818, 2018. https://doi.org/10.48550/arXiv.1802.02611

[6] H. Peng, C. Xue, Y. Shao, K. Chen, J. Xiong, Z. Xie, L. Zhang. Semantic Segmentation of Litchi Branches Using DeepLabV3+ Model. *IEEE Access*, volume 8, 2020. https://doi.org/10.1109/ACCESS.2020.3021739

[7] K. Chen, M. Wei, X. Chen, J. Yang, Y. Pei, S. Li. Fault Feature Extraction Method of Rotating Machinery Based on DeepLabV3+ Semantic Segmentation. *Global Reliability and Prognostics and Health Management (PHM-Nanjing)*, 2021. https://doi.org/10.1109/PHM-Nanjing52125.2021.9613104

[8] L. C. Chen, G. Papandreou, F. Schroff, H. Adam. Rethinking Atrous Convolution for Semantic Image Segmentation, *arXiv preprint*, 2017. https://doi.org/10.48550/arXiv.1706.05587

[9] I. Ahmed, M. Ahmad, F. A. Khan, M. Asif. Comparison of Deep-Learning-Based Segmentation Models: Using Top View Person Images. *IEEE Access*, 2020. https://doi.org/10.1109/ACCESS.2020.3011406

[10] Y. Chen, T. Yang, X. Zhang, G. Meng, X. Xiao, J. Sun. DetNAS: Backbone Search for Object Detection. *Advances in Neural Information Processing Systems 32 (NeurIPS)*, volume 32, 2019. https://doi.org/10.48550/arXiv.1903.10979

[11] R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 580-587, 2014. https://doi.org/10.48550/arXiv.1311.2524

[12] Z. X. Guo, W. K. Wong, S. Y. S. Leung, M. Li. Applications of artificial intelligence in the apparel industry: a review. *Textile Research Journal*, volume 81, no 18, 1871-1892, 2011. https://doi.org/10.1177/0040517511411968

[13] A. Hamja, M. Maalouf, P. Hasle. The effect of lean on occupational health and safety and productivity in the garment industry – a literature review. *Production & Manufacturing Research*, volume 7, no 1, 316-334, 2019. https://doi.org/10.1080/21693277.2019.1620652

[14] Y. J. Han, H. J. Yu. Fabric Defect Detection System Using Stacked Convolutional Denoising Auto-Encoders Trained with Synthetic Defect Data. *Applied Sciences*, volume 10, no 7, 2020. https://doi.org/10.3390/app10072511

[15] K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778, 2016. https://doi.org/10.1109/CVPR.2016.90

[16] H. Kim, H. Lee, J. S. Kim, S. H. Ahn. Image-based failure detection for material extrusion process using a convolutional neural network. *Proceedings of The International Journal of Advanced Manufacturing Technology*, 2020. https://doi.org/10.1007/s00170-020-06201-0

[17] H. Yaya, E. Irwansyah, E. Miranda, H. Soprano, K. Hashimoto. The effect of resnet model as feature extractor network on the performance of the DeepLabV3 model for semantic satellite image segmentation. *Proceedings of 2020 IEEE Asia-Pacific Conference on Geoscience, Electronics and Remote Sensing Technology (AGERS)*, 74-77, 2020. https://doi.org/10.1109/AGERS51788.2020.9452768

[19] N. Audebert, B. L. Saux, S. Lefèvre. Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images. *Remote Sensing*, Volume 9, no 4, 2017. https://doi.org/10.3390/rs9040368

[20] Z. Jun, C. Shiqiao, D. Zhenhua, L. Chabei. Automatic Detection for Dam Restored Concrete Based on DeepLabv3+. *Proceedings of IOP Conference Series: Earth and Environmental Science*, volume 571, no 1, 012108, 2020. https://doi.org/10.1088/1755-1315/571/1/012108

[21] K. Hanbaya, M. F. Talub, Ö. F. Özgüven. Fabric defect detection systems and methods—A systematic literature review. *Optik*, volume 127, 11960-11973, 2016. https://doi.org/10.1016/j.ijleo.2016.09.110

[22] H. Y. T. Ngan, G. K. H. Pang, N. H. C. Yung. Automated fabric defect detection—A review. *Image and Vision Computing*, volume 29, 442-458, 2011. https://doi.org/10.1016/j.imavis.2011.02.002

[23] H. Kim, W. K. Jung, Y. C. Park, J. W. Lee, S. H. Ahn. Broken stitch detection method for sewing operation using CNN feature map and image-processing techniques. *Expert Systems with Applications*, volume 188, 2022. https://doi.org/10.1016/j.eswa.2021.116014

[24] W. K. Wong, C. W. M. Yuen, D. D. Fan, L. K. Chan, E. H. K. Fung. Stitching defect detection and classification using wavelet transform and BP neural network. *Expert Systems with Applications*, volume 36, 3845-3856, 2009. https://doi.org/10.1016/j.eswa.2008.02.066

[25] W. K. Jung, H. Kim, Y. C. Park, J. W. Lee, E. S. Suh. Real-time data-driven discrete-event simulation for garment production lines. *Production Planning & Control*, volume 33, no 5, 2022. https://doi.org/10.1080/09537287.2020.1830194

[26] B. Hariharan, P. Arbeláez, R. Girshick, J. Malik. Simultaneous Detection and Segmentation. *Proceedings of European Conference on Computer Vision (ECCV)*, 297-312, 2014. https://doi.org/10.48550/arXiv.1407.1808

[27] L. Xu, J. S. Ren, C. Liu, J. Jia. Deep Convolutional Neural Network for Image Deconvolution. *Advances in Neural Information Processing Systems 27 (NIPS)*, volume 27, 2014. https://doi.org/10.48550/arXiv.1407.1808

[28] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 40, no 4, 834-848, 2018. https://doi.org/10.1109/TPAMI.2017.2699184

[29] F. Milletari, N. Navab, S. A. Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. *Proceedings of Fourth International Conference on 3D Vision (3DV)*, 565-571, 2016. https://doi.org/10.1109/3DV.2016.79

[30] A. F. Agarap. Deep Learning using Rectified Linear Units (ReLU). *arXiv preprint*, 2019. https://doi.org/10.48550/arXiv.1803.08375

[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510-4520, 2018. https://doi.org/10.1109/CVPR.2018.00474

[32] J. Lee, T. Won, T. K. Lee, H. Lee, G. Gu, K. Hong. Compounding the Performance Improvements of Assembled Techniques in a Convolutional Neural Network. *arXiv preprint*, 2020. https://doi.org/10.48550/arXiv.2001.06268

[33] J. Long, E. Shelhamer, T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431-3440, 2015. https://doi.org/10.1109/CVPR.2015.7298965

[34] L. Rui, S. Zheng, C. Zhang, C. Duan, J. Su, L. Wang, P. M. Atkinson. Multi-Attention-Network for semantic segmentation of fine-resolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing 60*, 1-13, 2021. https://doi.org/10.1109/TGRS.2021.3093977

[34] R. Olaf, P. Fischer, T. Brox. U-net: Convolutional networks for biomedical image segmentation. *Proceedings of International Conference on Medical image computing and computer-assisted intervention*, 234-241, 2015. https://doi.org/10.48550/arXiv.1505.04597