**Quantifying Nanoparticles and Monomer-Dimer Distribution from Uncorrelated Optical Images**

**Shadab H. Choudhury,a¥ Abu S. M. Mohsin a¥\***

a Nanotechnology, IoT and Applied Machine Learning Research Group, Brac University, Dhaka, Bangladesh

**Corresponding Author:** \* Abu S. M. Mohsin, email: [asm.mohsin@bracu.ac.bc](mailto:asm.mohsin@bracu.ac.bc)

¥ Both authors contributed equally.

**Abstract:** Nanoparticles embedded in polymer matrices play a critical role in enhancing the properties and functionalities of composite materials. Detecting and quantifying nanoparticles from optical images is crucial for understanding their distribution, aggregation, and interactions, which can lead to advancements in nanotechnology, materials science, and biomedical research. In this paper, we propose an ensembled deep learning approach for automatic nanoparticle detection and oligomerization quantification in a polymer matrix for optical images. We fine tune a deep neural network architecture, YOLOv8, on a carefully annotated dataset of images of 80nm gold nanospheres (AuNS) of varying oligomerization states. The performance of the model is evaluated using accuracy, mean average precision and intersection over union metrics, demonstrating its effectiveness in nanoparticle detection and oligomerization within the polymer matrix. This will be useful for analyzing nanoparticle uptake and aggregation kinetics in live cells, identifying membrane protein interactions, and drug delivery.

**Keywords:** Nanoparticle detection, Nanoparticle oligomerization, Deep Learning, Optical images, Convolutional Neural Network and YoloV8

1. **Introduction**

Gold Nanoparticles (AuNPs) play a significant role in various fields, including drug delivery, imaging, solar cells, biosensors, disease detection, nanoantenna design, etc. [1-6]. Localized surface plasmon resonance and field enhancement is dominated by nanoparticle size, shape, geometry, environment, and oligomerization state [3, 4, 6].

Identifying nanoparticles in a polymer matrix and determining their aggregation state from optical images is challenging due to the low contrast, anisotropic shape, overlapping particle instances and resolution limits [6, 7].

Though fluorescence or transmission electron microscopy can provide quantitative information, further analysis usually relies on manual processing or human expertise, which can be inaccurate and inefficient. Many semi-automatic computer aided microscopy image processing tools such as ImageJ or cisTEM require extensive user setup for image processing, such as selecting specific thresholds or filter properties. Due to ever-increasing volume of experimental data, a quick and effective method for quantitatively extracting the total number of NPs and their aggregation state from optical images is critical [6].

Compared to scanning electron microscopy (SEM) or tunneling electron microscopy (TEM) imaging, optical images have several advantages such as being usable for in situ studies, being non-destructive, having higher throughput, lower cost, and greater accessibility [6].

However, optical images suffer from many more issues compared to the crisp clear images returned by SEM or TEM microscopy. In optical images, particles of the same size may have inconsistent visible diameters. They tend to be present at varying distances from the microscope, making it challenging to focus on all particles in view. Reflections off the imaging surface can create bright lines and blurs around the edges of the images. The point spread function (PSF) of particles in the optical image will be identical due to nanoparticle plasmon coupling. The PSF of each particle is also much larger and brighter, causing distinct particles to blur together. In an SEM or TEM image, an oligomer with clearly distinct particles can appear as a single bright blob in an optical image. Additionally, the angle of polarization can have an impact on the overall intensity distribution. Data points such as quantum yield (dimer to monomer intensity ratio) and PSF are also unavailable in optical images.

Previous studies on nanoparticle quantification have been done mostly on SEM or TEM images [9-11]. Little to no quantification work regarding oligomerization has been done on optical images due to the complexity and randomness of the emitters as well as the myriad challenges described above.

In this paper, we describe a method for automating the nanoparticle counting process and identifying the oligomeric state in optical images using deep learning. Deep learning allows us to analyze optical images and extract quantitative data regarding particles. SEM images are used as a ground truth for comparison. This is a novel method that achieves high accuracy for particle detection and classification, comparable to SEM images.

1. **Background**

While the use of deep learning has been extensively studied in cell segmentation, cell imaging and other types of microscopies, it is still a developing field when it comes to applications in nanoparticle detection and oligomerization analysis. One challenge in the field is the wide variety of imaging modalities such as optical, SEM, TEM, etc., each of which gives very different outcomes in terms of visual features. Domain adaptation and transfer learning can be used to overcome this issue, but there are inevitable falloffs in accuracy. Therefore, models and methods specialized for each imaging modality are required.

Traditional methods such as geometric analysis and edge detection [9, 12] have been extensively studied and implemented in existing computer aided microscopy tools and software like ImageJ or CisTEM. However, these approaches are not as accurate and often fail when challenged by images featuring dense clustering, occlusion, etc. Deep learning approaches can resolve these issues.

Most prior work applying deep learning in this field has been focused on SEM and TEM imaging. The reason for this is obvious – those images are very easy to process and model and do not face the challenges optical images do. Qu et al [14] used a YOLO v5 convolutional neural network (CNN) model to quantify NPs of size ranging from 11nm to 50nm from TEM images. However, they were only able to extract position and diameter information from the particles, which is not enough to extract the oligomeric state. Oktay and Gurses [11] achieved an F1 score of 0.98 on detecting distinct particles in TEM images, but did not carry out any further analysis. Dahy et al [10] achieved a similar 0.98 F1 score on palladium nanoparticles in SEM images. Similar or weaker results were seen in [15-19], using a variety of different CNN models, different nanoparticles and either SEM or TEM images.

On the other hand, most work on optical images focuses on the micrometer scale – cells and their components [20]. However, some limited studies of optical images of nanoparticles do exist. Xu et al [21] used deep learning on optical images of silver NPs, but focused on individual particles rather than collections. They predicted morphologies with an accuracy of 82%.

Our work is novel in the sense that we focus on NP detection, aggregation, and oligomerization in optical images. There has been next to no work prior research on this type of analysis of optical images rather than SEM or TEM images.

1. **Methodology**

Our methods section is divided into two steps – preprocessing the dataset, and training the model.

Because of the lack of data, it’s difficult to achieve accurate results purely from an object detection model. Therefore, we use an ensemble approach – detect the bounding box of each nanoparticle using the object detection model, then extract spatial and pixel-wise features from it to classify the type of particle as a monomer or oligomer using a gradient boosted tree model.

Fig. 1. gives an overview of how the counting process works and the steps involved.

A close-up of a black and white photo

Description automatically generated

**Fig. 1.** Summary of the Method.

**3.1. Datasets and Preprocessing**

We initially began with a small dataset of 8 pairs of optical (confocal laser scattering) images and correlated SEM images. The SEM images provide the ground truth for the exact number of particles as well as their oligomeric state needed to train the models.

It is difficult to capture the optical image and its correlated SEM imaging simultaneously. The positions of the particles in the ground truth SEM images were slightly shifted with respect to the corresponding particle in the optical image. Each paired optical and SEM image were manually aligned before labeling. For labelling and counting particles in the SEM image, ImageJ’s particle counter was used.

We take six of these images as our training set and two as our validation set, giving us a 75:25 training-validation ratio. The sampling is carried out at the batch processing step, so every batch has a random set of training and validation images. For testing, we use a set of 20 optical images of white-light excited AuNPs that have no corresponding SEM image. Fig. 2 shows the training and validation sets, while Fig. 3 shows a sample of the test set.

[FIG 2 and FIG 3]

**3.1.1. Detection and Counting**

For detection and counting purposes, each particle in the training and validation sets were manually labelled using a bounding box and an oligomeric state label. Due to the low representation of higher oligomeric states such as trimers and tetramers, we used binary state labels of monomer and oligomer instead.

For detection and counting, extensive image augmentation was used to extend the limited dataset. For photometric augmentations, we use colour jitter and randomized pixel shift. For geometric augmentations, we use horizontal and vertical flips, masking to zero (erasing random parts of an image), and randomized resizing crops (cropping out a random part of the image and resizing it). All augmented images were returned as single-channel grayscale images before being fed into the CNN.

**3.1.2. Classification of Oligomers**

The data for the spatial and pixel-wise features of our oligomeric state classifier is extracted from the original ground truth bounding box labels. We extract the following features from each bounding box encompassing a nanoparticle: center point  (eq. 1), maximum intensity (eq. 2), total intensity (eq. 4), average intensity (eq. 5), distance to nearest particle , and pixel point density function expressed as a two-dimensional gaussian (eq. 6). The equations are given as follows:

(1)

Where is the coordinate of the top left corner of the bounding box and and are the width and height of the box respectively.

(2)

(3)

Where represents the set of pixel intensity values for each pixel coordinate within the area of the bounding box.

(4)

For the total intensity, the pixel intensity values of each pixel coordinate within the bounding box are summed up.

(5)

For the average pixel intensity, we simply divide the total intensity by the area of the bounding box.

(6)

Finally, to compute the pixel point density function, we use the two-dimensional gaussian function, where compute and are the standard deviations in the x-axis and y-axis respectively, with the mean located at the center point of the bounding box. The maximum value of the kernel is clamped to the maximum intensity of the particle.

For each bounding box, we compute and through gradient descent and pixelwise mean square error loss. We use a learning rate of 0.01 and update the values for 100 iterations. While it is difficult to generate a gaussian that approaches the original particle in terms of shape and intensity due to their irregular blob-shaped appearances, the gaussian approximates the particle’s total intensity. Brighter particles will have larger values of standard deviation, and vice versa for dimmer ones.

**3.2. Model Training**

In this section, we describe the architectures, hyperparameters, and training details of our models. There are two models here: an object detection model for nanoparticle detection, and a boosted tree model for classification.

**3.2.1. Nanoparticle Detection Model**

Our CNN model is a YOLO v8 model, which is state-of-the-art for object detection. It is an improvement on the YOLO v5 model [22, 23]. The specifics of the model are described here, based off [25]. Fig. 4 shows the full model diagram.

The model consists of a 5-layer feature pyramid network (FPN) backbone followed by a YOLO v8 head and a detection head. Each layer in the FPN is a convolution with a kernel size of 3x3, a stride of 2 and a padding of 1. The layers are labelled from largest to smallest as P1 to P5.

In the YOLO v8 head, the features extracted from P5 are upsampled and concatenated with the features from P4 and this combined array is passed into the Coarse-to-Fine Module (C2F). Let’s call the output from this module P4F. The output from P4F is then upsampled and concatenated with the output from P3 and passed into a C2F module, outputting P3F.

This process is repeated backwards moving upwards starting with P3F. Instead of an upsampling layer, a convolution of kernel size 3x3, stride of 2 and padding of 1 is used before concatenation. The final outputs of P3F, P4F and P5F are passed through another feature pyramid of P3, P4 and P5. The output from this goes into a Detection layer. In the detection layer, the output goes through two convolutions of kernel size 3x3, stride 1 and padding 1, then a single 2D Convolution of kernel size 1x1, stride of 1, padding of 1 and output channels equal to the number of anchors multiplied by 4 to give the Bounding Boxes output.

A diagram of a block diagram

Description automatically generated

**Fig. 4.** Model Architecture of YOLO v8.

Rather than train a model from scratch, we elected to fine-tune an existing model pretrained on the COCO dataset. A stochastic gradient descent (SGD) optimizer was used with a learning rate of 0.01 and a weight decay of 0.001. These hyperparameters were selected via grid search.

The model was trained for 50 epochs on a Tesla T4 GPU, after which no further improvement was seen.

**3.2.2. Oligomeric State Classifier**

To classify the oligomeric state from spatial and pixel-wise features, we built an extreme gradient boosted tree model using XGBoost [24]. The learning rate of this model was set to 0.3, the max depth to 20, and the number of estimators to 100. We optimized the hyperparameters using grid search.

Fig. 5. shows the full overview of the process of training both models.

A diagram of a diagram

Description automatically generated with medium confidence

**Fig. 5.** System Diagram for the training process

1. **Experimental Results**

The results of our experimentation are described in this section, divided into nanoparticle detection results and oligomeric state classification results.

**4.1. Detecting Particles**

For validation purposes, the confidence threshold for the bounding box prediction was set to 0.25. We calculated the precision, recall, and the mean average precisions at 0.50 IoU and 0.95 IoU. The full results for detecting each particle instance are given in Table 1. below.

**Table 1.** Performance of the Training, Validation and Testing process.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Precision** | **Recall** | **mAP@50** | **mAP@50:95** |
| Training | 0.948 | 0.926 | 0.925 | 0.458 |
| Validation | 0.975 | 0.951 | 0.962 | 0.520 |

As seen in the table, the model has a precision of 0.898 and a recall of 0.827 on the validation data. We calculate the F1-score to be 0.861, which is a good starting result especially for such a challenging detection task. Fig. S6 in the supplementary section shows the confusion matrix for this model.

Overall, the accuracy of this model, calculated from the true negative and true positive results, is 76.2%.

**4.2. Counting and Classifying Particles**

The XGBoost model characterizes the oligomeric state with a validation accuracy of 93.3%. This is an excellent result and shows that our pixel-wise features are effective at classifying oligomeric state. This number is, however, based off our labelled dataset, which has very accurate bounding boxes. Since the extraction of pixel-wise features depends on the boxes, poor bounding box localization could lead to lower downstream accuracy.

To validate the results of the full pipeline, we first carry out inference on an image from the validation dataset, then test the results of that with our particle classifier. The results of 8 validation images are given in Table 2. The true value from SEM was calculated using ImageJ, but we did not test ImageJ on optical images, as it completely fails to distinguish between monomers and oligomers.

**Table 2** Performance metrics of correlated images (laser-excited)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Image** | **True Value from SEM** | | **Predicted from Optical by our method** | | **Percentage (%) of Error** | |
| Monomer | Oligomers | Monomers | Oligomers | Monomer | Oligomers |
| 1 | 31 | 5 | 35 ± 4.5% | 5 ± 4.5% | 11.4% | 0.00% |
| 2 | 30 | 4 | 37 ± 4.5% | 6 ± 4.5% | 23.3% | 50.0% |
| 3 | 48 | 4 | 47 ± 4.5% | 3 ± 4.5% | 2.12% | 25.0% |
| 4 | 48 | 4 | 48 ± 4.5% | 1 ± 4.5% | 0.00% | 75.0% |
| 5 | 39 | 5 | 38 ± 4.5% | 2 ± 4.5% | 8.00% | 60.0% |
| 6 | 39 | 5 | 49 ± 4.5% | 1 ± 4.5% | 20.4% | 80.0% |
| 7 | 22 | 7 | 33 ± 4.5% | 2 ± 4.5% | 50.0% | 71.0% |
| 8 | 36 | 4 | 38 ± 4.5% | 9 ± 4.5% | 5.26% | 55.5% |

In this scenario, the RMSE for counting monomers was 6.245. The RMSE for counting oligomers is 3.62. The MAE for the count of monomers was 4.75 and for oligomers, it was 3.12.

The high error rate for detecting oligomers through this approach is expected due to limited data, but monomer detection is highly accurate at 84.9% accuracy. Weighing the overall accuracy by the average number of monomers and oligomers, we find that the average accuracy of the whole system is 80.7%.

This accuracy is an excellent starting point and may be improved by developing the particle detection model further and exploring other possible pixel-wise features.

1. **Testing with Non-Correlated Images**

The goal of our research was to find a way to count the number of AuNPs in optical images when there are no correlated SEM images available. As discussed in the introduction, optical images have advantages over SEM images such as higher throughput and speed, lower cost, and a non-destructive nature.

In this section, we show the results of applying our model in white-light excited optical images. This challenges the domain adaptability of our model as well as its basic particle detection and oligomeric state classification abilities.

In order to get a comparable set of results, we used ImageJ to detect the particles as well. We will discuss the differences between existing ‘manual’ methods and our method.

**5.1. Time Consumption**

With ImageJ, it took anywhere from half a minute to several minutes to fine-tune the thresholding levels and select the optimal filter parameters in order to get an accurate count. Variations in the image, such as the background being lighter than usual or particles being dimmer than usual, can lead to additional processing time.

Additionally, ImageJ does not return results for different types of AuNPs at the same time. Multiple runs of the program, with different settings, are needed to count monomers and oligomers.

By contrast, the deep learning model returned results in less than a second regardless of perturbations in the image.

**5.2. Uncorrelated Counting Comparison**

Does our method give results comparable to ImageJ?

To study this, we used our method and ImageJ to count the number of monomers and oligomers in the test dataset parallelly. As seen earlier in Fig. 2., the monomers and oligomers in these test images are much more distinct compared to the optical images used for training. Therefore, we’re able to easily distinguish between the two and note the number for counting accurately.

Table 3 shows the results of counting monomers and oligomers for each of the 20 images in this test set. We can see that the average divergence between our method and ImageJ for counting the number of particles is 16.56%.

**Table 3.** Counting Monomers and Oligomers in Uncorrelated Images

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Image No.** | **Monomers** | **Oligomers** | **Total** | **ImageJ Total** | **Divergence** |
| **1** | 191 | 67 | 258 | 285 | 9.94 |
| **2** | 223 | 90 | 313 | 217 | 36.23 |
| **3** | 228 | 96 | 324 | 247 | 26.97 |
| **4** | 220 | 31 | 251 | 250 | 0.39 |
| **5** | 231 | 58 | 289 | 236 | 20.19 |
| **6** | 416 | 79 | 495 | 300 | 49.06 |
| **7** | 239 | 77 | 316 | 288 | 9.27 |
| **8** | 205 | 61 | 266 | 239 | 10.69 |
| **9** | 237 | 71 | 308 | 244 | 23.19 |
| **10** | 127 | 10 | 137 | 157 | 13.61 |
| **11** | 260 | 70 | 330 | 298 | 10.19 |
| **12** | 231 | 76 | 307 | 308 | 0.33 |
| **13** | 192 | 64 | 256 | 297 | 14.83 |
| **14** | 206 | 78 | 284 | 239 | 17.21 |
| **15** | 193 | 89 | 282 | 256 | 9.67 |
| **16** | 233 | 101 | 334 | 259 | 25.30 |
| **17** | 172 | 72 | 244 | 236 | 3.33 |
| **18** | 165 | 27 | 192 | 209 | 8.48 |
| **19** | 109 | 46 | 155 | 186 | 18.18 |
| **20** | 128 | 25 | 153 | 195 | 24.14 |

Generally, ImageJ tends to count a lower total compared to our method. However, without actually labelling images to serve as a test dataset, it’s impossible to determine which one is the more accurate method.

**5.3. Accuracy in Uncorrelated Images**

It is impossible to draw clear conclusions about the accuracy of our methods compared to ImageJ without a quantitative comparison. Therefore, we processed the test set of images and labelled them manually. This process is not as accurate as that of our training set due to a lack of SEM ground truth, but still comes close to be an effective comparison.

A screenshot of a computer generated image

Description automatically generated

A screenshot of a computer generated image

Description automatically generated

**Fig. 6.** A sample processed test image and its pixel intensity heightmap. The heightmap is an easy way to visualize the difference between monomers and oligomers.

To preprocess the images, all pixels with an intensity value lower than or equal to 10 were zeroed out to remove the background. Non-local means denoising [26] was used to clean the image further, using a filter strength of 15, a 5x5 template window and a 15x15 search window.

After processing it, we carry out inference using our method and with ImageJ to detect the number of particles. The results are given in Table 4.

**Table 4.** Performance metrics for detection on non-correlated white-light excited images.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Images** | **Precision** | **Recall** | **mAP@50** | **mAP@50:95** |
| ImageJ | - | - | - | - |
| Our Method | 0.837 | 0.735 | 0.762 | 0.321 |

The F1 score for our method on this set was 0.783, and the overall accuracy on this dataset with our method was 52.3%. We attribute the lower accuracy here to the extreme dimness of many particles. This may be rectified in future work by using additional training data.

[INSERT IMAGEJ RESULTS AND DISCUSSION HERE]

1. **Conclusion**

In this study, we designed a novel method to detect the total number of gold nanoparticles in optical images and determine their oligomeric state. Previous research has neglected optical imaging due to the difficulty of analyzing the images. We show that this is not the case, and that using deep learning we can accurately detect and classify monomers and oligomers.

We achieved an accuracy of 76.2% for the detection process, and 93.3% for determining oligomeric state. Overall, the weighted average accuracy of our method is 80.7%. We also compare our results to using ImageJ, and find that our method is faster by an order of magnitude, does not require any manual intervention, while having either comparable or even much better accuracy depending on the type of optical image.

In the future, we hope to collect more images to increase the accuracy of these models and allow them to better differentiate distinct higher order oligomers such as trimers and tetramers, particularly for densely populated nanoparticle images. The proposed method was evaluated on spherical-shaped nanoparticles, but we also plan to extend our work to detect the boundaries and analyze other shaped NPs such as ellipses, dumbbells, bipyramids, triangles and nanocubes from optical images.

Our deep learning method offers an efficient and automated solution for detecting nanoparticles and quantifying oligomerization within polymer matrices using exclusively optical images. We expect that the proposed non-destructive imaging technique will be particularly valuable for assessing nanoparticle uptake and aggregation for studying membrane protein interactions and drug delivery in live cells.

**Author Contributions**

Author Abu S. M. Mohsin contributed to conception and design, data collection, analysis and interpretation of results, and manuscript preparation. Author Shadab H. Choudhury contributed to data labelling and preprocessing, deep learning model design and training, analysis and interpretation of results, and manuscript preparation.

**Data Availability Statement**

Data will be made available on reasonable request made to the corresponding author.

**Declaration of Competing Interest**

The authors declare no competing interests.

**Acknowledgment**

The authors would like to thank Brac University, Bangladesh and Swinburne University of Technology, Australia, for research support.

*References [UPDATE NEEDED]*

[1] A. S. M. Mohsin, Sujoy Mondal, Monica Mobashera, Afrida Malik, Maliha Islam, and Maisha Rubaiat, "Efficiency improvement of thin film solar cell using silver pyramids array and antireflective layer" *Heliyon,* vol. 9, no. 6, 2023.

[2] A. S. M. Mohsin, Monica Mobashera, Afrida Malik, Maisha Rubaiat, and Maliha Islam, "Light trapping in thin-film solar cell to enhance the absorption efficiency using FDTD simulation," *Journal of Optics* vol. 49, pp. 523-532, 2020.

[3] A. S. Mohsin, and Mariam B. Salim, "Probing the intracellular refractive index and molecular interaction of gold nanoparticles in HeLa cells using single particle spectroscopy," *International journal of nanomedicine,* pp. 6019-6028, 2018.

[4] A. S. Mohsin, and Mariam B. Salim, "Probing the plasmon coupling, quantum yield, and effects of tip geometry of gold nanoparticles using analytical models and FDTD simulation.," *IEEE Photonics Journal 10,* vol. 10, no. 3, pp. 1-10, 2018.

[5] A. S. M. Mohsin, and Ahmed, Fahmida, "Study the optical property of gold nanoparticle and apply them to design bowtie nanoantenna using FDTD simulation" *Journal of Optics,* vol. 51, no. 3, pp. 543-551, 2022.

[6] A. S. M. Mohsin, "Aggregation and uptake kinetics of gold nanoparticles in biological cells, using plasmon coupling and image correlation spectroscopy," PhD, Swinburne University of Technology Melbourne, Australia, 2015.

[7] A. Palmer and N. Thompson, "Molecular aggregation characterized by high order autocorrelation in fluorescence correlation spectroscopy " *Biophysical Journal* vol. 52(2), no. 2, pp. 257–270, 1987.

[8] N. O. Peterson, "Scanning fluorescence correlcation spectroscopy. i. theory and simulation of aggregation measurements," *Biophysical Journal* vol. 49, pp. 809–815, 1986.

[9] M. Mirzaei, & Rafsanjani, H. K. , "An automatic algorithm for determination of the nanoparticles from TEM images using circular hough transform. ," *Micron, ,* vol. 96,, pp. 86-95, 2017.

[10] G. Dahy, Soliman, M. M., Alshater, H., Slowik, A., & Hassanien, A. E. , "Optimized deep networks for the classification of nanoparticles in scanning electron microscopy imaging," *Computational Materials Science,* vol. 223, p. 112135, 2023.

[11] A. B. Oktay, & Gurses, A. , "Automatic detection, localization and segmentation of nano-particles with deep learning in microscopy images. ," *Micron,* vol. 120, 113-119., pp. 113-119, 2019.

[12] Y. Meng, Zhang, Z., Yin, H., & Ma, T. , "Automatic detection of particle size distribution by image analysis based on local adaptive canny edge detection and modified circular Hough transform. ," *Micron, ,* vol. 106, pp. 34-41, 2018.

[13] C. A. Schneider, Rasband, W. S., & Eliceiri, K. W., "NIH Image to ImageJ: 25 years of image analysis. ," *Nature Methods,,* vol. 9(7), pp. 671–675, 2012.

[14] E. Z. Qu, Jimenez, A. M., Kumar, S. K., & Zhang, K. , "Quantifying Nanoparticle Assembly States in a Polymer Matrix through Deep Learning. Macromolecules,," *Macromolecules,* vol. 54(7), , pp. 3034-3040., 2021.

[15] A. V. Nartova, Mashukov, M. Y., Astakhov, R. R., Kudinov, V. Y., Matveev, A. V., & Okunev, A. G. , "Particle Recognition on Transmission Electron Microscopy Images Using Computer Vision and Deep Learning for Catalytic Applications. Catalysts, 12(2), 135.," *Catalysts,* vol. 12(2), 135, p. 135, 2022.

[16] A. G. Okunev, Mashukov, M. Y., Nartova, A. V., & Matveev, A. V. , "Nanoparticle recognition on scanning probe microscopy images using computer vision and deep learning," *Nanomaterials, ,* vol. 10(7), p. 1285., 2020.

[17] J. D. López Gutiérrez, Abundez Barrera, I. M., & Torres Gómez, N., " Nanoparticle Detection on SEM Images Using a Neural Network and Semi-Synthetic Training Data," *Nanomaterials, ,* vol. 12(11), p. 1818, 2022.

[18] A. Kaphle, Jayarathna, S., Moktan, H., Aliru, M., Raghuram, S., Krishnan, S. and Cho, S.H., , "Deep Learning-Based TEM Image Analysis for Fully Automated Detection of Gold Nanoparticles Internalized Within Tumor Cell," *Microscopy and Microanalysis,* p. p.ozad066., 2023.

[19] P. Monchot, Coquelin, L., Guerroudj, K., Feltin, N., Delvallée, A., Crouzier, L., & Fischer, N. , " Deep learning based instance segmentation of titanium dioxide particles in the form of agglomerates in scanning electron microscopy," *Nanomaterials,* vol. 11(4), p. 968, 2021.

[20] S. K. Melanthota, Gopal, D., Chakrabarti, S., Kashyap, A. A., Radhakrishnan, R., & Mazumder, N., "Deep learning-based image processing in optical microscopy," *Biophysical Reviews,* vol. 14(2), pp. 463-481, 2022.

[21] Y. Xu, Xu, D., Yu, N., Liang, B., Yang, Z., Asif, M.S., Yan, R. & Liu, M. , " Machine Learning Enhanced Optical Microscopy for the Rapid Morphology Characterization of Silver Nanoparticles," *ACS Applied Materials & Interfaces,* vol. 15(14), pp. 18244-18251., 2023.

[22] A. C. Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, 曾逸夫(Zeng Yifu), Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel & Mrinal Jain, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation (v7.0)," *Zenodo,* 2022.

[23] J. a. C.-E. Terven, D., " A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. arXiv preprint arXiv:2304.00501," *arXiv preprint arXiv:2304.00501.*2023.

[24] T. Chen, & Guestrin, C., " Xgboost: A scalable tree boosting system," *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining,* pp. 785-794, 2016.

[25] RangeKing. (<https://github.com/RangeKing>). (2022). *Brief summary of YOLOv8 model structure,* .

[26] A. Buades, Coll, B. and Morel, J.M.,, "Non-local means denoising.," *Image Processing On Line, ,* vol. 1, pp. 208-212, 2011.