

# U-Net for wheel rim contour detection in robotic deburring

Hicham Ait El Attar, Hassan Samri, Moulay El Houssine Ech-Chhibat, Khalifa Mansouri,  
Abderrahim Bahani, Tarek Bahrar

Department of Mechanical Engineering, ENSET, University Hassan II Casablanca, Mohammedia, Morocco

## Article Info

### Article history:

Received Mar 6, 2024

Revised Oct 18, 2024

Accepted Nov 14, 2024

### Keywords:

Deep learning

OpenCV

Segmentation for robotic  
deburring

U-Net

Wheel rim detection

## ABSTRACT

Automating robotic deburring in the automotive sector demands extreme precision in contour detection, particularly for complex components like wheel rims. This article presents the application of the U-Net architecture, a deep learning technique, for the precise segmentation of the outer contour of wheel rims. By integrating U-Net's capabilities with OpenCV, we have developed a robust system for wheel rim contour detection. This system is particularly well-suited for robotic deburring environments. Through training on a diverse dataset, the model demonstrates exceptional ability to identify wheel rim contours under various lighting and background conditions, ensuring sharp and accurate segmentation, crucial for automotive manufacturing processes. Our experiments indicate that our method surpasses conventional techniques in terms of precision and efficiency, representing a significant contribution to the incorporation of deep learning in industrial automation. Specifically, our method reduces segmentation errors and improves the efficiency of the deburring process, which is essential for maintaining quality and productivity in modern production lines.

This is an open access article under the [CC BY-SA](#) license.



## Corresponding Author:

Hicham Ait El Attar

Department of Mechanical Engineering, ENSET, University Hassan II Casablanca

Mohammédia 28830, Morocco

Email: aitelattar.hicham@gmail.com

## 1. INTRODUCTION

The emergence of deep learning, combined with advances in robotics and automation, has revolutionized image processing, particularly in the automotive industry where precision is essential. Robotic deburring of wheel rims, a process that demands extreme accuracy, greatly benefits from these innovations. For instance, fast-U-Net has demonstrated its efficiency in orchard navigation [1], highlighting the progress made through deep learning. Our study follows this trend by aiming for optimal detection of wheel rim contours to enhance deburring.

Precise segmentation is crucial in various fields such as pathological imaging and industrial inspection. The work on Lite-UNet illustrates the importance of contour accuracy under complex lighting and background conditions [2], [3]. Other studies have also highlighted the importance of these conditions [4], [5], emphasizing the challenges to overcome for reliable segmentation. We build on this research to accurately define the external contours of wheel rims.

Speed and precision are essential qualities in robotic systems, as noted by various studies [6], [7]. We seek to maximize these qualities in our approach. Studies on the evaluation of microfractures and manufacturing defects guide us in developing solutions that meet the strict requirements of the automotive industry [8], [9], providing a solid foundation for our research.

Convolutional neural networks (CNNs) are widely used for image segmentation, as demonstrated by several works [10]. Research on plant disease recognition and tomato crop segmentation shows the effectiveness of CNNs [11], [12]. These diverse applications inform our use of CNNs for wheel rim contour detection, bringing proven techniques to a new application domain.

Managing trade-offs in robotic deburring systems and the impact of automation on machining forces offer valuable insights [13], [14]. Studies on deburring optimization reinforce our approach to integrating machining processes into our system [15], [16]. This research highlights the technical challenges and potential solutions, providing a framework for our experiments.

Advances in contour detection and classification, necessary for vehicle re-identification, are explored by various studies [17]. Work with U-Net++ and U-Net brings significant progress in complex segmentation, the foundation of our research [18], [19]. These studies demonstrate the enhanced capabilities of these deep learning architectures in complex scenarios, justifying our technological choice.

The integration of deep learning in robotics and image super-resolution frames our approach [20]–[22]. We refine this approach with edge detection techniques and precise sub-pixel edge localization [23], [24], ensuring increased accuracy in segmentation. This precision is crucial to meet the quality and efficiency requirements of the automotive industry.

Finally, research on industrial process automation and innovations in disease recognition and defect detection inform our method of automatic wheel rim contour detection [25]–[29]. We also rely on burr formation models and surface roughness predictions [30]–[32]. This knowledge enriches our understanding of industrial challenges and allows us to propose innovative solutions.

This article demonstrates how deep learning can be applied to address specific challenges in the automotive industry and be integrated into production systems to improve quality and efficiency. Through a series of experiments and validations, we establish new standards for contour detection in robotic deburring, paving the way for more advanced and precise industrial applications. These advancements not only enhance the accuracy of current systems but also offer scalable solutions for future automation in manufacturing.

## **2. MATERIALS**

### **2.1. Software environment**

The development and training of our U-Net segmentation model were conducted on Google Colab, which provides access to high-performance graphics processing units (GPUs). This platform was chosen for its ability to significantly accelerate training times and offer a flexible environment for development. The model was implemented using TensorFlow and Keras, leading libraries for constructing and optimizing deep neural networks. For the extraction and visualization of wheel rim contours, we used OpenCV version 4.8.1, chosen for its powerful image processing capabilities. Development was carried out in PyCharm version 2020.1.1, providing a robust integrated development environment. Execution was performed on a computer equipped with an Intel Core i7-6820HQ CPU at 2.70 GHz, 16 GB of RAM, and an NVIDIA Quadro M1000M graphics card, ensuring smooth handling of intensive computational operations.

### **2.2. Training configuration**

For the optimization of our modified U-Net model, we opted for training over 50 epochs. This choice was guided by the observation that beyond this threshold, the validation error began to increase, indicating the onset of potential overfitting. To stabilize training and prevent overfitting, we monitored loss curves to dynamically adjust hyperparameters. Starting with a learning rate of 0.001, we integrated a decaying learning rate scheduler to gradually fine-tune the weight updates of the network, thereby maximizing learning efficiency over the epochs. The batch size was set to 16, a balance that allows optimal utilization of computational resources while maintaining training stability and accuracy. This configuration helped reduce training times while improving model convergence.

To enhance model generalization, extensive data augmentation techniques were employed, including rotation, zoom, horizontal flipping, and lighting adjustments. These methods aimed to increase the diversity of the training set, exposing the model to a broader range of variations in wheel rim images. The inclusion of lighting adjustments was particularly crucial, as it enabled the model to maintain robust performance under various lighting conditions, thereby improving its ability to accurately identify wheel rim contours in diverse real-world lighting environments.

## **3. DATA COLLECTION AND DATASET PREPARATION**

### **3.1. Data collection**

Data collection is a critical step in the machine learning process. For this study, we acquired a comprehensive dataset of 220 wheel rim images from the public source Kaggle. The dataset was further

analyzed to ensure a balanced distribution of different rim shapes, enhancing the robustness of the model's training process. The images, as shown in Figure 1, were selected to reflect a variety of rim shapes, with 11 unique shapes and 20 representations for each shape, totaling a diverse sample for model training.



Figure 1. Example of the database

### 3.2. Data annotation

Image annotation was carried out using the (visual geometry group (VGG) image annotator (VIA)), a web-based tool designed for precise manual image annotation. Each image was annotated to delineate the external contour of the rims, resulting in the creation of 11 JavaScript object notation (JSON) files corresponding to the distinct rim shapes. Precise annotations are essential for training the model to recognize rim contours with high accuracy.

### 3.3. Data processing

The JSON files generated during annotation were used to create binary masks using a custom Python script. These masks serve to isolate the rims from the background, allowing the creation of image pairs: an original image and its corresponding mask. This process facilitates semantic segmentation and contour detection by the U-Net model, thereby improving detection accuracy.

### 3.4. Dataset preparation

To ensure a methodical distribution of data, the images were divided into three distinct sets: 70% for training, 20% for validation, and 10% for testing. This strategic division is crucial to maintain the integrity of the model evaluation, thereby ensuring that the model can effectively generalize on previously unseen data. Additionally, care was taken to randomize the distribution to avoid any potential bias in the training process.

### 3.5. Data augmentation

Data augmentation was implemented to improve the model's ability to generalize from a limited number of samples. Augmentation techniques include horizontal flipping, random adjustments of brightness, and contrast. These methods were systematically applied to produce image variants that simulate the natural variations encountered in a real production environment.

### 3.6. Data storage and organization

A structured directory was created to store the images and masks, with separate subfolders for each dataset (training, validation, and test). This organization allows for efficient access and retrieval during the model training phase, as shown in Figure 2. Additionally, this structure ensures that no overlap occurs between the datasets, maintaining the integrity of the training, validation, and testing processes.

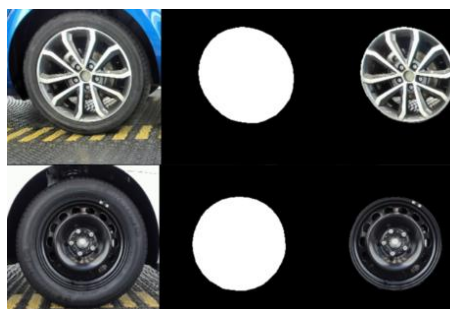


Figure 2. Example of an original image and associated binary mask

3.7. Summary of data preparation

The Table 1 details the various steps of data preparation, from the initial collection of images to their distribution into distinct sets for training, validation, and testing. This step-by-step breakdown ensures transparency in the data preparation process. It also highlights the careful consideration given to the organization and structure of the dataset to optimize model performance.

Table 1. Data preparation

Process	Description	Number of images
Preparation	Collected and resized images	220
Annotation	Used tools to annotate wheel rim contours	220
Augmentation	Applied augmentations such as horizontal flips, brightness, and contrast adjustments	880 (4 variants per image)
Distribution (%)	Split into training, validation, and test sets	Train: 70, Val: 20, Test: 10

The data preparation played a crucial role in the model's performance. Each image was meticulously annotated to precisely delineate the contours of the rims. Data augmentation techniques were applied to enrich the diversity of the data, which is essential for the model's robustness. The balanced distribution between the training, validation, and test sets ensures that the model is well-generalized and performs effectively on unseen data.

4. OPTIMIZED WHEEL RIM CONTOUR DETECTION ALGORITHM BASED ON U-NET

The U-Net model structure is designed for precise semantic segmentation. The first convolutional layer, forming the initial block, applies filters to detect low-level features. With 3×3 kernel sizes and "same" padding, it preserves the spatial dimensions while extracting features. Batch normalization stabilizes learning by normalizing the activations of each layer, and the rectified linear unit (ReLU) activation function is used to introduce non-linearity, which enhances the model's ability to learn complex patterns.

Following this are the encoder blocks, where each block first applies a convolution to extract more complex features, followed by a max pooling operation to reduce dimensionality. This reduction is crucial for capturing contextual information at larger scales. The encoder also progressively decreases the spatial resolution while increasing the depth of the feature maps, allowing the model to learn abstract and high-level representations.

In the decoder blocks, transposed convolutions are used to increase dimensionality, and concatenation with features from the encoder blocks helps recover spatial information that was lost. This step is essential for reconstructing the image with the fine details necessary for precise segmentation. The decoder, thus, plays a critical role in achieving accurate contour detection, especially in complex scenarios like wheel rim segmentation.

The hyperparameters were finely tuned to balance rapid learning and avoid overfitting. We monitored the training process with callbacks such as early stopping and learning rate reduction to ensure the model's generalization on unseen data. This approach helped prevent the model from over-specializing on the training data, maintaining its robustness.

Finally, illustrations and mathematical formulas are used to detail each step of the U-Net architecture, providing a deep understanding of its structure and functionality. These visual aids are crucial in explaining the inner workings of the model and its application to the precise detection of wheel rim contours. Furthermore, the diagrams help highlight the specific improvements made to the traditional U-Net model, ensuring clarity in the presentation of our optimization process.

4.1. Model structure

The U-Net model is structured into two main parts: the encoder for capturing features and the decoder for reconstructing the segmented image. The key operations of the model are mathematically defined as follows. Each part plays a crucial role in ensuring accurate segmentation by leveraging convolutional and deconvolutional layers to process the input data and reconstruct the output image.

4.1.1. Convolution

Here,  $W$  and  $b$  represent the weights and biases of the convolutional layer, respectively,  $x$  is the input, and  $*$  represents the convolution operation. Batch normalization (BN) stabilizes the learning process by normalizing the activations, and the ReLU activation function is used to introduce the necessary

non-linearity into the model. This combination of operations ensures that the model can efficiently learn complex patterns while maintaining stability during training.

$$y = ReLU(BN(W * x + b)) \quad (1)$$

#### 4.1.2. Dimensionality reduction

MaxPooling layers reduce the dimensionality of the features, allowing for information compression and reducing memory and computational power requirements. This operation is crucial in helping the model focus on the most important features while discarding irrelevant details. By reducing spatial dimensions, MaxPooling also speeds up the learning process, making the model more efficient.

$$y = MaxPool(x) \quad (2)$$

#### 4.1.3. Increasing spatial resolution

Transposed convolutions are used in the decoder to increase the spatial resolution of the features, preparing them for concatenation with the encoder features. This operation helps to restore the finer details lost during the downsampling process in the encoder. By gradually recovering the spatial dimensions, the model can accurately reconstruct the segmented image, ensuring precise contour detection.

$$y = ConvTranspose(x) \quad (3)$$

#### 4.1.4. Feature fusion

This concatenation operation merges the features extracted by the encoder  $x_{enc}$  with those upsampled by the decoder  $x_{up}$ . This allows for the recovery of spatial information lost during the pooling step. By combining both high-level and low-level features, the model achieves more accurate segmentation, especially in complex scenarios.

$$x_{concat} = Concat(x_{up}, x_{enc}) \quad (4)$$

#### 4.1.5. Model output

Finally, a convolutional layer is applied to  $x_{concat}$  to obtain the final segmentation mask. In this binary segmentation task, the activation function used at this stage is the sigmoid, which models the probability that a pixel belongs to the class of interest (e.g. the wheel rim). The sigmoid function  $\sigma$  is chosen for this final layer because it constrains the output to be between 0 and 1, which can be interpreted as the probability of belonging to the target class.

$$x_{final} = \sigma(W * x_{concat} + b) \quad (5)$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (6)$$

### 4.2. Training strategy

The training of the U-Net model was meticulously planned to achieve high-precision segmentation of wheel rim contours. We employed a series of techniques to optimize the training process and minimize the risk of overfitting. These techniques included early stopping, learning rate scheduling, and data augmentation, ensuring the model remained robust while generalizing well to new data.

**Optimizer:** the Adam optimizer was selected for its recognized efficiency, starting with an initial learning rate of 0.01. This rate is high enough to ensure rapid convergence while allowing precise adjustments during the later phases of training. Additionally, the adaptive nature of Adam helps balance the learning speed for each parameter, improving the overall stability of the training process.

**Loss function:** binary cross-entropy loss is appropriate for our binary segmentation task, where the model predicts whether a pixel belongs to the wheel rim contour or not. The loss function is mathematically defined as follows:

$$Loss = \frac{-1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (7)$$

Where  $N$  represents the total number of examples in the batch,  $y_i$  is the true label, and  $p_i$  is the predicted probability by the model. This loss function penalizes incorrect predictions, ensuring the model learns to differentiate between rim contours and the background effectively.

Safeguard mechanisms: to prevent overfitting, we integrated early stopping, halting training if the validation performance does not improve over a certain number of consecutive epochs. Additionally, the best model is automatically saved during training, and a learning rate reduction is implemented if no improvement in validation performance is observed over a predefined interval. This approach is illustrated by Figure 3, which shows an example of an image before and after segmentation.



Figure 3. Example of an image before and after segmentation

#### 4.3. U-Net architecture configuration

The configuration of the U-Net architecture, illustrated in Figure 4, details a systematic strategy for semantic segmentation, aiming to capture both the local and global contexts of the image. This structure ensures that fine details are preserved while maintaining a broader understanding of the scene. The balance between downsampling and upsampling in the model allows for accurate and efficient segmentation of complex shapes, such as wheel rim contours.

Each convolutional layer, followed by a ReLU activation function, preserves essential non-linear features, while max pooling layers reduce dimensionality, thereby increasing feature abstraction. The symmetrical architecture of U-Net, with its contraction and expansion paths, is crucial for precise localization in the segmented image, allowing detailed recovery of the wheel rim contours. This structure is particularly effective for images where the distinction between the object and the background is subtle, which is often the case in industrial applications like robotic deburring. The hyperparameters used are summarized in the Table 2.

These parameters were crucial for the model's performance. A controlled number of epochs prevented overfitting, while an optimized batch size balanced computational efficiency and model convergence. The initial learning rate was carefully set at 0.001 to ensure a rapid and stable gradient descent, with dynamic adjustment down to 0.000001 during training to fine-tune the weight updates as the validation error evolved. The image resolution was sufficient to capture the necessary details without imposing an excessive computational or memory load.

In addition to these essential hyperparameters, a series of callbacks was meticulously configured to enhance the robustness of training and refine model performance. These callbacks included saving the best model, adaptive learning rate adjustment based on the validation set error evolution, and early stopping to prevent overfitting. Table 3 illustrates the details and rationale behind the selection of these specific parameters, which played a crucial role in achieving an efficient and accurate model.

To ensure optimal convergence and prevent overfitting, callbacks were carefully selected during the model training. The ModelCheckpoint callback was configured with `save_best_only=True` to retain only the model state displaying the best performance on the validation dataset. This choice is driven by the desire to maximize storage efficiency and reduce computational complexity, avoiding the saving of suboptimal models at each epoch.

Learning rate reduction is managed by ReduceLROnPlateau, with a factor of 0.1, allowing for exponential decay, which is known to progressively refine the network weights once the improvement in validation error becomes less noticeable. The patience of 4 epochs strikes a balance between responsiveness to performance plateaus and preventing overreaction to normal variations during training. This mechanism ensures that the model does not stagnate at suboptimal performance levels.

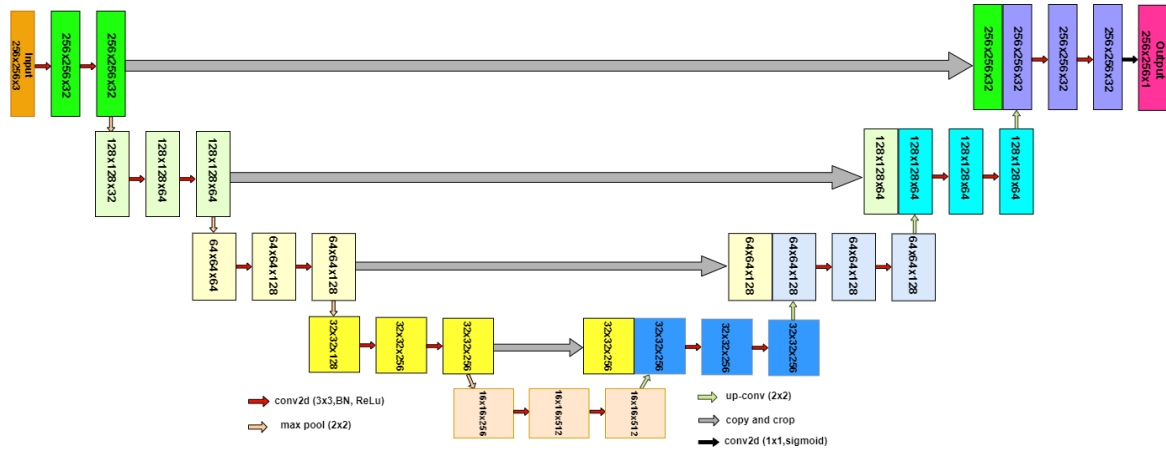


Figure 4. Configuration of the U-Net architecture for the segmentation of wheel rim contours

Table 2. Model U-Net hyperparameter configuration

Hyperparameter	Values
Epochs	50
Batch size	16
Learning rate	0.001
Image size	256×256

Table 3. Configuration of callbacks for the U-Net model training

Callback	Parametre	Value	Description
ModelCheckpoint	Verbose	1	Enables detailed messages during training
	Save-best-only	True	Saves only the model with the best performance on the validation set
ReduceLROnplateau	Factor	0.1	Reduction factor for the learning rate
	Ptience	4	Number of epochs without improvement before reducing the learning rate
Early Stopping	Ptience	15	Number of epochs without improvement before stopping
	Restore_best_weights	False	Requires restoring the best weights after stopping

Early stopping is implemented via EarlyStopping, with a patience of 15 epochs to give the model ample opportunity to overcome any temporary validation error plateau. The decision not to restore the best weights after stopping, `restore_best_weights=False`, is based on experimental results indicating that continuing training can sometimes lead to improved generalization by avoiding premature focus on a specific local minimum in the loss function space. This approach encourages a more thorough exploration of the loss function space.

The selected callback parameters reflect an optimization approach based on rigorous experimentation and adjustment. These decisions align with proven recommendations from deep learning literature. The relevance of these choices was confirmed by robustness tests, which demonstrated the model's ability to maintain high performance while generalizing well to unseen data. This experimental rigor ensures that the U-Net model training is both efficient and that the results are reliable.

## 5. RESULTS

### 5.1. Analysis of learning curves

The loss curve illustrates an initial rapid descent, followed by a promising stabilization. Mathematically, this can be interpreted as an effective minimization of the loss function  $L$ , which, in the case of binary cross-entropy, where  $y$  represents the true labels,  $\hat{y}$  the model predictions, and  $N$  the total number of pixels in the batch of images. This stabilization indicates that the model is learning effectively without overfitting, balancing accuracy and generalization, as formulated:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (8)$$



The learning curves for loss and accuracy provide significant insights into the behavior of the U-Net model during the initial and advanced phases of training. In the loss curve, a rapid drop in training loss is observed within the first few epochs, indicating that the model starts learning and adapting to the segmentation task from the early stages of training. However, the validation loss curve shows notable volatility during these initial stages, with a temporary increase followed by a decrease. This phenomenon can be explained by the model's adjustment process when encountering complex patterns in the validation data that it had not yet seen in the training data.

Figure 5 shows the loss curve of the U-Net model over 50 epochs, illustrating the rapid decrease in loss during training and the subsequent stabilization during validation. This trend highlights the effectiveness of the model in minimizing the error on the training set while gradually improving its performance on the validation set. The eventual stabilization of the validation loss curve indicates that the model is not overfitting and has reached a balance between learning and generalization.

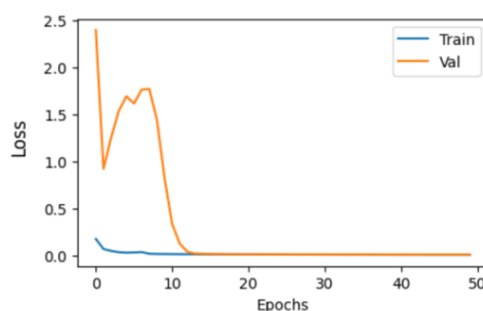


Figure 5. Loss curve of the U-Net model over 50 epochs

The accuracy curve, reaching a plateau at approximately 99%, demonstrates the model's excellence in pixel classification, suggesting a low occurrence of false positives and negatives. Figure 6 presents the accuracy curve of the U-Net model over 50 epochs, showing the continuous improvement of accuracy during training and validation. This steady increase in accuracy indicates the model's ability to consistently learn and adapt to the segmentation task, ultimately achieving high performance.

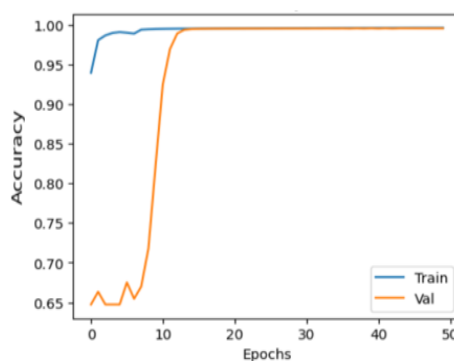


Figure 6. Accuracy curve of the U-Net model over 50 epochs

The early epochs of the learning curves show notable volatility in the loss, which is often observed as the model begins to learn and adjust to the complexity of the data. This initial instability reflects the process of adjusting the network weights from an initially untrained state to a configuration that minimizes the loss. While the training loss drops rapidly, the validation loss experiences some spikes, likely due to the optimization process where the model navigates through local minima before reaching a more stable convergence. After this initial phase, the validation loss curve stabilizes and closely follows the training loss curve, indicating that the model has become robust and less sensitive to the specifics of the training data, a sign of good generalization.



### 5.2. Prediction performance

The temporal efficiency of the model during the prediction phase is crucial for real-time applications. The average processing time  $T$  and the number of frames per second (FPS) are key performance indicators. These metrics are calculated as in (9) and (10):

$$T = \frac{1}{M} \sum_{i=1}^M t_i \quad (9)$$

$$FPS = \frac{1}{T} \quad (10)$$

Where  $t_i$  represents the time taken to predict the  $i$ -th image and  $M$  is the total number of images tested. These formulas allow us to evaluate the model's ability to function efficiently in an industrial setting where processing speed is as important as segmentation accuracy. The Table 4 presents the average prediction times and real-time performance of the model, indicating high efficiency for industrial applications.

Table 4. Prediction time and real-time performance

Metric	Value
Mean prediction time	0.097 seconds
Frames per second	10.24

### 5.3. Performance metrics

To comprehensively evaluate the performance of our U-Net model in segmenting wheel rim contours, we employed several standard metrics: accuracy (ACC), F1 score, Jaccard index (IoU), recall (R), and precision (P). The TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively. The high values obtained for these metrics confirm the precision of the U-Net model in segmenting wheel rim contours, highlighting its applicability in contexts where precision is paramount. Table 5 presents the model's performance on the test set, showing excellent values across various key metrics. These metrics are defined as follows:

$$Acc = \frac{TP+TN}{TP+FP+FN+TN} \quad (11)$$

$$F1 = 2 \times \frac{P \times R}{R + P} \quad (12)$$

$$IoU = \frac{TP}{TP+FP+FN} \quad (13)$$

$$R = \frac{TP}{TP+FN} \quad (14)$$

$$P = \frac{TP}{TP+FP} \quad (15)$$

Table 5. Model performance

Metric	Test (%)
Accuracy	99.45
F1 Score	0.99
Jaccard index (IoU)	0.98
Recall	0.99
Precision	0.99

The high values of accuracy, F1 score, IoU, recall, and precision reflect the excellent performance of the model on the test data. These results suggest that the model can segment wheel rims with high precision, minimizing pixel classification errors. The strong performance on the test set indicates effective generalization, which is crucial for the practical application of the model in real-world scenarios where segmentation accuracy is paramount.

Figure 7 illustrates the mean squared error (MSE) calculated for each image in our test set. The MSE measures the average squared difference between the pixels of the wheel rim contours detected by our U-Net model and the reference values. The results show low error for the majority of images, with a few

peaks that may indicate cases where the model encountered difficulties, possibly due to complex variations in texture or contrast in those specific images.

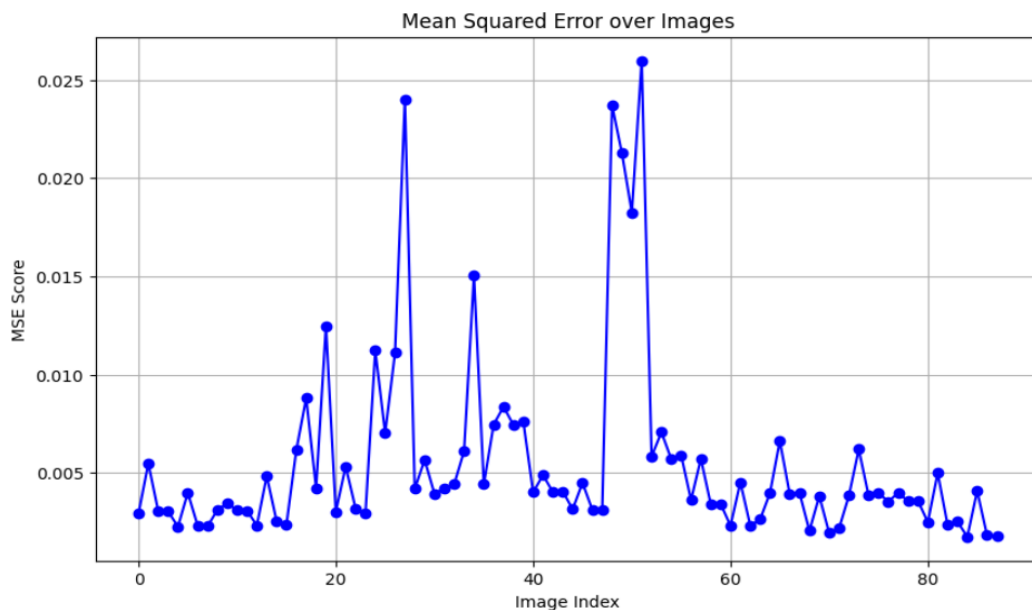


Figure 7. Variation of MSE per image

These higher error points provide valuable insights for future improvements of the model, highlighting situations that require more sophisticated processing or additional training. By analyzing these cases, we can identify specific conditions, such as unusual textures or lighting variations, that challenge the model's current capabilities. This understanding will guide future refinements in data augmentation and model architecture to further improve segmentation accuracy.

#### 5.4. Practical application with OpenCV and contour analysis

The second phase of the evaluation involves deploying the model in a practical context for extracting wheel rim contours. Processing images via OpenCV, using edge detection and dilation functions, allowed for effective contour extraction. The visual results, illustrated by the provided images, show precisely delineated wheel rim contours, demonstrating the accuracy of the U-Net model coupled with OpenCV contour analysis.

Figure 8 shows the results of wheel rim segmentation and contour detection by U-Net and OpenCV. This figure illustrates how the external contours of the rims were accurately detected after applying image processing and computer vision methods. The red circle represents the planned path for the deburring robot, indicating the areas the robot will follow to perform the deburring process with precision.

To ensure optimal wheel rim contour detection, our process relies on a systematic sequence of image processing and computer vision algorithms, detailed in Table 6. The pre-trained U-Net model serves as the foundation of our prediction system, enabling precise initial segmentation. The images are first converted to grayscale, followed by thresholding at 0.5, a value chosen to balance sensitivity and specificity in contour distinction. This preparatory step is crucial for enhancing the contrast necessary for effective contour extraction. The Canny algorithm is then applied to detect fine edges, complemented by a dilation operation to reinforce the continuity of the detected contours. OpenCV's find Contours function is used to accurately capture the external contours of the rims, which are subsequently highlighted with distinct markings on the output images.

This process is visually illustrated by marked points on the detected contours, facilitating the verification of segmentation accuracy. The overlay and marking provide a clear representation of the results, highlighting the synergy between our deep learning model and advanced image processing techniques for reliable application in an industrial setting. This visualization not only aids in accuracy validation but also helps identify areas for potential model refinement.

Table 6. Contour detection with OpenCV		
Parameter	Description	Value / method used
Prediction model	Trained U-Net model	Used for initial segmentation
Image processing	Conversion and thresholding	Conversion to grayscale and thresholding at 0.5
Contour detector	Canny and dilation	Use of the Canny algorithm followed by dilation
Contour extraction	«findContours» function	Detects the outer contours of the rims
Visualization	Overlay and marking	Points marked on the detected contours



Figure 8. Results of the segmentation and detection of wheel rim contours by U-Net and OpenCV

5.5. Error analysis

Despite the high performance of our model, some errors persist, primarily due to complex variations in texture or contrast in certain images. The higher error points identified in Figure 7 provide valuable insights for future improvements. Here, we present some examples of segmentation errors in Figure 9, explaining potential causes and areas for improvement.

The rims in the database are not newly molded but already mounted on cars, with internal elements such as brake plates and discs. These internal elements add additional variations in texture and contrast, complicating the segmentation task for the model. Ideally, rims after molding should not contain any internal elements and should be empty.



Figure 9. Examples of segmentation errors by the U-Net model

These images show the areas where the model encountered difficulties, illustrating false positives and false negatives in the detection of rim contours. These errors can be attributed to several factors, including complex variations in rim texture, the presence of internal elements such as brake plates and discs, as well as low or high contrasts in certain parts of the image. To improve the model's accuracy, approaches such as data augmentation with images featuring these specific variations could be explored. Despite several trials and adjustments of the hyperparameters, some limitations persist, highlighting the need for a more representative database. A database including images of industrial rims with varied lighting conditions and without internal elements could help reduce these errors and enhance the model's robustness.

5.6. Preliminary discussion on practical Impact4

The obtained results show that our U-Net model is capable of segmenting wheel rim contours with high precision. This accuracy is crucial for industrial applications, particularly in improving robotic

deburring processes, where clear and precise contours are necessary to guide deburring tools. Moreover, the application of this model can reduce production costs and increase the quality of finished products by minimizing human errors and automating a complex and repetitive process.

## 6. CONCLUSION

This study demonstrates the strong potential of U-Net architecture for the precise detection of wheel rim contours, a critical aspect in automating the robotic deburring process within the automotive industry. By achieving remarkable accuracy in image segmentation, with metrics such as 99.45% accuracy and F1 score of 0.99, the robustness and reliability of the model have been firmly established. The integration of OpenCV further enhances real-time image processing, achieving 10.24 FPS, making it highly applicable in industrial environments where speed and precision are essential. The implications of this research are significant for industrial automation, particularly in improving production efficiency and reducing human errors. However, to address the remaining challenges, particularly in diverse lighting conditions and rim variations, further work on dataset expansion and the application of transfer learning techniques will be necessary. Additionally, the exploration of internal rim contours could offer new avenues for more advanced applications in the future. Overall, this work contributes to the growing role of deep learning in industrial settings, showcasing its capacity to optimize processes, reduce costs, and improve product quality. The advancements achieved here signal a promising future for the continued integration of AI and deep learning in manufacturing, driving innovation and efficiency across the industry. As industries continue to adopt these technologies, further enhancements in model robustness and real-time processing will likely open up even more applications.




## REFERENCES

- [1] L. Zhang *et al.*, "Navigation path recognition between rows of fruit trees based on semantic segmentation," *Computers and Electronics in Agriculture*, vol. 216, Jan. 2024, doi: 10.1016/j.compag.2023.108511.
- [2] B. Li, Y. Zhang, Y. Ren, C. Zhang, and B. Yin, "Lite-UNet: a lightweight and efficient network for cell localization," *Engineering Applications of Artificial Intelligence*, vol. 129, Mar. 2024, doi: 10.1016/j.engappai.2023.107634.
- [3] W.-L. Mao *et al.*, "Integration of deep learning network and robot arm system for rim defect inspection application," *Sensors*, vol. 22, no. 10, May 2022, doi: 10.3390/s22103927.
- [4] X. Chen, K. Zhang, W. Wang, K. Hu, and Y. Xu, "Intelligent identification of tunnel water leakage based on super-resolution reconstruction and triple attention," *Measurement*, vol. 225, Feb. 2024, doi: 10.1016/j.measurement.2023.114009.
- [5] H. Du, H. Wang, C. Yang, L. Kabalata, H. Li, and C. Qiang, "Hand bone extraction and segmentation based on a convolutional neural network," *Biomedical Signal Processing and Control*, vol. 89, Mar. 2024, doi: 10.1016/j.bspc.2023.105788.
- [6] R. Staněk, T. Kerepecký, A. Novozámský, F. Šroubek, B. Zitová, and J. Flusser, "Real-time wheel detection and rim classification in automotive production," *Proceedings-International Conference on Image Processing, ICIP*, pp. 1410–1414, 2023, doi: 10.1109/ICIP49359.2023.10223161.
- [7] K. Muntarina, R. Mostafiz, F. Khanom, S. B. Shorif, and M. S. Uddin, "MultiResEdge: a deep learning-based edge detection approach," *Intelligent Systems with Applications*, vol. 20, Nov. 2023, doi: 10.1016/j.iswa.2023.200274.
- [8] Y. Wang, B. Jia, and C. Xian, "Machine learning and UNet++ based microfracture evaluation from CT images," *Geoenergy Science and Engineering*, vol. 226, Jul. 2023, doi: 10.1016/j.geoen.2023.211726.
- [9] X. Zhang, L. Liang, S. Zhao, and Z. Wang, "GRFB-UNet: A new multi-scale attention network with group receptive field block for tactile paving segmentation," *Expert Systems with Applications*, vol. 238, Mar. 2024, doi: 10.1016/j.eswa.2023.122109.
- [10] X. Soria, A. Sappa, P. Humanante, and A. Akbarinia, "Dense extreme inception network for edge detection," *Pattern Recognition*, vol. 139, Jul. 2023, doi: 10.1016/j.patcog.2023.109461.
- [11] S. J. Wei, D. F. Al Riza, and H. Nugroho, "Comparative study on the performance of deep learning implementation in the edge computing: Case study on the plant leaf disease identification," *Journal of Agriculture and Food Research*, vol. 10, Dec. 2022, doi: 10.1016/j.jafr.2022.100389.
- [12] M. Agarwal, S. K. Gupta, and K. K. Biswas, "Development of Efficient CNN model for tomato crop disease identification," *Sustainable Computing: Informatics and Systems*, vol. 28, Dec. 2020, doi: 10.1016/j.suscom.2020.100407.
- [13] I. F. Onstein, C. Haskins, and O. Semeniuta, "Cascading trade-off studies for robotic deburring systems," *Systems Engineering*, vol. 25, no. 5, pp. 475–488, Sep. 2022, doi: 10.1002/sys.21625.
- [14] K. Falandys, K. Kurc, A. Burghardt, and D. Szybicki, "Automation of the edge deburring process and analysis of the impact of selected parameters on forces and moments induced during the Process," *Applied Sciences*, vol. 13, no. 17, Aug. 2023, doi: 10.3390/app13179646.
- [15] Z. Liu, B. Guo, F. Wu, T. Han, and L. Zhang, "An improved burr size prediction method based on the 1D-ResNet model and transfer learning," *Journal of Manufacturing Processes*, vol. 84, pp. 183–197, Dec. 2022, doi: 10.1016/j.jmapro.2022.09.060.
- [16] Y. Zhang, H. Liu, W. Cheng, L. Hua, and D. Zhu, "A novel trajectory planning method for robotic deburring of automotive castings considering adaptive weights," *Robotics and Computer-Integrated Manufacturing*, vol. 86, Apr. 2024, doi: 10.1016/j.rcim.2023.102677.
- [17] S. Ghanem and R. A. Kerekes, "Robust wheel detection for vehicle re-identification," *Sensors*, vol. 23, no. 1, Dec. 2022, doi: 10.3390/s23010393.
- [18] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning skip connections to exploit multiscale features in image segmentation," *IEEE Transactions on Medical Imaging*, vol. 39, no. 6, pp. 1856–1867, Jun. 2020, doi: 10.1109/TMI.2019.2959609.
- [19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4\_28.




- [20] A. Verl, A. Valente, S. Melkote, C. Brecher, E. Ozturk, and L. T. Tunc, "Robots in machining," *CIRP Annals*, vol. 68, no. 2, pp. 799–822, 2019, doi: 10.1016/j.cirp.2019.05.009.
- [21] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 2472–2481, doi: 10.1109/CVPR.2018.00262.
- [22] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
- [23] L. Xuan and Z. Hong, "An improved canny edge detection algorithm," *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, pp. 275–278, 2017, doi: 10.1109/ICSESS.2017.8342913.
- [24] A. Trujillo-Pino, K. Krissian, M. Alemán-Flores, and D. Santana-Cedr s, "Accurate subpixel edge location based on partial area effect," *Image and Vision Computing*, vol. 31, no. 1, pp. 72–90, Jan. 2013, doi: 10.1016/j.imavis.2012.10.005.
- [25] H. Wang *et al.*, "Deep-learning-based workflow for boundary and small target segmentation in digital rock images using UNet++ and IK-EBM," *Journal of Petroleum Science and Engineering*, vol. 215, Aug. 2022, doi: 10.1016/j.petrol.2022.110596.
- [26] J. C. Aurich, D. Dornfeld, P. J. Arrazola, V. Franke, L. Leitz, and S. Min, "Burs—analysis, control and removal," *CIRP Annals*, vol. 58, no. 2, pp. 519–542, 2009, doi: 10.1016/j.cirp.2009.09.004.
- [27] F. Domroes, C. Krewet, and B. Kuhlenkoetter, "Application and analysis of force control strategies to deburring and grinding," *Modern Mechanical Engineering*, vol. 03, no. 02, pp. 11–18, 2013, doi: 10.4236/mme.2013.32A002.
- [28] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network," *Computers and Electronics in Agriculture*, vol. 154, pp. 18–24, Nov. 2018, doi: 10.1016/j.compag.2018.08.048.
- [29] L. Yang, S. Xu, J. Fan, E. Li, and Y. Liu, "A pixel-level deep segmentation network for automatic defect detection," *Expert Systems with Applications*, vol. 215, Apr. 2023, doi: 10.1016/j.eswa.2022.119388.
- [30] A. A. Toropov, S. L. Ko, and J. M. Lee, "A new burr formation model for orthogonal cutting of ductile materials," *CIRP Annals*, vol. 55, no. 1, pp. 55–58, 2006, doi: 10.1016/S0007-8506(07)60365-5.
- [31] Y. Chen, R. Sun, Y. Gao, and J. Leopold, "A nested-ANN prediction model for surface roughness considering the effects of cutting forces and tool vibrations," *Measurement*, vol. 98, pp. 25–34, Feb. 2017, doi: 10.1016/j.measurement.2016.11.027.
- [32] X. Soria, E. Riba, and A. Sappa, "Dense extreme inception network: towards a robust CNN model for edge detection," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2020, pp. 1912–1921, doi: 10.1109/WACV45572.2020.9093290.

## BIOGRAPHIES OF AUTHORS






**Hicham Ait El Attar**    is an engineer and researcher in deep learning, specializing in convolutional neural networks and their applications in computer vision. He is a technology teacher, with expertise in Python, Arduino, and deep learning. Currently a Ph.D. candidate, his research focuses on the application of deep learning for mobile robot control. Dedicated to industrial automation and the improvement of computer vision systems, he is passionate about technological innovation. He can be contacted at email: aitelattar.hicham@gmail.com.






**Hassan Samri**    is Associate Professor and Doctor in fluid and energy mechanics, head of the Department of Mechanical Engineering at ENSET in Mohammedia. He holds the Approval to Direct Research (ADR) at the Laboratory of Modeling and Simulation of Intelligent Industrial Systems of ENSET Mohammedia, Hassan II University of Casablanca, Morocco. He is interested in applications of fluid mechanics, robotics, automation of industrial systems, industry 4.0 and artificial intelligence. He can be contacted at email: samrih127@gmail.com.






**Moulay El Houssine Ech-Chhibat**    is an Associate Professor and Doctor of mechanical engineering. He has the Accreditation to Direct Research (ADR) in the Laboratory of Modeling and Simulation of Intelligent Industrial Systems at ENSET Mohammedia, Hassan II University of Casablanca, Morocco. He is interested in maintenance, reliability, robotics, automatics, industry 4.0, and artificial intelligence applications. He can be contacted at email: echchhibate@gmail.com.






**Khalifa Mansouri**    was born in 1968 in Azilal, Morocco. He is currently a Researcher-Professor in computer science, Training Director and Director of the M2S2I Research Laboratory at ENSET of Mohammedia, Hassan II University of Casablanca. His research interests include information systems, e-learning systems, real time systems, artificial intelligence, and industrial systems (modeling, optimization, numerical computation). Graduated from ENSET of Mohammedia in 1991, CEA in 1992 and Ph.D. (computation and optimization of structures) in 1994, HDR in 2010 and National Ph.D. (computer science) in 2016. He is the author of 10 books in computer science, a scientific book with the publisher Springer, 441 research papers including 248 in the Scopus library and supervised 36 defended doctoral theses. He can be contacted at email: [khalifa.mansouri@enset-media.ac.ma](mailto:khalifa.mansouri@enset-media.ac.ma).



**Abderrahim Bahani**    was born in Morocco. He received the B.Sc. and M.Sc. degrees in mechanical engineering from the University of Mohammed V, Rabat, Morocco, in 2018 and 2020, respectively. He also received an external aggregation degree in mechanical engineering in 2015. Currently, he is a lecturer of mechanical engineering at preparatory classes for engineering schools in Mohammedia, Morocco. His research interests are in the area of mechanical modeling, robotics, and artificial intelligence (PDF) the inverse kinematics evaluation of 6-DOF robots in cooperative tasks using virtual modeling design, and artificial intelligence Tools. He can be contacted at email: [abdeer.bahani@gmail.com](mailto:abdeer.bahani@gmail.com).



**Tarek Bahrar**    holds a degree in mathematical sciences, state engineering diploma in industrial engineering. Currently a Ph.D. student in the Modeling and Simulation of Intelligent Industrial Systems (M2S2I) département of ENSET Mohammédia Hassan 2 University and Prototype and Launch Manager in the automotive field. His research areas of interest include mechanics, energy, artificial intelligence particularly in the automobile industry. He can be contacted at email: [tarek.bahrar1@gmail.com](mailto:tarek.bahrar1@gmail.com).