Proceedings of the Design Society, Volume 6: DESIGN 2026
https://doi.org/10.1017/pds.2026.10581

Object detection in technical drawings for data-driven design: the case of patents

Marco Consoloni[1,2,✉], Gabriele Marino[1], Denny Meini[1], Luciano Socci[3], Vito Giordano[1,2] and Gualtiero Fantoni[1,2]

[1] Università di Pisa, Italy, [2] Business Engineering for Data Science (B4DS) research group, Italy, [3] Coesia, Italy

✉ marco.consoloni@phd.unipi.it

ABSTRACT: Data-Driven Design (DDD) is emerging as a transformative approach in engineering design, leveraging AI tools to extract knowledge from design data that drive product development and innovation. While large language models have advanced DDD through the analysis of textual data, technical drawings remain largely unexplored. To address the limitations of current vision-language models, this study presents a novel object detection pipeline that automatically identifies components in patent images, enabling data-driven analysis of component geometries, interfaces, and spatial configurations.

KEYWORDS: data-driven design, artificial intelligence (AI), vision-language model, object detection, patent drawings

1. Introduction

In the current digital era, the high volume, variety and velocity of data have driven a pervasive transformation, redefining the operational and decision-making processes of organizations (Cantamessa et al., 2020). This increase of data, combined with the increasing computational capabilities of current AI technologies, is changing the way organizations are developing and designing new products and services (Chiarello et al., 2021).

In this scenario, Data-Driven Design (DDD) has emerged as a paradigm shift that will profoundly re-shape design practice and innovation processes within organizations (Quiñones-Gómez et al., 2025) and its potential is increasingly acknowledged by practitioners and researchers (Johnson et al., 2023). DDD is a multifaced concept and in literature has not been defined in a unified way (Bertoni, 2020). A comprehensive definition of DDD is beyond the scope of this work. However, to contextualize its use in this study, several definitions from the literature are presented below. Trauer et al. (2020) describe DDD as framework in which the collection and analysis of data drive decisions and applications in the product design and development process. Similarly, Cantamessa et al. (2020) describe DDD as the process supporting the transformation of design-related data to effective design decision-making processes. Quiñones-Gómez et al. (2025) define DDD as a design approach in which the extraction of knowledge from data is used to inform design decisions, moving design practice from an intuition-driven process to one informed by real-world data.

In the field of DDD, most studies rely on natural language processing to support design decision-making, where the "data" predominantly consists of text (Bertoni, 2020; Johnson et al., 2023). This emphasis on textual data has been further reinforced by the advent of Large Language Models (LLMs), which have facilitated the extraction and analysis of insights from unstructured text. However, we argue that using only text is limiting and the success of DDD applications also depends on the ability to extract information from visual data sources, such as sketches and technical drawings. In fact, throughout the engineering design process, images provide synthetic representation of design artifacts, and they emerge as the primary mode of communications among innovators and engineers (Jiang et al., 2021). In the early

https://doi.org/10.1017/pds.2026.10581 Published online by Cambridge University Press

stages of engineering design abstract representations such as free-hand sketches enable designers to rapidly explore the design space and communicate their ideas. In later stages, high-fidelity representations such as detailed technical drawings are used to support requirements specification, design optimization and manufacturing. Moreover, images can offer complementary information that text alone cannot capture. For instance, textual descriptions may not adequately capture structural details of design artifacts, often missing crucial information such as geometric features, aesthetic qualities and spatial arrangements of components. In addition, text is affected by linguistic ambiguity and terminological inconsistency. For instance, different terms (e.g., "bracket," "arm," or "joint") may describe components that are geometrically and functionally equivalent, while identical terms may refer to components with entirely different shapes and functions. By contrast, images can effectively address these issues by visually disambiguating component features and spatial arrangements, supporting a more systemic understanding of design artifacts.

For these reasons, this study develops a novel pipeline for analyzing technical drawings to support DDD processes and it aligns with the research stream on analytics and tools for DDD identified by Cantamessa et al. (2020, p. 2). Specifically, this work focuses on developing a novel computer-based system capable of automatically extracting components from technical drawings. In the field of computer vision, this problem is commonly known as Object Detection (OD). The OD task involves both identifying and localizing specific objects within an image; an OD model determines not only which objects are present (e.g., a gear, bolt, or lever) but also their precise positions within an image.

OD on technical drawings, holds significant value for DDD because it allows to: (1) reduce the cognitive load and time required to analyze complex technical drawings, allowing engineers to devote more time to creative design work rather than manual document parsing; (2) facilitate comparative analysis by isolating individual components, thereby improving the ability to recognize functional and structural similarities across alternative designs, and supporting more informed design decisions; and (3) support design-by-analogy practices by automatically revealing component boundaries and interface regions within technical drawings, allowing designers to identify specific elements that may be reusable or adaptable to new design solutions.

The OD pipeline proposed in this study is designed to extract components from patent drawings. This choice is motivated by two main reasons: first, patent documents are publicly available, which increases the reproducibility of the study (Ruiz & Maier, 2017); and second, patent drawings are a specific type of technical drawings characterized by intricated lines, complex shapes, absence of color, limited texture information, and embedded textual elements. These characteristics make OD particularly challenging and non-trivial. Developing an OD model capable of addressing these challenges can lead to more effective tools for analysing also other types of technical images such as CAD drawings, sketches and 2D renderings, with implications that are highly relevant for the whole field of DDD.

2. Related work and literature gaps

2.1. Patent drawings analysis

Among the different kinds of data sources for design, the patent database is probably the biggest digitized and publicly available design repository (Ruiz & Maier, 2017). For this reason, patents documents have been extensively used to support DDD using various data mining techniques (Cantamessa et al., 2020). However, most of existing works on patent analysis for DDD only focused on text analysis and ignored patent drawings (Jiang et al., 2021). In the existing literature of DDD, patent drawings have been primary used to support patent retrieval and similarity. In the early work of Vrochidis et al. (2012), images were employed for image-based patent retrieval to facilitate prior art mapping and provide design stimuli. Jiang et al. (2021) proposed a computer-based system for patent retrieval based on patent drawings. Their approach supports a data-driven design-by-analogy process in which designers reuse existing technical knowledge and ideas from prior patents to develop new design solutions. Similarly, Higuchi and Yanai (2023) develop an application that enables us to search for patent drawings using any input images, allowing a rapid DDD exploration of existing design solutions. Finally, Lin et al. (2023) utilized patent images to assess patent similarity, enabling the identification of potential conflicts and opportunities for design around existing patents. The primary limitation in the existing literature is that the unit of analysis is the entire image; that is, each patent drawing is compared as a whole to other drawings to assess similarity (i.e., overall image similarity). This approach ignores what is inside the drawings, including the

https://doi.org/10.1017/pds.2026.10581 Published online by Cambridge University Press

specific components and features they contain. For this reason, exiting patent drawings analysis remains too coarse-grained to provide actionable insights for DDD. By contrast, a finer level of detail, such as the component-level analysis enabled by the OD task, can identify and locate specific components within technical drawings that could be reusable or adaptable for developing new design solutions and informing design decisions. To the best of our knowledge, no existing approaches have been developed to perform OD on technical drawings. Therefore, this study proposes a novel OD pipeline designed to automatically extract components from patent drawings, enabling fine-grained analysis for DDD applications.

2.2. Object detection for technical drawings

Recent advances in the field of OD have been largely driven by machine learning, specifically by the YOLO family of models (Redmon et al., 2016). These general-purpose models are trained on large-scale annotated datasets, where human annotators have manually labeled natural images by drawing bounding boxes around everyday objects such as cats, bicycles, and chairs, to both identify and locate them within each image. Once these models are trained, they can be adapted to perform OD of other objects provided that a sufficiently large and well-annotated dataset is available for training. This adaptation process, known as fine-tuning, is extensively used in the background literature, but it is not feasible for OD on technical drawings, because: 1) while there are many annotated datasets for natural images (e.g., COCO dataset, ImageNet), no annotated dataset for OD in technical drawings exists; 2) creating such a dataset manually would be extremely expensive and time-consuming, as it requires domain-specific expertise to correctly identify and label every component in each drawing; 3) unlike common objects (e.g., cats, cars, chairs), technical components (e.g., joint, valve, bearings) vary widely across technological domains, and they have different shapes, scales, and configurations depending on the perspective (e.g., front, cross-section, isometric view) so defining consistent categories and annotating them across drawings is largely unfeasible at scale.

To address the lack of training data, Vision-Language (VL) models can be used to perform OD while bypassing the need for manual data annotation and finetuning. VL models such as OpenAI's chatGPT-4V (vision), chatGPT-5, DALL-E and SORA can process and integrate visual information (images or video) with textual information (natural language text). These models can generate images from textual descriptions (i.e., image generation) and can also answer text-based questions about the content of images (i.e., visual question answering). These capabilities can be used to directly prompt a VL model for identifying and locating an object inside a technical drawing. For example, as illustrated in Figure 1, we prompted ChatGPT-5 with a textual query asking the model to identify specific components, namely the nose bridge and the lenses, within patent drawings of eyeglasses. ChatGPT-5 successfully recognized the bridge when the drawing was properly oriented and depicted from a standard isometric view (a). However, the model failed to correctly identify the bridge when the same image was rotated (b). Similarly, when the eyeglasses were shown in a foldable or non-standard spatial configuration (c), or when the drawing presented a cross-sectional view of the hinge mechanism (d), ChatGPT-5 was unable to correctly identify the lenses in the input images. In addition, VL models often respond using text only, and when they produce visual outputs, these are typically modified images rather than pixel-level segmentation masks. Consequently, their outputs are not directly actionable for follow-up analyses and quantitative measurement. Figure 1 is not intended to provide an extensive assessment of current VL models capabilities but rather to illustrate both their potential applicability and their current limitations that directly motivated the present study.

These identified limitations are also consistent with findings from previous research. Picard et al. (2025) evaluated VL models for their spatial understanding of technical drawings using prompt-based approaches and found that models such as GPT-4V and LLaVA-1.6-34B exhibit strong sensitivity to prompt wording, limited reproducibility, and poor spatial reasoning performing no better than random guessing. Doris et al. (2025) reported similar results, demonstrating that GPT-4 and GPT-4o struggle to interpret spatial relationships in CAD drawings, particularly in engineering applications such as Formula SAE regulation analysis. Finally, Consoloni et al. (2025) noted that most VL models are trained on general-purpose datasets rather than on design data. As a result, they struggle with the knowledge-intensive and schematic nature of technical drawings, which require precise spatial reasoning and design understanding.

https://doi.org/10.1017/pds.2026.10581 Published online by Cambridge University Press

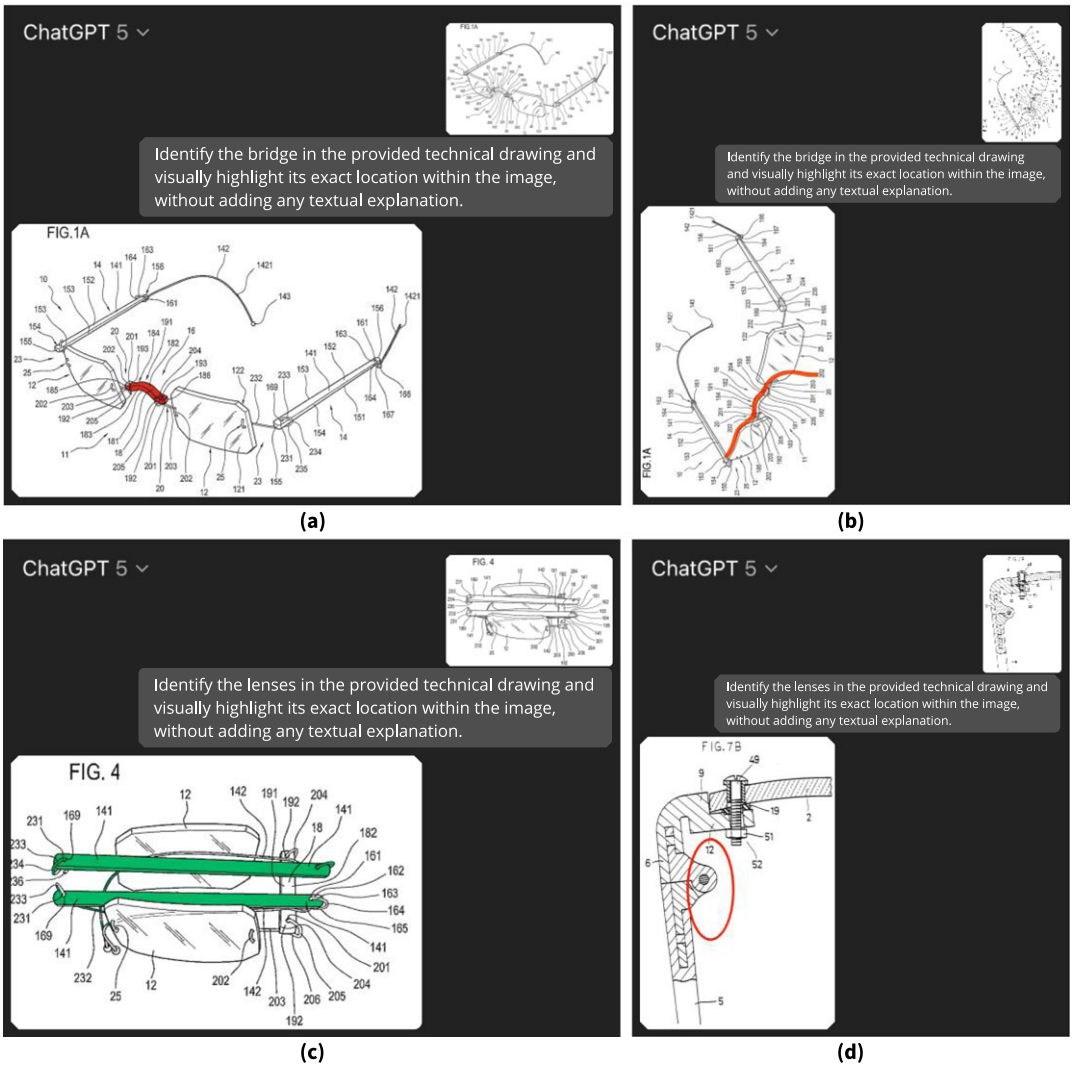


Figure 1. Examples of ChatGPT-5 failures in the object detection task on patent drawings

3. Methodology

To address the limitations of existing approaches, we propose a novel pipeline for OD to identify and locate components illustrated in patent drawings of utility patents. The key novelty of our methodology is the adaptation of the Segment Anything Model (SAM) developed by Meta (Kirillov et al., 2023) for patent drawings, as illustrated in Figure 2.

SAM is a computer vision model specifically trained to locate arbitrary objects within an image. Specifically, SAM takes input points, namely segmentation points, that must be specified by the user to indicate regions of interest. These points guide and condition the model to draw precise object boundaries around those points. To provide the segmentation points of SAM for patent drawings, we first extract the component numbers that are included in patent drawings to denote individual components. This step is automatically performed using Optical Character Recognition (OCR), which enables the identification of text within images. Then we develop an algorithm named Follow-The-Arrow (FTA) algorithm to trace the arrows that typically extend from each component number to the exact position of the component within the drawing. The tip of each arrow is subsequently used as the segmentation points for SAM.

This approach for OD has several advantages. First, it eliminates the need for large, manually annotated datasets because it leverages (pre-)existing reference numbers and arrows in patent drawings as visual indicators for component identification and location, respectively. Second, unlike prompt-based VL models that necessitate explicit instructions such as "*locate the nose bridge in the image*", this approach does not require prior knowledge of which components to search for, making the pipeline adaptable across different technological domains. Finally, SAM does not produce a modified version of the input

https://doi.org/10.1017/pds.2026.10581 Published online by Cambridge University Press

image as in the case of VL models (see Figure 1); instead, it produces a segmentation mask over the input image that identifies which pixels correspond to specific components, enabling quantitative analysis and metric computation.

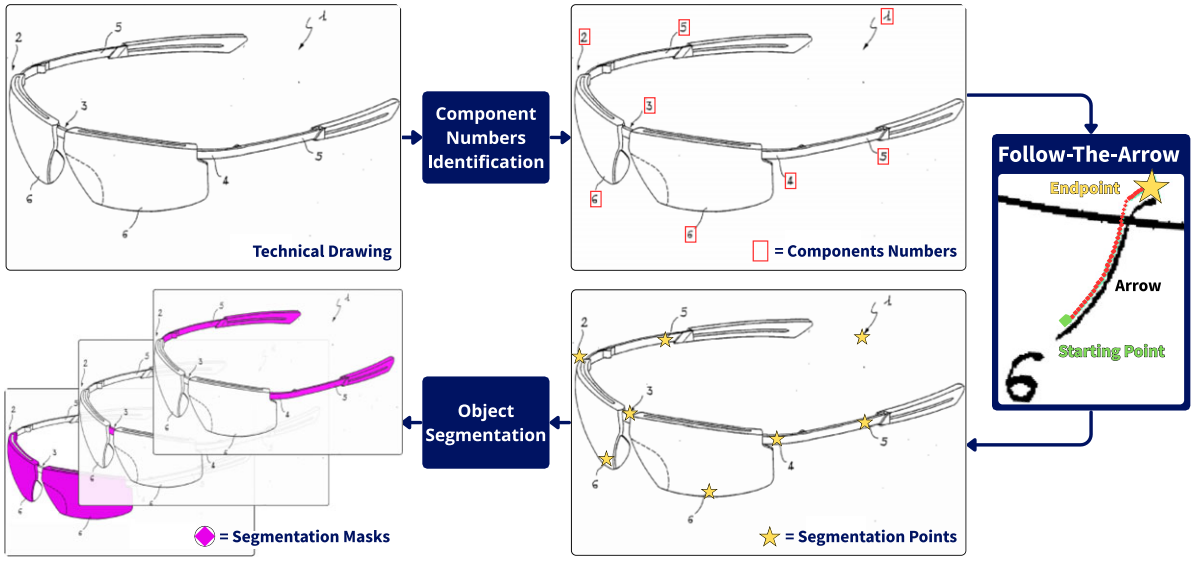


Figure 2. Methodology workflow

The following sections provides a detailed description of the methodology which consists of seven steps: (1) data collection, where patent drawings are collected; (2) image preprocessing, in which only technical drawings are retained while other types of patent drawings such as plots, flowcharts, and chemical formulas are excluded; (3) component numbers identification, where numeric identifiers corresponding to individual components are detected within each drawing; (4) FTA algorithm, where an algorithm traces arrows from each component number to determine segmentation points; (5) object segmentation, where the identified segmentation points are used as inputs to SAM to locate components within each drawing; (6) test dataset creation, where a manually annotated dataset is created to assess the performance of our pipeline; and (7) performance evaluation, in which quantitative metrics are defined to evaluate the outputs of steps (3), (4), and (5) against the test dataset.

3.1. Data collection

We retrieved patents from the PATSTAT database, maintained by the European Patent Office. Specifically, we extracted all eyeglass-related patents and their corresponding drawings using the query: *IPC = "G02C1" AND PUL = "en"*, where G02C1 refers to the International Patent Classification (IPC) category *"Assemblies of lenses with bridges or browbars"* while PUL = "en" restricts the results to English-language patents. The query yielded 535 patents, of which 322 included patent drawings, resulting in a total of 2,980 associated drawings. The G02C1 class was selected because eyeglasses are a relatively simple mechanical object with a limited number of components, facilitating the assessment of our OD pipeline.

3.2. Image preprocessing

To exclude non-technical drawings from further analysis, two PhD students with expertise in engineering design manually selected technical drawings from the collected patent drawings. The number of images that were identified as technical drawings and retained for subsequent analysis is 669.

Technical drawings of patents often contain multiple sub-images, which show the same object from different views such as front-view, isometric-view and cross-section. To standardize the input for the OD task and ensure that performance was evaluated independently of the number of sub-images in each technical drawing, we cropped individual sub-images from the collected technical drawings. In total, 1,223 single-image technical drawings were cropped for subsequent analysis by drawing the contours of sub-images using bounding boxes. We decided to perform these two steps manually for two main reasons. First, we aimed to evaluate the OD pipeline on a golden dataset, avoiding the propagation of

https://doi.org/10.1017/pds.2026.10581 Published online by Cambridge University Press

errors from these preliminary steps. Second, both the preprocessing steps are well-established in patent literature and are not the focus of this study. For reference, see the work of Jiang et al. (2021) for the classification of patent drawings, and Joshua et al. (2023) for the segmentation of sub-images.

3.3. Component numbers identification

Technical drawings typically include component numbers which are numerical identifiers assigned to specific parts of an invention. These identifiers are referenced in the accompanying patent text to uniquely denote individual components. As illustrated in Figure 2, we identified component numbers using DocTR, a state-of-the-art Optical Character Recognition (OCR) model. OCR is a technique that automatically detects and converts text contained in images or scanned documents into machine-readable form. In our case, DocTR outputs both the coordinates of each component number within the drawings and its recognized numerical label. The coordinates are then used by the FTA algorithm to locate the starting points of arrows, while the numerical labels are used to match each component number with its corresponding description in the patent text.

3.4. Follow-The-Arrow (FTA)

Technical drawings in utility patents typically include arrows that link component numbers to the corresponding parts of the drawing. These arrows serve as visual cues that indicate which parts of the drawing correspond to the components described in the patent text, establishing an explicit relationship between text and image. By detecting the endpoints (tips) of the arrows, we can determine the positions of the individual components within the technical drawings. Based on the idea proposed by Chen et al., 2015, we developed the Follow-The-Arrow (FTA) algorithm, a method designed to automatically trace arrows and identify their endpoints, as illustrated in Figure 2.

The algorithm proceeds in three steps. **(i) identify the starting points:** to identify the starting points of the arrows, the algorithm begins with the coordinates of the component numbers obtained during the component number identification step using DocTR. Around each component number, the algorithm incrementally traces concentric circles, searching for non-white pixels that may belong to an arrow. For expert readers, it seeks pixels with a non-zero gradient. Once such a pixel is detected, it is assumed to be part of an arrow, and its coordinates are recorded as the starting point (shown as green point in Figure 2); **(ii) follow the arrow:** to trace each arrow, the algorithm proceeds step by step from the previously identified starting point, following the connected black pixels that form the arrow body. Mathematically, this process involves iteratively moving in a direction perpendicular to the local image gradient, reconstructing the arrow's line pixel by pixel (shown as red points in Figure 2). To prevent deviations when arrows intersect other lines in the drawing, a control mechanism was implemented. It tracks the last W movement directions and, if the next direction deviates significantly from their average, the algorithm continues along the average direction; and **(iii) identify the endpoints**: to identify the arrow endpoints, the algorithm relies on the empirical observation that endpoint regions typically exhibit low local variance in pixel intensity. In technical drawings, these regions are generally uniformly white, as arrow tips often terminate on the solid (white) areas representing component bodies. When the algorithm reaches such a region, it records the corresponding coordinates as the arrow's endpoint (indicated by a star in Figure 2). The detected endpoints are then used as segmentation points to guide the subsequent object segmentation step.

3.5. Object segmentation

In this step, we employed the Segment Anything Model (SAM) (Kirillov et al., 2023). For reference, see the official project page at https://segment-anything.com/. SAM is a model specifically trained to segment arbitrary objects within an image. Specifically, SAM takes input points, referred to here as segmentation points, that are provided by the user to indicate regions of interest within an image. These points guide and condition the model to create segmentation masks around segmentation points. Each segmentation mask defines the set of pixels of the input image corresponding to an individual object. In this work, we adapted this approach to segment the components within technical drawings. As illustrated in Figure 2, we used the endpoints extracted by the FTA algorithm as segmentation points for SAM. For each segmentation point in the input image, SAM produced a segmentation mask containing the pixels of the input technical drawing corresponding to an individual component.

https://doi.org/10.1017/pds.2026.10581 Published online by Cambridge University Press

Since the SAM model was originally trained on color images, we hypothesized that its performance might be limited when applied to black-and-white technical drawings. To examine this effect, we automatically colorized technical drawings using different generative AI models: Komiko, Colorize Diffusion, Next Diffusion, and SORA. Among them, OpenAI's SORA model (https://sora.chatgpt.com) demonstrated strong capabilities in colorizing technical drawings, without modifying the input image. Therefore, each technical drawing in our dataset was colorized using SORA with the following textual prompt: "*Color the attached technical drawing by adding color strictly within the existing outlines. Keep all lines, shapes, proportions, and structural elements exactly as they are in the original image. Do not add or modify any elements. The background must remain completely white, with no gradients or textures.*" This allowed us to evaluate SAM's segmentation performance on both the original black-and-white drawings and their colorized versions.

3.6. Test dataset creation

For the evaluation of our object detection (OD) pipeline, we created a manually annotated test dataset to serve as ground truth. A random sample of 120 images was selected from the 1,223 technical drawings to keep the annotation workload manageable and ensure that annotations were completed carefully and consistently. We provide open access to the dataset at https://github.com/gabrielemarino-gm/technical-drawings-object-detection-data.

The annotation process was conducted by two PhD students with expertise in engineering design and patent analysis. The two annotators were instructed to perform three annotation tasks. **Component Number Identification**: annotators identified all component numbers within the technical drawings, recording their corresponding numeric identifiers. This task was designed to evaluate the performance of the DocTR model. A total of 1,643 component numbers were annotated. **FTA:** annotators marked the endpoints of all arrows present in each image. This task aimed to generate a ground truth dataset for assessing the output of the FTA algorithm. A total of 1,643 endpoints were annotated. **Object Segmentation**: annotators delineated the boundaries of each component by manually placing a sequence of points along its perimeter, producing closed polygons that precisely define the set of pixels corresponding to each component and serve as ground-truth segmentation mask for evaluating the segmentations generated by the SAM model. This annotation task was performed using the CVAT tool (https://www.cvat.ai/) and a total of 1,643 distinct components were annotated.

Creating this test dataset was one of the most demanding and time-consuming aspects of the study because annotators had to read and interpret both the drawings and the accompanying technical text to infer which image parts were referenced by the arrows. In addition, placing multiple points along each component's contour to isolate individual components was both cognitively demanding and time-consuming, as annotators had to carefully distinguish overlapping and densely intricated lines. Overall, this process required approximately 66.1 hours to complete (about 30 minutes per image), demonstrating that component-level annotation is impractical to perform at scale for AI training.

3.7. Performance evaluation

Component Number Identification: to evaluate the performance of the component number identification step, we compared the component numbers detected by DocTR with their corresponding ground-truth annotations in the test set. The accuracy of DocTR was computed as the percentage of correctly identified component numbers relative to the total number of component numbers in the test set (1,643). An accuracy of 100% corresponds to perfect identification of all component numbers, while lower percentages reflect a greater number of missed or incorrectly recognized components. **FTA:** to evaluate the performance of the FTA algorithm, we compared its predicted endpoints with ground-truth annotations using a distance-based metric ranging from 0 to 1, as defined by Pascual-Marqui et al. (2013). The metric is defined as $e^{-\lambda d}$,where d is the Euclidean distance (in pixel) between an FTA-predicted and its corresponding ground-truth endpoint, and λ is a decay parameter controlling sensitivity to distance. The decay parameter λ was empirically set to 0.002. This formulation prevents excessively large pixel distances from disproportionately lowering the metric. Moreover, it preserves the intuitive interpretation of accuracy: the metric equals 1 when the two endpoints coincide ($d = 0$) and decays exponentially toward 0 as their distance increases. The overall accuracy of the FTA algorithm was computed by averaging the metric values across all the endpoints in the test dataset. **Object Segmentation:** to evaluate object segmentation step of SAM, we computed the Intersection

https://doi.org/10.1017/pds.2026.10581 Published online by Cambridge University Press

over Union (IoU) metric. IoU is a standard evaluation metric for OD tasks, quantifying the degree of overlap between two regions relative to their combined area. The IoU is defined as $IoU = |A \cap B|/|A \cup B|$, where A is the ground-truth region and B is the predicted segmentation mask. In this context, it quantifies the overlap between the manually annotated area of a component and the corresponding segmentation mask produced by the SAM model. This metric ranges from 0 to 1 and quantifies segmentation accuracy; higher values indicate more precise segmentations, with a value of 1 representing perfect overlap between the predicted and ground-truth regions. To evaluate the overall performance of the object segmentation step, we computed the mean Intersection over Union (mIoU) across all the annotated components.

4. Results

4.1. Component numbers identification

DocTR successfully detected 1,093 component numbers out of 1,643, achieving an overall accuracy of 67%. This indicates that the model correctly identifies the majority of components present in the technical drawings. The detected component numbers can then be matched with their corresponding mentions in the patent text, allowing the retrieval of detailed component descriptions. This represents a first-level outcome of our pipeline for DDD, enabling designers to analysis both visual representations of components along with their functional and descriptive information.

4.2. FTA

The FTA algorithm achieves an overall accuracy of 0.78, demonstrating its effectiveness in tracing arrows and showing that most predicted arrow endpoints closely match the ground truth. However, a standard deviation of 0.21 suggests variability across predictions. The main errors observed in FTA algorithm's predictions include: (1) inaccurate detection of starting points caused by noise around component numbers that misleads the radial search; (2) difficulty in following arrows with atypical shapes, where sharp angles confuse the tracing process and restrict deviation from the intended path; (3) errors in detecting endpoints when the arrow tips fall in regions that are not uniformly white or low in pixel variance, causing the FTA to misidentify the true endpoint and continue moving forward past it.

4.3. Object segmentation

A qualitative comparison of the segmentation results under both experimental conditions—on the original black-and-white technical drawings and on the colorized versions—is presented in Figure 3. Panel (a) shows the human-annotated ground-truth mask of the lens components across two different eyeglasses configurations; (b) shows the SAM-predicted mask using the original black-and-white technical drawing and FTA-predicted endpoints; (c) shows the SORA-colorized version of the original drawing; and (d) shows the SAM-predicted mask generated using the colorized image and the FTA-predicted endpoints. Both the drawings in panels (b) and (d) present yellow stars indicating the endpoints of the lens components identified by the FTA and provided to SAM as segmentation points. Panels b) and d) show that segmentation accuracy greatly varies between the two settings. In panel (b), SAM fails to segment lens components, achieving IoU values of 0.436 and 0.385 for the upper and lower drawings, respectively. In contrast, panel (d) shows a significant improvement in segmentation performance when SAM receives the same segmentation points applied to the colorized versions of the drawings, yielding IoU values of 0.961 and 0.810.

Figure 3 presents two examples in which our pipeline successfully segments the lens components, illustrating the beneficial effect of color on the SAM's performance. We also performed a quantitative evaluation of the SAM's performance across all components in the test dataset. Note that the total number of unique components, after accounting for DocTR's errors, is 869. This is because components sharing the same component number (e.g., right lens and left lens) are treated as identical instances. Table 1 presents the mIoU and its standard deviation (std) values for different component groups. For all components, including nose pads, hinge screws, and lens clips (n=869), the mIoU is 0.135 without colors and 0.150 with colors. For the main components, such as the frame, lenses, bridge, and temples (n=75), the mIoU is 0.230 without colors and 0.375 with colors.

https://doi.org/10.1017/pds.2026.10581 Published online by Cambridge University Press

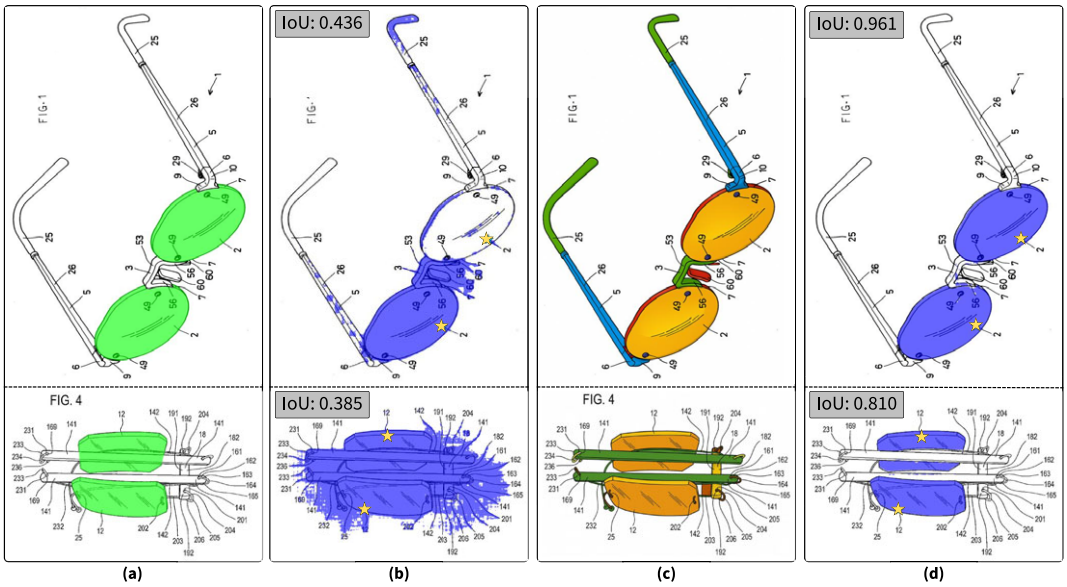


Figure 3. Comparison of segmentation results: a) ground-truth masks; b) SAM segmentation masks without colours; c) SORA-colorized drawings and d) SAM segmentation masks with colours

Table 1. Object segmentation performance of SAM

Components (n)	Example	mIoU (std) without colors	mIoU (std) with colors
all components (869)	nose pads, hinge screws, lens clip, etc.	0.135 (0.230)	0.150 (0.265)
main components (75)	frame, lenses, bridge, temples.	0.230 (0.264)	0.375 (0.351)

To validate the impact of color on SAM's performance, we analyzed whether the distributions of the mIoU metrics differ significantly between the with-colors and without-colors images. As the metric distributions were found to be non-normal, we applied the non-parametric Mann–Whitney U test with a 0.05 confidence level. The results showed p-values below 0.05 for the main components, indicating that the two mIoU distributions differ significantly. In contrast, across all components, color was found to have little influence on segmentation performance. This happens because the SORA model correctly colors the main components but fails to assign distinct colors to smaller sub-components, such as pivots or hinges, limiting the potential benefit of color information.

Although the SAM model achieved accurate segmentation in certain cases (see Figure 3, panel d) and the colorization step offered some benefits, its overall performance remained low, as indicated by the poor mIoU values in Table 1. Through manual inspection of both successful and failed cases, we identified three main sources of error: (1) SAM model is highly sensitive to the placement of segmentation points; even slight variations in their location can substantially affect the resulting segmentation masks. This issue is particularly evident in technical drawings containing small or closely nested components, such as bolts and pivots, leading to a significant reduction in mIoU; (2) in many cases, arrows indicate components from the side rather than the center; using their endpoints as segmentation points can therefore produce misaligned segmentation masks; (3) since SAM was trained on natural images, when applied to technical drawings, it often struggles with complex shapes or intricated lines, producing fragmented or scattered segmentation masks.

5. Conclusions and future developments

This work presents a novel pipeline to perform OD on technical drawings. Current AI models show some potential but also limited technological readiness to analyze technical drawings beyond basic image classification and similarity tasks. This reveals a critical technological gap in the field of DDD, as fine-grained analysis of technical drawings is not yet capable of supporting design decision-making. To address this gap, we propose a novel approach that does not require data annotation and leverages existing component numbers and arrows in patent drawings to perform OD for arbitrary components.

https://doi.org/10.1017/pds.2026.10581 Published online by Cambridge University Press

Moreover, we proposed a novel approach to colorizing patent drawings, which can aid in visually distinguishing and understanding component boundaries. This may support engineers in interface discovery, morphological analysis of component shapes, and image-to-CAD modeling. With this preliminary study we also aim to stimulate further research and offer potential pathways for advancing automated analysis of technical drawings. In future work, we plan to carry out a case study to assess the impact of our OD pipeline on design tasks. We also aim to examine SAM's sensitivity to segmentation points by employing point clusters to create segmentation masks and exploring colorization techniques for direct component segmentation.

References

Bertoni, A. (2020, May). Data-driven design in concept development: systematic review and missed opportunities. In *Proceedings of the Design Society: DESIGN Conference* (Vol. 1, pp. 101–110). Cambridge University Press. https://doi.org/10.1017/dsd.2020.4

Cantamessa, M., Montagna, F., Altavilla, S., & Casagrande-Seretti, A. (2020). Data-driven design: the new challenges of digitalization on product design and development. *Design Science*, 6, e27. https://doi.org/10.1017/dsj.2020.25

Chen, J., Li, M., Jin, Q., Bao, S., Su, Z., & Yu, Y. (2015). Lead curve detection in drawings with complex cross-points. *Neurocomputing*, 168, 35–46. https://doi.org/10.1016/j.neucom.2015.06.018

Chiarello, F., Belingheri, P., & Fantoni, G. (2021). Data science for engineering design: State of the art and future directions. *Computers in Industry*, 129, 103447. https://doi.org/10.1016/j.compind.2021.103447

Consoloni, M., Giordano, V., Galatolo, F. A., Cimino, M. G. C. A., & Fantoni, G. (2025). Uncovering the limits of visual-language models in engineering knowledge representation. *Proceedings of the Design Society*, 5, 3261–3270. https://doi.org/10.1017/pds.2025.10340

Doris, A. C., Grandi, D., Tomich, R., Alam, M. F., Ataei, M., Cheong, H., & Ahmed, F. (2025). Designqa: A multimodal benchmark for evaluating large language models' understanding of engineering documentation. *Journal of Computing and Information Science in Engineering*, 25(2), 021009. https://doi.org/10.1115/1.4067333

Higuchi, K., & Yanai, K. (2023). Patent image retrieval using transformer-based deep metric learning. *World Patent Information*, 74, 102217. https://doi.org/10.1016/j.wpi.2023.102217

Jiang, S., Luo, J., Ruiz-Pava, G., Hu, J., & Magee, C. L. (2021). Deriving design feature vectors for patent images using convolutional neural networks. *Journal of Mechanical Design*, 143(6), 061405. https://doi.org/10.1115/1.4049214

Johnson, J., Hurst, A., & Safayeni, F. (2023). Managing data-driven design: A survey of the literature and future directions. *Proceedings of the Design Society*, 3, 2525–2534. https://doi.org/10.1017/pds.2023.253

Joshua, M. J., Ragav, V., & Ibrahim, S. S. (2023, July). Advanced knowledge extraction of physical design drawings, translation and conversion to cad formats using deep learning. In *International Conference on Creative Technology* (pp. 343–356). Cham: Springer Nature Switzerland.https://doi.org/10.1007/978-3-031-59164-8_28

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment Anything. https://doi.org/10.48550/arXiv.2304.02643

Lin, W., Yu, W., & Xiao, R. (2023). Measuring Patent Similarity Based on Text Mining and Image Recognition. *Systems*, 11(6), 294.https://doi.org/10.3390/systems11060294

Pascual-Marqui, R. D., Lehmann, D., Kochi, K., Kinoshita, T., & Yamada, N. (2013). A measure of association between vectors based on "similarity covariance". arXiv preprint arXiv:1301.4291. https://doi.org/10.48550/arXiv.1301.4291

Picard, C., Edwards, K. M., Doris, A. C., Man, B., Giannone, G., Alam, M. F., & Ahmed, F. (2025). From concept to manufacturing: Evaluating vision-language models for engineering design. *Artificial Intelligence Review*, 58(9), 288. https://doi.org/10.1007/s10462-025-11290-y

Quiñones-Gómez, J. C., Mor, E., & Chacón, J. (2025). Data-driven design in the design process: A systematic literature review on challenges and opportunities. *International Journal of Human–Computer Interaction*, 41(4), 2227–2252. https://doi.org/10.1080/10447318.2024.2318060

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788). https://doi.org/10.1109/CVPR.2016.91

Ruiz, P. P., & Maier, A. (2017, August). Data-driven engineering design research: opportunities using open data. In *21st International Conference on Engineering Design (ICED17)* (pp. 41–50).

Trauer, J., Schweigert-Recksiek, S., Onuma Okamoto, L., Spreitzer, K., Mörtl, M., & Zimmermann, M. (2020). Data-driven engineering–definitions and insights from an industrial case study for a new approach in technical product development. DS 101: Proceedings of NordDesign 2020, Lyngby, Denmark, 12th-14th August 2020, 1-12. https://doi.org/10.35199/NORDDESIGN2020.46

Vrochidis, S., Moumtzidou, A., & Kompatsiaris, I. (2012). Concept-based patent image retrieval. *World Patent Information*, 34(4), 292–303. http://doi.org/10.1016/j.wpi.2012.07.002

https://doi.org/10.1017/pds.2026.10581 Published online by Cambridge University Press