Instant Particle Size Distribution Measurement Using CNNs Trained on Synthetic Data

Yasser Eljarida Youssef Iraqi Loubna Mekouar
College of Computing, University Mohammed VI Polytechnic
Benguerir, Morocco
{yasser.eljarida, youssef.iraqi, loubna.mekouar}@um6p.ma

Abstract

Accurate particle size distribution (PSD) measurement is important in industries such as mining, pharmaceuticals, and fertilizer manufacturing, significantly influencing product quality and operational efficiency. Traditional PSD methods like sieve analysis and laser diffraction are manual, time-consuming, and limited by particle overlap. Recent developments in convolutional neural networks (CNNs) enable automated, real-time PSD estimation directly from particle images. In this work, we present a CNN-based methodology trained on realistic synthetic particle imagery generated using Blender's advanced rendering capabilities. Synthetic data sets using this method can replicate various industrial scenarios by systematically varying particle shapes, textures, lighting, and spatial arrangements that closely resemble the actual configurations. We evaluated three CNN-based architectures—ResNet-50, InceptionV3, and EfficientNet-B0—for predicting critical PSD parameters (d10, d50, d90). Results demonstrated comparable accuracy across models, with EfficientNet-B0 achieving the best computational efficiency suitable for real-time industrial deployment. This approach shows the effectiveness of realistic synthetic data for robust CNN training, which offers significant potential for automated industrial PSD monitoring.

1. Introduction

Accurate particle size distribution (PSD) directly impacts material properties, process efficiency, and product consistency across multiple industries, including mining, pharmaceuticals, and agriculture. Pharmaceutical manufacturing relies heavily on accurate PSD control for drug uniformity, bioavailability, and effectiveness [8], while fertilizer production requires consistent PSD management for optimal nutrient release and spreading performance [5]. Given these implications, accurate and reliable real-time PSD measurement methods are essential.

Conventional PSD measurement approaches, such as sieve analysis and laser diffraction [5], have significant limitations. Sieving is straightforward but struggles with accuracy, especially for fine particles [5]. Laser diffraction provides precise measurements but relies heavily on the assumption of spherical particles, limiting its effectiveness for irregular particle shapes common in practical scenarios [2]. While microscopy-based methods can deliver detailed particle morphology, they remain labor-intensive, slow, and prone to human annotation errors [10].

Recent advancements in convolutional neural networks (CNNs) have opened new opportunities for automating PSD estimation directly from particle images [9, 14]. Unlike traditional segmentation-based computer vision methods, CNNs are robust to overlapping particles and variations in lighting and texture, significantly reducing annotation complexity and improving accuracy. However, CNN training generally requires large, well-annotated datasets, which are expensive and challenging to obtain through manual labeling.

Synthetic data generation addresses this issue by enabling scalable creation of fully annotated datasets with precise control over particle characteristics [4, 13]. Earlier synthetic approaches, such as discrete element method (DEM) simulations, accurately captured particle physics but often lacked visual realism, limiting model generalization [13]. Modern graphics engines like Blender [3] provide sophisticated physics simulations and photorealistic rendering capabilities, enabling generation of highly realistic synthetic particle imagery. Recent studies also emphasize the significant impact of particle shape and size distribution on material properties [6], further motivating the creation of realistic synthetic datasets.

In this work, we leverage Blender's advanced rendering engine to generate realistic synthetic datasets closely matching actual fertilizer granulation scenarios, with a particular focus on critical PSD metrics (d10, d50, d90). We comprehensively evaluate three popular CNN

architectures—ResNet-50, InceptionV3, and EfficientNet-B0—for their ability to accurately predict PSD parameters directly from these synthetic images.

Our key contributions include: (i) introducing a robust CNN-based PSD estimation method optimized for real-time industrial monitoring; (ii) developing a flexible, automated synthetic data generation pipeline using Blender, capable of replicating diverse industrial scenarios with high fidelity; (iii) providing comprehensive evaluations demonstrating CNN performance comparable to traditional PSD measurement methods; and (iv) highlighting the practical feasibility and computational efficiency of deploying CNN models, especially EfficientNet-B0, in resource-constrained environments.

The remainder of this paper is structured as follows: Section 2 reviews related PSD estimation methods. Section 3 details our synthetic data generation pipeline. Section 4 outlines CNN training and model selection strategies. Section 5 presents experimental results and analyses. Section 6 discusses practical implications and domain adaptation strategies, and Section 7 provides concluding remarks.

2. Background and Related Works

2.1. Classical PSD Measurement Methods

Classical methods for measuring PSD remain widely adopted despite inherent limitations [5]. Sieve analysis is the most common method due to its simplicity and low cost, yet it struggles with accurately handling fine particles or scenarios involving significant overlap [5]. Laser diffraction provides higher accuracy and a broader measurable range but assumes spherical particle shapes, leading to inaccuracies when analyzing irregularly shaped particles [2]. Sedimentation methods and Coulter counters offer precision, particularly for fine particle ranges, but both methods require meticulous sample preparation and extensive analysis time, thus limiting their application in real-time industrial scenarios [1, 8].

2.2. Computer Vision Methods for PSD Estimation

Computer vision-based methods provide solutions to many challenges posed by classical PSD techniques, enabling automated and rapid analysis directly from images [10, 14]. Early vision-based PSD estimation relied heavily on explicit particle segmentation from images. However, these methods frequently failed under challenging conditions such as overlapping particles, varied lighting, or complex backgrounds [5].

Recent advances using CNNs address these limitations by learning to directly estimate PSD metrics (e.g., d10, d50, d90) from raw image data without explicit segmentation [9, 14]. CNN-based regression models have shown significant robustness to common real-world issues such as overlapping particles, irregular shapes, and variable lighting conditions [8, 9]. However, training effective CNN models typically requires large annotated datasets, which are difficult and costly to acquire from real-world scenarios [13].

2.3. Synthetic Data Generation for PSD

Synthetic data generation has emerged as a practical solution to overcome the data acquisition bottleneck by allowing researchers to generate large-scale annotated datasets through computational simulations [13]. Previous synthetic generation methods, such as those based on DEM simulations, produced accurately annotated data but often lacked visual realism in particle appearance and texture [13].

Recent progress by Dahmen et al. [4] highlighted the potential of neural rendering techniques for generating realistic synthetic scanning electron microscopy (SEM) images efficiently. This approach significantly improved realism while maintaining low computational costs compared to traditional methods. Motivated by these developments, our work adopts Blender, a 3D graphics engine [3], to generate synthetic particle images with detailed particle geometries, realistic textures, and highly controllable environmental parameters. By incorporating advanced rendering techniques and systematic randomization of particle properties such as size distribution and spatial arrangements, we produce synthetic datasets closely aligned with real-world particle imaging scenarios. This capability significantly enhances the effectiveness of CNN-based PSD estimation methods, enabling robust and accurate predictions in practical applications.

3. Synthetic Data Generation

Manually annotating large-scale datasets for PSD analysis is costly, error-prone, and challenging to scale. To overcome these limitations, we developed a fully automated synthetic data generation pipeline using Blender, an open-source 3D graphics software [3]. Blender makes it possible to have detailed control over particle geometries, materials, and environmental conditions, enabling the creation of diverse and realistic particle scenes.

All the configuration values that we will present in this section are just specific to our generated dataset and can be changed to simulate any given environment from particles shape and size to the environment setup.

3.1. Simulation Environment

Our virtual simulation environment replicates a realistic setup for taking sample images from processes such as fertilizer granulation or aggregate handling. The scene includes a flat square surface for particle deposition measuring 300 mm $\times$ 300 mm, and invisible collision walls surrounding the table to ensure particles remain within the

simulation area, consistent with the generated metadata (Fig. 1).

Particles were modeled using detailed 3D meshes imported as FBX files with physically accurate textures, including base color, metallic properties, roughness, and normal maps. The complexity of these particle models, shown in Fig. 2, significantly enhances the realism of the synthetic data, surpassing traditional DEM-based simulations, which typically lack such visual fidelity.

3.2. Automated Dataset Generation Pipeline

The dataset generation involves two interconnected Python scripts integrated within Blender's Python API:

1. **Metadata Generation**: This script randomly selects parameters for each simulation, including particle count, particle sizes, and positions. Particle sizes follow a truncated normal distribution defined as:

$$\text{size} \sim \text{TruncNormal}(a, b, \mu, \sigma), \tag{1}$$

   - (a, b) : Truncation bounds.
   - (μ) : Mean particle size.
   - (σ) : Standard deviation.

   For our dataset we used the configuration ranges shown in Table 1:

	Ranges
(μ)	[6 mm, 12 mm]
(σ)	[6 mm, 8 mm]
(a, b)	[0.1 mm, 20 mm]
Particle count	[700, 1000]

Table 1. Data configuration

   The metadata for each simulation are stored as structured JSON files containing the following information: shape type, mean and standard deviation of particle sizes, table dimensions, particle count, and a list of individual particle parameters (size and x-y coordinates). Listing 1 shows an example of a sample metadata file.

Listing 1. Example JSON metadata describing a synthetic particle scene.

```
{
  "shape_type": "crushed_rock",
  "size_mean": 10.5,
  "size_sigma": 7.2,
  "table_size": 300,
  "samplesize": 920,
  "particles": [
    {"size": 11.80, "x": -11.12, "y": -2.31},
    {"size": 7.97, "x": -118.32, "y": 119.38},
    ...
  ]
}
```

2. **Scene Construction and Rendering**: The second script reads the generated JSON metadata, places particles accordingly, and initiates Blender's rigid-body physics simulation. Once particles settle, the final scene state is rendered using Blender's Cycles rendering engine with GPU (CUDA) acceleration, significantly enhancing rendering efficiency. The rendered images accurately reflect their corresponding metadata annotations.

This pipeline was executed on a High Performance Computing (HPC) system using NVIDIA A100 GPUs, facilitating rapid and scalable dataset generation. For our training dataset, we rendered approximately 4500 images (4668 images across three rendering jobs), each taking in average one minute to render, depending on the sample size.

3.3. Dataset Realism and Diversity

Our synthetic images exhibit realistic lighting, detailed textures, and diverse particle arrangements (Fig. 3). Compared to previous DEM-based synthetic datasets [13], our Blender-based dataset achieves significantly greater visual realism, closely approximating real-world scenarios and thus enhancing CNN model generalizability.

3.4. Advantages over Previous Methods

Our Blender-based synthetic data generation method offers several key advantages:

- *Enhanced Realism*: Complex particle geometries and detailed textures produce visually realistic datasets.
- *Flexibility and Control*: Adjustable parameters (lighting, particle properties, environment) enable replication of a broad range of real-world conditions.
- *Scalable Automation*: GPU-accelerated rendering combined with automated metadata handling allows efficient large-scale dataset creation.
- *Accurate Annotation*: Automated metadata generation ensures consistent and precise annotations, eliminating manual labeling errors.

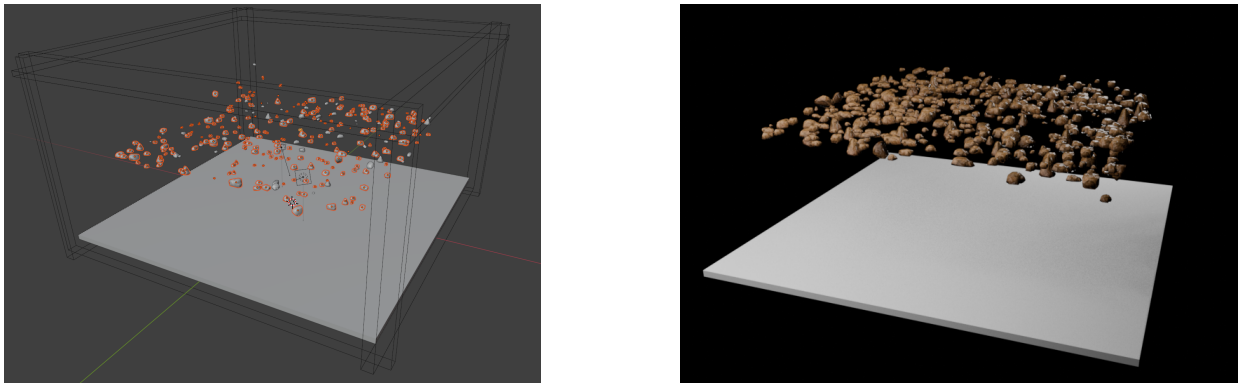
This pipeline efficiently generates realistic synthetic datasets essential for training robust and reliable CNN models for practical PSD estimation tasks.

4. CNN Model Selection and Training

To evaluate the effectiveness of our synthetic dataset for PSD estimation, we selected CNNs known for strong performance, computational efficiency, and suitability for regression tasks. Our primary goal was to accurately predict critical PSD parameters (d10, d50, d90) directly from particle images.

4.1. Model Selection

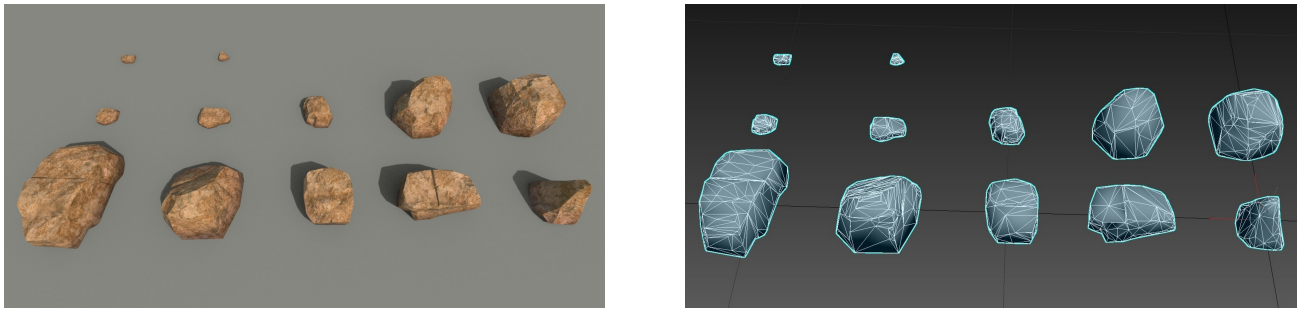
We selected three widely-adopted CNN architectures, each chosen for specific strengths relevant to particle analysis:



(a) Setup inside the Blender interface.

(b) Rendered image depicting particle deposition before fall.

Figure 1. Simulation environment setup in Blender.



(a) Rendered particle models with realistic textures.

(b) Wireframe mesh showing particle geometric complexity.

Figure 2. Particle model geometry and texture realism.

- *ResNet-50* [7]: Chosen for its robust feature extraction capabilities, using residual connections to effectively learn complex visual patterns without suffering from vanishing gradients.
- *EfficientNet-B0* [12]: Selected because of its balanced trade-off between computational efficiency and predictive accuracy, making it particularly suitable for real-time industrial monitoring scenarios.
- *InceptionV3* [11]: Included for its ability to extract multi-scale features via parallel convolutional operations, effectively capturing variations in particle sizes, shapes, and textures.

4.2. Training Procedure

All models were pretrained on the ImageNet dataset, as preliminary experiments demonstrated improved convergence speed and predictive accuracy compared to training from scratch.

The models were fine-tuned to perform regression directly on the PSD parameters (d10, d50, d90). After some experimentation, we ended up using these Training hyperparameters:

- *Loss Function*: Mean Squared Error (MSE).
- *Optimizer*: Adam optimizer with an initial learning rate of 1×10^{-4}, adaptively reduced with respect to the validation plateau.
- *Data Augmentation*: Random rotations (0°, 90°, 180°, 270°), horizontal flipping were employed to improve generalization and mitigate overfitting.
- *Epochs*: Training ran up to 50 epochs which showed satisfactory convergence.

TensorBoard was utilized throughout training to monitor convergence, facilitating real-time hyperparameter tuning and performance analysis.

4.3. Computational Resources

Model training was conducted on an HPC cluster. Specifically, GPU-accelerated training was performed on nodes equipped with NVIDIA A100 GPUs, enabling rapid iteration, efficient experimentation, and significant reductions in overall training duration which took approximately 3 hours for each model.

The effectiveness of this model selection and training approach is evaluated in detail in the following experimental section.

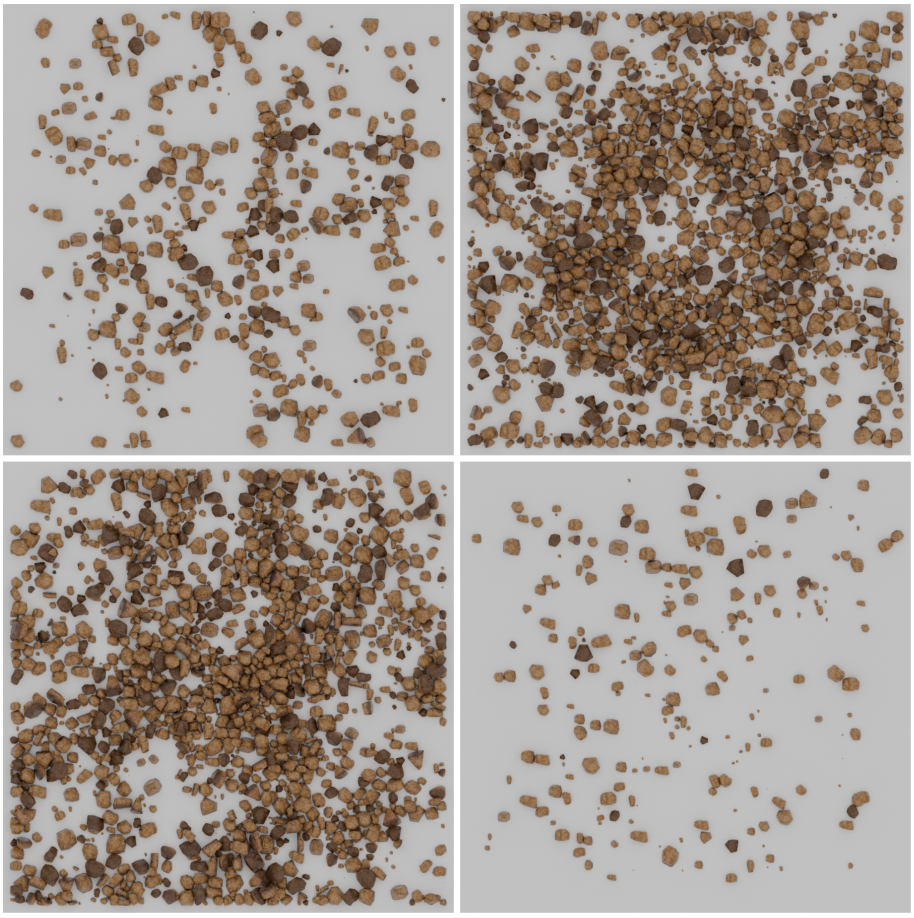


Figure 3. Examples of synthetic images demonstrating realistic variability in particle sizes, shapes, textures, and spatial distributions.

5. Experimental Results and Analysis

In this section, we compared the predictive accuracy and computational efficiency between the three CNN architectures (ResNet-50, InceptionV3, and EfficientNet-B0) using our synthetic dataset comprising approximately 5,000 annotated images. The evaluation focused on the prediction of key PSD parameters: d10, d50, and d90.

The performance of the models was quantitatively assessed using the coefficient of determination (R^2), mean squared error (MSE), and mean absolute error (MAE), providing robust measures of accuracy and consistency.

5.1. Detailed Quantitative Results

Table 2 presents detailed predictive performance metrics (R^2, MSE, MAE) for each CNN model across individual PSD parameters and overall. Although ResNet-50 showed marginally better predictive performance, the differences between the three models' results are minor (less than $\approx$ 0.1 mm error), indicating similar predictive capabilities.

Figure 4 visualizes the predicted versus ground truth PSD values for each model, we can see that the ResNet-50 model is more aligned, with minor variations in prediction quality with the other models.

Figure 5 shows training and validation loss curves monitored via TensorBoard. EfficientNet-B0 demonstrates efficient and rapid convergence behavior, which got close final validation loss to ResNet-50 and InceptionV3.

5.2. Computational Efficiency

Given the importance of computational speed for real-time PSD monitoring in industrial settings, we tested the inference speed of each CNN model under realistic hardware conditions. We conducted inference tests using a consumer-grade laptop equipped with an NVIDIA RTX 3060 GPU and an AMD Ryzen 7 5800H CPU, providing a practical and accessible benchmark scenario.

Inference speeds are summarized in Table 3. Although ResNet-50 showed slightly better accuracy, EfficientNet-B0 significantly outperformed other models in inference speed on both GPU and CPU. Specifically, EfficientNet-B0 can process approximately 37.75 frames per second (FPS) on GPU and 21.51 FPS on CPU, markedly faster

Table 2. Detailed performance comparison for CNN models on synthetic test data (GPU inference). The best results are in Bold.

CNN Model	Overall			d10			d50			d90		
	R^2	MSE	MAE	R^2	MSE	MAE	R^2	MSE	MAE	R^2	MSE	MAE
ResNet-50	**0.9987**	**0.0204**	**0.0773**	**0.9591**	**0.0315**	**0.0913**	**0.9904**	**0.0160**	**0.0747**	**0.9888**	**0.0137**	**0.0660**
InceptionV3	0.9966	0.0550	0.1796	0.8933	0.0821	0.2212	0.9738	0.0438	0.1651	0.9680	0.0391	0.1525
EfficientNet-B0	0.9959	0.0660	0.1973	0.8777	0.0941	0.2357	0.9667	0.0555	0.1841	0.9604	0.0484	0.1721

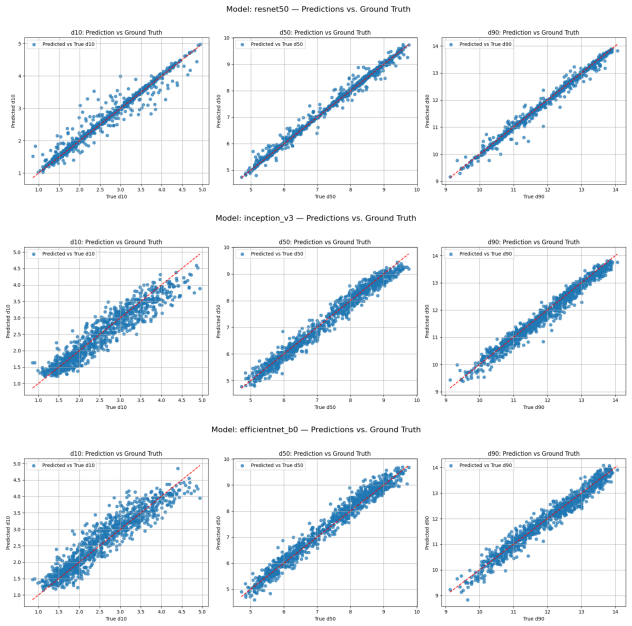


Figure 4. Predicted vs. ground truth PSD comparison.

compared to ResNet-50 (27.31 FPS GPU, 14.66 FPS CPU) and InceptionV3 (23.18 FPS GPU, 12.00 FPS CPU). This substantial improvement in inference speed, combined with its considerably lower parameter count, underscores EfficientNet-B0's practical advantage, particularly in resource-constrained environments.

Thus, despite minor accuracy trade-offs, EfficientNet-B0 offers the optimal balance between accuracy and computational efficiency, making it particularly suitable for real-time PSD estimation tasks in industrial applications.

Table 3. Inference speed comparison and model complexity on realistic laptop hardware (RTX 3060 GPU, AMD Ryzen 7 5800H CPU). The best results are in Bold.

CNN Model	GPU FPS	CPU FPS	GPU Time/Image (s)	Parameters (M)
ResNet-50	27.31	14.66	0.0366	25.6
InceptionV3	23.18	12.00	0.0431	23.9
EfficientNet-B0	**37.75**	**21.51**	**0.0265**	**5.3**

6. Discussion

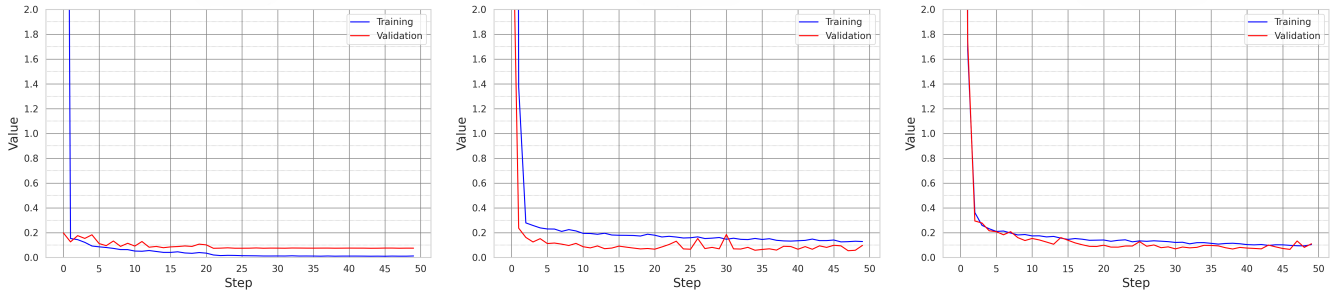
This study demonstrates the potential of using CNNs trained on realistic synthetic data generated via Blender to accurately estimate PSD. Our approach leverages Blender's advanced rendering capabilities, allowing precise control over particle geometries, textures, spatial arrangements, and lighting conditions. This meticulous control enables the creation of synthetic images that closely match real-world industrial scenarios, thereby significantly improving model robustness compared to previous approaches relying on simplified particle representations.

One key advantage of our method is the ease with which diverse scenarios can be generated. By adjusting parameters such as particle shape, size distribution (through truncated normal distributions), lighting intensity, and camera perspectives, the synthetic data pipeline becomes highly adaptable to various industrial processes or materials. Additionally, automation within Blender facilitates rapid, scalable, and cost-effective dataset generation, removing the reliance on labor-intensive manual annotations typically required for training robust CNN models.

In evaluating three popular CNN architectures—ResNet-50, InceptionV3, and EfficientNet-B0—we found minor differences in predictive performance. Although ResNet-50 achieved slightly better numerical results, EfficientNet-B0 stood out in computational efficiency, offering substantially faster inference speeds, particularly on CPU hardware. This computational advantage suggests that EfficientNet-B0 provides the most practical balance between accuracy and real-time processing, making it highly suitable for deployment in resource-constrained industrial environments or edge devices.

The realism of our synthetic dataset substantially narrows the traditional "domain gap" between synthetic and real-world imagery, yet minor discrepancies inevitably remain. Integrating targeted domain adaptation methods, such as incremental fine-tuning on limited real data or employing adversarial training techniques, offers promising ways to further enhance model generalization. These strategies can be seamlessly incorporated into the synthetic data pipeline, helping to continuously refine and align synthetic scenarios more closely with real-world operational conditions.

In practical terms, while high-performance computing resources such as GPU clusters accelerate synthetic dataset creation, our methodology does not inherently require such infrastructure. The pipeline remains operationally feasible on standard GPU-equipped workstations, but with longer rendering times. which means that computational resources



(a) ResNet50 (b) Inception V3 (c) EfficientNet-B0

Figure 5. Training and validation loss curves showing model convergence for (a) ResNet50, (b) Inception V3, and (c) EfficientNet-B0.

should not pose a barrier to adopting our approach broadly, although they directly influence dataset generation speed.

To encourage trust in industrial deployment, we suggest initially validating CNN models alongside conventional PSD measurement techniques. This hybrid approach allows comprehensive verification and fosters confidence in automated systems, paving the way for full integration into real-time process control loops. Further enhancing model interpretability, such as employing explainability techniques (e.g., saliency maps), can also support industrial adoption by increasing transparency in model predictions.

Ultimately, the flexibility, realism, and automation offered by synthetic data generation position CNN-based PSD estimation as a highly promising alternative to traditional measurement methods, presenting substantial opportunities for improving quality control, optimizing production efficiency, and reducing operational costs across various industries.

7. Conclusion

This study introduced an efficient and accurate approach to the estimation of PSD in real time using CNNs trained exclusively on synthetic datasets. Using Blender's advanced rendering capabilities, we generated highly realistic particle imagery, addressing critical limitations associated with traditional PSD measurement techniques, such as manual sampling, offline analysis, and challenges related to overlapping particles.

Experimental evaluations demonstrated that CNN architectures—ResNet-50, InceptionV3, and EfficientNet-B0—can effectively predict PSD parameters (d10, d50, and d90) directly from images. Among these models, EfficientNet-B0 emerged as particularly well suited for industrial deployment due to its superior inference speed, especially on resource-constrained hardware, despite marginal and statistically insignificant accuracy differences compared to ResNet-50.

The flexibility of the synthetic data generation pipeline allows rapid adaptation to various industrial scenarios by simply adjusting particle properties, textures, and environmental settings, significantly improving training robustness and reducing data acquisition costs.

Further integration of targeted domain adaptation strategies, incremental fine-tuning on limited real-world data, and enhanced model interpretability techniques hold promise for further narrowing the synthetic-to-real gap. In general, the proposed method offers a practical, scalable, and cost-effective path to automated PSD monitoring, improving process control and product consistency in various industrial applications.

References

[1] Zinah Alabdali and F. Alkalid. Analysis of particle size in composite materials using image processing. *International Journal of Engineering*, 37:579–587, 2024. 2

[2] Isaac Capone, Kevin Hurlbutt, Andrew J. Naylor, Albert W. Xiao, and Mauro Pasta. Effect of the particle-size distribution on the electrochemical performance of a red phosphorus–carbon composite anode for sodium-ion batteries. *Energy & Fuels*, 33(5):4651–4658, 2019. PMID: 32063668. 1, 2

[3] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 1, 2

[4] Tim Dahmen. A neural model for high-performance scanning electron microscopy image simulation of porous materials. In *Synthetic Data for Computer Vision Workshop @ CVPR 2024*, 2024. 1, 2

[5] Francesco Di Maria, Francesco Bianconi, Caterina Micale, Stefano Baglioni, and Moreno Marionni. Quality assessment for recycling aggregates from construction and demolition waste: An image-based approach for particle size estimation. *Waste Management*, 48:344–352, 2016. 1, 2

[6] Jian Gong, Xianwei Pang, Yu Tang, Ming Liu, Jie Jiang, and Xiaoduo Ou. Effects of particle shape, physical properties and particle size distribution on the small-strain stiffness of granular materials: A dem study. *Computers and Geotechnics*, 165:105903, 2024. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian

Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 4

[8] Donggeun Kim, Jisu Song, and Jaesung Park. Quantifying soil particle settlement characteristics through machine vision analysis utilizing an rgb camera. *Agriculture*, 13(9), 2023. 1, 2

[9] Phongsathorn Kittiworapanya, Kitsuchart Pasupa, and Peter Auer. Particle size estimation in mixed commercial waste images using deep learning. In *Proceedings of the 12th International Conference on Advances in Information Technology*, New York, NY, USA, 2021. Association for Computing Machinery. 1, 2

[10] Ruggero Labati, Angelo Genovese, Enrique Ballester, Vincenzo Piuri, and Fabio Scotti. 3d granulometry using image processing. *IEEE Transactions on Industrial Informatics*, PP:1–1, 2018. 1, 2

[11] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. 4

[12] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019. 4

[13] Pin Zhang and Zhen-Yu Yin. A novel deep learning-based modelling strategy from image of particles to mechanical properties for granular materials with cnn and bilstm. *Computer Methods in Applied Mechanics and Engineering*, 382: 113858, 2021. 1, 2, 3

[14] Bowen Zheng, Guiling Sun, Zhaonan Meng, and Ruili Nan. Vegetable size measurement based on stereo camera and keypoints detection. *Sensors*, 22(4), 2022. 1, 2