STUDIA INFORMATICA
Nr 1 (32) Systems and information technology 2025

Rafał Jednoróg
ORCID: 0009-0000-6366-6422

Kozminski University
ul. Jagiellońska 57/59, 03-301 Warsaw, Poland

`rjednorog@kozminski.edu.pl`

A Vision-Based System for Rapid DXF Generation in Metalworking: Image Processing Pipeline and Economic Feasibility Case Study

DOI: 10.34739/si.2025.32.05

Abstract. Low-volume and custom production jobs in small and medium-sized metalworking enterprises are often declined due to the high fixed costs and time requirements of manual CAD modeling. This paper presents a practical, production-ready desktop application for the automated generation of DXF files from photographs of physical parts, designed to support rapid laser cutting and prototyping workflows. The system combines a user-friendly GUI with a robust computer vision pipeline including adaptive thresholding, contour extraction, geometric scaling, and vector simplification. Unlike generic raster-to-vector tools, it targets the specific needs of SMEs: offline use, closed contours, minimal training, and compatibility with CAM systems. A real-world case study demonstrates the successful replication of an agricultural machine parts, showing ~90% reduction in preparation time and ~45% profit margin on the order. Interviews with industry stakeholders confirm the potential of the proposed solution to recover previously unviable jobs and increase monthly revenues by an estimated 5-7%. This interdisciplinary contribution integrates computer vision and applied economics, offering a scalable solution to improve productivity and digital inclusion in SMEs.

Keywords: Computer vision, DXF generation, SME automation, Economic impact

1 Introduction

Small and medium-sized manufacturing enterprises (SMEs) are vital contributors to industrial output, yet they often lag in adopting advanced automation and digitalization technologies. Resource constraints mean that many SMEs still rely on manual processes for design and production preparation, especially in high-mix, low-volume scenarios [1]. For example, diverse product variants, small batch sizes, and short product life cycles make it difficult for SMEs to justify expensive automation investments [1]. Traditional computer vision solutions and Industry 4.0 tools promise efficiency gains, but their implementation typically demands substantial capital and specialized expertise that SMEs may not possess [2, 4]. Studies have identified limited investment capacity, shortages of skilled personnel, and uncertain returns on investment as key barriers preventing SMEs from embracing technologies like machine vision in manufacturing [2, 4, 11]. As a result, there is a clear need for low-cost, easy-to-use automation solutions that can operate within the constraints of SME environments while still delivering productivity and cost benefits.

One costly and time-consuming task in low-volume production is the creation of computer-aided design (CAD) [13] models for each custom part. In many small workshops, if a client provides a physical sample or an informal drawing (e.g., a sketch or a photograph of a part), technicians must manually re-create a precise CAD model or 2D drawing for fabrication. This manual drafting process requires expertise and can significantly add to lead time and cost for one-off or small-batch orders. Automating this process using computer vision can fill an important gap: by directly converting images of parts or templates into a CAD-compatible format (such as DXF), SMEs can save engineering hours and reduce errors. DXF (Drawing Exchange Format) is one of the most widely adopted standards for representing 2D vector graphics in manufacturing workflows. Its open, text-based structure allows seamless integration with CNC machines, laser cutters, and most CAM software, making it an ideal target format for automated shape extraction [14]. Existing tools for raster-to-vector conversion (e.g., generic image tracing software) do exist, but they are not tailored to manufacturing needs and often require clean inputs or extensive manual post-editing. Automatic generation of DXF files from images would allow rapid creation of laser cutting patterns, CNC outlines, and other fabrication instructions without the need for a human CAD designer in the loop, thus skilled engineers could focus on more complex tasks.

Prior work on vision-based CAD automation is limited. Some research has explored converting scanned technical drawings into editable CAD files using image processing and pattern recognition. For instance, Intwala [3] developed a method to translate raster images of engineering drawings into DXF format by extracting line features and text via machine learning. Practical applications also include commercial software that vectorizes images (e.g., for architectural plans or logos), but these often require very clean inputs or still need manual correction, and many are cloud-based solutions that raise concerns about data security or require internet connectivity. There remains a lack of a lightweight, offline tool specifically aimed at manufacturing SMEs for quickly generating usable part drawings from arbitrary images or photos. Such a tool should handle moderate image noise or background clutter, produce vector outputs that align with fabrication requirements (for example, closed contours suitable for laser cutting), and be operable by staff with minimal training.

To develop such a solution, this work builds upon well-established techniques from computer vision and image processing. For instance, Otsu's algorithm [7] is a classic thresholding method from the literature [12] used here to distinguish the part from its background. Similarly, the Douglas–Peucker algorithm [8] provides a reliable method of contour simplification and is applied to reduce noise and unnecessary detail in the resulting vector output. General-purpose raster-to-vector tools are also available (e.g., Inkscape's tracing functionality [10]), but these often require very clean input images and still involve manual correction. The widespread availability of high-level libraries such as OpenCV [5] in Python [9] makes it feasible to implement a custom pipeline tailored specifically to the needs of fabrication in SME contexts. These foundations are combined to create a lightweight, standalone application optimized for practical manufacturing scenarios.

This work presents an offline vision-based system for automated DXF generation from part images, designed with the operational constraints and needs of small and medium-sized enterprises (SMEs) in mind. The system constitutes an interdisciplinary innovation that combines computer vision techniques with economic feasibility analysis. The key novelty of the proposed approach lies in the integration of fast, low-cost image processing methods to address a practical manufacturing problem, coupled with an assessment of their economic impact on small businesses. The image-to-DXF conversion pipeline is described in detail, and its effectiveness is demonstrated through a real-world case study involving the repair of a custom metal component in an SME environment. Furthermore, the evaluation includes quantitative analysis of time and cost savings as well as qualitative feedback from industry interviews, providing a comprehensive view of how a targeted computer vision tool can enhance productivity and profitability in small-scale manufacturing.

2 Methodology: Image-to-DXF Conversion Tool

The proposed system is implemented as a stand-alone application with a graphical user interface (GUI) for ease of use, capable of converting a photograph of a part into a vector outline in DXF format suitable for immediate laser cutting or similar fabrication processes. The conversion process consists of several image processing and geometric transformation steps, which are outlined below. The design philosophy relies on proven, classical algorithms to ensure reliability and speed, thereby avoiding the need for computationally intensive machine learning models or large training datasets. This choice results in a lightweight tool that can be easily deployed on standard office computers without dedicated hardware. The system was developed in Python [9], employing OpenCV [5] for image processing operations and the `ezdxf` library [6] for DXF file generation.

Conversion Pipeline Overview

The system processes an input image through the following main stages (Fig. 1 illustrates selected steps on an example image captured directly from the application's GUI):

1. **Image Preprocessing:** The input photograph (Fig. 1a) is first loaded and converted to grayscale (Fig. 1b) to simplify the information content and reduce computational complexity. Instead of a traditional Gaussian blur, the system uses *bilateral filtering*, a

non-linear technique that smooths homogeneous areas while preserving sharp edges. This is especially beneficial in technical imagery, where preserving part contours is critical for accurate boundary extraction. In case of implementation, a bilateral filter is used with following parameters $d = 9$, $\sigma_{\text{color}} = 75$, and $\sigma_{\text{space}} = 75$ selected empirically based on tests with typical input photos.

2. **Binarization with Otsu's Method:** To distinguish the part from its background, the grayscale image is converted into a binary image (Fig. 1c). This is achieved using Otsu's thresholding method, which automatically determines an optimal threshold by minimizing intra-class variance (i.e., the weighted sum of variances of the foreground and background pixel distributions). In practice, Otsu's method is particularly effective when the image histogram is bimodal, as is often the case in industrial imagery with strong object–background contrast. The method is applied in inverse binary mode, rendering the foreground (i.e., the object or part) in white and the background in black. This representation facilitates clean contour extraction in subsequent processing steps.
3. **Morphological Cleaning:** Despite thresholding, binary images often contain noise, small holes, or disconnected edge fragments. To address this, a sequence of morphological transformations is applied. First, a *closing operation* (Fig. 1d) which consists of a dilation followed by an erosion. It is used to bridge small gaps and fill narrow voids within the object's silhouette, ensuring that edge continuity is intact. This is especially helpful in cases where surface imperfections or shadows create interruptions in the object's shape. Next, an *opening operation* is applied (not shown here), an erosion followed by a dilation, to remove small, isolated white regions (noise) that do not correspond to meaningful object features. Together, these operations produce a clean, coherent binary mask that accurately reflects the part's outline while suppressing background interference. For both steps, a 4×4 kernel is used, the size of which was selected empirically based on testing with real-world part photographs and templates. This kernel offers a balanced trade-off: it is large enough to connect edge fragments and eliminate noise, yet small enough to preserve the fine geometric detail of the contours.
4. **Contour Extraction:** The next step is to extract continuous outlines (contours) from the cleaned binary image. Using OpenCV's `findContours` function (with the `RETR_TREE` mode), the application identifies all closed regions in the image, returning them as point sequences. These contours are sorted by area, and it is assumed that the largest contour(s) correspond to the main part outline and any significant internal cutouts. Very small contours are discarded as noise based on a dynamic area threshold defined as:

$$\text{Threshold} = \max(200\,\text{px}^2,\ 10^{-4} \times A_{\max}) \tag{1}$$

   where $A_{\max}$ is the area of the largest detected contour. This two-pass filtering makes sure that only meaningful contours remain: after the first removal of small contours, the largest remaining one is reassessed, and the threshold is recalculated to refine the selection.
5. **Scaling to Real Dimensions:** In order to produce a DXF at true scale, the user provides two reference dimensions (the known width and height of the part in millimeters). Using these inputs, the system computes scaling factors by comparing the pixel bounding box of the detected contour(s) to the real-world dimensions. Specifically, if $(\Delta x, \Delta y)$ is the width and height of the contour's bounding box in pixels and (W, H) are the real width and height in millimeters (entered by the user), program computes scale factors $s_x = W/\Delta x$

and $s_y = H/\Delta y$. All contour coordinates are then transformed to millimeter units by translating the origin to $(x_{\min}, y_{\min})$, multiplying x values by s_x, and y values by $-s_y$ to convert from image coordinates (where y increases downward) to a conventional Cartesian system used in CAD software. The result of the scaling process is visualized in Fig. 1e.

6. **Automatic and Manual Epsilon Selection.:** To control the trade-off between geometric accuracy and vector simplification in the final DXF output, the application provides two modes of setting the approximation tolerance ε used in the Ramer-Douglas-Peucker algorithm:
   - **Automatic Mode:** By default, the application computes a recommended ε based on the diagonal of the part's bounding box:

     $$\varepsilon_{\text{auto}} = \alpha \cdot \sqrt{W^2 + H^2}$$

     where W and H are the real-world width and height of the part (in millimeters), and α is an empirically tuned coefficient set to 0.001 during the implementation to ensure high-fidelity vector output. A looser tolerance of 0.005 × diagonal is used for smaller features. Thanks to this the tolerance scales proportionally with part size.
   - **Manual Mode:** Advanced users can override the automatic suggestion by entering a custom ε value via the GUI. This is useful when extreme precision is required or when the user wishes to aggressively simplify the geometry to reduce cutting time.

   The GUI provides a slider and numeric entry box for manual adjustments, along with descriptive labels (e.g., *"Accuracy: high"*, *"Simplified"*) to guide users without requiring them to understand the mathematical underpinnings. Regardless of the selected mode, the application internally applies a two-tier simplification strategy. For each contour, a primary tolerance $\varepsilon_{\text{base}}$ is used if the contour's area exceeds 1% of the largest contour's area ($A > 0.01 \cdot A_{\max}$); otherwise, a looser threshold $\varepsilon_{\text{loose}} = 2 \cdot \varepsilon_{\text{base}}$ is used. This ensures that large, structural contours are preserved with higher accuracy, while minor or nested features are simplified to reduce node count and improve processing efficiency.
7. **Vectorization and DXF Generation:** The final stage is to generate vector geometry from the scaled contour points and save the output as a DXF file. For each extracted contour, the application differentiates between circular and arbitrary shapes. If a contour's shape is approximately circular (determined by checking whether the contour area deviates less than 20% from the area of its minimum enclosing circle), it is represented as a DXF `CIRCLE` entity (with center and radius). Otherwise, the contour is approximated by a polyline where the Ramer–Douglas–Peucker algorithm is applied (via OpenCV's `approxPolyDP`) to reduce the number of points, using either the automatically calculated or manually defined ε. The resulting polyline vertices are then written to the DXF file as a closed `LWPOLYLINE` entity. All DXF geometry is exported in millimeter units. The final file (Fig. 1f) can be opened in standard CAD or CAM software for inspection or immediate fabrication.

The above pipeline enables a user to go from a raw photograph to a ready-to-cut CAD outline with minimal intervention. The processing from image loading to DXF export typically takes only a few seconds on a standard PC for a 2D part outline. To validate the quality of the extracted geometry, Figure 2 shows a comparison between an original steel part and its replicated counterpart, which was laser-cut directly from the generated DXF file. The entire image-processing pipeline is computationally efficient. Each step (filtering, thresholding, contour finding, etc.) operates in O(N) or O(N log N) time relative to the number of pixels or contour points, so the processing scales linearly with image size for practical purposes. On a typical laptop (e.g. Core i5 processor), a 1024×768 resolution image is processed in under one second of computation.

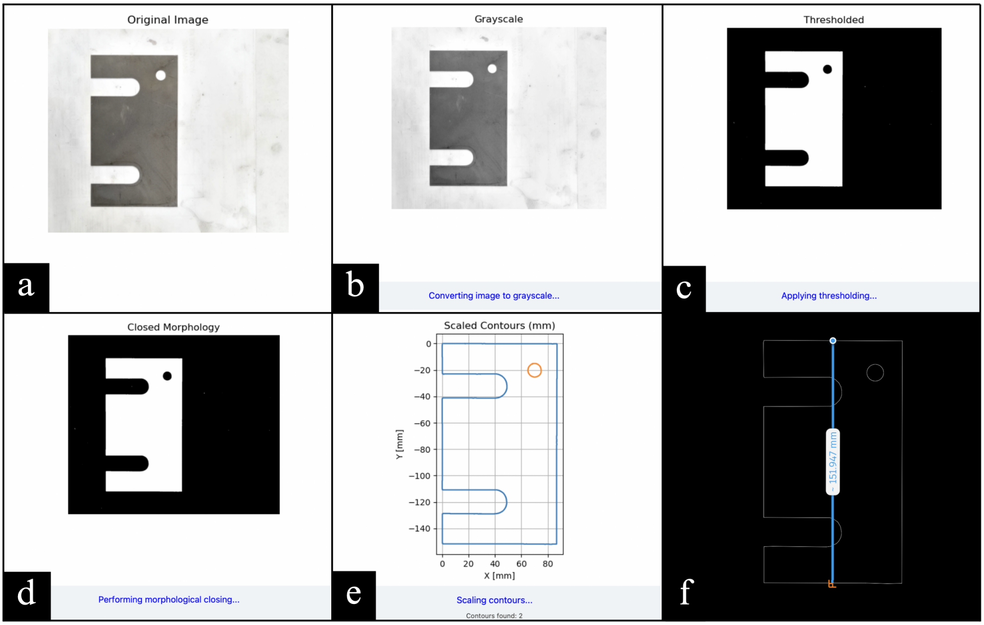


Figure 1. Selected steps of the DXF-generation pipeline. (a) Original image loaded into the application; (b) grayscale version for intensity-based processing; (c) binary mask generated by inverse Otsu thresholding; (d) closed binary mask after morphological closing; (e) scaled contours in real-world dimensions (mm); (f) final DXF file preview opened in CAD software with a measurement for validation.

User Interface and Accessibility To ensure ease of use in real-world conditions, the system includes a graphical user interface (GUI) designed for non-expert users. As shown in Figure 3, users can upload part images, input real-world dimensions, adjust geometric approximation parameters, and initiate conversion through buttons. A live preview of processes and descriptions including total contour count support operations control. The interface was built using Python’s Tkinter framework and tested by technicians without prior CAD experience.

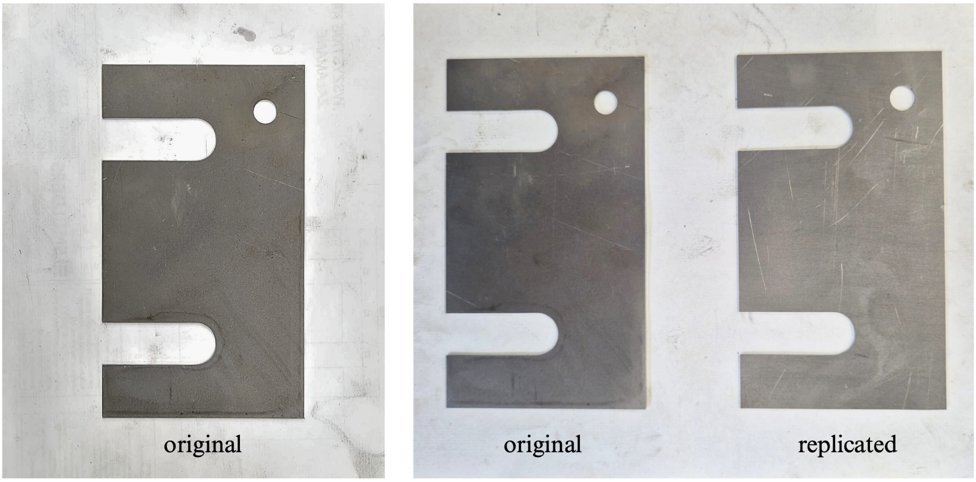
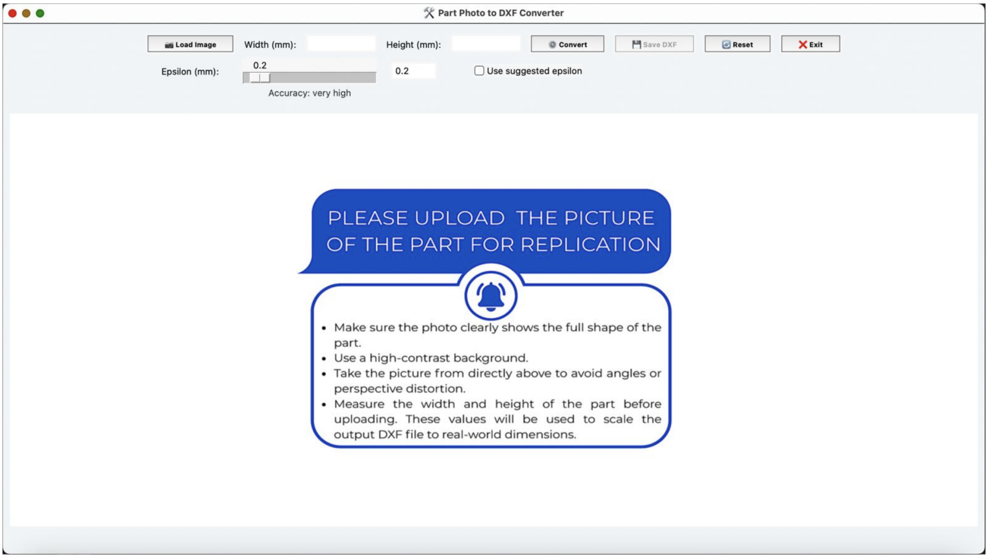


Figure 2. Comparison of geometry fidelity before and after DXF conversion. Left: the original steel part; Right: original and replicated parts side-by-side - replicated part laser-cut from the DXF file. The shapes demonstrate near-identical accuracy sufficient for fabrication.



Width (mm):

Height (mm):

Epsilon (mm): 0.2 0.2

☐ Use suggested epsilon

Accuracy: very high

PLEASE UPLOAD THE PICTURE OF THE PART FOR REPLICATION

- Make sure the photo clearly shows the full shape of the part.
- Use a high-contrast background.
- Take the picture from directly above to avoid angles or perspective distortion.
- Measure the width and height of the part before uploading. These values will be used to scale the output DXF file to real-world dimensions.

Figure 3. Screenshot of the DXF converter application GUI. Users can load an image, specify real-world dimensions, adjust contour simplification, and generate the DXF file using intuitive controls. A visual accuracy indicator and instructions help guide input quality.

3 Case Study: Rapid Repair of a Mixing Auger

To validate the proposed tool in a practical setting, a case study was conducted in collaboration with a local metalworking SME, focusing on a real-world repair task. The subject of the repair was a component of an agricultural feed grinder: a mixing auger screw with several cutting blades ("wały tnące") that had become damaged through wear and required replacement. No CAD models or technical drawings of the auger or its blades were available, a common situation for older or custom-built machinery. The objective was to fabricate replacement blade sections by capturing the geometry of the originals and cutting new parts from sheet metal. This scenario provided an ideal test case for the image-to-DXF pipeline, as it required generating a precise 2D outline for laser cutting in the absence of an existing CAD drawing.

In the repair process, the technicians first recreated the shapes of the missing blade sections using simple physical templating: cardboard pieces were cut and manually fitted onto the auger in place of the broken segments to approximate the original shape and curvature (Fig. 4a). Once satisfied with the fit, each cardboard template was photographed on a flat, contrasting background. These photographs of the blade outlines served as input to the DXF conversion application. Using the tool, the images were processed to extract the contour of each blade profile, and DXF files were generated. In this case, the technicians measured the dimensions of the templates (length and width) and entered those values to scale the DXF output to true size. The resulting DXF profiles (example shown in Fig. 4b) were then imported into the company's laser cutting planning software.

New blade pieces were laser-cut from steel plate and later welded onto the auger screw to complete the repair. Photographs of the auger are shown in Fig. 4c and Fig. 4d, respectively before and after the intervention. The entire process from template creation to finished part installation was completed in a short time frame, demonstrating the potential of rapid turnaround using the vision-based DXF tool. This case study setup also allowed for evaluation of the tool's usability in a real-world context. The operator who performed the DXF conversion was a technician with no involvement in the development of the system. After a brief introduction and user guide, the operator was able to run the software on the template photos. A small learning curve was observed in understanding how the image quality affects the outcome; for instance to avoid harsh shadows on the photograph by adjusting lighting, which improved the edge detection results. After one or two trials, obtaining a clean contour became repeatable.

From a financial perspective, this repair job was sold to the client for 5500 PLN, while the material and cutting costs amounted to approximately 320 PLN (119 PLN for laser cutting and around 201 PLN for steel). Additional operational and labor costs including handling, welding, and assembly totaled about 2200 PLN - bringing the overall cost to roughly 2520 PLN. This resulted in a gross profit of 2980 PLN and a profit margin of approximately 54%. Such a margin significantly exceeds the typical 7–15% range achieved by the company on standard orders. A key contributor to this result was the elimination of costly CAD modeling time, thanks to the automated DXF workflow.

In summary, the case study reflects a realistic use-case: a user can take a picture of a part or template and, within minutes, obtain a CAD drawing ready for fabrication without specialized CAD skills. The next section details the results of this case study, focusing on the time saved and the economic implications compared to the traditional approach for such a repair job.

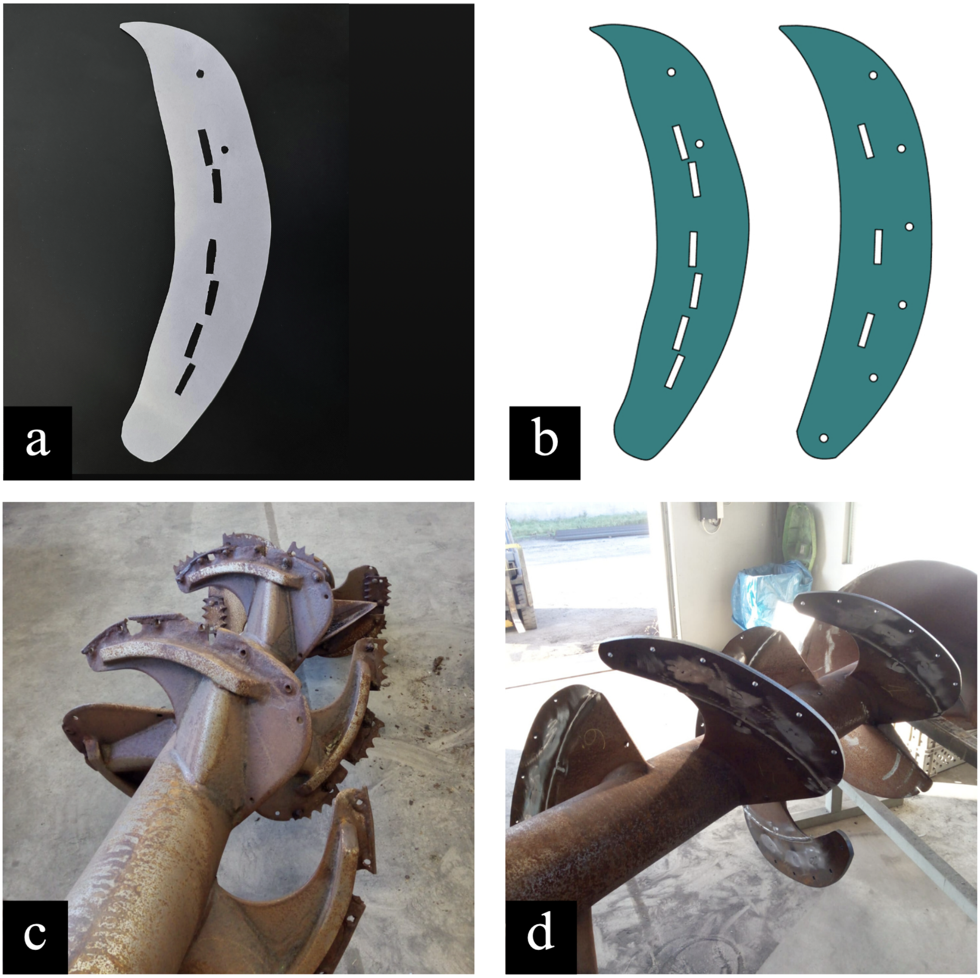


Figure 4. Blade repair workflow using the image-to-DXF system. (a) Cardboard template cut to match one of the original blade segments; (b) digital DXF visualization of two different blade shapes after contour extraction and scaling; (c) the worn mixing auger screw prior to repair, with damaged or missing cutting blades; (d) the auger after successful repair with new laser-cut blade segments mounted and welded in place.

4 Results and Discussion

4.1 Efficiency Gains in CAD Drawing Generation

The introduction of the automated image-to-DXF tool in the case study yielded significant improvements in efficiency. Table 1 summarizes a comparison between the traditional manual process and the automated vision-based process for generating the CAD drawing of one replacement part (1 out of 4 different shapes of blades in this project in total). The comparison is broken down in terms of the time required, the level of labor skill needed, and the approximate costs associated with each approach (mainly labor cost, assuming an engineer or technician hourly rate).

Table 1. Comparison of manual vs. automated CAD drawing generation for the case study part.

Aspect	**Manual Process (Traditional)**	**Automated Process (Our Tool)**
Time to obtain CAD model	~2 hours (measuring & drafting)	5 minutes
Personnel required	Skilled CAD designer or engineer	Any technician (basic training)
Labor cost for design	~approx. 200 PLN (engineering rate)	Negligible (production worker)
Iterations needed	Possibly multiple (for adjustments)	One (capture image & verify)
Total cost for CAD model	approx. 200 PLN	approx. 10 PLN

One can observe the drastic reduction in the time and skilled labor required to produce a usable CAD drawing. In the manual scenario, an experienced CAD designer might spend on the order of a couple of hours measuring the parts and drafting it in CAD software. Using the proposed system, the total effective time was on the order of minutes (including taking the photo and verifying the output). This corresponds to a time savings of roughly 95–98% in this example. In a production setting, such time saved can shorten delivery schedules and allow the company to respond faster to customer needs. In terms of labor, the manual approach requires the attention of a skilled CAD designer or engineer. That person's time is a costly resource; while they are occupied with drafting a single part, they are not available for other high-value engineering tasks. The automated approach, in contrast, can be executed by a technician with only basic computer skills, since the specialized knowledge of CAD drawing is effectively embedded in the software. This frees the skilled engineer to focus on more complex tasks that truly require their expertise.

Translating the time savings into monetary terms, and assuming a typical hourly wage for a design engineer, the cost of manually producing a single technical drawing can be estimated at approximately 200 PLN. The automated method's direct labor cost is essentially the few minutes of a technician's time (perhaps only a few PLN worth of labor). Given that the system is built on existing standard hardware and open-source software, the amortized cost per use is very low, especially as the number of uses increases. The case study also revealed qualitative benefits. The generated DXF outline was sufficiently precise that no additional editing was required in CAD software before cutting. This means the risk of human error (for example, mis-measuring a dimension or drawing a line at the wrong angle) was greatly reduced or eliminated. The consistency of results is another advantage: if multiple similar parts or variations need to be processed, the algorithm will apply the same procedure each time, leading to standardized outputs. This consistency can indirectly reduce costs by avoiding mistakes that might cause a part to be fabricated incorrectly.

From a broader economic feasibility perspective, if the case study results are extrapolated, an SME that frequently engages in custom fabrication or repair jobs could save a substantial amount of labor over the course of a year. For instance, suppose a company typically spends N hours per week on manual CAD drawing for custom orders, and the proposed system cuts that down by around 90%. The annual labor savings (in hours) would be approximately:

$$\text{AnnualHoursSaved} \approx 0.90 \times N \times 52\ , \tag{2}$$

which, when multiplied by the hourly wage W of a designer, translates to an annual labor cost saving of approximately $0.90 \times N \times 52 \times W$. Depending on the workload and wage level, this can amount to several thousand PLN per year. The cost-effectiveness of the proposed system is thus potentially high, especially in environments where custom part preparation is frequent. In such contexts, the initial implementation cost could be recovered over a relatively small number of jobs through cumulative time savings and reallocation of skilled labor to other tasks.

4.2 Feedback from Industry Interviews

Beyond the performance observed in the case study, qualitative feedback was gathered from industry professionals to assess the broader applicability and perceived value of the solution for SMEs. Interviews were conducted with four experts responsible for laser cutting operations at small metal fabrication firms. All interviewees confirmed a common phenomenon: potential clients who lack CAD drawings or design capabilities are often effectively excluded from the customer base, because the cost (or time) for the company's engineering department to create the necessary drawings can outweigh the revenue of a small order. In other words, preparing a CAD model for a one-off part often incurs a fixed overhead that makes such jobs unprofitable. As a result, a number of inquiries for custom-cut parts are turned down each year. In the interviews, one company estimated that roughly 10–15% of small incoming orders were declined for this reason, which is in line with the 14% figure noted in the case study firm's records.

The interviewees also pointed out that when such custom jobs are accepted, they are usually charged at a much higher margin to justify the engineering effort. In fact, they indicated that these kinds of "difficult" orders, if handled, tend to carry significantly higher profit margins than regular production orders that have competitive pricing. One practical strategy mentioned was to treat these jobs as *fillers*: for example, custom part outlines can sometimes be inserted into empty spaces of sheet metal that would otherwise go unused in a large laser cutting job. By producing the custom pieces in areas considered scrap (and thus material cost accounted for at scrap value rather than full sheet cost), the company can generate additional revenue with very little incremental cost. This requires having the part outlines readily available to nest into the cutting plan, which the automated DXF generation would facilitate.

The concept of an easy-to-use image-based quoting tool was met with strong interest. One interviewee (a technology department manager at *LP Konstal*) noted that integrating such a system with a web interface could allow a broad range of customers (farmers, independent repair shops, small construction contractors, etc.) to submit photos of needed parts and receive fabrication services. This could open up a new stream of orders that are currently not fully tapped. He estimated that implementing the solution could result in a revenue increase on the order of 4-7% for his company, associated with a profit increase of 8-10%, given the high margins of these special orders and additional income from providing quick turnaround and possibly even logistics or delivery services for the fabricated parts. Another interviewee emphasized the importance of high automation and speed in the conversion process. In his experience, preparing a DXF drawing from scratch based on manual measurements or even from customer-provided sketches can take anywhere from 20 minutes to an hour, depending on complexity. If this could be reduced to under 10 minutes with an automated tool, it would save on the order of 40–70 PLN in labor cost per job (assuming typical technician/engineer hourly rates in the range of 90–130 PLN/hour). Those savings directly improve the profitability of each small order. Moreover, faster turnaround can make the service more attractive to clients, potentially increasing order volume.

Overall, the industry feedback suggests that a vision-based DXF generation tool addresses a real pain point in the SME metalworking sector. It has the potential to not only reduce internal costs but also to enable new orders that was previously avoided or overlooked. The combination of quantitative projections (a few percent boost in revenue and several percent higher overall profit margin) and qualitative enthusiasm (interest in integration and new use cases like web-based customer outreach) indicates a strong case for further development and deployment of the system.

5 Limitations and Future Work

While the results are promising, the current approach has several limitations that point to future improvements. First, it is limited to generating 2D outlines. The system works well for flat parts or planar profiles, but it cannot capture any three-dimensional features of a part. If material thickness or other 3D geometry is important (for example, angled holes or bends), a single outline photograph will not capture those. Extending the method toward 3D would require additional inputs such as multiple images from different angles or depth sensing, or at least user-specified parameters (e.g., part thickness) to incorporate some 3D information.

Another limitation is the need for controlled imaging conditions. The pipeline assumes a high-contrast photo of the part (or template) against a clean background. Poor lighting, strong shadows, or clutter can cause the contour extraction to fail or introduce noise. During experiments, using a plain backdrop and diffuse lighting produced good results. In the future, more robust vision techniques (like background subtraction or learning-based segmentation) could help isolate the object in more challenging scenes. Similarly, adding a perspective correction step (for instance, by detecting a reference shape or markers in the image) would mitigate errors when the photo is taken at an angle.

Scaling and accuracy currently rely on user input. The user must enter the true dimensions of the object, which introduces a potential source of error if the measurements are wrong. This step could be improved by including a reference object of known size (such as a ruler or calibration marker) in the photo, allowing the software to automatically infer scale without manual entry.

Additionally, the contour simplification uses fixed heuristics for curve approximation. This works for the parts we tested, but more complex outlines with many small features might benefit from a tunable approach. An enhancement could be to let the user adjust the simplification tolerance via an interactive slider, updating the vector result in real time. This would provide finer control for complex shapes while still being far faster than manual drawing.

Another potential extension would be to incorporate the Hough Transform for detecting basic geometric primitives particularly lines and circles in the contour extraction phase. This method could help improve accuracy for parts with highly regular features, such as bolt holes or straight edges, where pixel-level segmentation may produce jagged or irregular shapes. Given that the current GUI already includes parameter sliders, the Hough Transform could be integrated in a way that lets the user fine-tune detection sensitivity interactively. Such hybrid geometric detection may complement the pixel-based pipeline and further reduce post-processing effort.

Finally, the study focused on internal use of the tool within a company. Deploying the system directly to external end-users (for instance, through a web interface for customers) would require an intuitive user interface and additional safeguards. For example, the software would need to verify that uploaded images meet quality requirements and perhaps provide a preview of the detected contour for customer approval before fabrication. These considerations extend beyond the core algorithm, but they would be important for integrating the tool into a real-world production workflow.

6 Conclusion

This paper presented a vision-based approach to automating CAD drawing generation for low-volume manufacturing, with a particular emphasis on suitability for small and medium-sized enterprises (SMEs). The study began by identifying a gap in technology adoption among small fabrication firms, where the high costs and complexity of conventional automation have limited the uptake of advanced digital tools. Addressing the specific task of converting part images to DXF outlines, a lightweight, offline system was developed using classical image processing techniques capable of delivering results within minutes. The introduction and background sections established the dual context of the work: the technical landscape of computer vision applications in manufacturing, and the economic constraints faced by SMEs that demand low-cost, user-friendly automation solutions.

By framing the problem in an interdisciplinary manner, the study highlights how a technological innovation can address practical business needs, aligning with economic considerations such as cost reduction and efficiency improvement. The methodology section details the system's step-by-step processing pipeline—from image preprocessing and thresholding to contour extraction, geometric scaling, and DXF generation—demonstrating how classical computer vision techniques can be effectively applied in a manufacturing context. The approach prioritizes reliability and ease of use, avoiding heavy machine learning in favor of well-understood algorithms that perform robustly on the target problem. This design philosophy proved effective in the case study, where the system was applied in a real manufacturing repair scenario. The case study demonstrated that proposed tool could drastically cut down the time and expertise needed to go from a physical part (or a template) to a CAD model. The results showed tangible savings in labor and time, validating the hypothesis that such a tool can improve the economics of custom production work. The case study also provided a template for how other SMEs might implement and benefit from the system in their own operations.

The evaluation and discussion explored broader implications, noting that the system's offline, fast, and user-friendly nature addresses many of the concerns SMEs have regarding advanced technology adoption. Current limitations were identified, such as the focus on 2D outlines and the requirement for reasonably good image quality, along with suggestions for future enhancements, including support for 3D reconstruction and more robust image handling. Importantly, the interdisciplinary contribution of this project was emphasized: it not only advances the engineering field through the development of an application based on computer vision, but also contributes to economic efficiency in manufacturing processes. In conclusion, the vision-based DXF automation system developed in this work offers a promising solution for small manufacturers seeking to improve their operations without large investments. It exemplifies how a narrowly focused, well-crafted technological solution can create substantial value in a specific niche of manufacturing. Furthermore, it demonstrates that "smart automation" does not always require expensive or complex AI—sometimes a clever utilization of existing algorithms in a targeted way can deliver most of the value. For SMEs, this can mean the difference between manually performing a tedious, skill-intensive task or having it done at the push of a button. This approach, bridging computer vision and economic feasibility, may serve as a model for future innovations aimed at empowering smaller industry players. As the system continues to be refined and its limitations addressed, it is anticipated that it will encourage wider adoption of automation in the SME sector and inspire similar interdisciplinary efforts to tackle the unique challenges faced by these businesses—often family-owned companies with limited access to high-end digital tools. By lowering the entry barrier to CAD automation, such solutions can help democratize access to advanced manufacturing technologies, enhance competitiveness among smaller market participants, and make it easier for them to compete with so-called big players.

Acknowledgments. The author gratefully acknowledges the cooperation and valuable insights provided by the participating companies, including LP Konstal and other manufacturing SMEs associated with the local Special Economic Zone (SSEMP). Special thanks go to Krzysztof D., Artur J., and Mateusz B. for their contributions during interviews and the case study implementation, as well as to Piotr S., whose expertise was helpful in refining the economic estimates.

References

1. Herps, K., Dang, Q.-V., Martagan, T. G., & Adan, I. J. B. F.: A simulation-based approach to design an automated high-mix low-volume manufacturing system. *Journal of Manufacturing Systems* `https://doi.org/10.1016/j.jmsy.2022.05.013`
2. Müller, J. M., Islam, N., Kazantsev, N., Romanello, R., Olivera, G., Das, D., & Hamzeh, R.: Barriers and Enablers for Industry 4.0 in SMEs: A Combined Integration Framework. *IEEE Transactions on Engineering Management* (2024). `https://doi.org/10.1109/TEM.2024.3365771`
3. Intwala, A., et al.: Image to CAD: Feature extraction and translation of raster engineering drawings to DXF format. In: *Computer Vision and Image Processing (CVIP 2019)*, CCIS, vol. 115 Springer, Singapore (2020). `https://doi.org/10.1007/978-981-15-4015-8_18`

4. European Commission: Digital Economy and Society Index (DESI) 2022 – overall progress but digital skills, SMEs and 5G lag behind. (2022). `https://digital-strategy.ec.europa.eu/en/news/digital-economy-and-society-index-2022-overall-progress-digital-skills-smes- -skills-smes-and-5g-networks-lag`
5. Bradski, G.: The OpenCV Library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer* **25**(11)
6. ezdxf Developers: *ezdxf: DXF for Python*, `https://ezdxf.readthedocs.io`. Last accessed 20 Jan 2025
7. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Trans. Sys. Man Cybern.* **9**(1), 62–66 (1979).
8. Douglas, D.H., Peucker, T.: Algorithms for the reduction of the number of points required to represent a line. *The Canadian Cartographer* **10**(2), (1973).
9. Python Software Foundation: *Python 3.11 Documentation*, `https://docs.python.org/3.11/`. Last accessed 28 May 2025.
10. Paulsen, G.: How to Convert Image Files to DXF for Sheet Cutting. Xometry (Blog), 25 May 2022 (updated 14 May 2025). `https://www.xometry.com/resources/sheet/converting-images-to-dxf/`. (Accessed 28 May 2025).
11. SME & CESMII: SME and CESMII release 2022 Smart Manufacturing Market Survey results. Press Release, 18 May 2022. `https://www.prnewswire.com/news-releases/sme-cesmii-release-2022-smart-manufacturing-market-survey-301561585.html`
12. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*, 4th edn. Pearson (2018).
13. Autodesk, Inc.: *AutoCAD 2023 User Guide*. Available online: `https://images.autodesk.com/adsk/files/autocad_aca_user_guide_english.pdf`
14. Autodesk DXF Reference: *The DXF File Structure Explained*. `https://ezdxf.readthedocs.io/en/stable/dxfinternals/filestructure.html`. Last accessed 24 Oct 2025.