Neighborhood Attention Makes the Encoder of ResUNet Stronger for Accurate Road Extraction

Ali Jamali, Swalpa Kumar Roy, *Member, IEEE,*
Jonathan Li, *Fellow Member, IEEE,* and Pedram Ghamisi, *Senior Member, IEEE*

***Abstract*—In the domain of remote sensing image interpretation, road extraction from high-resolution aerial imagery has already been a hot research topic. Although deep CNNs have presented excellent results for semantic segmentation, the efficiency and capabilities of vision transformers are yet to be fully researched. As such, for accurate road extraction, a deep semantic segmentation neural network that utilizes the abilities of residual learning, HetConvs, UNet, and vision transformers, which is called `ResUNetFormer`, is proposed in this letter. The developed `ResUNetFormer` is evaluated on various cutting-edge deep learning-based road extraction techniques on the public Massachusetts road dataset. Statistical and visual results demonstrate the superiority of the `ResUNetFormer` over the state-of-the-art CNNs and vision transformers for segmentation. The code will be made available publicly at https://github.com/aj1365/ResUNetFormer**

***Index Terms*—Vision transformers, road extraction, attention mechanism, UNet, neighbourhood attention transformer (NAT).**

I. Introduction

ONE of the most profound tasks in the area of remote sensing is the accurate road extraction. Despite substantial interest in the last decade, road extraction from high-resolution imagery remains difficult due to occlusions, noise, and the difficulty of the surrounding features in remotely sensed imagery [1]. Deep neural network-based techniques have achieved a high level of performance on a broad range of computer vision tasks, including graph neural networks [2] and multi-modal transformer networks [3]. Nevertheless, due to issues such as vanishing gradients, training a very deep architecture is incredibly challenging [1]. To address this issue, He *et al.* [4] proposed the deep residual learning method, which employs identity mapping to aid in training. Ronneberger *et al.* [5] introduced the UNet, which concatenates feature maps from various levels to enhance segmentation performance, rather than using skip connections in fully convolutional networks (FCNs) [6]. UNet incorporates low-level detailed information with high-level semantic representation, resulting in promising biomedical image segmentation performance [5]. On the other hand, ViTs utilize self attention mechanism rather than the widely used convolutional operations employed by standard deep models [7]. Consequently, unlike CNNs, ViTs capture global contextual information in a better way with the utilization of self-attention at the cost of quadratic complexity. This helps the transformers to outperform the CNN algorithms in terms of feature generalization capabilities. Moreover, because of their flexible attention window, ViTs, such as the neighborhood attention transformer (NAT) [8], have demonstrated the potential of linear computational costs and gains more attention in the vision community. As such, we propose the ResUNetFormer that integrates the capabilities of heterogeneous convolution (HetConv), residual learning, UNet, and NAT for accurate prediction of road information from high-resolution aerial imagery. The contributions of this letter can be explained as:1) We developed a deep learning UNet based semantic segmentation framework that effectively utilizes HetConv operation to leverage heterogeneous kernels within the residual learning unit for degradation free feature representation learning, 2) In contrast with the conventional vanilla ViTs, the proposed model utilizes NAT, which replaces the computationally expensive self-attention mechanism for enhancing the feature generalization ability limited within a local neighborhood that substantially reduces the computation cost, and 3) The decoder network's capacity to determine where to search for the discriminative and task-specific necessary data is significantly improved by using the local attention mechanism.

This letter introduces the ResUNetFormer in Section II, presents the experiments and analysis in Section III, and highlights the concluding remarks in Section IV.

II. Proposed segmentation Framework

Given an image $\mathbf{X} \in R^{H \times W \times C}$ where H and W represent spatial height and width and C is the number of channels, respectively. The goal is to predict $y = \mathcal{F}(X)$ that corresponds to the pixel label classification map of input $\mathbf{X}$ having size of $(H \times W)$. The easiest way to perform such a task is to use a convolutional U-network that maps input images into high-level feature representations in encoding stages, and then decode back to the full spatial resolution to produce a pixel-wise label map. In this paper, we introduce the ResUNet-Former model for semantic segmentation of the road extraction task that incorporates the advantages of HetConv, U-network,

This research was funded by the Institute of Advanced Research in Artificial Intelligence (IARAI).(Corresponding author: *Pedram Ghamisi*)

A. Jamali is with the Department of Geography, Simon Fraser University, British Columbia 8888, Canada (e-mail: alij@sfu.ca).

S. K. Roy is with the Department of Computer Science and Engineering, Alipurduar Government Engineering and Management College, West Bengal 736206, India (e-mail: swalpa@cse.jgec.ac.in).

J. Li is with the University of Waterloo Department of Geography and Environmental Management, Waterloo, Ontario, Canada (e-mail: junli@uwaterloo.ca).

P. Ghamisi is with the Helmholtz-Zentrum Dresden-Rossendorf (HZDR), Helmholtz Institute Freiberg for Resource Technology, 09599 Freiberg, Germany, and is also with the Institute of Advanced Research in Artificial Intelligence (IARAI), 1030 Vienna, Austria (e-mail: p.ghamisi@gmail.com).

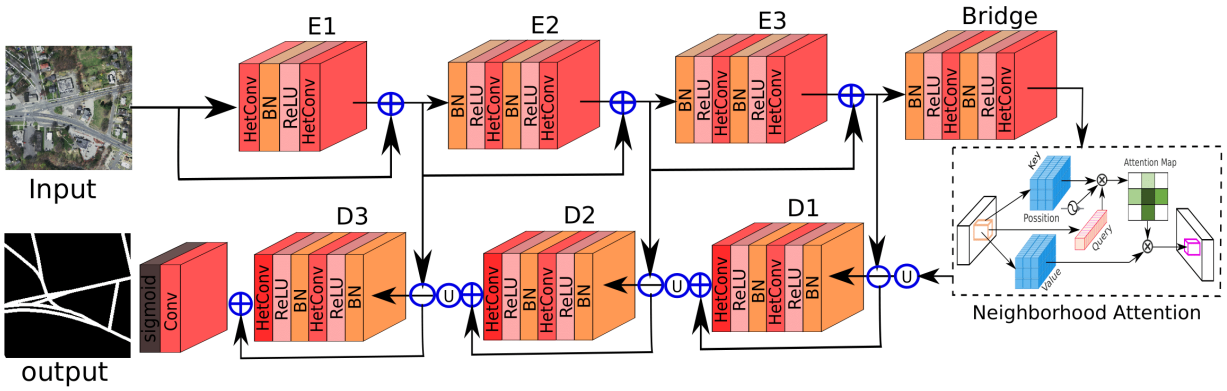


Fig. 1: Proposed ResUNetFormer model for accurate extraction of road information where $\oplus$ and $\ominus$ represent element wise addition and concatenation operations and circular U denotes the up-sampling operation of the feature maps.

residual skip connection, and vision transformers (ViT). The ResUNetFormer provides several advantages: (1) HetConv combines group-wise convolution and point-wise convolution to increase the efficiency of the generalized representation; (2) the residual learning will enable effective network training; (3) the skip connections in a residual unit, which connects the low level feature to its corresponding high level feature, improve information propagation without degradation, enabling us to construct lower complexity framework that captures better semantic segmentation information with the limited amount of reference data; (4) NAT allows a pixel-level operation that localizes each pixel to its neighborhood and improves the acquisition of global and local contextual information from satellite imagery.

1) **Residual learning:** To increase the projection power of CNNs, one of the traditional ways is to add more layers to the network, which may hampers accurate propagation of models information during back-propagation and yields degradation gradient [4]. To tackle these shortfalls, a residual unit was introduced in the place of the conventional convolutional blocks, which carefully monitor the vanishing gradient issue with a skip connection and residual learning. The skip connection of a residual unit allows to map low level feature representation to its high level representation in an easier way. Suppose $\mathbf{X_j^{l-1}}$ represents the input jth feature map of the residual blocks which is parameterized with $F_{RN}(\mathbf{X}^{l-1};\theta_1,\theta_2)$ using HetConv filter banks $W=\{W^{l+i}|1\leq i\leq 2\}$ of kernel size $k_1^l\times k_2^l$ in the $(l-1)$th and $(l)^{th}$ layers, respectively. The output feature map X^{l+1} obtained in the $(l+1)$th layer can be calculated as follows:

$$X^{l+1}=I(X^{l-1})+F_{RN}(X^{l-1};\theta_1,\theta_2) \tag{1}$$

where $I(X^{l-1})$ the identity mapping in a residual unit.

$$\begin{aligned} F_{RN}(X^{l-1};\theta_1,\theta_2)&=ReLU(BN(X^l\circledast W^{l+1}+b^{l+1})\\ X^l&=ReLU(BN(X^{l-1}\circledast W^l+b^l)) \end{aligned} \tag{2}$$

where $\circledast$ represents the *HetConv* operation and X^l and X^{l+1} are output feature maps in lth and $(l+1)$th layers, respectively. θ_1 and θ_2 denote the weight and bias parameters associated with the jth unit of the lth and $(l+1)$th of *HetConv* layers, respectively.

2) **Neighborhood attention transformer (NAT):** NAT limits the receptive field of each query token to fixed-sized neighboring pixels in its local area of the neighborhood. The NAT is driven by the goal of creating a local neighborhood region in which the smaller neighboring area receives more local attention, and the wider neighboring area obtains more global attention. The local neighborhood region of all points $\forall_y$ such that $dist(x,y)\leq r$ where r shows the radius of the local window, which can be expressed as $\lambda(i,j)=\{y\in x: dist(x,y)\leq r\}$.

The NAT consider a pixel at location (i,j) of the feature map extracted from the bridge layer of encoder is expressed as the linear projections Φ of the input features $X\in R^{(48\times 48\times 512)}$, the queries $Q=\Phi_q X$ whereas keys $K=\Phi_k X$ and the values $V=\Phi_v X$ functions for all the ith input patch with local wind of size r and the size of the patch matrix is expressed as 48×48, whereas 512 is the embedding dimension of the matrix of the input feature, and Φ_q, Φ_v, and Φ_k represent the parameters of the projections i.e., query, value, and key, respectively. The Φ will be optimized utilizing the Adam [9] optimizer in the phase of model's training. To produce weights of the attention $\mathbf{A_i^k}$, the scaled dot product is utilized whiting ith input queries ($\mathbf{Q}$) with $(r\times r)$ neighboring keys ($\mathbf{K}$) as follows:

$$A_i=\begin{bmatrix} Q_iK^T_{\lambda(i,1)}+B_{\lambda(i,1)}\\ Q_iK^T_{\lambda(i,2)}+B_{\lambda(i,2)}\\ \vdots\\ Q_iK^T_{\lambda(i,r^2)}+B_{\lambda(i,r^2)}\end{bmatrix} \tag{3}$$

where $\lambda(i,j)$ defines the jth neighboring region of the ith query which is dependent on their relative positions, the corresponding location bias is expressed by $B_{\lambda i,j}$, which is added to each point of attention weights. Afterwards, $V_{\lambda(i,j)}=[V_{\lambda(i,1)},V_{\lambda(i,2)},\dots V_{\lambda(i,r^2)}]$ are presented by the r^2 localized region with the values of the ith input query. The output of ith attention weights A_i in Eq. (3) is passed through a `softmax` function to calculate the attention map. Thus, the `NAT` for the ith input token of the Y_i^r the output map is expressed as:

$$\texttt{NAT}(\mathbf{Y_i^r})=\texttt{softmax}\left(\frac{\mathbf{A_i^r}}{\sqrt{\mathbf{D}}}\right)\mathbf{V}_{\lambda(\mathbf{i},\mathbf{j})^{\mathbf{r}}} \tag{4}$$

where the scale is defined by $1/\sqrt{D}$, which is utilized to enhance the `softmax` function's small gradient propagation. It should be noted that we used neighboring window $r = 3$ for the experimental analysis in this letter.

3) **ResUNetFormer:** In this work, we adopted the 7-layer ResUNet architecture initially developed by Zhang *et al.* [1] to address the accurate road extraction task as the backbone, illustrated in Fig. 1. The ResUNetFormer is build with four components which includes encoding, bridge, NAT, and decoding. In the encoding stage, i.e., $E1$ to $E3$, the input images are compressed into compact representations which is achieved through the consecutive use of the residual unit as shown in Eq. (1). The final section, i.e., $D1$ to $D3$, is responsible for restoring the representations to perform pixel-by-pixel classification, i.e., semantic segmentation. The bridge section acts as a link between the encoding and NAT sections and helps the decoder for smooth recovering of the features. The input of the NAT block is the feature map of the bridge section, and the resulting attention maps will be passed into the decoder section. The encoder, bridge, and decoder are constructed by residual units, which have two 3×3 HetConv blocks and an identity mapping (defined by Eq. (1)) followed by a Batch Normalization layer, a ReLU activation layer, and a HetConv layer in each block of convolutions. It should be noted that the identity mapping connects the low-level input feature and high-level output feature in a residual unit.

The encoding part has three units of the residual function. It should be mentioned that rather than employing a pooling operation to reduce the size of output maps, a stride of 2 was utilized in the first block of HetConvs in each unit, decreasing the ratios of output maps by 50%. Similarly, the decoding part has three residual units. Before each unit, output maps from the lower level will be up-sampled and concatenated with their relative encoding path output maps. A 2D convolution with kernel size (1×1) and a sigmoid activation function are employed after the last level of decoding for projecting the multichannel output maps into the targeted segmentation road map. It should be noted that we developed two versions of ResUNetFormer. In ResUNetFormer-V1, similar to the ResUNet model, we utilized Conv2D operations, while in ResUNetFormer-V2, instead of using Conv2D functions, we employed HetConv as seen in Fig. 1. In the HetConv layers, there are three depth-wise convolutional groups with kernel sizes of 3×3, while there is a point-wise Conv2D with a kernel size of 1×1. The feature map of depth-wise convolutions is added to the point-wise convolution to produce the results of HetConv functions. The details of parameters and size of feature maps for the ResUNetFormer-V1 and ResUNetFormer-V2 are illustrated in Table I.

TABLE I: The layer-wise architecture of the ResUNetFormer-V1 and ResUNetFormer-V2 segmentation algorithms.

Unit level	Filter-V1	Filter-V2	Stride	Output size-V1	Output size-V2
Input	-	-	-	384 * 384 * 3	384 * 384 * 3
E1	3 * 3 * 64	[3*(3 * 3*22)]+ [1*(1*1* 66)]	1	384 * 384 * 64	384 * 384 * 66
	3 * 3 * 64	[3*(3 * 3*22)]+ [1*(1*1* 66)]	1	384 * 384 * 64	384 * 384 * 66
E2	3 * 3 * 128	[3*(3 * 3*42)]+ [1*(1*1* 126)]	2	192 * 192 * 128	192 * 192 * 126
	3 * 3 * 128	[3*(3 * 3*42)]+ [1*(1*1* 126)]	1	192 * 192 * 128	192 * 192 * 126
E3	3 * 3 * 256	[3*(3 * 3*84)]+ [1*(1*1* 252)]	2	96 * 96 * 256	96 * 96 * 252
	3 * 3 * 256	[3*(3 * 3*84)]+ [1*(1*1* 252)]	1	96 * 96 * 256	96 * 96 * 252
Bridge	3 * 3 * 512	[3*(3 * 3*84)]+ [1*(1*1* 510)]	2	48 * 48 * 512	48 * 48 * 510
	3 * 3 * 512	[3*(3 * 3*84)]+ [1*(1*1* 510)]	1	48 * 48 * 512	48 * 48 * 510
NAT	-	-	1	48 * 48 * 512	48 * 48 * 510
D1	3 * 3 * 256	[3*(3 * 3*84)]+ [1*(1*1* 252)]	1	96 * 96 * 256	96 * 96 * 252
	3 * 3 * 256	[3*(3 * 3*84)]+ [1*(1*1* 252)]	1	96 * 96 * 256	96 * 96 * 252
D2	3 * 3 * 128	[3*(3 * 3*42)]+ [1*(1*1* 126)]	1	192 * 192 * 128	192 * 192 * 126
	3 * 3 * 128	[3*(3 * 3*42)]+ [1*(1*1* 126)]	1	192 * 192 * 128	192 * 192 * 126
D3	3 * 3 * 64	[3*(3 * 3*22)]+ [1*(1*1* 66)]	1	384 * 384 * 64	384 * 384 * 66
	3 * 3 * 64	[3*(3 * 3*22)]+ [1*(1*1* 66)]	1	384 * 384 * 64	384 * 384 * 66
Output	1 * 1	1 * 1	1	384 * 384 * 1	384 * 384 * 1

III. Experimental Results

The developed model, ResUNetFormer, is evaluated against several other state-of-the-art segmentation models, including UNet [5], UNet++ [10], UNet+++ [11], Attention UNet [12], SwinUNet [13], and ResUNet [1], respectively.

A. Experimental Data and Settings

Mihn *et al.* [14] created the Massachusetts roads data (MRD). The road benchmark contains 1171 high-resolution images, which include 1108 images for training, 14 for validation, and 49 for testing. We trained the model with images of size 384×384 in this letter. It should be mentioned that throughout training, no data augmentation was used. In this letter, we utilized a learning rate, batch size and number of epoch of 0.0001, 1 and 40, respectively.

B. Segmentation Results

To validate the efficiency of the proposed ResUNetFormer for accurate road extraction, we consider the MRD dataset to create two experimental settings. In scenario 1 (MRD100), we only used 100 images as the training data, whereas in scenario 2 (MRD800), 800 images were utilized to train the segmentation algorithms. We employed binary cross entropy as the loss function in scenarios 1 and 2. On the other hand, we have also used intersection over union (IoU) ($IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$) loss function in scenario 1 (MRD100IOU) with 100 training images and scenario 2 (MRD800IOU) with 800 training images.

TABLE II: Segmentation results of the MRD100 dataset in terms of F1-score, Precision, Recall, and Dice coefficient.

Algorithm	F-1×100 ↑	Precision×100 ↑	Recall×100 ↑	Dice coefficient ↑	Time (min)
UNet [5]	48.53	82.03	25.91	0.3149	10
UNet++ [10]	53	79.56	32.52	0.3315	9.2
UNet+++ [11]	49.71	80	26.34	0.3392	12
AttUNet [12]	54.64	86.64	27.18	0.3879	8
SwinUNet [13]	44.38	69.42	29.73	0.3728	12.5
ResUNet [1]	65.6	91.41	41.3	0.469	10
ResUNetFormer_V1	63.07	**95.3**	35.79	0.4394	10
ResUNetFormer_V2	**65.82**	88.99	**45.59**	**0.5113**	12.5

As seen in Table II and Fig. 2, the best results in terms of recall (45.59%), dice coefficient (0.513), F1-score (65.82%), and visual interpretation were achieved by the ResUNetFormer-V2 model with the HetConv operations. Moreover, the highest precision was obtained by the ResUNetFormer-V1 (95.3%). The ResUNetFormer-V2 results increased the F1-score, dice coefficient, and recall of the ResUNet by about 1%, 9%, and 10%, respectively. In addition, the ResUNetFormer-V2 with HetConv opertions model showed significantly less noise compared to other segmentation techniques, including vision-based SwinUNet and Attention UNet.

In scenario 2, as seen in Table III and Fig. 3, statistical result analysis and visual interpretation illustrated the superiority of the ResUNetFormer-V2 with the HetConv operations over the other vision-based algorithms using a recall,

TABLE III: Segmentation results of MRD800 dataset in terms of F1-score, Precision, Recall, and Dice coefficient.

Algorithm	F-1×100 ↑	Precision×100 ↑	Recall×100 ↑	Dice coefficient ↑	Time (min)
UNet [5]	51.38	87.96	26.64	0.4671	78
UNet++ [10]	49.74	89.89	23.19	0.4584	80
UNet+++ [11]	51.91	87.15	28.29	0.4811	100
AttUNet [12]	61.56	92.53	32.08	0.4608	68
SwinUNet [13]	61.52	86.14	43	0.5171	97
ResUNet [1]	57.2	92.45	31.94	**0.5706**	79
ResUNetFormer_V1	64.65	**98.13**	34.1	0.5143	80
ResUNetFormer_V2	**65.62**	97.12	**36.66**	0.5522	120

dice coefficient, F1-score, and precision of 36.66%, 0.5522, 66.62%, and 97.12%, respectively. The ResUNet segmentation algorithm obtained the highest dice coefficient (0.5706). The ResUNetFormer-V2 results enhanced the ResUNet model's precision, F1-score, and recall by around 5%, 13%, and 13%, respectively.

TABLE IV: Segmentation results in terms of F1-score, Precision, Recall, and Dice coefficient for MRD100IOU dataset.

Algorithm	F-1×100 ↑	Precision×100 ↑	Recall×100 ↑	Dice coefficient ↑
UNet [5]	52.64	81.46	35.94	0.4929
UNet++ [10]	54.36	73.05	46.62	0.5142
UNet+++ [11]	56.63	73.87	51.44	0.53
SwinUNet [13]	50.74	78.02	37.33	0.4808
ResUNet [1]	64.57	92.23	48.08	0.6227
ResUNetFormer_V1	65.19	90.75	49.48	0.6238
ResUNetFormer_V2	**65.3**	**92.52**	**53.19**	**0.6269**

In scenario 1 with IoU loss function, statistical analysis and visual interpretation showed better segmentation capability of the ResUNetFormer-V2 over the other semantic segmentation models that achieved a recall, dice coefficient, F1-score, and precision of 53.19%, 0.6269, 65.53%, and 92.52%, respectively, as reported in Table. IV and Fig. 4. The ResUNetFormer-V2 enhanced the semantic segmentation performance of the ResUNet by approximately 1%, 1%, 1%, and 10%, in terms of F1-score, precision, dice coefficient, and recall, respectively.

TABLE V: Segmentation results in terms of F1-score, Precision, Recall, and Dice coefficient for MRD800IOU dataset.

Algorithm	F-1×100 ↑	Precision×100 ↑	Recall×100 ↑	Dice coefficient ↑
UNet [5]	53.28	80.03	39.78	0.6296
UNet++ [10]	53.66	78.19	43.29	0.6278
UNet+++ [11]	53.99	71.58	49.87	0.5992
SwinUNet [13]	50.25	87.57	34.11	0.6273
ResUNet [1]	58.37	90	42	0.6833
ResUNetFormer_V1	**67.94**	94.94	**50.33**	**0.6860**
ResUNetFormer_V2	65.62	**96.02**	46.10	0.6602

In scenario 2 with IoU loss function, as seen in Table V and Fig. 5, the ResUNetFormer-V1 shows superior performance over other semantic segmentation models that obtained a recall, F1-score, and dice coefficient of 50.33%, 67.94%, and 0.686%, respectively. The ResUNetFormer-V2 algorithm also achieved the highest precision (96.02%). The results of the ResUNet in terms of dice coefficient, precision, F1-score, and recall were improved by the ResUNetFormer-V1 by approximately 1%, 5%, 14%, and 17%, respectively. Moreover, as shown in Fig. 6, the results demonstrated high Area under the ROC Curve (AUC) values of 0.988, 0.988, 0.966, and 0.961 for MRD100, MRD800, MRD100IOU, and MRD800IOU, respectively. Results illustrated that the proposed ResUNetFormer-V1 and ResUNetFormer-V2 with the use of NAT led to much better segmentation accuracy and produces much less noisy road maps as compared with the vision and attention-based models like SwinUNet and Attention UNet. Overall, the utilization of the HetConv operations over the standard Conv2D resulted in a better segmentation accuracy.

C. Ablation Study

The use of residual learning and UNet with NAT resulted in better segmentation accuracy and much lower segmentation noises as compared to the base ResUNet segmentation algorithm. For example, in scenario 1 with IoU loss function, the ResUNetFormer-V2 utilizing both the HetConv and NAT enhanced the semantic segmentation performance of the ResUNet by approximately 1%, 1%, 1%, and 10%, in terms of F1-score, precision, dice coefficient, and recall, respectively, as shown in Table IV. In scenario 2 with the IoU loss function, as seen in Table V, the ResUNetFormer-V2 algorithm achieved the highest precision (96.02%). In addition, the results of the ResUNet in terms of dice coefficient, precision, F1-score, and recall were improved by the ResUNetFormer-V1 by approximately 1%, 5%, 14%, and 17%, respectively, through the utilization of the NAT mechanism.

D. Computational Cost of Segmentation Models

The computational cost of the implemented segmentation models can be seen in Table II and Table III. The least and the highest required training time belonged to Attention UNet (8 min), Swin UNet (12.5 min), and ResUNetFormer-V2 (12.5 min), respectively, utiltizing 100 images as the training data. Moreover, in using 800 training images, the highest and least training time were for Attention UNet (68 min) and ResUNetFormer-V2 (120 min). As the results indicate, due to the use of HetConv and local attention mechanism functions, the computational cost of the developed ResUNetFormer increased compared to the base ResUNet algorithm, while the segmentation accuracy of the ResUNet considerably improved. The ResUNetFormer-V1 using only the local attention mechanism resulted in a better trade-off in terms of segmentation accuracy improvement and computation cost complexity.

IV. Conclusion

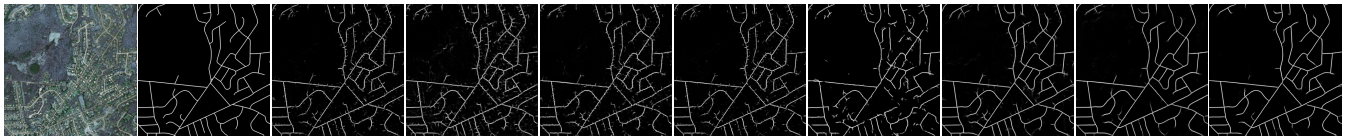
This letter proposes and discussed a deep vision transformer-based technique for semantic segmentation, which employs NAT to enhance feature extraction capabilities locally whereas significantly lowering computation costs. The results on the Massachusetts road data demonstrate that the developed model, ResUNetFormer, outperforms statistically and visually the state-of-the-art semantic segmentation models, including UNet, UNet++, UNet+++, Attention UNet, SwinUNet, and ResUNet. ResUNet with HetConv and local attention mechanism operations resulted in much lower noise than that of current CNN and transformer-based semantic segmentation techniques.

References

[1] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

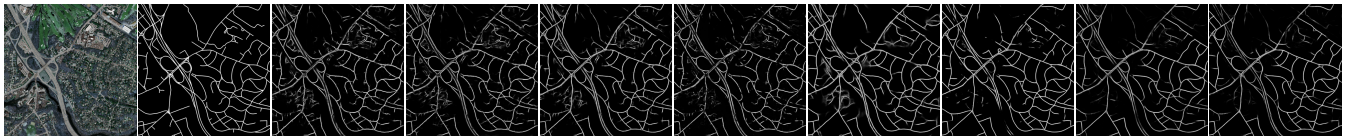
[2] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5966–5978, 2021.

[3] S. K. Roy, A. Deria, D. Hong, B. Rasti, A. Plaza, and J. Chanussot, "Multimodal fusion transformer for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–20, 2023.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 2: Segmentation maps over MRD100 dataset using (a) RGB image (b) Ground Truth, (c) UNet, (d) UNet++, (e) UNet+++, (f) Attention UNet, (g) SwinUNet, (h) ResUNet, (i) ResUNetFormer-V1, and (j) ResUNetFormer-V2.

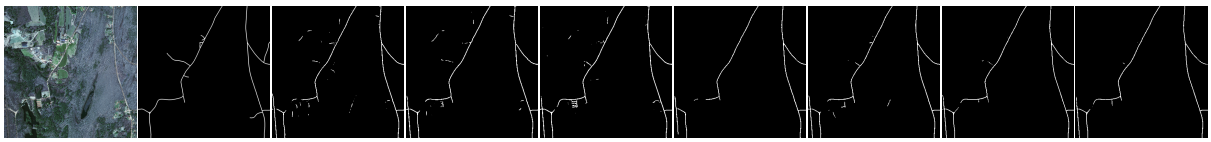


(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 3: Segmentation maps over MRD800 dataset using (a) RGB image (b) Ground Truth, (c) UNet, (d) UNet++, (e) UNet+++, (f) Attention UNet, (g) SwinUNet, (h) ResUNet, (i) ResUNetFormer-V1, and (j) ResUNetFormer-V2.

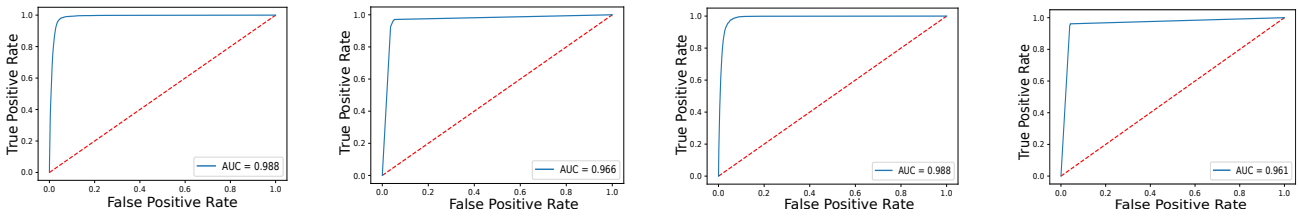
(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 4: Segmentation maps over MRD100IOU dataset using (a) RGB image, (b) Ground Truth, (c) UNet, (d) UNet++, (e) UNet+++, (f) SwinUNet, (g) ResUNet, (h) ResUNetFormer-V1, and (i) ResUNetFormer-V2.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 5: Segmentation maps over MRD800IOU dataset using (a) RGB image, (b) Ground Truth, (c) UNet, (d) UNet++, (e) UNet+++, (f) SwinUNet, (g) ResUNet, (h) ResUNetFormer-V1, and (i) ResUNetFormer-V2.



(a) (b) (c) (d)

Fig. 6: The area under the ROC Curve of the ResUNetFormer-V2 for dataset of (a) MRD100, (b) MRD100IOU, and (c) MRD800, and (d) MRD800IOU.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[8] A. Hassani, S. Walton, J. Li, S. Li, and H. Shi, "Neighborhood attention transformer," 2022. [Online]. Available: https://arxiv.org/abs/2204.07143

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[10] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: Redesigning skip connections to exploit multiscale features in image segmentation," *IEEE Transactions on Medical Imaging*, vol. 39, no. 6, pp. 1856–1867, 2020.

[11] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, "Unet 3+: A full-scale connected unet for medical image segmentation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 1055–1059.

[12] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[13] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," *arXiv preprint arXiv:2105.05537*, 2021.

[14] V. Mnih and G. E. Hinton, "Learning to detect roads in high-resolution aerial images," in *European conference on computer vision*. Springer, 2010, pp. 210–223.