Deep Segmentation of Fuel Rod End Plugs for Nuclear Assembly Inspection

Vojtěch Bláha[1,2], Jan Blažek[1,3]

[1]Research Centre Řež, Czech Republic
[2]Charles University, Faculty of Mathematics and Physics, Department of Software and Computer Science Education, Prague, Czech Republic
[3]Institute of Information Theory and Automation, Czech Academy of Sciences, Prague, Czech Republic

Abstract

In nuclear power generation, precise geometric conformity of fuel rods is essential for reactor safety and optimal performance. We present an end-to-end image processing system built around deep learning-based segmentation for automated inspection of fuel rod end plugs in grayscale side-view images. The system handles challenges posed by metallic surfaces with complex light reflections and structural ambiguity. We train U-Net and U-Net 3+ decoders with EfficientNet-V2-S and ConvNeXt-Tiny encoders on both real and synthetically rendered datasets. The proposed approach is evaluated across multiple configurations, with the best model achieving an average positional error of 0.42 mm and standard deviation of 0.39 mm. Our results demonstrate the feasibility of deploying convolutional architectures in real-world industrial inspection workflows.

Keywords

nuclear fuel inspection, semantic segmentation, convolutional neural networks, deep learning, U-Net, synthetic data, industrial automation

Introduction

Nuclear power remains a key contributor to low-carbon electricity generation, offering high reliability and efficiency. The safe and effective operation of a nuclear reactor depends significantly on the structural integrity and precise geometry of its core components, in our case the fuel rods. These rods are sealed with metallic end plugs and arranged in grid-like structures within fuel assemblies. Accurate measurement of plug positions is essential for verifying manufacturing quality by determining the length of each fuel rod, which in turn is a key parameter for assessing structural conformity and fuel burnup. Detecting deviations allows operators to identify design flaws or damaged rods, enabling targeted replacements during scheduled outages and reducing the risk of in-service failures.

Our experience with the quality assurance (QA) involves visual inspection by human operators which are responsible for the image processing and measurement of rod growth. In past years, we developed our first image processing system that utilizes edge detection, morphological operations, and geometric heuristics and produces reproducible results with automation about 80%. These approaches, however, often fall short in robustness and precision, especially in the presence of complex reflections on metalic surfaces. Consequently, we register an increasing demand for automated, reliable, and accurate inspection systems that generate reproducible inspection data under industrial constraints.

The recent success of deep learning, particularly convolutional neural networks (CNNs), in computer vision tasks has paved the way for their application in industrial inspection. Semantic segmentation using CNNs allows for pixel-precise classification of image regions and is well-suited for tasks requiring precise object localization and measurement. However, deploying these methods in real-world nuclear inspection scenarios presents three main challenges:

- **Data scarcity:** Annotated data is limited due to confidentiality and cost of expert labeling.
- **Domain complexity:** As illustrated in Figure 1, reflections on metallic surfaces — common in real fuel assembly imagery — pose challenges for accurate segmentation. Figure 2 shows

ITAT'25: Information technologies – Applications and Theory, September 26–30, 2025, Telgart, Slovakia
vojtechblaha00@gmail.com (V. Bláha); blazek@utia.cas.cz (J. Blažek)

an example of a synthetically rendered fuel assembly designed to closely resemble the visual characteristics of real inspection data.

- **Industrial constraints:** The system must be reliable and verifiable, producing correct outputs or clearly indicating uncertainty, as required in safety-critical environments.

Figure 1: Photograph of a real nuclear fuel assembly showing reflective metallic surfaces that resemble the visual complexity encountered during inspection [1].

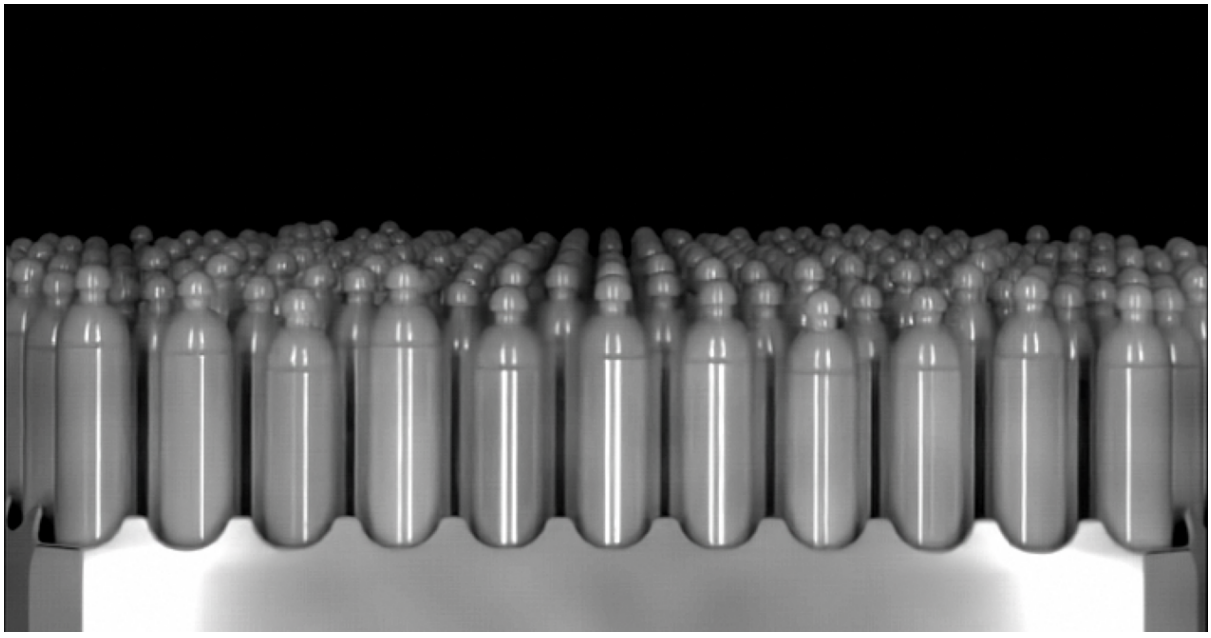

Figure 2: Example of a synthetically rendered nuclear fuel assembly designed to mimic real inspection conditions [2].

This paper proposes an image processing pipeline based on deep learning segmentation models, trained on nuclear industry specific data. We focus on the segmentation of fuel rod end plugs and adjacent grid structures from a grayscale side-view images produced during visual inspections. Our

method integrates modern CNN architectures — U-Net and U-Net 3+ — tailored for image segmentation with high-performing backbones such as EfficientNetV2-S and ConvNeXt-Tiny.

To address data scarcity and promote generalization, we develop a synthetic image generation pipeline using Blender, capable of rendering annotated fuel rod images with realistic geometry, textures, and lighting. We also apply targeted data augmentation strategies to better capture the variability of real-world data.

The core contributions of this work include:

1. A detailed semantic segmentation pipeline optimized for fuel rod inspection.
2. A synthetic dataset simulating images acquired during real fuel inspection.
3. A rigorous evaluation of model variants and training strategies, achieving sub-millimeter accuracy.
4. A comparison with legacy rule-based systems, showing significant improvements in speed and precision.
5. A deployment-ready model export with GPU acceleration.

In the following sections, we describe the industrial requirements and image characteristics, review related work, explain our data preparation and model architectures, present experimental results, and discuss implications for industrial deployment and future extensions.

1. Industrial Context and Problem Setup

Visual inspection of nuclear fuel rod end plugs is typically performed during scheduled outages of nuclear power plants. The goal is to assess the condition and assembly quality of fuel rods, including both those in active use and those nearing the end of their fuel cycle.

An example frame is shown in Figure 3, highlighting end plug boundaries and the grid's upper edge. Variations in appearance arise due to surface oxidation and irradiation reflections.

Each image captures the upper part of the fuel assembly at the point where the fuel rods terminate, using a high-resolution grayscale radiation-resistant camera under controlled lighting. Our scene typically contains eleven vertically arranged fuel rods end plug and the top edge of a horizontal spacer grid. The imaging setup is designed so that the camera sensor is aligned parallel to the plane formed by the peripheral rods, although slight deviations may occur due to mechanical tolerances, resulting in minor left-right geometric skew.

The primary goal is to estimate the full 2D position and shape of each visible end plug head, as well as to accurately localize the upper edge of the spacer grid which serves as a reference point. The perspective transformation used to correct geometric skew is identical to the original rule-based system; the main advantage of our approach lies in the improved robustness and accuracy of segmentation through deep learning.

For each rod depicted in the image, we generate binary masks for two specific regions:

- The visible plug head
- The upper edge of the horizontal spacer grid

The choice to use masks is a design decision; the 2D position of plugs could be estimated through other methods. While the shape of the mask could, in principle, serve as an additional check for assembly correctness, our current approach primarily relies on the size of the masked area rather than detailed shape information.

Although no strict specification is provided, the acceptable positional deviation is typically around 1.5 mm, based on downstream processing requirements and empirical tolerances. The expected detection success rate is not formally quantified. Real-time performance is not a critical constraint, as the fuel assembly is scanned over several minutes, allowing sufficient time for repeated measurements and evaluation of rod lengths across the image sequence.

In this context, the segmentation model should achieve high localization precision, robustness to visual variability, and sufficient inference speed to support real-time on-site decision-making.

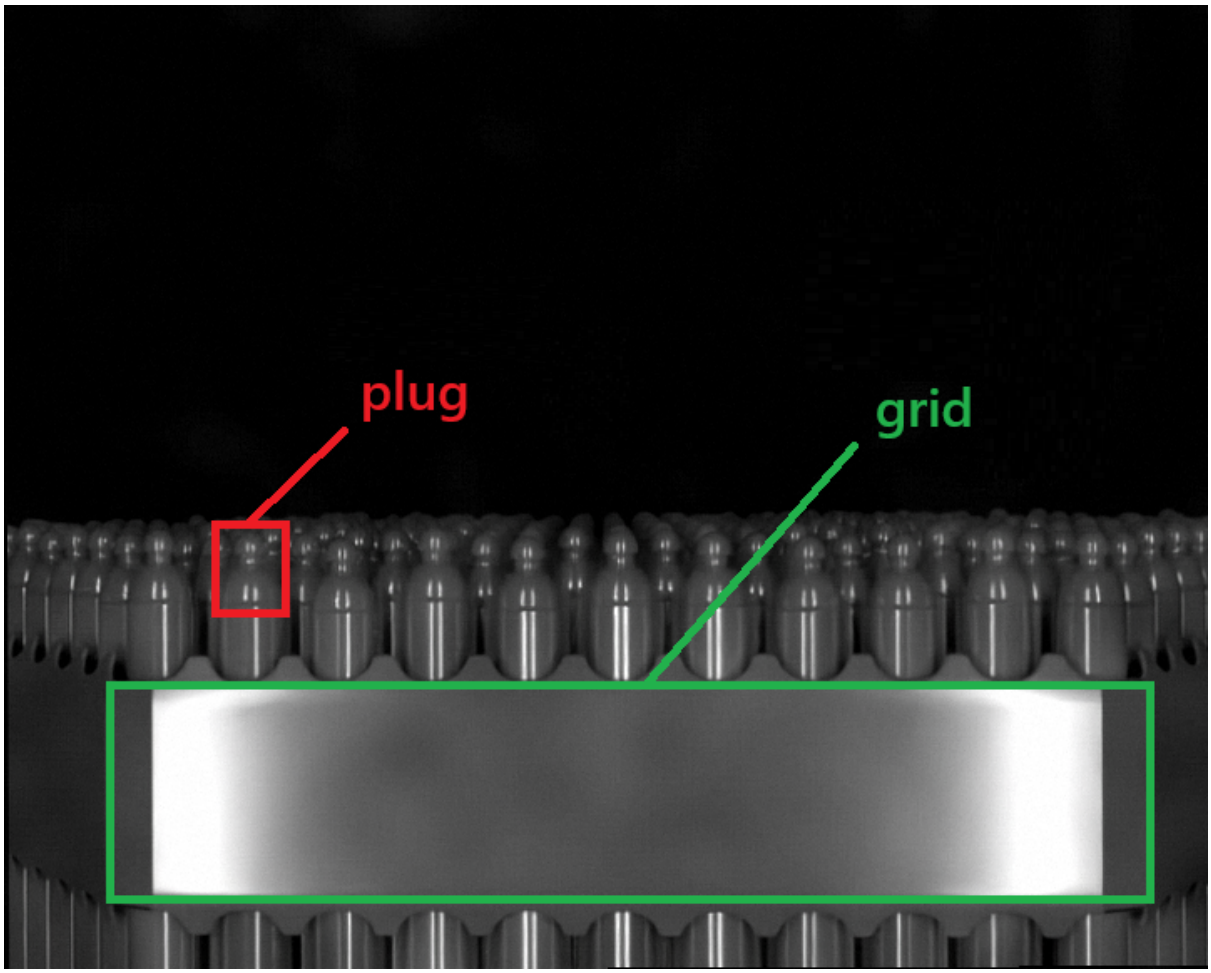

Figure 3: Example inspection frame with end plug heads and top edge of the spacer grid marked.

2. Related Work

The problem of object segmentation in industrial settings, particularly for metallic components, has been addressed by several computer vision and machine learning approaches. This section reviews the key developments in semantic segmentation, synthetic data generation, and applications in nuclear and manufacturing domains.

2.1. Semantic Segmentation in Industrial Vision

Since the introduction of UNet [3], semantic segmentation has seen significant advancements aimed at improving contextual understanding, feature representation, and efficiency. UNet's encoder-decoder structure with skip connections effectively combines spatial and semantic information but is limited in capturing global context. Subsequent models like PSPNet [4] and DeepLabV3 [5] addressed this by introducing pyramid pooling and atrous spatial pyramid pooling, respectively, to incorporate multi-scale contextual information. Deep residual networks replaced shallow encoders, enhancing feature extraction capabilities, as seen in DeepLabV3+ [6]. Attention mechanisms and, more recently, Transformer-based architectures such as TransUNet [7], SegFormer [8], and Mask2Former [9] have further improved segmentation by modeling long-range dependencies. Additionally, lightweight models like ENet [10], BiSeNet [11], and Fast-SCNN [12] have been developed for real-time applications, while refined decoders and self-supervised techniques [13] continue to enhance boundary precision and reduce annotation dependence. These developments collectively mark a shift toward more context-aware, accurate, and efficient segmentation architectures.

To address the demands of a new industry application, we consider architectures that combine U-Net-style decoders with modern encoder backbones such as EfficientNet [14] and ConvNeXt [15], offering a strong balance between performance, scalability, and modular design.

2.2. Synthetic Data and Domain Adaptation

Due to confidentiality constraints surrounding real-world datasets, we propose training convolutional neural networks (CNNs) using a synthetic dataset that can be openly shared with the research community. Synthetic data not only circumvent privacy issues but also address challenges related to precise annotations and the limited availability of labeled images. Our prior work [16] introduced a synthetic generator for fuel assemblies, which we build upon in this study.

Recent advances demonstrate that high-quality synthetic data generated via 3D modeling and rendering tools, such as Blender, can effectively support CNN training. Studies like [17, 18] show that synthetic-to-real transfer is feasible when rendered data incorporate diverse and realistic variations in lighting, textures, and object poses. Furthermore, domain randomization [19] and techniques like structured synthetic pipelines [20] and texture transfer [21] have proven valuable in bridging the domain gap between synthetic and real images. In our approach, we leverage both photorealistic rendering and domain variation strategies to simulate operational conditions and maximize model generalizability.

2.3. Deep Learning in Nuclear and Manufacturing Inspection

In the nuclear industry, most applications of machine learning have focused on reactor monitoring, anomaly detection in sensor networks [22], or predictive maintenance. Visual inspection of fuel rods remains underexplored due to strict confidentiality and domain-specific constraints.

However, deep learning has been successfully applied to weld inspection [23], defect detection in turbine blades [24], and PCB quality control [25]. These tasks share similar challenges, such as small defect size, reflective surfaces, and class imbalance.

To our knowledge, this paper is among the first to address deep segmentation of nuclear fuel rod end plugs in a production-grade setting. Our work contributes a detailed, reproducible pipeline and highlights the viability of CNN-based inspection under safety-critical constraints.

3. Data Preparation and Augmentation

Deep learning-based semantic segmentation models rely on diverse, well-annotated training data. In our study, we use a combination of real inspection images with manual annotations and synthetically generated images. Since the real dataset is small and labor-intensive to annotate consistently, we apply data augmentation to enhance model generalization. The same augmentation techniques are applied to the synthetic data to maintain consistency across the training set.

3.1. Real Dataset

The real dataset consists of 523 manually annotated grayscale images captured from a nuclear fuel rod assembly line. Each 720×576 image contains 11 vertically aligned peripheral rods. Expert annotators labeled two key regions: the end plug and the corresponding grid bar. To ensure consistency, all annotations were reviewed and refined based on inter-rater agreement. The limited size of the dataset highlights the need for artificial diversity through synthetic data or augmentation. Due to a non-disclosure agreement (NDA), these images cannot be publicly shown in this work.

3.2. Synthetic Dataset Generation

To complement real images, we developed a parametric 3D rendering pipeline using Blender, building upon the publicly available codebase introduced in [16]. The synthetic data generator is capable of

simulating diverse rod configurations, illumination conditions, and camera viewpoints. Key features include:

- Photorealistic rendering using physically-based shaders and ray-traced lighting.
- Automatic generation of ground truth masks for plug and grid regions.
- Variable camera roll, pitch, and distance to simulate mechanical misalignment.
- Simulation of various lighting intensities and realistic image noise artifacts to enhance domain variability.

A total of 523 synthetic images were rendered with random configurations. Post-rendering, all images were converted to 8-bit grayscale and resized to match real frame specifications.

3.3. On-the-Fly Augmentations

To further increase variability, we used the Albumentations library [26] to perform random augmentations, applied only to training dataset. The pipeline included:

- **Geometric:** Small affine transformations (rotation ±3°, scaling ±25%, horizontal flip).
- **Photometric:** Random brightness/contrast changes (±80%).
- **Noise:** Additive Gaussian noise, blur, and simulated compression artifacts.
- **Cutout:** Random erasure of random patch to simulate occlusions.

Each image passed through 10–15 transformations during training. All transformations were label-preserving, meaning masks were transformed in parallel with input images.

3.4. Data Splits and Normalization

Models were trained and evaluated separately on both real and synthetic datasets to assess their performance under controlled and practical conditions. We split the dataset into 60% training, 15% validation, and 25% test sets.

This data pipeline provided a rich training signal while simulating the diversity encountered in real manufacturing scenarios, ultimately contributing to the segmentation model's robustness and generalization.

4. Model Architectures and Training Protocols

In designing our semantic segmentation framework, we prioritize three key criteria: fast training and inference, support for rapid prototyping through modular design, and strong performance in terms of convergence and generalization. To meet these goals, we adopt an encoder-decoder architecture that enables clear separation between feature extraction and spatial reconstruction. This modular structure facilitates easy substitution of encoder backbones and decoder strategies, allowing for flexible experimentation and optimization. Additionally, we favor architectures that are computationally efficient yet capable of learning robust representations from both real and synthetic data, making them suitable for deployment in industrial inspection scenarios with limited training resources.

4.1. Network Architectures

We implemented and benchmarked four model variants:

1. U-Net with EfficientNetV2-S encoder
2. U-Net with ConvNeXt-Tiny encoder
3. U-Net 3+ with EfficientNetV2-S encoder
4. U-Net 3+ with ConvNeXt-Tiny encoder

All backbones were initialized with ImageNet-1k weights. The decoder was constructed with 2D convolutions, batch normalization, ReLU activations, and bilinear upsampling blocks. U-Net 3+ variants included dense skip connections and deep supervision at multiple decoder stages.

For each dataset combination, two separate models were trained: one for predicting the plug regions and another for the spacer grid. Each model outputs a probability map corresponding to either the `plug` or `grid` class, with a final softmax layer applied to normalize the pixel-wise logits.

4.2. Loss Functions

To train the segmentation models, we used a composite loss function that combines Binary Cross-Entropy (BCE) and Dice loss:

$$\mathscr{L}_{\text{total}}(p_i, y_i) = \mathscr{L}_{\text{BCE}}(p_i, y_i) + \mathscr{L}_{\text{Dice}}(p_i, y_i), \tag{1}$$

where y_i is the ground truth label for the i-th sample, p_i is the predicted probability for the positive class.

Binary Cross-Entropy ensures accurate pixel-wise classification, while Dice loss promotes shape consistency and robustness to class imbalance. Their combination helps the model capture fine details and produce coherent segmentation masks.

The individual components are defined as:

$$\mathscr{L}_{\text{BCE}}(p_i, y_i) = -\alpha(1 - p_i)^{\gamma} y_i \log(p_i) \tag{2}$$

$$\mathscr{L}_{\text{Dice}}(p_i, y_i) = (1 - \alpha) p_i^{\gamma} (1 - y_i) \log(1 - p_i) \tag{3}$$

We used hyperparameters $\alpha = 0.05$, $\gamma = 2.0$.

4.3. Training Settings

All models were trained using the Adam optimizer with its default hyperparameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1\text{e}{-7}$. Training proceeded for a maximum of 100 epochs, with early stopping based on validation loss employed to prevent overfitting. The Adam optimizer's adaptive learning rate adjustment contributed to stable and efficient convergence across all tested architectures and dataset variants, even in scenarios involving reflective artifacts, class imbalance, or synthetic-to-real domain shifts.

Batch size was set to 32 and training was performed on an NVIDIA RTX A5000 GPU with 24 GB of GDDR6 memory. Each model variant required approximately 1–3 hours to train.

4.4. Input Resolution

Although the original images have a resolution of 720×576 pixels, the input resolution was downscaled using bilinear interpolation to 128×128 pixels due to the computational demands of the chosen backbone architectures. We are aware that downscaling the images can lead to a loss of precision. Although this issue has been addressed, a satisfactory solution is under development.

5. Experiments and Results

We evaluated our segmentation models on the both datasets using standard accuracy metrics, as well as domain-specific evaluation criteria relevant to industrial quality control. This section presents our experimental design, benchmark results, and visual analysis.

5.1. Evaluation Metrics

We measured segmentation performance using the following metrics:

- **Intersection over Union (IoU)** for plug/grid masks
- **F1-score**-*X*-*Y* for plug/grid positions, computed with spatial tolerance thresholds X and Y (in pixels) relative to the original image resolution of 720×576.

5.2. Model Performance

Table 1 and Table 2 summarize the quantitative results of real dataset for all four model variants on the held-out test set (25% of data) for plug prediction. Table 3 and Table 4 summarize results of real dataset for grid spacer prediction.

Table 1
Summary of training results of real dataset for plug detection using EfficentNet-V2-S, averaged over five runs (mean ± std). Used metrics: F1-X-Y at original image size 720 x 576 px [2].

	Eff-UNet	**Eff-UNet3**
Train. time	125 ± 11 m	118 ± 23 m
Pred. time	0.021 s	0.027 s
Train. Epochs	100 ± 0	29 ± 6
Total Params	29.3 M	**24.8 M**
Train. Params	**9.0 M**	24.7 M
Test iou	79.03 ± 1.18	**83.19** ± 0.93
Test F1-8-5	95.30 ± 0.20	96.18 ± 0.91
Test F1-6-4	92.99 ± 0.79	**94.98** ± 0.72
Test F1-4-3	78.60 ± 1.72	83.58 ± 1.71
Test F1-3-2	66.20 ± 1.90	69.85 ± 1.76
Test F1-3-1	53.88 ± 1.91	52.95 ± 4.37

Table 2
Summary of training results of real dataset for plug detection using ConvNeXt-Tiny, averaged over five runs (mean ± std). Used metrics: F1-X-Y at original image size 720 x 576 px [2].

	Con-UNet	**Con-UNet3**
Train. time	**100** ± 52 m	186 ± 35 m
Pred. time	**0.012** s	0.013 s
Train. Epochs	39 ± 10	98 ± 4
Total Params	43.0 M	30.8 M
Train. Params	43.0 M	30.8 M
Test iou	79.94 ± 1.01	81.67 ± 1.37
Test F1-8-5	95.28 ± 0.56	**96.66** ± 0.94
Test F1-6-4	94.36 ± 0.80	94.65 ± 0.92
Test F1-4-3	**84.13** ± 2.16	81.09 ± 1.01
Test F1-3-2	**70.13** ± 1.45	66.27 ± 2.31
Test F1-3-1	**56.37** ± 4.48	51.81 ± 1.62

The highest sub-millimeter precision for plug detection was achieved using the ConvNeXt-Tiny encoder with a U-Net 3+ decoder, while the best performance in coarse detection was obtained with the ConvNeXt-Tiny and U-Net combination.

The highest sub-millimeter precision across all metrics for grid spacer detection was achieved using the EfficientNet-V2-S encoder with a U-Net 3+ decoder

Table 5 summarize the quantitative results of synthetic dataset for EfficientNet-V2-S + U-Net 3+ combination on the held-out test set (25% of data) for plug and grid spacer prediction.

5.3. Qualitative Results

Figure 4 shows representative examples from the test set. The models exhibit strong boundary delineation, even under challenging illumination and reflections.

Failure cases were rare and typically involved extremely low-contrast plugs or plugs partially visible.

Table 3
Summary of training results of real dataset for grid spacer detection using EfficentNet-V2-S, averaged over five runs (mean ± std). Used metrics: F1-X-Y at original image size 720 x 576 px [2].

	Eff-UNet	**Eff-UNet3**
Train. time	122 ± 17 m	159 ± 42 m
Pred. time	0.022 s	0.027 s
Train. Epochs	79 ± 17	57 ± 15
Total Params	37.5 M	**28.2 M**
Train. Params	37.3 M	28.0 M
Test iou	92.56 ± 1.05	**93.70** ± 0.83
Test F1-8-5	98.70 ± 0.40	**99.68** ± 0.40
Test F1-6-4	96.26 ± 1.51	**97.72** ± 0.95
Test F1-4-3	88.46 ± 1.74	**92.36** ± 2.10
Test F1-3-2	67.48 ± 4.82	**74.15** ± 5.15
Test F1-3-1	30.41 ± 4.98	**36.91** ± 4.56

Table 4
Summary of training results of real datasets for grid spacer detection using ConvNeXt-Tiny, averaged over five runs (mean ± std). Used metrics: F1-X-Y at original image size 720 x 576 px [2].

	Con-UNet	**Con-UNet3**
Train. time	**25** ± 13 m	176 ± 12 m
Pred. time	0.014 s	**0.011** s
Train. Epochs	12 ± 7	100 ± 0
Total Params	48.0 M	30.8 M
Train. Params	20.1 M	**3.0 M**
Test iou	87.31 ± 0.88	90.55 ± 1.07
Test F1-8-5	98.37 ± 0.51	97.07 ± 3.90
Test F1-6-4	96.59 ± 0.61	93.82 ± 6.10
Test F1-4-3	87.97 ± 2.38	86.67 ± 6.35
Test F1-3-2	68.46 ± 5.45	66.67 ± 9.34
Test F1-3-1	32.52 ± 3.37	32.85 ± 8.33

5.4. Comparison with Legacy Systems and Deployment

Prior to the adoption of our CNN-based segmentation pipeline, a rule-based image processing system was used for inspection. This legacy approach relied on edge detection and periodicity of the FA rods in the image to estimate rod's welds positions from pixel intensities.

Table 6 compares plug detection performance between the legacy system and our best model (EfficientNet-V2-S + U-Net 3+), showing significant improvements in F1 scores across test cases.

Figure 5 illustrate improvements in rod height estimation, where the proposed method achieves a mean error of $\mu = 0.424$ mm compared to 7.094 mm with the legacy system. Standard deviation is also markedly reduced (0.393 mm vs. 10.567 mm), confirming better robustness.

While the CNN-based system introduces slightly higher latency, GPU acceleration enables batch processing without affecting throughput. Known limitations include minor segmentation degradation near image borders and sensitivity to extreme camera angles. These are mitigated through data augmentation, synthetic training examples, and postprocessing heuristics. Future work may explore transformer-based architectures for enhanced global context modeling.

Table 5
Summary of training results of synthetic datasets using EfficientNet-V2-S + U-Net 3+ combination, averaged over five runs (mean ± std). Used metrics: F1-X-Y at original image size 720 x 576 px [2].

	Plug prediction	**Gridline prediction**
Train. time	1h 13m ± 20m	1h 10m ± 40m
Pred. time	0.039 s	0.037 s
Train. Epochs	14 ± 1	4 ± 1
Test iou	90.28 ± 0.28	94.14 ± 0.10
Test F1-8-5	99.99 ± 0.03	99.73 ± 0.38
Test F1-6-4	99.99 ± 0.03	98.78 ± 0.41
Test F1-4-3	99.62 ± 0.24	95.66 ± 1.12
Test F1-3-2	93.51 ± 3.01	82.52 ± 5.07
Test F1-3-1	64.33 ± 8.68	45.12 ± 5.82

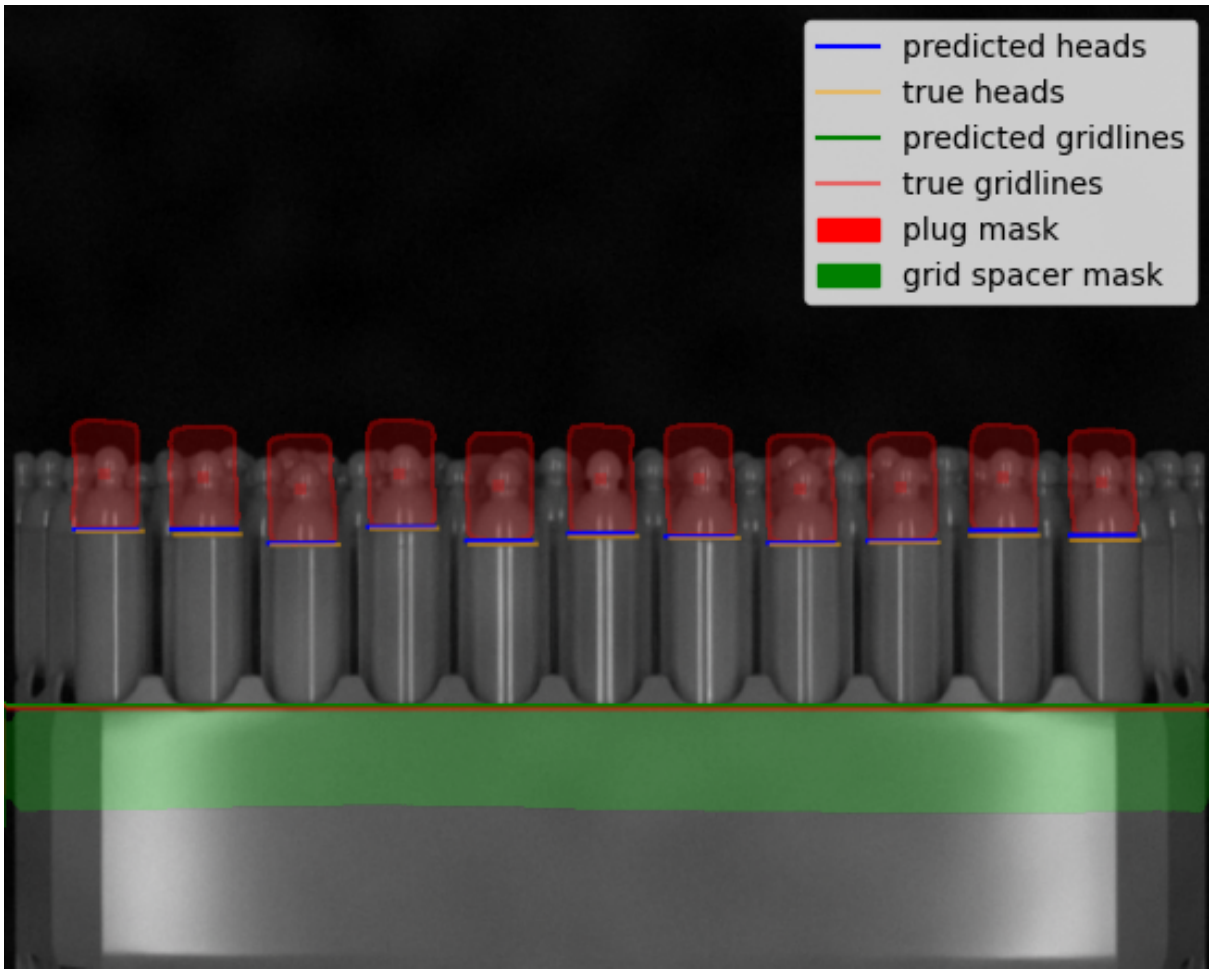

Figure 4: Example test image with predicted plug head positions and grid spacer location. The visualization also includes the corresponding segmentation masks for both plugs and the grid spacer.

Conclusion and Future Work

We have presented a complete, data-driven segmentation pipeline for automated control of nuclear fuel rod end plugs. By leveraging modern deep learning architectures, synthetic data augmentation, and domain-specific evaluation, our system delivers robust, sub-millimeter plug and grid segmentation under diverse industrial conditions.

Among four tested model variants, the U-Net 3+ decoder with a EfficientNet-V2-S encoder offered the

Table 6
Comparison with legacy inspection system

	Baseline	Our approach
Test F1-8-5	61.94	**96.18**
Test F1-6-4	60.03	**94.94**
Test F1-4-3	57.29	**83.58**
Test F1-3-2	50.80	**69.85**
Test F1-3-1	47.83	**52.92**

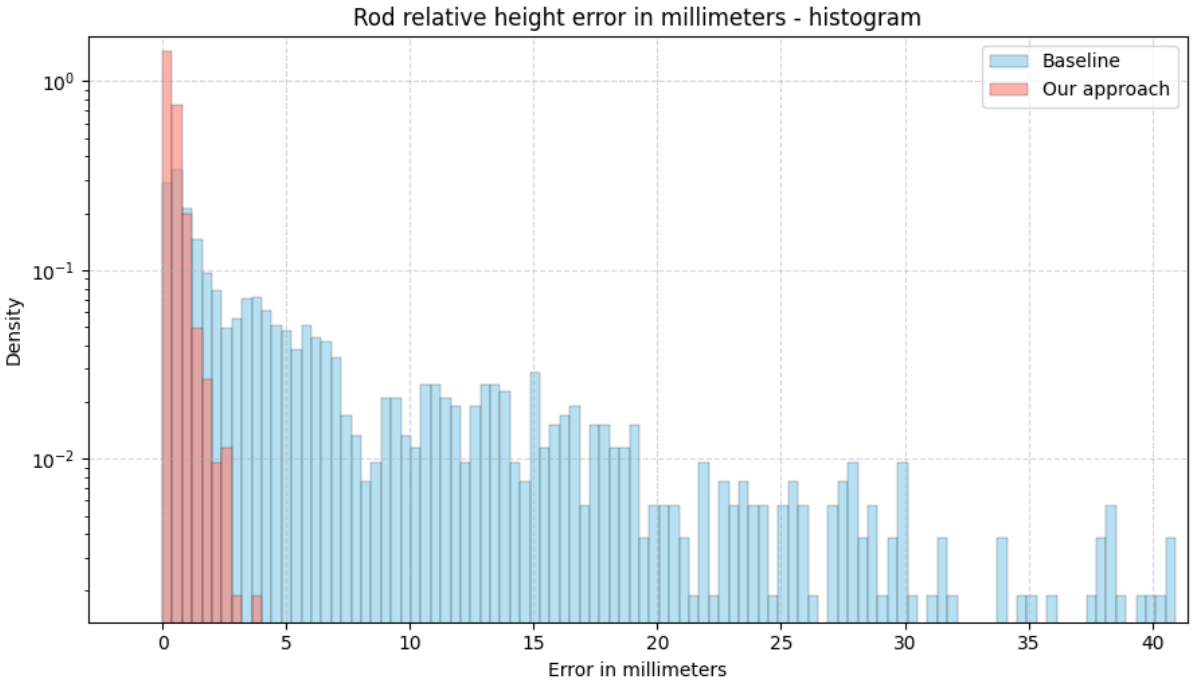

Figure 5: Error distribution for rod height prediction using baseline vs. deep learning approach [2].

best trade-off between accuracy and speed. Our pipeline enabled the system to generalize across plug types, lighting setups, and slight misalignments — a limitation of the previous rule-based framework. The best model achieved an average positional error of just 0.424 mm, significantly outperforming the legacy rule-based system (7.094 mm). On the test set, it also improved the F1-score from 61.94 to 96.18 for loose spatial tolerance (F1-8-5), and from 50.80 to 69.85 for stricter tolerance (F1-3-2). These results confirm that the proposed method meets the sub-millimeter precision required in nuclear inspection workflows.

The CNN-based inspection module was successfully tested and has demonstrated stable performance and real-time processing capabilities. It is well-positioned to serve as a practical enhancement to the current inspection workflow, significantly increasing the level of automation in video analysis. Furthermore, the potential for continuous model monitoring and periodic retraining offers a promising path toward further improving accuracy and adaptability, ultimately supporting a more robust and fully automated inspection process in the future.

Future Work

Our future research and development efforts will focus on the following directions:

- **Exploration of Alternative Segmentation Architectures:** Evaluate architectures tailored for high-precision segmentation on small datasets, such as HRNet [27] and DeepLabv3+ [5]. These

models offer strong performance through high-resolution feature preservation and multi-scale context aggregation.
- **Custom Backbone Design:** Develop a flexible custom backbone with tunable width (filter multiplier) and depth (number of layers) parameters. This would allow optimization for larger input resolutions and better balance between computational efficiency and segmentation accuracy.
- **Enhanced Plug Annotation Protocol:** Augment the current center-point plug annotations to include plug width or bounding boxes, enabling more precise mask generation. Future work may also explore full contour annotations if labeling cost is justified.
- **Cross-Dataset Evaluation:** Test the trained models on real-world datasets with different plug types and assembly geometries to assess generalization performance and identify adaptation needs.

These research directions aim to systematically improve both the precision and adaptability of the segmentation system. By enhancing annotation quality, exploring advanced architectures, and evaluating cross-domain generalization, we intend to develop a more robust, fully automatic, reproducible and precise solution that can meet the evolving demands of industrial nuclear fuel inspection.

Final Remarks

Our work demonstrates that modern convolutional neural networks — when combined with photorealistic synthetic data and targeted augmentations — are not only effective for academic benchmarks but also mature enough for industrial deployment. The methodology and insights presented here may serve as a blueprint for future visual inspection systems across high-stakes manufacturing domains.

Acknowledgments

We thank to Jaroslav Knotek for generator of the synthetic fuel dataset and Marcin Kopeć for domain expertise. This work was supported by Centre of Recherche Řež, ČEZ a.s. and Charles University. We also acknowledge the state support of the Technology Agency of the Czech Republic within the National Competence Centre Programme, project TN02000012 „Center of Advanced Nuclear Technology II“ which is partially co-financed within the National Recovery Plan from the European Instrument for recovery and resilience.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4 and in order to: Grammar and spelling check. After using these tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] U.S. Energy Information Administration, The nuclear fuel cycle, https://www.eia.gov/energyexplained/nuclear/the-nuclear-fuel-cycle.php, 2025. Accessed: 2025-06-24.

[2] V. Bláha, Segmentation of Fuel Rod End Plugs, Master's thesis, Charles University, Department of Software and Computer Science Education, Prague, Czech Republic, 2025.

[3] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, 2015. URL: https://arxiv.org/abs/1505.04597. arXiv:1505.04597, [Accessed: 2025-04-28].

[4] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, 2017. URL: https://arxiv.org/abs/1612.01105. arXiv:1612.01105.

[5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018. URL: https://arxiv.org/abs/1802.02611. arXiv:1802.02611, [Accessed: 2025-04-28].

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 801–818. doi:10.48550/arXiv.1802.02611.

[7] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, Y. Zhou, Transunet: Transformers make strong encoders for medical image segmentation, arXiv preprint arXiv:2102.04306 (2021). doi:10.48550/arXiv.2102.04306.

[8] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, P. Luo, Segformer: Simple and efficient design for semantic segmentation with transformers, arXiv preprint arXiv:2105.15203 (2021). doi:10.48550/arXiv.2105.15203.

[9] B. Cheng, A. Schwing, A. Kirillov, Masked-attention mask transformer for universal image segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1290–1299. doi:10.48550/arXiv.2112.01527.

[10] A. Paszke, A. Chaurasia, S. Kim, E. Culurciello, Enet: A deep neural network architecture for real-time semantic segmentation, arXiv preprint arXiv:1606.02147 (2016). doi:10.48550/arXiv.1606.02147.

[11] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, N. Sang, Bisenet: Bilateral segmentation network for real-time semantic segmentation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 334–349. doi:10.48550/arXiv.1808.00897.

[12] R. P. K. Poudel, S. Liwicki, R. Cipolla, Fast-scnn: Fast semantic segmentation network, arXiv preprint arXiv:1902.04502 (2019). doi:10.48550/arXiv.1902.04502.

[13] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738. doi:10.48550/arXiv.1911.05722.

[14] M. Tan, Q. V. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, CoRR abs/1905.11946 (2019). URL: http://arxiv.org/abs/1905.11946. arXiv:1905.11946, [Accessed: 2025-04-28].

[15] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s, 2022. URL: https://arxiv.org/abs/2201.03545. arXiv:2201.03545, [Accessed: 2025-04-28].

[16] J. Knotek, J. Blažek, M. Kopeć, Simulating nuclear fuel inspections: Enhancing reliability through synthetic data, Nuclear Engineering and Technology 57 (2025) 103571. URL: https://www.sciencedirect.com/science/article/pii/S1738573325001391. doi:https://doi.org/10.1016/j.net.2025.103571, [Accessed: 2025-04-28].

[17] S. R. Richter, V. Vineet, S. Roth, V. Koltun, Playing for data: Ground truth from computer games, in: European Conference on Computer Vision (ECCV), Springer, 2016, pp. 102–118. doi:10.1007/978-3-319-46475-6_7.

[18] J. Tremblay, T. To, S. Birchfield, Falling things: A synthetic dataset for 3d object detection and pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018, pp. 2038–2041. doi:10.1109/CVPRW.2018.00257.

[19] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, Domain randomization for transferring deep neural networks from simulation to the real world, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 23–30. doi:10.1109/IROS.2017.8202133.

[20] A. Kar, C. Häne, J. Malik, Meta-sim: Learning to generate synthetic datasets, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 4551–4560. doi:10.1109/ICCV.2019.00465.

[21] S. Zheng, T. Xiao, Z. Li, H. Yang, Y. Song, P. Luo, Unbiased synthetic data generation via texture disentanglement for semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 15765–15775. doi:10.1109/CVPR52688.2022.01536.

[22] Y. Shi, X. Xue, Y. Qu, J. Xue, L. Zhang, Machine learning and deep learning methods used in safety management of nuclear power plants: A survey, in: 2021 International Conference on Data Mining Workshops (ICDMW), 2021, pp. 917–924. doi:10.1109/ICDMW53433.2021.00120.

[23] Y. Chang, W. Wang, A deep learning-based weld defect classification method using radiographic

images with a cylindrical projection, IEEE Transactions on Instrumentation and Measurement 70 (2021) 1–11. doi:10.1109/TIM.2021.3124053.

[24] J. Liu, J. Liu, D. Yu, M. Kang, W. Yan, Z. Wang, M. G. Pecht, Fault detection for gas turbine hot components based on a convolutional neural network, Energies 11 (2018). URL: https://www.mdpi.com/1996-1073/11/8/2149. doi:10.3390/en11082149.

[25] L. Zhou, X. Ling, S. Zhu, Z. Sun, J. Yang, An self-supervised learning self-attention based method for defects classification on pcb surface images, in: 2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT), 2021, pp. 229–234. doi:10.1109/CECIT53797.2021.00047.

[26] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A. A. Kalinin, Albumentations: Fast and flexible image augmentations, Information 11 (2020) 125. URL: http://dx.doi.org/10.3390/info11020125. doi:10.3390/info11020125.

[27] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, B. Xiao, Deep high-resolution representation learning for visual recognition, 2020. URL: https://arxiv.org/abs/1908.07919. arXiv:1908.07919, [Accessed: 2025-04-28].