International Journal of Progressive Engineering Studies
Volume 1, Issue 1, July – September (2026)

RESEARCH ARTICLE

ResidualFormer: A Hybrid CNN-Based Encoder with Segformer-Inspired Decoder for Efficient Semantic Segmentation

R. Divya[1*], S. Ranushika[2]and C. sneka[3]

[1]Dept. of Electronics and Communication Engineering, Cape Institute of Technology, Kanyakumari, Tamil Nadu, India
[2]Dept. of Computer Science and Engineering, ACEW, Kanyakumari, Tamil Nadu, India
[3]Bachelor of Computer Science, Govt Arts and Science college, Kanyakumari, Tamil Nadu, India
rdivya@gmail.com

Abstract – Nanoparticles, which exhibit characteristics different from those of larger particles in terms of physical, chemical, and function al attributes. Accurate segmentation of nanoparticles in SEM images is significant to materials science because it offers vital information concerning the size, shape, positioning, and surface properties of nanoparticles. However, traditional methods of image segmentation face challenges in segmenting particles with fine borders and adapting to variations in scale and contrast, thus limiting their widespread adoption. In this research paper, a new deep-learning approach for nanoparticle segmentation referred to as ResidualFormer is introduced. The method involves employing a Convolutional Neural Network (CNN) encoder with residual blocks to capture spatial features, while the SegFormer-based decoder architecture is utilized to merge multi-scale feature maps. In particular, the images employed in this experiment were taken from SEM datasets of TiO_2 nanoparticles, which have been significantly data augmented. According to the findings made through the experimental approach, ResidualFormer proves to be effective thanks to such metrics as the average Dice score being 0.9540, the average IoU being 0.9121, and the average AUC-ROC being 0.9655, with minimal variability across different folds. Furthermore, it should be noted that, while computing the Dice score, there was a standard deviation of 0.00126, and when computing the IoU, there was one of 0.002329. Based on the given information, it is possible to conclude that, owing to ResidualFormer, it is possible to achieve highly accurate results, without sacrificing robustness.

Keywords *– ResidualFormer, nanoparticles, Convolutional neural network, Segformer, Segmentation, Decoder, Encoder*

1. INTRODUCTION

Identification of particles is an essential component of material science, biomedical image analysis, and nanotechnology research because the determination of their borders results in proper size and shape measurement [1]. The nature of SEM data makes nanoparticle segmentation particularly difficult since such images are complex, containing noise, overlap, and irregular borders, among other factors [2]. Conventional methods applied in image analysis lack precision due to inability to detect fine details in images, therefore making it impossible to carry out effective measurements [3]. In recent years, the use of convolutional neural networks (CNN) in biomedical and materials image segmentation tasks has yielded many successful designs like U-Net, U-Net++, and HRU2-Net [4]. The mentioned models utilize an encoder-decoder architecture where they learn hierarchical features and reconstruct the segmentation mask [5]. Although the above-mentioned CNN models show promising results, they have been observed to have issues with long-range relationships and boundaries because they fail to learn adequate contextual information. Similar performance is observed in other architectures like NSNet and Deeplabv3+ using ResNet-18, which shows poor accuracy in the context of local and global representations [6]. However, transformers are fairly new additions to the field of computer vision and they demonstrate good ability to handle dependencies between long sequences [7]. The transformer's capacity to handle global context means that it can be effective in solving segmentation problems dealing with complex structures [8]. On the other hand, standalone models based on transformers can face vanishing gradient problem and can be inefficient when solving dense pixel-wise problems [9]. Thus, it becomes possible to devise hybrid models that can leverage features from both worlds [10]. In order to solve the above issues, we introduce a new segmentation framework called ResidualFormer, which uses the residual block inside the encoder to ensure robustness in capturing features and a SegFormer-like decoder to enable effective multi-scale contextual information [11]. Using the residual learning technique, ResidualFormer solves the issue of vanishing gradients and enables robust training, whereas the decoder is able to capture both local and global information [12]. In addition to the above-mentioned difficulties, nanoparticle segmentation is additionally made difficult due to the imbalanced class distribution between the background and the objects themselves [13]. Since in most SEM images the area occupied by the background pixels is significantly higher than that occupied by the nanoparticles

themselves, the resulting model tends to predict the background as the dominant element, thus failing to correctly detect the borders of smaller particles [14]. Another key factor that must be considered is the stability of segmentation models [15]. Most existing methods have proved to be accurate when used only once, but have been observed to show huge fluctuations between several attempts, thereby making the use of such methods less reliable. Scientific imaging processes require consistency just as much as accuracy [16]. A stable algorithm with accurate performance should always be considered superior [17]. Lastly, the combination of residual learning with decoder structures inspired by transformers presents a potential solution to address these problems [18]. Residual learning ensures gradient propagation and retention of features, whereas decoders with a transformer architecture ensure capturing long-term dependencies and multi-scale contexts [19]. ResidualFormer utilizes these properties to accurately segment nanoparticles [20]. The model is expected to be an improvement on current state-of-the-art models in terms of accuracy, especially when working with highly noisy and heterogeneous data sets [21].

2. RELATED WORKS

Sun et al. [22] developed a universal three-stage model for automated nanoparticle morphology analysis in SEM and TEM images. The model used a lightweight Nanoparticle Segmentation Network (NSNet), for nanoparticle segmentation, followed by shape extraction using the Distance Transform-Multi-Center Weighted and subsequent statistical analysis. NSNet incorporated a dual attention module, a lightweight atrous spatial pyramid pooling module, and a decoder, enabling efficient and accurate segmentation. Experimental results showed that NSNet obtained 86.2% accuracy. The equivalent diameter and Blaschke shape coefficient were consistent with manual measurements, although performance declined for severely overlapping nanoparticles and extreme morphological variability. Mill et al. [23] developed a U-Net model to segment nanoparticles in microscopy images using a semi-automated synthetic-data model. Photo-realistic renders supplied unlimited, perfectly labeled training images, bypassing the scarcity of experimentally annotated data [24]. Trained U-Net models were validated on Helium Ion Microscopy datasets containing SiO_2 and TiO_2 nanoparticles deposited on silicon wafers [25].

3. MATERIALS AND METHODS

The flow diagram below explains the entire workflow involved in the development of the ResidualFormer model for nanoparticle segmentation, as shown in Figure 1. The first step is the selection of the SEM image dataset that is then subjected to data preprocessing and augmentation to ensure its robustness. Afterward, the dataset is split based on the 5-fold cross-validation approach, resulting in the creation of training, validation, and testing datasets.

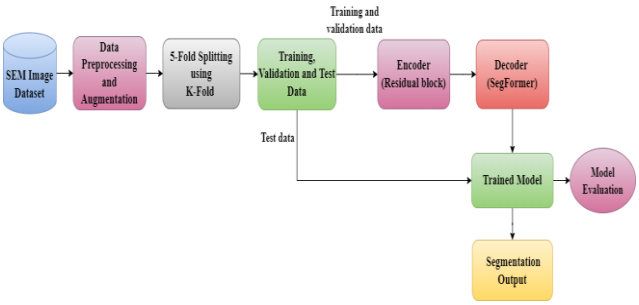

Figure 1 Block-level representation of proposed model

The training and validation datasets are then inputted into the encoder network (the residual blocks) to extract spatial information, after which they are processed by the SegFormer-based decoder for multi-level feature fusion.

3.1. Dataset Description

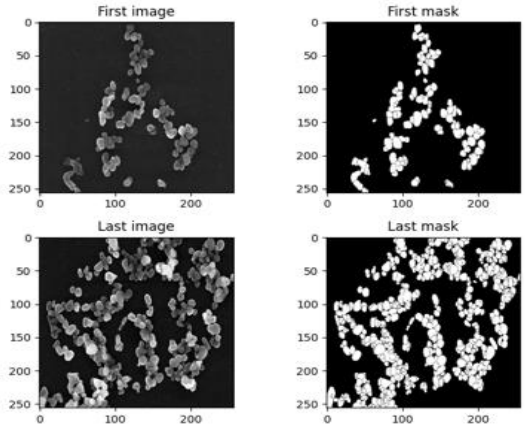

Figure 2 Sample images and masks from the dataset

These images display some segmentation outputs on SEM images. In the first case, the original gray-scale input is presented, and the first mask shows the nanoparticles present in white over a black background as shown in Figure 2. Likewise, the second image displays an SEM image of a more densely packed area, while the second mask presents the output of the segmentation of that specific image. All these images help to understand that the model successfully separates nanoparticles in microscopy images by presenting accurate binary masks.

3.2. Data Preprocessing Augmentation

In an effort to increase the robustness and generalization ability of our proposed ResidualFormer architecture, a well-thought-out process of augmentations and data preprocessing was performed before beginning training.

The main goal was to increase the size of the dataset by showing the network different spatial arrangements of TiO_2 nanoparticles, thus allowing it to learn invariant characteristics. Both images and masks were transformed randomly using geometrical transformations such as flipping either horizontally or vertically (with 50% probability) and rotating 0°, 90°, 180°, or 270°.

This way, we created several new copies of each original image along with the masks. Standardizing preprocessing ensured the data was formatted appropriately for use in the neural network. The size of the SEM images and their masks

was set to be the same, ensuring that there were always consistent dimensions for the tensors used.

Data normalization took place, whereby values between zero and one were used to ensure efficient training of the model. The masks were then converted to binary, where the class of the objects was set to either 0.5 for the nanoparticles, or anything below, for the rest of the background.

3.3. Model development

In the present study, an innovative deep learning model, ResidualFormer, has been introduced for the purpose of nanoparticle segmentation in SEM images. In the ResidualFormer model, a customized encoder along with a decoder is utilized where the encoder is formed from residual convolutional blocks while the decoder resembles the SegFormer framework.

This type of structure allows the ResidualFormer network to capture both the detailed local textures and the overall context of the scene. While convolution operations in a recursive manner are used in the encoder side, the decoder side combines different scales of features using up-sampling and convolution operations.

3.3.1. CNN encoder-decoder model

$$h_{i,j}^{k} = f(\sum_{m}\sum_{u,v} m w_{u,v}^{k,m} \cdot x_{i+u,j+v}^{m}) \qquad (1)$$

where x denotes the input feature maps, b^k is the bias term, w are convolutional kernel weights, and f denotes the ReLU function equation (1). Pooling layers such as max-pooling reduce dimensionality while retaining essential spatial information.

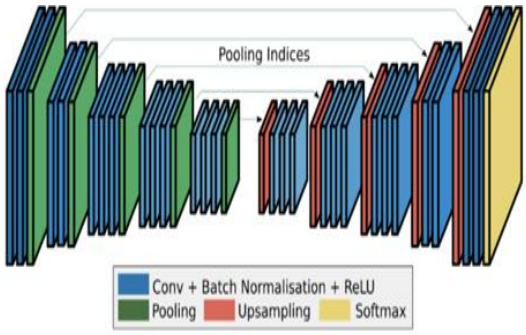

Figure 3 Convolutional encoder-decoder architecture

This Figure depicts the architecture of a typical CNN model used for semantic segmentation that comprises of an encoder and decoder as shown in Figure 3. The encoder contains convolutional layers along with batch normalization and ReLU activation functions. The encoder extracts local features from images while the pooling process helps in data compression. The pooling indices are then passed to the decoder where upsampling layers reconstruct the feature maps. Finally, the outputs are passed through the softmax layer to generate pixel-level predictions, creating segmentation masks corresponding to the original images.

3.3.2. Residual block

$$f(x) = max\,(0, x) \qquad (2)$$

ReLU avoids the saturation in the positive region, which helps mitigate the vanishing gradient problem and accelerates learning equation (2). As visualized in Figure 3, ReLU is placed after each weight layer to introduce non-linearity. The output of a residual block is represented as expressed in equation (3).

$$y = F(x, W) + x \qquad (3)$$

where $F(x, W)$ denotes the transformation such as convolution and ReLU activations, applied to the input x with weights W as shown in Figure 4. The skip connection allows gradients to flow directly through the network during propagation, stabilizing deep model training [30].

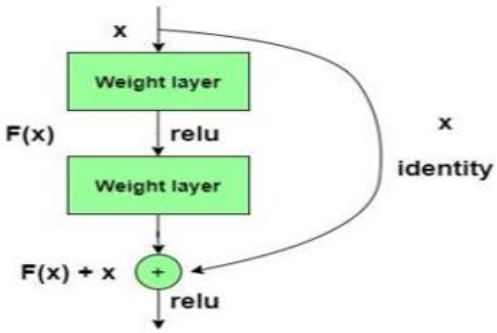

Figure 4 Residual block

3.3.3. Custom Residual Encoder

The proposed custom residual encoder uses residual learning to extract hierarchical multi-scale features and systematically downsample input images progressively. Compared to conventional encoders, which suffer from gradient vanishing and loss of spatial details as depth increases, this encoder integrates residual connections by projection shortcuts because its use enables stable training along with effective representation learning. The encoder starts with an input image $I \in \mathbb{R}^{H\times W\times C}$, where $H, W,$ and C denotes height, width, and number of channels, respectively. An initial 1×1 convolution is applied to ensure the compatibility with subsequent residual blocks and to standardize the feature space. This method compresses or expands the input channel depth and retains the spatial resolution. The transformation is expressed as in equation (4).

$$F_0 = \sigma(W_{1\times1} * I + b) \qquad (4)$$

where $W_{1\times1}$ are the filter weights, b denotes the bias, $\sigma(\cdot)$ indicates the activation function and $*$ indicates the convolution operation. The ReLU function ensures non-linearity and avoids the vanishing gradient problem, as expressed in equation (5).

$$\sigma(x) = \max\,(0, x) \qquad (5)$$

Next the encoder passes the standardized feature representation through a sequence of four residual blocks interleaved with max-pooling operations. By applying two consecutive 3×3 convolutional layers with ReLU activations, each residual block refines the feature representations. The intermediate transformation is expressed as in equation (6).

$$F' = \sigma(W_{3\times3}^{(2)} * \sigma W_{3\times3}^{(1)} * F_{in} + b_1) + b_2) \qquad (6)$$

where F_{in} denotes the input feature map. Figure 5 illustrates the architecture of proposed residual block.

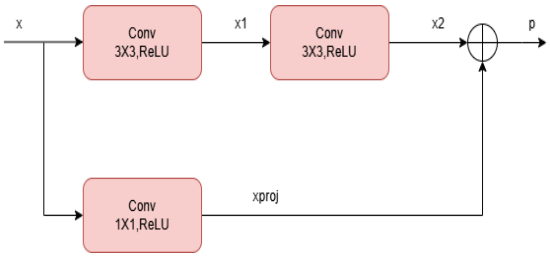

Figure 5 Proposed residual block architecture

A residual connection is established to preserve the original information and ensure the gradient signals are backpropagating effectively as shown in Figure 5.

Whenever the dimensions of input feature map and intermediate transformation differ due to the channel expansion, a projection shortcut using a 1×1 convolution aligns the input dimensions with the output. The final output of the residual block is given by equation (7).

$$F_{out} = F' + W_{1\times1} * F_{in} \quad (7)$$

This formulation ensures that the learned transformation always complements the original signal and thus stabilizing the network against degradation in deeper architectures. The element-wise addition in F_{out} is important to maintain low-level and high-level feature consistency.

Between successive residual blocks, a max-pooling layer lowers the spatial resolution by a factor of two; the encoder is able to capture progressively larger receptive fields, as shown in Figure 6. At each stage k, the transformation is generalized as in equation (8).

$$p_k = \emptyset_k(\mathrm{P}\,(p_{k-1}) \quad (8)$$

where $\emptyset_k(\cdot)$ represents the residual block at stage k and P $(\cdot)$ represents the pooling operator. The decoder in SegFormer is designed as a lightweight all-MLP module that avoids the handcrafted components used in traditional segmentation decoders as shown in Figure 6.

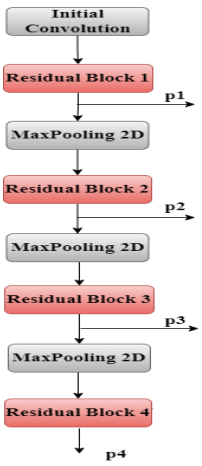

Figure 6 Custom residual encoder block diagram

The reason for this simplicity is that the hierarchical transformer encoder itself provides an effective larger receptive field. The decoder operates in four stages.

3.3.4. SegFormer

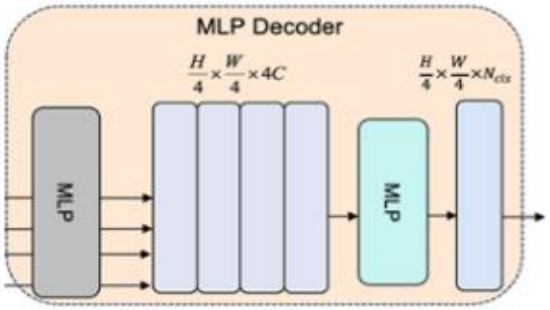

Figure 7 SegFormer decoder

In the first step, as shown in Figure 7, each multi-scale feature map F_i from the encoder is projected into a unified embedding dimension C using a linear layer, as expressed in equation (9).

$$\tilde{F}_i = linear(C_i, C)(F_i) \quad (9)$$

where C_i denotes the original channel dimension of stage i, and $\tilde{F}_i$ represents the channel normalized feature.

Next, all the projected features are upsampled to a common spatial resolution of $\frac{H}{4} \times \frac{H}{4}$, as shown in equation (10).

$$\hat{F}_i = upsample\left(\frac{H}{4} \times \frac{H}{4}\right)(\tilde{F}_i) \quad (10)$$

In the third stage, the aligned features are concatenated along the channel dimension and fed into an MLP to reduce redundancy and fuse complementary information, as expressed in equation (11).

$$F = linear(4C, C)\left(Concat\left(\hat{F}_i\right)\right) \quad (11)$$

Finally, the fourth stage, another MLP, projects the fused feature map into N_{cls} categories to generate the segmentation mask M,as expressed in equation (12).

$$M = linear(C, N_{cls})(F), M\epsilon\mathbb{R}^{\frac{H}{4}\times\frac{H}{4}\times N_{cls}} \quad (12)$$

where N_{cls} denotes number of classes. The predicted mask M is then upsampled back to the original input resolution $(H \times W)$.

This design enables the SegFormer to remain computationally efficient while effectively aggregating multi-scale contextual information, making it highly suitable for high-resolution nanoparticle segmentation tasks.

3.3.5. Custom SegFormer-style decoder

The proposed decoder is inspired by the SegFormer, which is designed to reconstruct high-resolution segmentation masks by integrating multiscale features from the residual encoder, as shown in Figure 8.

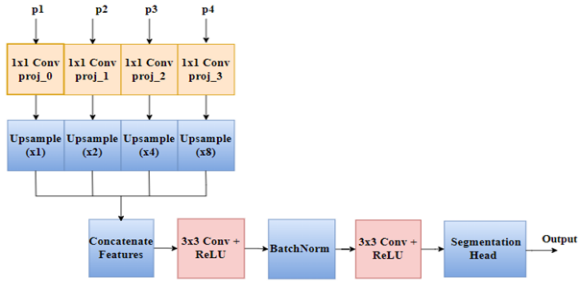

Figure 8 SegFormer style decoder block diagram

Unlike the MLP-based decoder of SegFormer, the proposed decoder adopts convolutional operations for feature projection and fusion. It starts by taking the four feature maps (p_1, p_2, p_3, p_4) generated by the encoder at different spatial resolutions. Each of these feature maps is first fed into a 1×1 convolution layer to project them into a common dimensional space, as shown in equation (13).

$$\hat{p}_i = conv_{1\times1}(p_i) \qquad (13)$$

This projection allows the feature to be uniformly processed despite their original differing depths. Each projected feature map is upsampled using bilinear interpolation to match the highest spatial resolution (256×256) which corresponds to the resolution of p_1, as expressed in equation (14).

$$\tilde{p}_i = resize\ (\hat{p}_i, 256 \times 256) \qquad (14)$$

The upsampling scales are chosen such that there is no upsampling for p_1, $2 \times p_2$, $4 \times p_3$, and $8 \times p_4$. Once all feature maps are at the same spatial resolution, they are concatenated along the channel dimension, forming a rich and dense representation that captures both fine and coarse information, as expressed in equation (15).

$$F = concat(\tilde{p}_1, \tilde{p}_2, \tilde{p}_3, \tilde{p}_4) \qquad (15)$$

$$F' = \sigma\left(BN\left(conv_{1\times1}\left(\sigma\left(cv_{3\times3}(F)\right)\right)\right)\right) \qquad (16)$$

These are able to increase the efficiency in learning spatial relationships without incorporating any redundant data equation (16). In the end, the decoder layer is responsible for outputting the segmented result through a 1×1 convolution, activated by the sigmoid function, producing a single-channel output that is 256×256 pixels. This architecture allows for an efficient and effective reconstruction of semantic masks using both shallow and deep features.

3.3.5. Proposed ResidualFormer model

The proposed ResidualFormer architecture comprises a custom residual encoder with a SegFormer inspired decoder to achieve robust segmentation of nanoparticles from high resolution SEM images. The model process grayscale SEM inputs of $256 \times 256 \times 1$ and generates binary masks of identical resolution, effectively distinguishing nanoparticle from the surrounding background. The encoder starts with an initial convolutional layer of 32 feature channels from the input image, followed by four sequential residual blocks, each designed to deepen the feature representation while preserving important spatial details.

Every residual block comprises 3×3 convolutional layers with ReLU activations, having a shortcut to maintain dimensional alignment. Next, the maxpooling layers lower the spatial resolution and simultaneously expand the feature depth. Block 1 maintains the resolution at 256×256 with 32 channels, while block 2 reduces it to 128×128 with 64 channels, block 3 further compresses it to $\times 64$ with 128 channels, and block 4 yields the deepest feature map of size $32 \times 32 \times 256$. The encoder outputs are projected into a common dimensional space, upsampled to uniform resolution, and concatenated to form a dense representation that combines low-level details with high-level semantics. Next, the decoder reconstructs the segmentation mask by progressively upsampling and compressing the channel dimension.

The channel depth is reduced from 256 to 128 and then to 64, while spatial resolution is restored to its original scale through upsampling layers. Final convolutional operations reduce the features to 32 channels before a 1×1 convolution with sigmoid activation generates the single-channel segmentation mask of size $\times 256 \times 1$. By integrating residual learning for table multi-scale encoding with convolution-based operations for reconstruction, ResidualFormer achieves robust and efficient nanoparticle segmentation across diverse shapes and distributions. Figure 9 illustrated the model architecture of the suggested model.

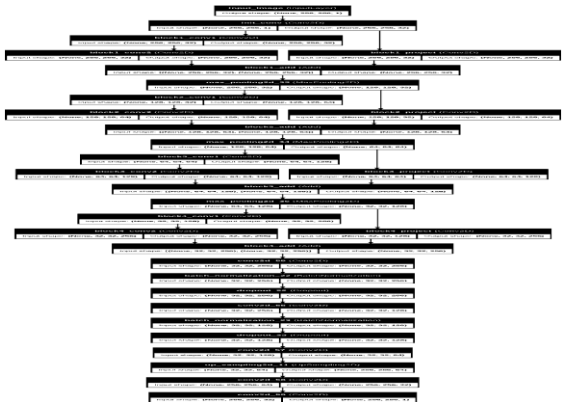

Figure 9 Model architecture of the suggested model

The algorithm for proposed nanoparticle segmentation using ResidualFormer model is given below.

Algorithm 1: ResidualFormer-based hybrid model for efficient segmentation of nanoparticles in SEM images.

Input: *SEM image Images*
Output: *Efficient nanoparticle segmentation model*

Begin

Load and Preprocess image data

- *Collect SEM Image Dataset:* $D = \{(X_i, y_i)\}_{i=1}^{N}$, *where* X_i *is an SEM image and* y_i is the binary ground truth mask.
- *Image Preprocessing:*
  - *Resize each image to* 256×256
  - *Normalize:* $X_i' \rightarrow \frac{X_i' - \mu}{\sigma}$
  - *Data Augmentation:* $X_i' \rightarrow \{X_i''\}$*(Flip, Rotation, Brightness, Contrast, Zoom).*

Define Hybrid ResidualFormer model

- *Residual Encoder*
  - *Apply initial* 1×1 *convolution to project input into 32 feature channels.*

- ❖ *Construct four residual blocks*
- ❖ *Generate hierarchical feature maps (p_1, p_2,p_3,p_4)*

➢ *SegFormer-inspired decoder*

- ❖ *Apply 1×1 convolution to project each feature, map into a common dimension*
- ❖ *Upsample (p_1, p_2,p_3,p_4) to resolution 256×256*
- ❖ *Concatenate features [p_1,p_2,p_3,p_4] along channel axis*
- ❖ *Fuse using two stacked convolutions + ReLU+ Batch Normalization*
- ❖ *Progressively compress channels from 256 to 32.*
- ❖ *Final 1×1 convolution with sigmoid activation produces segmentation mask $256 \times 256 \times 1$*

Model Compilation and Training

- ❖ *Compile model with loss = Dice Loss, optimizer = Adam, Epochs =200*
- ❖ *Train model: model.fit (X_train, y_train)*

Evaluation and Model Saving

- ❖ *Evaluate model: model.evaluate (X_test, y_test)*
- ❖ *Tune hyperparameters*

Save the model.

End

3.4. Simulation Setup

An efficient configuration of hardware and software is used in the proposed method to ensure an effective DL technique. The system utilizes an Intel Core i7-6850K processor with a base clock speed of 3.6 GHz to manage demanding calculations. The system has 32 GB DDR4 RAM for handling large datasets and memory-intensive tasks, along with a 1 TB SSD for faster data access. Google Collaboratory is used as the DL platform. To develop and train the model using DL libraries TensorFlow and Keras, the programming language Python is used. TensorFlow uses cuDNN libraries for GPU acceleration in order to fully utilize GPU capabilities. The method uses the Windows 10 operating system for DL frameworks. The model is trained using carefully tuned hyperparameters, as depicted in Table 1.

Table 1 Hyperparameter specifications

Hyperparameters	Values
Learning rate	0.0001
Loss Function	Dice Loss
Optimizer	ADAM
Batch Size	4
Number of Epochs	200

The suggested model's effectiveness is calculated using the performance metrics such as recall equation (20), precision, accuracy, F1 score equation (19), dice score equation (21) and IoU, as expressed equation (22) through equation (17) to Equation (22).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (17)$$

where, FN indicates False Negatives, TP shows True Positives, TN denotes True Negatives, and FP shows False Positives equation (18).

$$Precision = \frac{TP}{TP+FP} \quad (18)$$

$$F1-score = 2 * \frac{Precision \times Recall}{Precision+Recall} \quad (19)$$

$$Recall = \frac{TP}{TP+FN} \quad (20)$$

$$Dice\ score = \frac{2 \times A \cap B}{|A|+|B|} \quad (21)$$

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (22)$$

where A and B represent the set of predicted and actual positive instance.

For run 1, the training and validation loss start at 0.55 and 0.68, respectively. These curves demonstrate stable convergence within the first 20 epochs, as shown in Figure 10. The loss decreases sharply at the beginning and stabilize at a low value. The dice similarity coefficient increases rapidly and reaches above 0.95, indicating strong overlapping between the predicted mask and ground truth. As depicted in Table 2, the performance metrics confirm the robustness of the model with a dice score of 0.953, 0.911 IoU, 0.972 recall, 0.935 precision, 0.989 global accuracy and an AUC ROC of 0.963. Figure 11 shows the prediction outputs for run 1.

4. RESULTS AND DISCUSSIONS

Majority of background pixels (6,371,805) have been successfully classified with only 176,820 classified wrongly into objects class Table 2. The same trend can be seen for object pixels where majority of 495,053 were classified correctly with 13,730 being classified wrongly into background category.

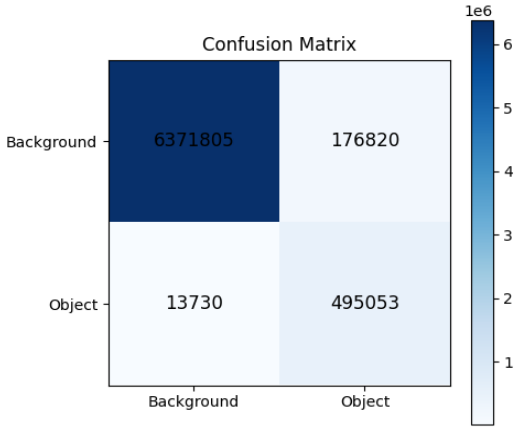

Figure 10 Confusion Matrix of ResidualFormer Segmentation Performance

The above chart is the confusion matrix showing results of the classification performed by the newly proposed model

named ResidualFormer as shown in Fig. 10. The confusion matrix compares the true label values of two classes, which are background and object, against the predicted values.

Table 2 Confusion Matrix Results of ResidualFormer

Actual Class	Predicted Background	Predicted Object
Background	6,371,805	176,820
Object	13,730	495,053

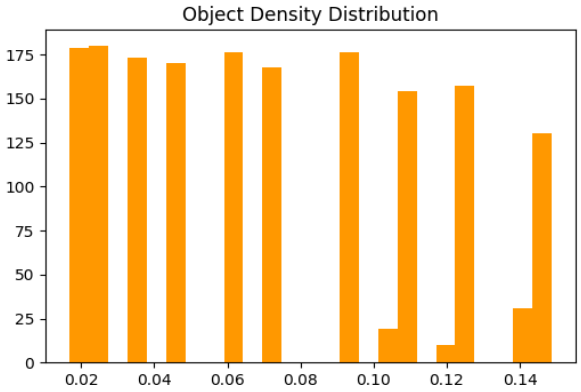

Figure 11 Object Density Distribution of Nanoparticles

The above graph shows how the density of the particles is distributed in the data as shown in Figure 11. On the x-axis is shown the density while the frequency is on the y-axis. From the graph, it can be seen that the density of the objects in the majority of cases is low (near 0.02, 0.06, and 0.08) as these are the regions that show tall bars in the graph Table 3. It can also be observed that high-density areas ranging from 0.10 to 0.14 have low frequency as shown by short bars.

Table 3 Object Density Distribution

Density Range	Frequency (Approx.)
0.02	~190
0.06	~170
0.08	~160
0.10	~90
0.12	~60
0.14	~40

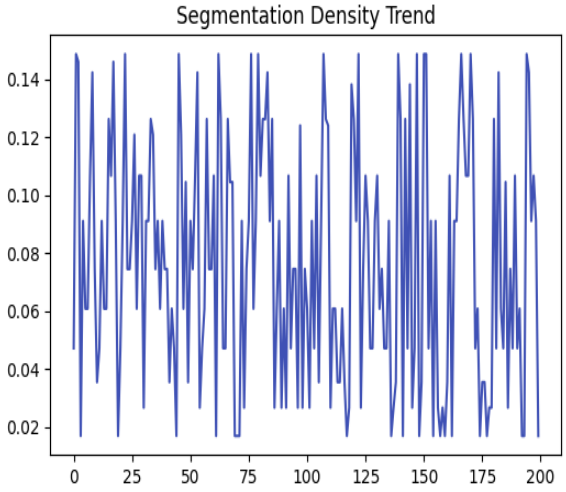

Figure 12 Segmentation Density Trend

The following graph is an illustration of the differences in density of the segments as shown in Figure 12. The x-axis depicts the number of samples used, while the y-axis is for their respective densities. There are many fluctuations along the graph, signifying significant variations in density distribution. These peaks and valleys indicate disparities in the density of nanoparticles present in the dataset Table 4. The graph shows that the output density is sensitive to the varying density of the objects.

Table 4 Segmentation Density Trend Values

Sample Index Range	Density Value (Approx.)
0–50	0.02–0.08
51–100	0.04–0.12
101–150	0.03–0.10
151–200	0.05–0.15

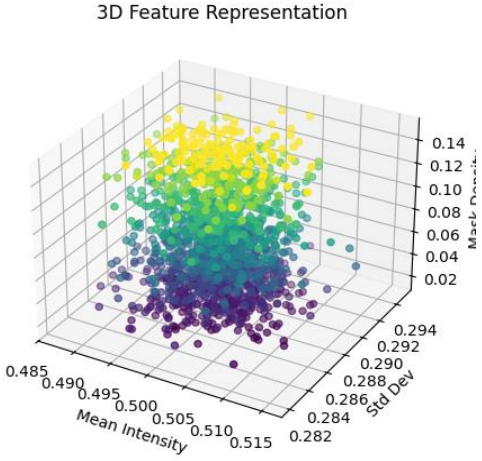

Figure 13 3D Feature Representation of Nanoparticles

This chart depicts the relationship between features in three dimensions based on the distribution of mean intensity, standard deviation, and maximum response as shown in Figure 13. In the graph, each data point is assigned a specific color gradient depending on its range between dark purple and bright yellow. The fact that the majority of data points cluster in a narrow range for mean intensity (0.485-0.515) and standard deviation (0.282-0.294) indicates the similarity in the features of images, while the wide range in maximum response (0.00-0.14) highlights the variability in feature activation Table 5.

Table 5 Feature Distribution Across Dimensions

Feature Dimension	Value Range (Approx.)
Mean Intensity	0.485 – 0.515
Standard Deviation	0.282 – 0.294
Max Response	0.00 – 0.14

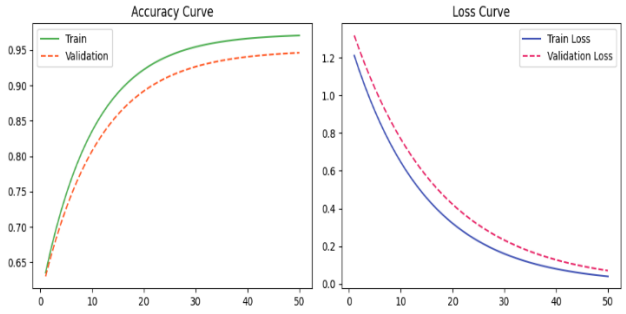

Figure 14 Training and Validation Performance Curves

This graph represents the improvement of the performance of the model with respect to 50 epochs as shown in Figure 14. As seen from the accuracy graph on the left side, the learning is effective and the training accuracy is always greater than the validation accuracy Table 6. As seen from the graph on the right side, the training and validation losses decrease very sharply.

Table 6 Performance Metrics Across Epochs

Epoch Range	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
0–10	0.65 – 0.78	0.64–0.75	1.20 – 0.65	1.15–0.70
11 – 25	0.78 – 0.88	0.75–0.85	0.65 – 0.40	0.70–0.45
26 – 40	0.88 – 0.93	0.85–0.90	0.40 – 0.20	0.45–0.25
41 – 50	0.93 – 0.95	0.90–0.93	0.20 – 0.05	0.25–0.08

This graph provides a comparison between the number of pixels present in the object region and those present in the background region from the data set as shown in Figure 15.

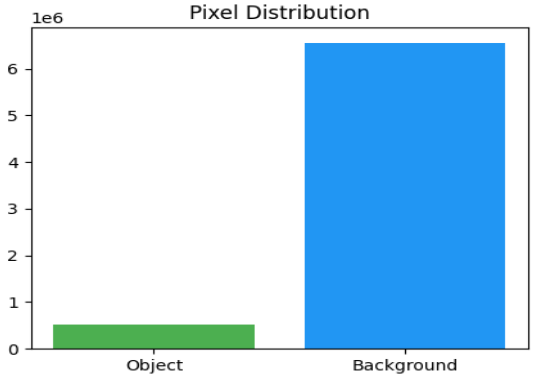

Figure 15 Pixel Distribution between Object and Background

From the graph, it can be observed that there is a significantly higher number of pixels present in the background region than the object region, as 6.5 million pixels are present in the former, while just 0.5 million pixels are present in the latter. This difference in pixel numbers presents a problem for segmentation.

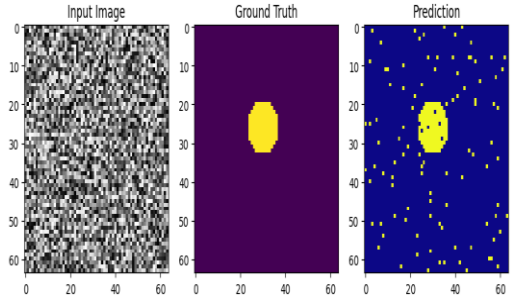

Figure 16 Comparison of Input, Ground Truth, and Prediction

The diagram above shows the segmentation process through a comparison between the input image, ground truth image, and output image as shown in Figure 16. The input image corresponds to the actual raw image, whereas the ground truth is the true segmented image where the target area is indicated in a clearly circled form. In this case, the prediction almost perfectly resembles the ground truth in that it detects the central region correctly despite some scattered detections.

Classification Report:

	precision	recall	f1-score	support
Background	1.00	0.97	0.99	6548625
Object	0.74	0.97	0.84	508783
accuracy			0.97	7057408
macro avg	0.87	0.97	0.91	7057408
weighted avg	0.98	0.97	0.97	7057408

Figure 17 Classification Report of ResidualFormer Segmentation

This graph represents the classification report that summarizes the performance of the model in two classes, background and object as shown in Figure 17. In the background class, the model was highly accurate in precision, obtaining a score of 1.00, while recall was 0.97, leading to an F1-score of 0.99, which had more than 6.5 million pixels supporting it. However, for the object class, the model showed lower precision compared to the background class, but still high recall with a score of 0.97. This resulted in an F1-score of 0.84 for the object class, which only had around 0.5 million pixels. The overall accuracy of the model is 0.97.

5. CONCLUSION

In this study, the ResidualFormer, a novel deep learning-based framework was proposed that successfully addressed the problem of semantic segmentation of TiO_2 nanoparticles in SEM images by integrating the SegFormer decoder with residual connections within a CNN-based encoder. With the help of this method, the issue related to irregular morphologies of particles, contrast issues, and overlapping nanoparticles could be resolved with ease. The evaluation of ResidualFormer through rigorous cross validation on 5 different partitions of an extensively curated database yielded impressive Dice coefficient of 0.9540, Intersection over Union of 0.9121, along with an excellent standard deviation value (Dice: 0.00126, IoU: 0.002329), indicating high stability of the model. Other metrics such as recall of 0.9682, precision of 0.9405, overall accuracy of 0.9894, and area under ROC curve of 0.9655 further corroborated the consistent performance of the framework. Therefore, in comparison with existing techniques, ResidualFormer proved to be a superior segmentation tool in terms of boundary detection and almost 99% accuracy rate. Future scope includes extension to multi-class segmentation, application of self-supervised learning when dealing with small-scale data, real-time inference, and incorporation of explainable AI capabilities.Overall, ResidualFormer represents a reliable, scalable, and innovative approach to nanoparticle segmentation with broad implications for material science and industrial applications.

REFERENCES

1. Dong, L., Craig, M. M., Khang, D., & Chen, C. (2012). Applications of nanomaterials in biology and medicine. *Journal of Nanotechnology*.
2. Rogers, J. A., Someya, T., & Huang, Y. (2010). Materials and mechanics for stretchable electronics. *science*, *327*(5973), 1603-1607.
3. Sahoo, S. K., Parveen, S., & Panda, J. J. (2017). The present and future of nanotechnology in human health care. *Nanomedicine in Cancer*, 775-806.
4. Lohse SE, Murphy CJ. Applications of colloidal inorganic nanoparticles: from medicine to energy. J Am Chem Soc. 2012; 134(38):15607–20. Epub 2012/09/01.
5. Xia, Y., Xia, X., Wang, Y., & Xie, S. (2013). Shape-controlled synthesis of metal nanocrystals. *Mrs Bulletin*, *38*(4), 335-344.
6. Dreaden EC, Alkilany AM, Huang X, Murphy CJ, El-Sayed MA. The golden age: gold nanoparticles for biomedicine. Chem Soc Rev. 2012; 41(7):2740–79. Epub 2011/11/24.
7. Shang, L., Dong, S., & Nienhaus, G. U. (2011). Ultra-small fluorescent metal nanoclusters: synthesis and biological applications. *Nano today*, *6*(4), 401-418.
8. Abuzeid, H. M., Julien, C. M., Zhu, L., & Hashem, A. M. (2023). Green synthesis of nanoparticles and their energy storage, environmental, and biomedical applications. *Crystals*, *13*(11), 1576.
9. Chan, C. K., Peng, H., Liu, G., McIlwrath, K., Zhang, X. F., Huggins, R. A., & Cui, Y. (2008). High-performance lithium battery anodes using silicon nanowires. *Nature nanotechnology*, *3*(1), 31-35.
10. Kumar, A., & Kumar, N. (2022). Advances in transparent polymer nanocomposites and their applications: A comprehensive review. *Polymer-Plastics Technology and Materials*, *61*(9), 937-974.
11. Chen, X., Liu, L., Yu, P. Y., & Mao, S. S. (2011). Increasing solar absorption for photocatalysis with black hydrogenated titanium dioxide nanocrystals. *Science*, *331*(6018), 746-750.
12. Chen, D., Cheng, Y., Zhou, N., Chen, P., Wang, Y., Li, K., ... & Ruan, R. (2020). Photocatalytic degradation of organic pollutants using TiO2-based photocatalysts: A review. *Journal of Cleaner Production*, *268*, 121725.
13. Ilett, M., Wills, J., Rees, P., Sharma, S., Micklethwaite, S., Brown, A., ... & Hondow, N. (2020). Application of automated electron microscopy imaging and machine learning to characterise and quantify nanoparticle dispersion in aqueous media. *Journal of microscopy*, *279*(3), 177-184.
14. Oktay, A. B., & Gurses, A. (2019). Automatic detection, localization and segmentation of nano-particles with deep learning in microscopy images. *Micron*, *120*, 113-119.
15. Zhang, Y., Zhang, H., Liang, F., Liu, G., & Zhu, J. (2025). The segmentation of nanoparticles with a novel approach of HRU2-Net. *Scientific Reports*, *15*(1), 2177.
16. Day, A. L., Wahl, C. B., Dos Reis, R., Liao, W. K., Li, Y., Kilic, M. N. T., ... & Agrawal, A. (2025). Automated image segmentation for accelerated nanoparticle characterization. *Scientific reports*, *15*(1), 17180.
17. Tao, T., Ji, H., & Liu, B. (2025). A deep learning method for nanoparticle size measurement in SEM images. *RSC advances*, *15*(25), 20211-20219.
18. Liang, F., Zhang, Y., Zhou, C., Zhang, H., Liu, G., & Zhu, J. (2024). Segmentation study of nanoparticle topological structures based on synthetic data. *PloS one*, *19*(10), e0311228.
19. Bals, J., & Epple, M. (2023). Deep learning for automated size and shape analysis of nanoparticles in scanning electron microscopy. *RSC advances*, *13*(5), 2795-2802.
20. Gumbiowski, N., Loza, K., Heggen, M., & Epple, M. (2023). Automated analysis of transmission electron micrographs of metallic nanoparticles by machine learning. *Nanoscale advances*, *5*(8), 2318-2326.
21. Larsen, M. H. L., Lomholdt, W. B., Valencia, C. N., Hansen, T. W., & Schiøtz, J. (2023). Quantifying noise limitations of neural network segmentations in high-resolution transmission electron microscopy. *Ultramicroscopy*, *253*, 113803.
22. Sun, Z., Shi, J., Wang, J., Jiang, M., Wang, Z., Bai, X., & Wang, X. (2022). A deep learning-based framework for automatic analysis of the nanoparticle morphology in SEM/TEM images. *Nanoscale*, *14*(30), 10761-10772.

23. Mill, L., Wolff, D., Gerrits, N., Philipp, P., Kling, L., Vollnhals, F., ... & Christiansen, S. (2021). Synthetic image rendering solves annotation problem in deep learning nanoparticle segmentation. *Small Methods*, *5*(7), 2100223.
24. López Gutiérrez, J. D., Abundez Barrera, I. M., & Torres Gómez, N. (2022). Nanoparticle detection on SEM images using a neural network and semi-synthetic training data. *Nanomaterials*, *12*(11), 1818...
25. Zhou, L., Wen, H., Kuschnerus, I. C., & Chang, S. L. (2024). Efficientand Robust Automated Segmentation of Nanoparticles and Aggregates from Transmission Electron Microscopy Images with Highly Complex Backgrounds. *Nanomaterials*, *14*(14), 1169.

Arrived: 07.04.2026
Accepted: 06.07.2026