ENSEMBLE METHOD FOR SMALL OBJECT SEGMENTATION

LAM KEN LUN
AZIZI ABDULLAH

Fakulti Teknologi & Sains Maklumat, Universiti Kebangsaan Malaysia

ABSTRAK

Object detection on tiny object is an emerging research topic useful for tracking tiny object taken from UAV (Unmmaned Aerial Vehicle). There are two prominent deep-learning approaches when it comes to object detection using convolutional neural network and fully convolutional neural network. Existing fully convolutional network network usually results in better detection but single fully convolutional network is unable to localize tiny object. The objective of this project is to study 3 existing fully convolutional network such FCN-8s, U-Net and FPN. Besides that, suggest and develop a ensemble fully convolutional network. Finally, create a web interface for visualization. Based on this project, the suggested model is an ensemble fully convolutional network of two fully convolutional network, U-Net and FPN using product or concatenation operation from the output of these two networks. The experimental results on this projects who promising results. U-Net FPN (concatenation operation) scored 0.679 in IoU higher as compared FPN that scored 0.678 in the same metric on ship class in Seagull dataset. Besides, combination of U-Net and FPN (concatenation operation) scored 0.538 in IoU which is higher as compared to the best results from a single fully convolutional network FPN that scored 0.242 on human class in UAVid dataset.

1 INTRODUCTION

Image segmentation is an computer vision technique where a system take RGB image as an input to product segmented images of each class. Segmenting millions of images is a labor intensive task to perform manually therefore automated image segmentation system was developed to be used object segmentation using images collected from UAV.

Before the introduction of deep learning, unsupervised methods were used to solve image segmentation problem in computer vision field. These methods are categorized into two groups, threshold based, and edge based. Threshold based method separate two sections of an image by using a threshold value or pixel history (Glasbey 1993). Edge based method detection the change in brightness of a pixel with its surrounding neighbor by apply filter such as Sobel (Kanopouplos, Vasanthavada & Baker 1993) and Canny (Canny 1986) to generate an edge for each section. After the introduction of deep learning to solve computer vision problem, CNN (Krizhevsky 2012) became the state-of-the-art method to extract semantic information from a given image using its deep convolutional neural network architecture.

But CNN network alone cannot be used to address image segmentation problem therefore Fully Convolutional Network (Long, Shelhamer & Darrel 2015) was introduced to solve image segmentation problem. FCN uses multiple up-sampling layer to increase the output size of the convoluted layer from down-sampling layer. After the introduction of FCN, other state of the art was proposed to solve image segmentation such as UNET (Ronnberger, Fischer & Brox 2015), UNet++ (Zhou et al. 2019), FPN (Seferbekov et al. 2018) and ensembel model (Wu et al. 2019). Therefore, 3 single model such as FCN, UNET, FPN was trained, evaluated, and analyzed. An ensemble model was proposed.

2 PROBLEM STATEMENT

Convolutional neural network is unable to detect small object because of sub-sampling Ronnberger, Fischer & Brox 2015) layer. Therefore, fully convolutional network became the most popular method to detect small object because the network contains up-sampling and down-sampling layer (Long, Shelhamer & Darrel 2015). Variant of fully convolutional network was proposed to improve existing method such as UNet dan FPN. But single fully convolutional network is unable obtain good results (Wu et al. 2019). Therefore, this paper proposed method is ensemble fully convolutional network with various combination method such as element-wise summation, element-wise multiplication, and concatenation.

3 OBJECTIVE

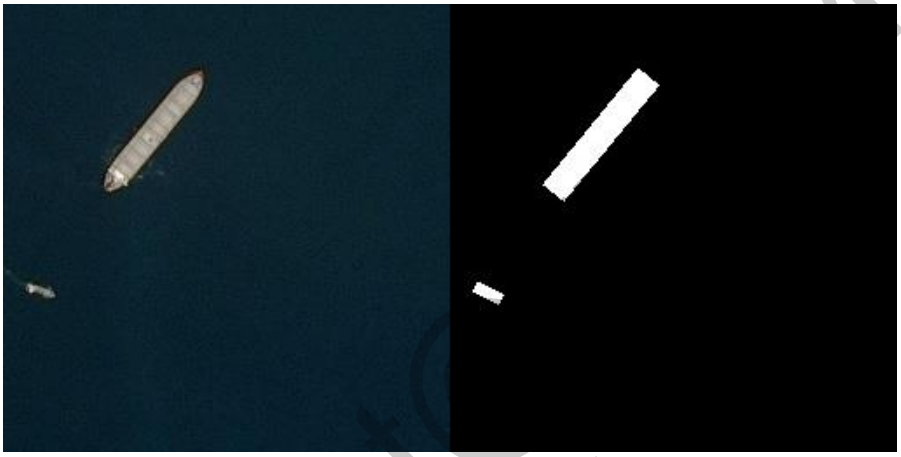
The objective of this project is to proposed an ensemble fully convolutional network for small object detection with multiple combination of fully convolutional network. In details,

I. Compare the performance of 3 popular fully convolutional network in Seagull and UAVid dataset.
II. Proposed an ensemble fully convolutional network
III. Develop a user interface for visualization

4 METHODOLOGY

To prepare for the training, each dataset is first preprocess with its respesctive methods and then trained on baseline model such as FCN8s, UNet, FPN with a predefined hyper-parameter.

4.1. DATA PREPROCESSING



Figur1: Sample of Seagull dataset image and label

To train and test all the model on Seagull dataset, images is obtained from the author with each image size of 256 x 256 pixels. Seagull dataset consisted of RGB and hyperspectral images which in grayscale while represented in RGB format. Label given is an grayscale of the of the segmented object such as ship or background as show in Figure 1. Seagull dataset contain only two classes, background and ship. In total, there is 23123 images and label.

The dataset is splitted into 80 per cent (18499 images/labels) for training and 20 per cent (4624 images/labels) for testing.

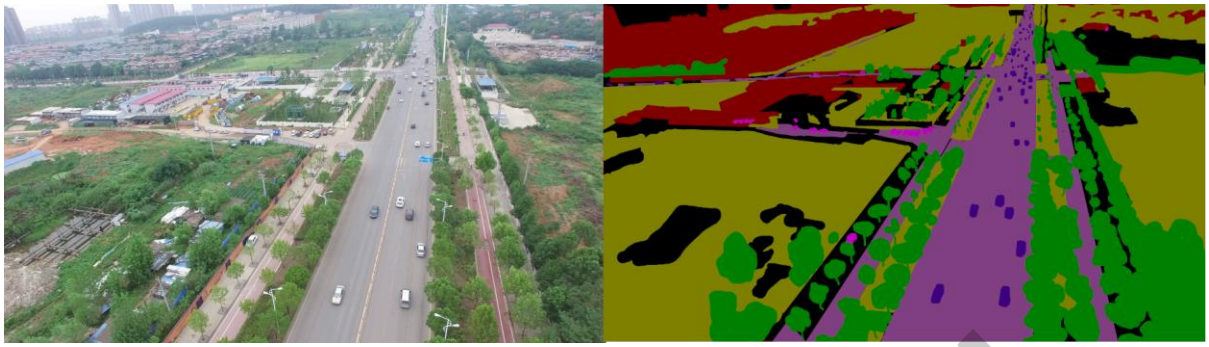


Figure 2: Sample of UAVid dataset image and label

While UAVid dataset, images is obtained from the author with image size of 3840 × 2160 pixels. This dataset consisted of only RGB images. Label is given in an one image with all 8 classes segmented in different color. As shown in Figure 2, image on the right. UAVid dataset contain 8 classes, background, building, road, static car, tree, low vegetation, human and moving car. The dataset is splitted into 80 per cent (6400 images/labels) for training and 20 per cent (2240 images/labels) for testing.

4.2. FCN8S ARCHITECTURE

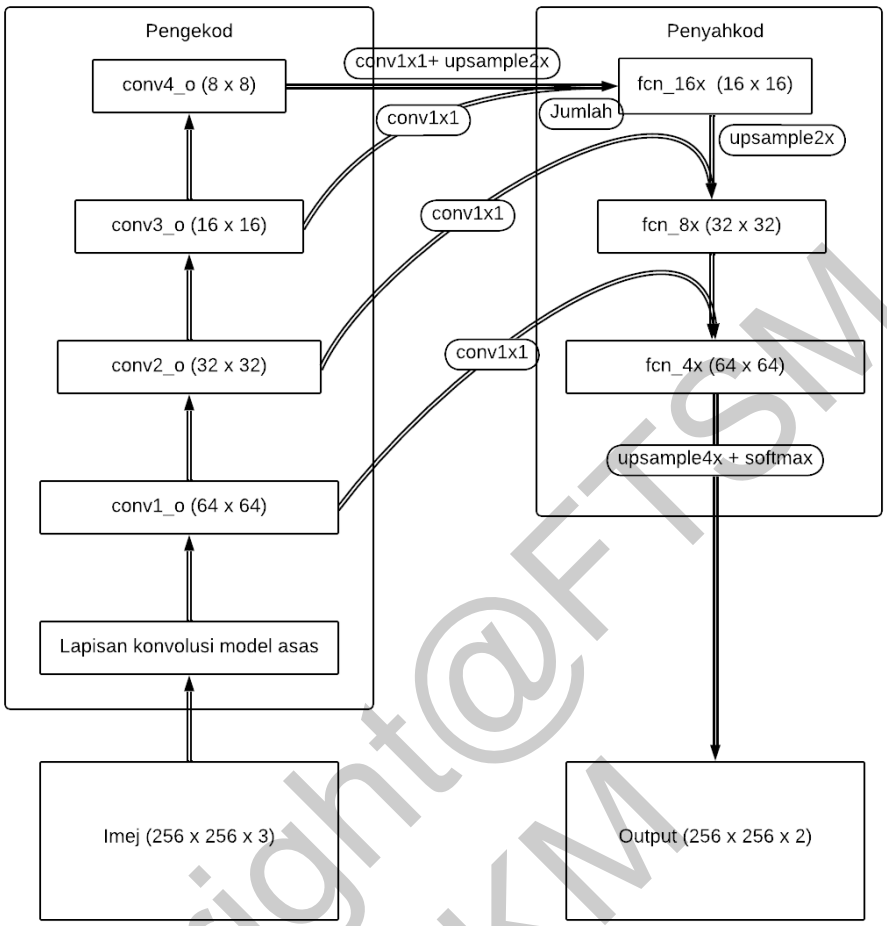


Figure 3: FCN8s Architecture

Based on Figure 3, FCN8s has an encoder, the encoder is an EfficientNetB0 pretrained on Imagenet Dataset. Four output layer is obtained from the encoder with each 1 / 2 of the spartial size of previous convolutional layer in the encoder. The output layer we gathered are conv4_o, conv3_0, conv2_o dan conv1_0. To product fcn_16, convolutional layer of with kernel of size of 1x1 is apply to conv4_o and then upsample by 2 times using a transpose convolutional layer with stride of 2 and sum with the output from conv3_0 forming fcn_16x. To get fcn_8x, fcn_16x is upsample and then sum with output from conv2_0 and to get the final output of size 256 x 256, fcn_4x is upsample with strides of 8 and softmax is then applied.

UNET ARCHITECTURE

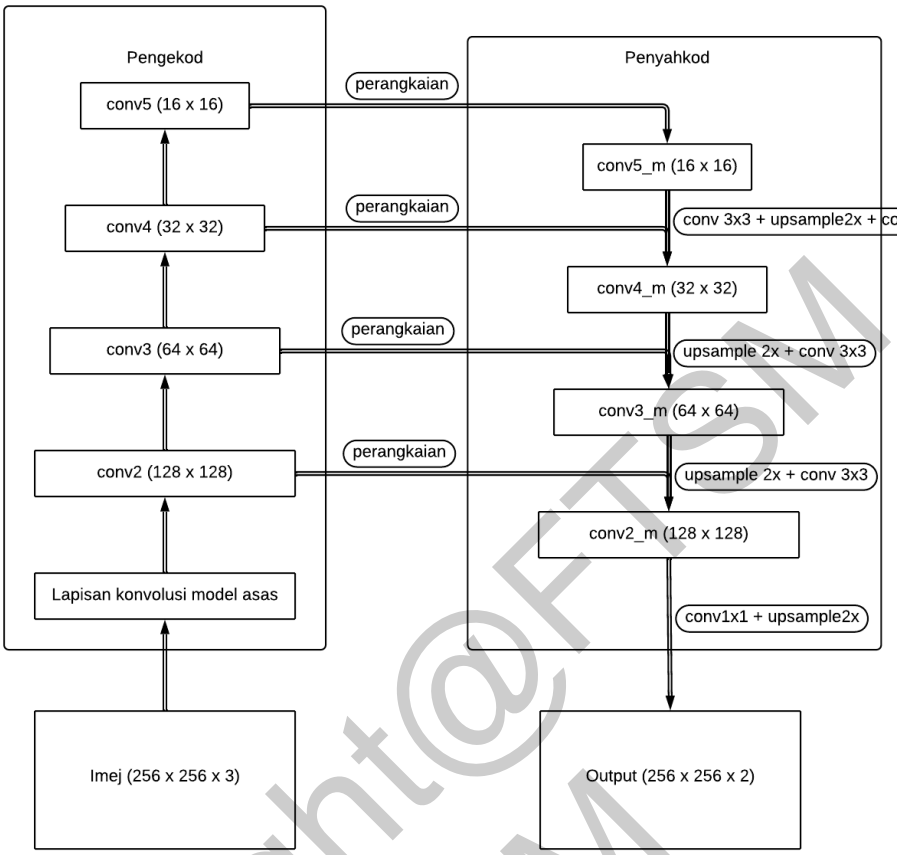


Figure 4: UNet Architecture (Ronnberger, Fischer & Brox 2015)

Based on Figure 4, UNet uses the same type of encoder as FCN8s in the previous example in Figure 3. Four output laye is obtrained from the endoer with each 1 / 2 of the spatial size of previous convolutional layer in the encoder. The output layer we gathered are conv5, conv4, conv3 dan conv2. Each output except for output conv5 is concatenated with is corresponding layer in decoder. To form conv5_m, convolutional layer is apply on conv5 to produce conv5_m. To form the subsequent layer in decoder, conv4_m is form when conv5_m was applied with convolutional layer of kernel size of 3x3 and upsample and another convolutional layer of kernel size of 3x3. For the final output, conv2_m is applied with a convolutional layer of kernel size of 1 x1 and then upsample by 2 times with a bilinear interpolation upsampling.

4.3. FPN ARCHITECTURE

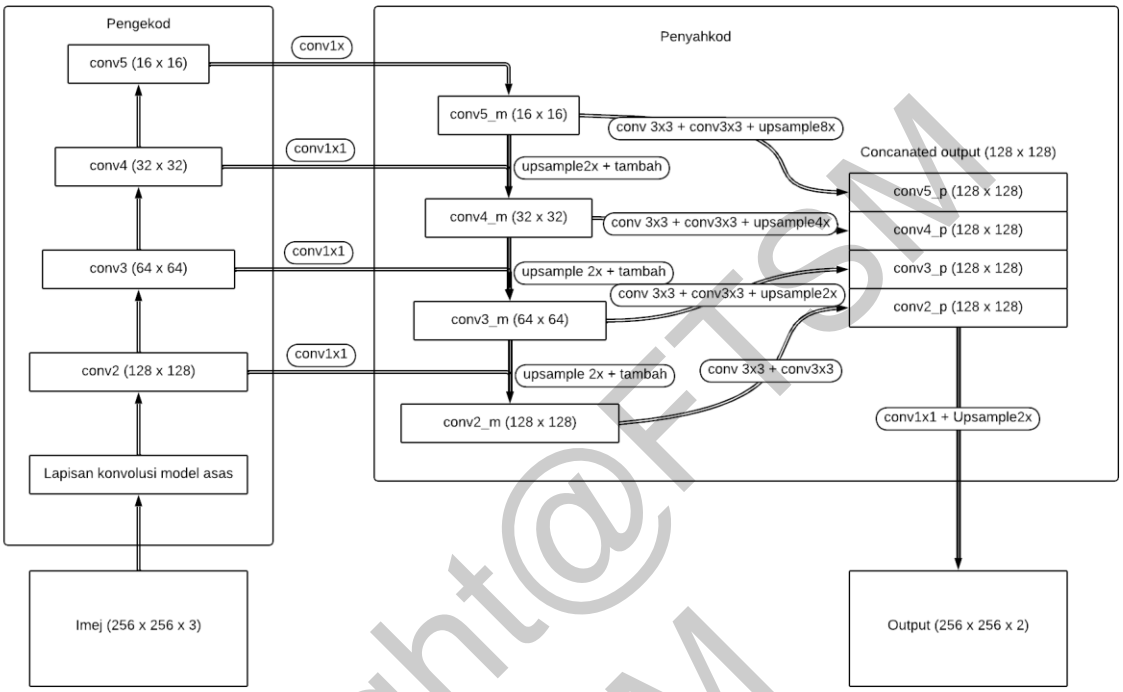


Figure 5: FPN Architecture (Seferbekov et al. 2018)

Based on Figure 5, FPN uses the same type of encoder as FCN8s and UNet in the previous example in Figure 3 and Figure 4. Four output laye is obtrained from the endoer with each 1 /2 of the spatial size of previous convolutional layer in the encoder. The output layer we gathered are conv5, conv4, conv3 dan conv2. Each output except for output conv5 is concatenated with is corresponding layer in decoder. To form conv5_m, convolutional layer is apply on conv5 to produce conv5_m. To form the subsequent layer in decoder, conv4_m is form when conv5_m was applied with convolutional layer of kernel size of 3x3 and upsample and another convolutional layer of kernel size of 3x3. Instead of forming final output from the last layer of the decoder, FPN combined all the output from each scale of the decoder and generate an output. Firstly, all layers in encoder, conv5_m, conv4_m, conv3_m and conv2_m was applied with their own convolutional 3x3 twice and then upsample 8x, 4x and 2x while leavning conv2_m untouched. All intermediate output is then concanated and then a convolutional layer of 1x1 is applied and then upsample 2x to form the final output.

Upsampling used in the intermediate layer were a nearest neigbor upsampling while the final output uses bilinear interpolation upsampling.

4.4. ENSEMBEL ARCHITECTURE

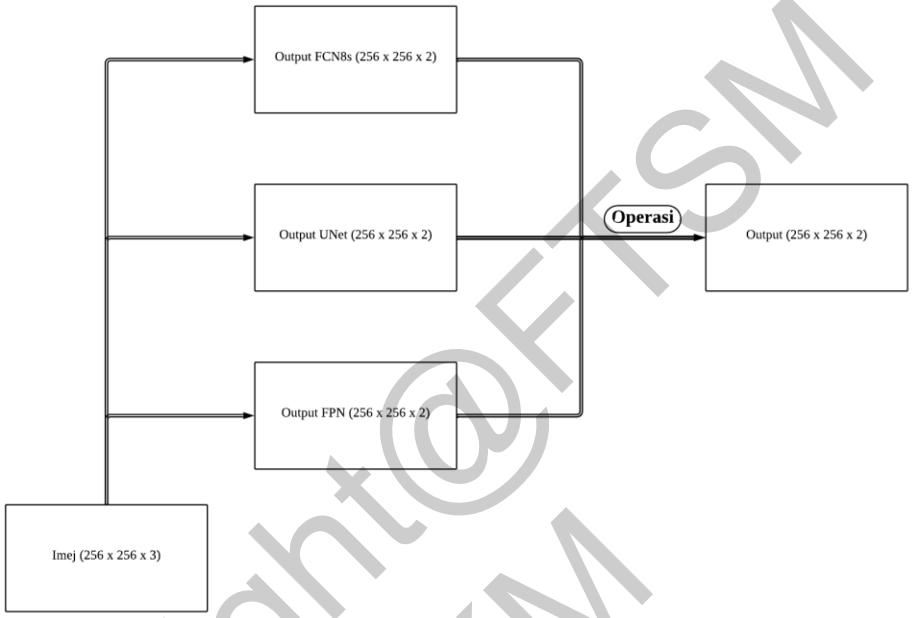


Figure 6: Ensemble Architecture

Based on Figure 6, ensemble method combine the output of each single fully convolutional network to form a ensemble model. There are several method to combined the output, in this project. The proposed method was summation, multiplication and concatenation.

4.5. PARAMETER SETUP

Every model was trained with a Adam as optimizer at the learning rate of 0.00005 and batch size of 8. For Seagull dataset, training amount is set to 35 epochs while UAVid is set to 20 epochs. Based on training observation, all the method reaches convergence at those number of epochs. No additional training is done during the training of ensemble model. Every single model in ensemble was trained together in model or end-to-end training.

5 EXPERIMENTAL RESULTS

$$skorIoU = \frac{1}{C}\sum_{c=1}^{C}\sum_{i=1}^{N}\left(\frac{y_i^c \hat{y}_i^c}{y_i^c + \hat{y}_i^c - y_i^c \hat{y}_i^c}\right)$$

Figure 7: IoU score Function

By using the same IoU score metric used by previous approaches (Seferbekov et al. 2018) and (Ronneberger, Fischer & Brox 2015) to evaluate the performance of single fully convolutional network and our proposed ensembel network. IoU measure the area of intersection between two images. IoU is also known as Jaccard Loss. Equation is shown in Figure 7. yic is the label image while y^ic is the predicted image. $yic + y\hat{}\, ic - yic\, y\hat{}\, ic$ is the area of union of two images. The closer the IoU score to 1.0, the better the model performance.

5.1. QUANTITATIVE COMPARISON OF OVERALL CLASSES

Table 1 IoU score of training and testing set for Seagull dataset. The best model score is highlighted.

Model	IoU Score (training set)	IoU Score (testing set)
FCN	0.494	0.498
UNET	0.826	0.823
FPN	**0.843**	**0.838**
UNET + FPN (multiplication)	0.838	0.838
UNET + FPN (summation)	0.821	0.826
UNET + FPN (concatenation)	0.827	0.827
FCN + FPN (multiplication)	0.762	0.763

Based on Table 1, FPN achieved the best IoU score with 0.843 as compared to other single fully convolutional model while FCN achieved the lowest IoU score of 0.494 in training set of Seagull. In the Seagull test data set, FPN also produced the highest IoU score of 0.838 while FCN producedthe lowest IoU score with 0.498 in the Seagull test data set. For ensemble convolution network, UNET + FPN (multiplication method) yields a score IoU 0.838 in the Seagull learning data set and 0.838 in the testing data set

Seagull. Whereas FCN + FPN (multiplication method) produces an IoU score lowest with 0.762 in the Seagull learning data set and an IoU score of 0.763 in the Seagull test data set.

Table 2 IoU score of training and testing set for UAVid dataset. The best model score is highlighted.

Model	IoU Score (training set)	IoU Score (testing set)
FCN	0.258	0.192
UNET	0.516	0.405
FPN	0.604	0.476
UNET + FPN (multiplication)	0.639	0.566
UNET + FPN (summation)	0.589	0.523
UNET + FPN (concatenation)	**0.646**	**0.574**
FCN + FPN (multiplication)	0.394	0.446

Based on Table 2, FPN produced the highest IoU score of 0.604 in the convolutional networksingle while FCN produced the lowest IoU score with 0.258 in the dataset UAVid learning. In the UAVid testing dataset, FPN also produced the highest IoU score which is 0.476 while FCN produces the lowest IoU score with 0.192 in the data set UAVid testing. For the ensemble convolutional network, UNET + FPN (multiplication method) yields an IoU score of 0.646 in the UAVid learning dataset and 0.574 in the testing dataset UAVid. While FCN + FPN (multiplication method) produces the lowest IoU score with 0.394 in the UAVid learning dataset and an IoU score of 0.446 in the UAVid testing dataset.

5.2. QUANTITATIVE COMPARISON OF PER CLASS

In seagull dataset, ship is defined as small object while in UAVid, human, static vehicle and moving vehicle are defined as small object for better comparison.

Table 3: IoU score of training and testing set for Seagull dataset per class. The best model score is highlighted. RK1 is FCN, RK2 is UNET, RK3 is FPN, RK4 is UNET + FPN (multiplication), RK5 is UNET + FPN (summation), RK6 is UNET + FPN (concatenation), RK7 is FCN + FPN (multiplication). TS1 is the testing IoU score for background object, TS2 is the testing IoU score for ship object. Underlined label are defined as small object.

Model	TS1	TS2
RK1	0.964	0.031
RK2	**0.998**	0.648
RK3	**0.998**	0.678
RK4	**0.998**	**0.679**
RK5	**0.998**	0.653
RK6	**0.998**	0.659
RK7	0.997	0.526

Table 4: IoU score of training and testing set for Seagull dataset per class. The best model score is highlighted. RK1 is FCN, RK2 is UNET, RK3 is FPN, RK4 is UNET + FPN (multiplication), RK5 is UNET + FPN (summation), RK6 is UNET + FPN (concatenation), RK7 is FCN + FPN (multiplication). TS1, TS2, TS3, TS4, TS5, TS6, TS7, TS8 are the testing IoU score for class background, building, road, tree, low vegetation, moving vehicle, static vehicle, human. Underlined labels are defined as small object.

Model	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8
RK1	0.296	0.538	0.478	0.472	0.158	0.021	0.020	0.001
RK2	0.421	0.699	0.589	0.633	0.397	0.337	0.335	0.140
RK3	**0.516**	**0.778**	0.656	**0.699**	0.495	0.369	0.397	0.242
RK4	0.501	0.768	**0.669**	0.687	0.467	0.412	**0.495**	0.528
RK5	0.482	0.760	0.659	0.697	0.476	0.350	0.222	**0.538**
RK6	0.445	0.748	0.607	0.670	0.421	**0.715**	0.448	0.534
RK7	0.003	0.688	0.609	0.530	**0.496**	0.291	0.349	0.193

Referring to Table 3 and 4, For the Seagull testing data set. The highest IoU score of 0.679 achieved by UNET + FPN (multiplication method) in the ship class while the second FPN high reaching an IoU score of 0.678 in the same class. While for the testing data set UAVid, IoU score of 0.538 achieved by UNET + FPN (addition method), higher than IoU score of 0.376 from FPN in the human class. Compared to above focus on small objects is the ship class from the Seagull dataset and the human class from the UAVid dataset.

5.3. DISCUSSION

With the results of the study available, it shows that FPN has an IoU score of 0.476 highest relative to other single models in UAVID. While the FCN score lowest with 0.192 in UAVID. The reason is because FCN is lacking side connection between encoder and decoder. Side information is important

for space information. To improve performance, UNET was introduced with side connection between encoder and decoder. Even a new upgrade, UNET only averaged with an IoU score of 0.405. This is also because of information the spatial is lost during the decoder phase as the decoder only outputs the volume previous layer and side connections and output on the last layer decoder.

To further improve the performance of the existing network, FPN introduced inference on each scale summing the output of the previous layer and the side connections. Each scale is consolidated and passes through the other convolution layers and then
increased by 4 times.

When comparing the proposed network, which is a combination of multiple single models. The best performing network is UNET + FPN with (unification method) with IoU by 0.574. Compared to the benchmark from (Lyu et al. 2020) which scored IoU 0.570. For small object in both Seagull and UAVid, the best performing model are the ensemble when referring to Figure 3 and 4.

6 KESIMPULAN

In conclusion, proposed model score higher IoU as compared to single model.

7 REFERENCES

Bischke, B., Helber, P., Borth, D., & Dengel, A. (2018). Segmentation of imbalanced classes in satellite imagery using adaptive uncertainty weighted class loss. IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, 6191–6194. IEEE.

Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), 679–698.

Glasbey, C. A. (1993). An analysis of histogram-based thresholding algorithms. CVGIP: Graphical models and image processing, 55(6), 532–537.

Gupta, S., Arbelez, P., Girshick, R., & Malik, J. (2015). Indoor scene understanding with rgb-d images: Bottom-up segmentation, object detection and semantic segmentation. International Journal of Computer Vision, 112(2), 133–149.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778.

Kanopoulos, N., Vasanthavada, N., & Baker, R. L. (1988). Design of an image edge detection filter using the Sobel operator. IEEE Journal of solid-state circuits, 23(2), 358–367.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

Liang, Z., Chi, Z., Fu, H., & Feng, D. (2012). Salient object detection using content-sensitive hypergraph representation and partitioning. Pattern Recognition, 45(11), 3886–3901.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision, 2980–2988.

Liu, C., Ming, Y., & Zhu, J. (2020). Improving the Performance of Seabirds Detection Combining Multiple Semantic Segmentation Models. IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium, 1608–1611. IEEE.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 3431–3440.

Lyu, Y., Vosselman, G., Xia, G.-S., Yilmaz, A., & Yang, M. Y. (2020). UAVid: A semantic segmentation dataset for UAV imagery. ISPRS Journal of Photogrammetry and Remote Sensing, 165, 108–119.

Nguyen, N.-D., Do, T., Ngo, T. D., & Le, D.-D. (2020). An evaluation of deep learning methods for small object detection. Journal of Electrical and Computer Engineering, 2020.

Ribeiro, R., Cruz, G., Matos, J., & Bernardino, A. (2017). A data set for airborne maritime surveillance environments. IEEE Transactions on Circuits and Systems for Video Technology, 29(9), 2720–2732.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention, 234–241. Springer.

Seferbekov, S., Iglovikov, V., Buslaev, A., & Shvets, A. (2018). Feature pyramid network for multi-class land segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 272–275.

Silva, S. H., Rad, P., Beebe, N., Choo, K.-K. R., & Umapathy, M. (2019). Cooperative unmanned aerial vehicles with privacy preserving deep vision for real-time object identification and tracking. Journal of parallel and distributed computing, 131, 147–160.

Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. International conference on machine learning, 6105–6114. PMLR.

Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016). Understanding data augmentation for classification: when to warp? 2016 international conference on

digital image computing: techniques and applications (DICTA), 1–6. IEEE.

Wu, G., Guo, Y., Song, X., Guo, Z., Zhang, H., Shi, X., … Shao, X. (2019). A stacked fully convolutional networks with feature alignment framework for multi-label land-cover segmentation. Remote Sensing, 11(9), 1051.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv.

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2019). Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. IEEE transactions on medical imaging, 39(6), 1856–1867.

Lam Ken Lun(A175960)
Azizi Abdullah
Fakulti Teknologi & Sains Maklumat,
Universiti Kebangsaan Malaysia