ICIC Express Letters
Volume **16**, Number **1**, January **2022**

ICIC International ©2022 ISSN 1881-803X
pp. **75–82**

UNET++ WITH SCALE PYRAMID FOR CROWD COUNTING

Marcellino[1], Tjeng Wawan Cenggoro[2,3,*] and Bens Pardamean[1,3]

[1]Computer Science Department, BINUS Graduate Program – Master of Computer Science
[2]Computer Science Department, School of Computer Science
[3]Bioinformatics and Data Science Research Center
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
marcellino003@binus.ac.id; bpardamean@binus.edu
*Corresponding author: bdsrc@binus.edu

Received March 2021; accepted June 2021

Abstract. *Crowd counting is a popular study that offers beneficial applications in various fields. Despite the benefits, developing a crowd counting application with high accuracy is challenging because of the variation in the training images, such as the density of the crowd, the perspective distortion, and the camera position. To address this challenge, a better crowd counting method needs to be developed. In this study, we propose a deep learning method based on UNet++ for crowd counting. We used VGG as the backbone and add the Scale Pyramid module as the transition between VGG and UNet++. The result of the experiment reveals that our proposed method achieved state-of-the-art performance in the ShanghaiTech Part A dataset with an MAE of 54.8 and an MSE of 85.4.*

Keywords: Crowd counting, Deep learning, Transfer learning, UNet++

1. **Introduction.** The application of computer vision is massively deployed in real-world cases, especially in developing countries [1]. Among the sub-field of computer vision, crowd counting is one of the most prolific generators of real-world applications. Examples include camera surveillance, seeing the crowd of a place, traffic, and the estimated number of people in a place. Many factors influence crowd counting such as the weather condition [2], the crowd density [3], the people size variation [4], and the camera placement [5,6]. The common datasets for crowd counting are ShanghaiTech Part A and B [7], World Expo [8], UCF_CC_50 [9], UCSD [10], and Mall dataset [11]. Among those datasets, the state-of-the-art method for ShanghaiTech Part A dataset has noticeably large MAE (Mean Absolute Error) and MSE (Mean Squared Error) [12].

The method used for crowd counting is dominated by the use of Convolutional Neural Networks (CNN) [13] and Generative Adversarial Networks (GAN) [14]. Several methods have a higher level of accuracy compared to the previous method such as the Adaptive Dilated Self Correction (ADSCNet) method [15]. This method produces an MAE of 55.4 and an MSE of 97.7 in the ShanghaiTech Part A dataset. In the image, crowd counting has a human shape pattern. The high density of people makes the people shown in the picture only the head, this also forms a pattern of uniformity. In the image, the cerebral tissue also has pattern tissue that has uniformity. In researching cerebral tissue images using UNet++, it produces segmentation with a dice score of 0.91 [16], so this method is suitable for use in crowd counting. This method is widely used for segmentation in the medical field, such as making architecture for segmenting medical images [7,8].

Because of the similarity between image segmentation and crowd counting, we proposed to use UNet++ for crowd counting. To enhance the performance of UNet++ for scale variation, we added the Scale Pyramid module [6] between the backbone and the UNet++

DOI: 10.24507/icicel.16.01.75

head. The backbone we used is a pretrained VGG [18], which is suitable for transfer learning in crowd counting [15]. The transfer learning approach is employed in this study because it has shown a benefit to improve deep learning models in various domains [19,20]. The proposed method achieved 54.8 MAE and 85.40 MSE.

The rest of the paper is organized as follows: Section 2 presents the related work, Section 3 presents the preparation and methodology of results, Section 4 presents the outcome and discussion, and the final section holds the conclusion.

2. **Related Work.**

2.1. **Crowd counting methods.** CNN architecture is a deep learning architecture that is often used in crowd counting. Liu et al. [21] used Context-Aware Network (CAN) that makes it possible to detect based on context by getting contrast from the image and used in training and detection processes, making it possible to divide the type of image based on color contrast such as weather. Sindagi and Patel [22] used the Hierarchical Attention-based Crowd Counting Network (HA-CCN) method. This method overcomes the disadvantages that previous methods using multi-scale were not effective for crowd counting. Scale determination used features obtained from image extraction. Next is Ma et al. [23] by using the Bayesian Loss (BL) method. This method used an entropy map that makes a density map use the color of the entropy so that the color is warmer, the density is higher. Then there are also Shi et al. [24] who use the Counting with Focus for Free (CFF) method. This method combines the Focus from segmentation method with Focus from global density which combines segmentation map with density map and produces an annotation map. The results have higher accuracy than using only one method that tends to have different accuracy in different datasets. Next, Chen et al. [25] used Scale Pyramid Network (SPN). What distinguishes this method from other neural networks is that they use multiple scales in each image. However, not only that, they use different kernels for each scale, so the size of the kernel affects the scale. Then there is also research by combining the Scale Preserving Network (SPN) method with the Learning to Scale Module (L2SM) [26]. L2SM gets input in the form of a density map generated by the previous SPN method. Other researchers, namely Liu et al. [27] used the Deep Structured Scale Integration Network (DDSINet) method. The method uses three subnetworks, namely by making two scales that are lower than the original image scale for training. The function of conducting training on data with different scales is also to vary the scale of the training process so that it can also do crowd counting at different scales.

Sindagi and Patel [28] used the Multi-level Bottom-up and Top-Bottom Fusion (MBT-TBF) method. In this method, they add layers at the time of the fuse so that the fuse becomes multi-level or has a second fuse layer. So, the fuse layer from the bottom-top and top-bottom will be connected to the final layer. Whereas Cheng et al. [2] used the Spatial Awareness Network (SPANet) method. The method adds a feature extraction to limit the background and existing noise so that what is seen or can be recognized is a human object. The method also conducts training in parallel as a branch called multi-branch architecture. Regression, which is a method of estimation, is also used in crowd counting research such as that conducted by Xiong et al. [29]. They detect the test interval values, and then take the median as the value from the density map. Yan et al. [30] prioritize perspective estimation, and the goal is to get a good perspective and better scale detection. Furthermore, the development was carried out by Bai et al. [15] by using the adaptive dilated convolution method and adding a self correction module to make corrections to the resulting density map values. This method succeeded in reducing the MAE value from the previous method.

2.2. **UNet++.** UNet++ is a method that is often used in deep learning in medical research such as Electron Microscopy (EM) image segmentation in the brain which aims to understand brain networks [16], architectural development for segmentation in medical images [31], and began to develop research in other fields such as change detection [17]. Change detection research involves detecting changes in ground level from satellite imagery. This research compares the results of segmentation before and after changes that occur in the soil surface. UNet++ is a nested or systematic iteration of the UNet architecture and consistently outperforms UNet and wide UNet [31].

3. Proposed Method.

3.1. **Dataset.** We used ShanghaiTech Part A [7] in this study. Table 1 shows the statistics of the ShanghaiTech Part A crowd image dataset for model evaluation. The ground truth density maps are provided in the dataset file as a CSV file with density value per pixel.

Table 1. Statistics of ShanghaiTech Part A dataset

Dataset	**Type**	**Total images**	**Min**	**Max**	**Mean**
ShanghaiTech A	Train	300	32	3,135	541
ShanghaiTech A	Test	182	65	2,255	433

The training dataset contains 300 images. The number of people ranged between 32 and 3,135 people with an average of 541 people. The testing dataset contains 182 images. The number of people ranged between 65 and 2,255 people with an average of 433 people. With the size of the ShanghaiTech Part A dataset, it is assumed to be large enough for a downstream task like crowd counting. The size of the dataset needs to be considered because it plays an important role in the performance of a trained deep learning model [32].

3.2. **Deep learning architecture.** Our proposed method is UNet++ with VGG backbone and Scale Pyramid module as the transition between VGG and UNet++. The advantage of our method is combining the Scale Pyramid module that can generate variations of image scales with UNet++ that can segment objects with the same pattern. Figure 1 shows the training process of our proposed method that used UNet++ with a VGG backbone and the Scale Pyramid module. Repetition is performed on each image used for training. Each image will be taken with a pixel value and then converted to a grayscale value which will then be input for subsequent processing. The grayscale result will take the value of each pixel and turn these values into the input layer. The first process at the input layer is to pool the layers on the VGG backbone first. The output layer from VGG will be the input layer for the Scale Pyramid module. In the Scale Pyramid module, CNN conducted parallel with dilation 1, 12, 24, 36 and the average pool. The five layers will be concatenated into one layer. In addition, the results of the previous VGG backbone are also inputted to the Contextual module.

The results of the concatenate of the five previous layers will be combined with the results from the Contextual module layer and become one output layer before becoming the input layer in the UNet++ layer module, which will be upscaling with a scale of 8 times. Upscaling is made to equalize the dimensions with the dimensions of the first input image of 576 × 768 and it is easy to downscale.

The result of the upscaled becomes the input layer on UNet++ which is marked with the $x^{0,0}$ symbol. At $\mathcal{L}1$ it is shown that $x^{0,0}$ as the input layer is executed downsampling to $x^{1,0}$, then concatenate between $x^{0,0}$ and $x^{1,0}$ which has been upscaled by 2 times to $x^{0,1}$, and this result becomes the first output. $\mathcal{L}2$ is equal to $\mathcal{L}1$ with the addition of downsampling from $x^{1,0}$ to $x^{2,0}$. After that, concatenate between $x^{1,0}$ and $x^{2,0}$ which has been scaled up

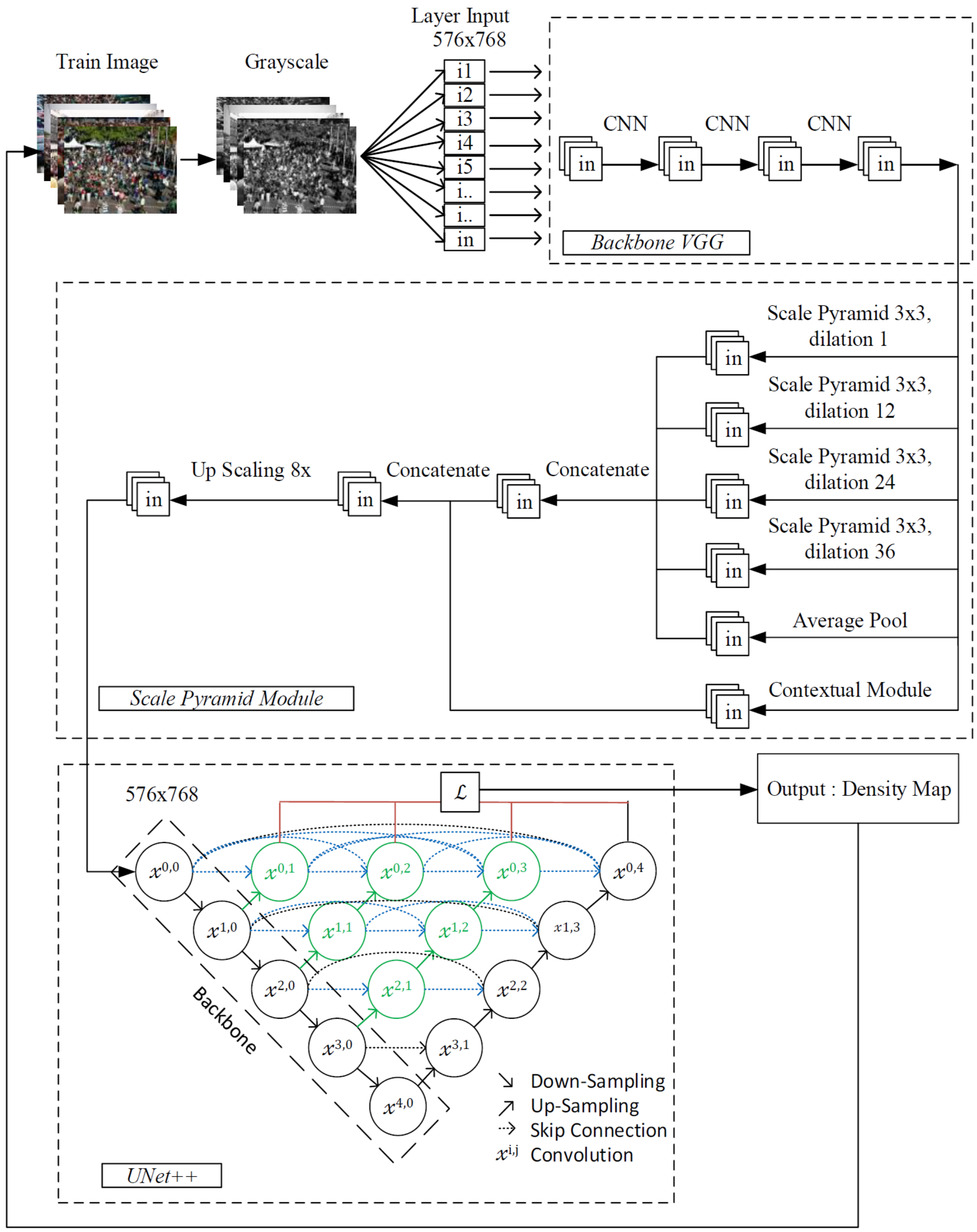


FIGURE 1. UNet++ with Scale Pyramid module

to 2 times to $x^{1,1}$. Concatenate is done again between $x^{0,1}$ with $x^{1,1}$ which has been upscaled times 2 to $x^{0,2}$, and this result becomes the second output. $\mathcal{L}3$ is equal to $\mathcal{L}2$ with the addition of downsampling from $x^{2,0}$ to $x^{3,0}$. After that, concatenate between $x^{2,0}$ and $x^{3,0}$ which has been upscaled by 2 times to $x^{2,1}$. After that, do the concatenate between $x^{1,1}$ with $x^{2,1}$ which has been up the scale times 2 to $x^{1,2}$. Concatenate is executed again between $x^{0,2}$ with $x^{1,2}$ which has been upscaled times 2 to $x^{0,3}$, and this result becomes the third output. $\mathcal{L}4$ is equal to $\mathcal{L}3$ with the addition of downsampling from $x^{3,0}$ to $x^{4,0}$. After that, concatenate between $x^{3,0}$ and $x^{4,0}$ which has been upscaled by 2 times to $x^{3,1}$. Then do the concatenate between $x^{2,1}$ with $x^{3,1}$ which has been up the scale times 2 to $x^{2,2}$. Next do the concatenate between $x^{1,2}$ with $x^{2,2}$ which has been up the scale times

2 to $x^{1,3}$. Concatenate is executed again between $x^{0,3}$ and $x^{1,3}$ which has been upscaled times 2 to $x^{0,4}$, and this result becomes the fourth output. The output layer is marked with the $\mathcal{L}$ symbol. This output layer becomes the density map of the UNet++ method. The density map is calculated to obtain the density count value from the image.

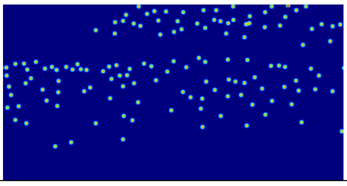
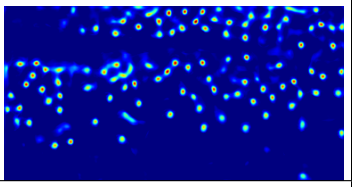
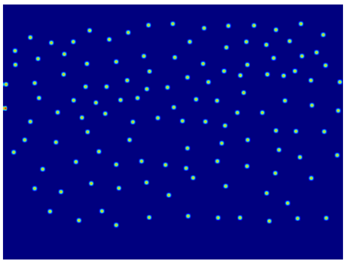
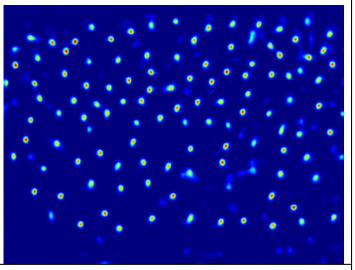
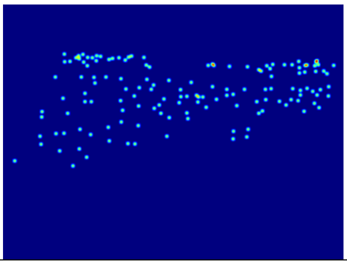
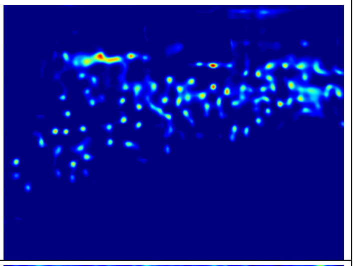
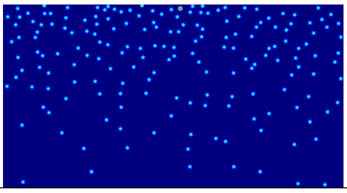
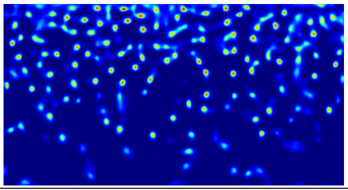
3.3. **Evaluation metric.** For crowd counting, Mean Absolute Error (MAE) and Mean Squared Error (MSE) are two metrics widely adopted to evaluate the performance as shown in Equations (1) and (2) where n is the total number of the testing images, f_i and γ_i are the estimated count and the ground truth count of the ith image respectively. In particular, f_i is calculated by summing the approximate density map values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - \gamma_i| \tag{1}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f_i - \gamma_i)^2 \tag{2}$$

4. **Experimental Results.** Table 2 shows four samples of original images, ground truth density maps with the density map derived from the UNet++ method. The first column is the original image testing ShanghaiTech Part A dataset. The second column is the ground truth density map provided by the dataset. The third column is the density map that is generated from the UNet++ model. The density of people in the image is visualized in a jet color map, which more intense red color indicates more density and more intense blue color indicates less density. In the case of monochrome print, more density is indicated

Table 2. (color online) Comparison density map ground truth and UNet++

Original image	**Ground truth**	**UNet++**
		
		
		
		

with a lighter color. The number of people is obtained by adding up the density value. As shown by the visualization of the results of the UNet++ density map, it approximates the visualization of the ground truth of the density map.

Table 3 shows the MAE and MSE comparison result between the top five previous methods and the UNet++ method for ShanghaiTech Part A dataset. The best performance is styled with bold and the second best is styled with bold italic. Our proposed method achieved the best MAE and MSE compared to the top 5 previous methods. Specifically, our proposed method improves the MAE of the previous state-of-the-art method, ADSCNet, from 55.40 to 54.80, or 1.08% relatively. Our proposed method also improves the MSE of the previous state-of-the-art method, PGCNet, from 86.00 to 85.40, or 0.70% relatively.

Table 3. Performance comparison for ShanghaiTech Part A dataset. The best performance is styled with bold and the second-best performance is styled with bold italic.

Method	**MAE**	**MSE**
MBTTBF-SCFB [28]	60.20	94.10
SPANet+SANet [2]	59.40	92.50
S-DCNet [29]	58.30	95.00
PGCNet [30]	57.00	***86.00***
ADSCNet [15]	***55.40***	97.70
UNet++ (Proposed method)	**54.80**	**85.40**

5. **Conclusions.** Our proposed method successfully reduces MSE and MAE among the methods from previous studies for ShanghaiTech Part A dataset. The proposed method achieved 54.80 MAE and 85.40 MSE. The MAE is improved from the previous state-of-the-art method, ADSCNet, which achieved 55.40 MAE. Additionally, our proposed method also has a better MSE than the previous state-of-the-art method, PGCNet, which has 86.00 MSE. Relatively, our proposed method improves the performance of MAE and MSE for the ShanghaiTech Part A dataset from the previous state-of-the-art method by 1.08% and 0.70%, respectively. A promising future direction is to employ guided filtering [33] to reduce noises in the images for improved performance. It has been proved to increase performance for vehicle counting [34].

REFERENCES

[1] K. Muchtar, F. Rahman, T. W. Cenggoro, A. Budiarto and B. Pardamean, An improved version of texture-based foreground segmentation: Block-based adaptive segmenter, *Procedia Comput. Sci.*, vol.135, no.9, pp.579-586, 2018.

[2] Z.-Q. Cheng, J.-X. Li, Q. Dai, X. Wu and A. G. Hauptmann, Learning spatial awareness to improve crowd counting, *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp.6152-6161, 2019.

[3] Z. Shi et al., Crowd counting with deep negative correlation learning, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.5382-5390, 2018.

[4] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu and X. Yang, Crowd counting via adversarial cross-scale consistency pursuit, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.5245-5254, 2018.

[5] B. Pardamean, H. H. Muljo, F. Abid, Herman, A. Susanto and T. W. Cenggoro, RHC: A dataset for in-room and out-room human counting, *Procedia Comput. Sci.*, vol.179, pp.33-39, 2021.

[6] B. Pardamean, H. H. Muljo, T. W. Cenggoro, B. J. Chandra and R. Rahutomo, Using transfer learning for smart building management system, *J. Big Data*, vol.6, no.1, p.110, 2019.

[7] Y. Zhang, D. Zhou, S. Chen, S. Gao and Y. Ma, Single-image crowd counting via multi-column convolutional neural network, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.589-597, 2016.

[8] C. Zhang, H. Li, X. Wang and X. Yang, Cross-scene crowd counting via deep convolutional neural networks, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.833-841, 2015.

[9] K. S. Venkatesh and A. Bansal, People counting in high density crowds from still images, *Int. J. Comput. Electr. Eng.*, vol.7, no.5, pp.316-324, 2015.

[10] A. B. Chan, Z.-S. J. Liang and N. Vasconcelos, Privacy preserving crowd monitoring: Counting people without people models or tracking, *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, DOI: 10.1109/CVPR.2008.4587569, 2008.

[11] K. Chen, C. C. Loy, S. Gong and T. Xiang, Feature mining for localised crowd counting, *Proc. of the British Machine Vision Conference*, pp.21.1-21.11, 2012.

[12] T. W. Cenggoro, Deep learning for crowd counting: A survey, *Eng. Math. Comput. Sci. J.*, vol.1, no.1, pp.17-28, 2019.

[13] Y. LeCun et al., Backpropagation applied to handwritten zip code recognition, *Neural Comput.*, vol.1, no.4, pp.541-551, 1989.

[14] I. J. Goodfellow et al., Generative adversarial networks, *Adv. Neural Inf. Process. Syst.*, vol.3, no.11, pp.2672-2680, 2014.

[15] S. Bai, Z. He, Y. Qiao, H. Hu, W. Wu and J. Yan, Adaptive dilated network with self-correction supervision for counting, *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.4594-4603, 2020.

[16] S. Roy, A. Panda and R. Naskar, Unsupervised ground truth generation for automated brain EM image segmentation, *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp.66-71, 2019.

[17] D. Peng, Y. Zhang and H. Guan, End-to-end change detection for high resolution satellite images using improved UNet++, *Remote Sens.*, vol.11, no.11, DOI: 10.3390/rs11111382, 2019.

[18] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *The International Conference on Learning Representations*, pp.1-14, 2015.

[19] B. Pardamean, T. W. Cenggoro, R. Rahutomo, A. Budiarto and E. K. Karuppiah, Transfer learning from Chest X-Ray pre-trained convolutional neural network for learning mammogram data, *Procedia Comput. Sci.*, vol.135, pp.400-407, 2018.

[20] T. W. Cenggoro, S. M. Isa, G. P. Kusuma and B. Pardamean, Classification of imbalanced land-use/land-cover data using variational semi-supervised learning, *2017 International Conference on Innovative and Creative Information Technology (ICITech)*, pp.1-6, 2017.

[21] W. Liu, M. Salzmann and P. Fua, Context-aware crowd counting, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[22] V. A. Sindagi and V. M. Patel, HA-CCN: Hierarchical attention-based crowd counting network, *IEEE Trans. Image Process.*, vol.29, pp.323-335, 2019.

[23] Z. Ma, X. Wei, X. Hong and Y. Gong, Bayesian loss for crowd count estimation with point supervision, *Proc. of IEEE Int. Conf. Comput. Vis.*, pp.6141-6150, 2019.

[24] Z. Shi, P. Mettes and C. G. M. Snoek, Counting with focus for free, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp.4200-4209, 2019.

[25] X. Chen, Y. Bin, N. Sang and C. Gao, Scale pyramid network for crowd counting, *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp.1941-1950, 2019.

[26] C. Xu, K. Qiu, J. Fu, S. Bai, Y. Xu and X. Bai, Learn to scale: Generating multipolar normalized density maps for crowd counting, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp.8382-8390, 2019.

[27] L. Liu, Z. Qiu, G. Li, S. Liu, W. Ouyang and L. Lin, Crowd counting with deep structured scale integration network, *Proc. of IEEE Int. Conf. Comput. Vis.*, pp.1774-1783, 2019.

[28] V. A. Sindagi and V. M. Patel, Multi-level bottom-top and top-bottom feature fusion for crowd counting, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp.1002-1012, 2019.

[29] H. Xiong, H. Lu, C. Liu, L. Liu, Z. Cao and C. Shen, From open set to closed set: Counting objects by spatial divide-and-conquer, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp.8362-8371, 2019.

[30] Z. Yan et al., Perspective-guided convolution networks for crowd counting, *Proc. of IEEE Int. Conf. Comput. Vis.*, pp.952-961, 2019.

[31] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh and J. Liang, UNet++: A nested U-Net architecture for medical image segmentation, in *Deep Learning in Medical Image Analysis and Multimodal Learning*

for Clinical Decision Support. DLMIA 2018, ML-CDS 2018. Lecture Notes in Computer Science, D. Stoyanov et al. (eds.), Cham, Springer, 2018.

[32] T. W. Cenggoro, F. Tanzil, A. H. Aslamiah, E. K. Karuppiah and B. Pardamean, Crowdsourcing annotation system of object counting dataset for deep learning algorithm, *IOP Conf. Ser. Earth Environ. Sci.*, vol.195, no.1, 2018.

[33] K. He, J. Sun and X. Tang, Guided image filtering, *European Conference on Computer Vision*, pp.1-14, 2010.

[34] B. Setiyono, D. R. Sulistyaningrum, IGN R. Usadha and A. P. Nusantara, The rain noise reduction using guided filter to improve performance of vehicle counting, *International Journal of Innovative Computing, Information and Control*, vol.16, no.4, pp.1353-1370, 2020.