A Web-Based Platform for Interactive Comparison of Neural Network Image Segmentation Models

Viktoriia Vovchenko, Sergej Schultenkämper, and Frederik Simon Bäumer
Bielefeld University of Applied Sciences and Arts, Applied AI Working Group
Bielefeld, Germany
email: {viktoriia.vovchenko|sergej.schultenkaemper|frederik.baeumer}@hsbi.de

***Abstract*—Selecting an appropriate semantic segmentation model for a given application domain remains a challenging and time-consuming task for practitioners and researchers. This paper presents an interactive, web-based platform that enables side-by-side visual comparison of multiple neural network segmentation models applied to identical images. The system integrates three transformer-based segmentation models: a face-parsing network producing 19 semantic classes, a SegFormer-B3 clothing segmentation model with 18 classes, and a Mask2Former model for general-purpose scene segmentation spanning 150 ADE20K categories. Key contributions include side-by-side evaluation of model outputs across multiple architectures and image categories, with real-time segment highlighting and a scalable inference caching system that enables model comparisons without requiring repeated graphics processing unit (GPU) computation. The platform organizes a curated dataset of images under a hierarchical category taxonomy, supporting structured evaluation across demographic and contextual variables. As a practical use case, the system is applied within the ADRIAN project to assist in verifying identity consistency across images through segmentation-based analysis. The platform thus contributes a specialized artificial intelligence (AI) tool for systematic segmentation and object detection model evaluation within media analysis pipelines, where selecting appropriate models is a recurring challenge across tasks from identity verification to content moderation. It is available under https://github.com/vika-v-v/neural-networks-for-image-segmentation and designed to lower the barrier for comparative model evaluation in applied computer vision workflows.**

Keywords-segmentation; model comparison; face parsing.

I. Introduction

Semantic image segmentation, the task of assigning a class label to every pixel in an image, has seen rapid progress driven by deep learning, with transformer-based architectures [1] in particular achieving state-of-the-art results on standard benchmarks [2]. However, the rapid growth of publicly available pre-trained segmentation models, each trained on different datasets and optimized for different domain assumptions, makes it increasingly difficult to identify the most suitable model for a given task without empirical comparison [3]. A model that excels on face parsing may perform poorly on general scene understanding, and vice versa. Practitioners must therefore evaluate candidate models against their target data distribution before committing to a deployment choice.

Existing evaluation workflows are fragmented: researchers typically run inference scripts locally, produce static result images, and compare them manually. This process is labor-intensive, not easily shareable, and does not support interactive exploration of individual segment classes. There is a clear need for tooling that centralizes comparative evaluation in an accessible, visual, and reproducible manner.

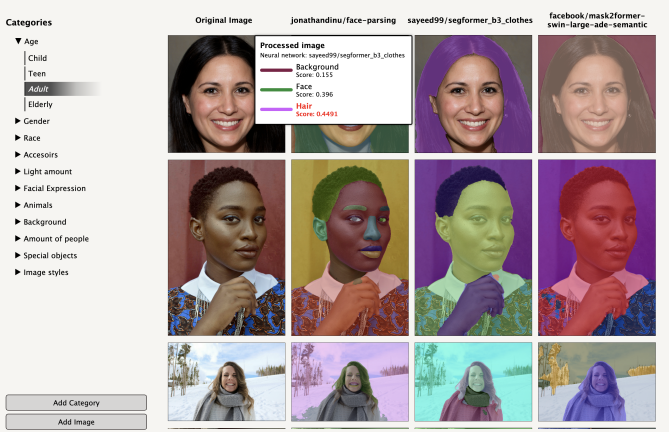

Figure 1. Platform showing a selected image compared across all three segmentation models, with a hovered segment highlighted and its class label indicated in the overlay panel.

This paper addresses that gap by presenting a web-based platform for the interactive comparison of segmentation models. The system allows a user to select an image from a curated, categorized dataset and immediately visualize the outputs of all three integrated models side by side on an interactive canvas, as shown in Figure 1. The three-column layout places each model's overlay next to the others for direct visual comparison; hovering over any region highlights the corresponding segment and displays all class labels, as well as the highlighted label and confidence score for the segments, if available, in the floating panel. Individual segments can be explored via this hover-based highlighting in real time. Additional segmentation models can be integrated with minimal effort and automatically appear alongside existing models for direct comparison.

The platform integrates three representative pre-trained models from the Hugging Face Hub [4]: a face-parsing model [5], a SegFormer-B3 clothing segmentation model [6], and Meta AI's Mask2Former model fine-tuned on ADE20K [7]. These three models cover complementary segmentation granularities (facial anatomy, clothing and body parts, and scene-level objects), making cross-model comparison particularly informative for tasks that involve human subjects.

The platform was developed in the context of the Authority-Dependent Risk Identification and Analysis in Online Net-

 ISBN: 978-1-68558-403-0

works (ADRIAN) project [8], which is dedicated to exploring and developing machine-learning-based methods for detecting potential threats to individuals based on online datasets and Digital Twins. Within ADRIAN, segmentation models are used to extract and compare structural face and body features across image pairs, supporting identity consistency verification. The platform described here provides a dedicated environment for evaluating and selecting segmentation models suited to this verification task.

The main contributions of this work are:

- An open, interactive web platform for side-by-side visual comparison of multiple semantic segmentation models on shared image inputs.
- An optimized canvas-based rendering pipeline with a downscaled label map enabling real-time hover-based segment inspection without redundant redraws.
- A batch segment retrieval endpoint and client-side cache that reduce Hypertext Transfer Protocol (HTTP) round-trips per category load from $N \times M$ (images $\times$ models) to two requests.
- An extensible system architecture supporting heterogeneous model runtimes (Node.js with Transformers.js and Python with Flask), unified under a common Representational State Transfer (REST) application programming interface (API) and result caching layer, allowing additional segmentation models to be integrated with minimal effort and compared directly against existing ones.
- A curated, hierarchically organized image dataset with demographic and contextual annotations, enabling structured evaluation across variables, such as age, gender, accessories, lighting, and background.

Collectively, these contributions address a gap in the tooling available for segmentation research, where the absence of interactive, multi-model evaluation environments forces practitioners into manual, non-reproducible comparison workflows.

This paper is structured as follows. Section II covers related work on segmentation models and evaluation tools. Section III describes the system architecture. Section IV details the integrated segmentation models. Section V presents the frontend and visualization design. Section VI characterizes the image dataset. Section VII presents two case studies. Section VIII addresses limitations and future work, followed by conclusions in Section IX.

II. Related Work

This section reviews prior work on semantic segmentation models and approaches to their evaluation and comparison.

A. Semantic Segmentation Models

Semantic segmentation has evolved from fully convolutional networks [9] and dilated convolution approaches, such as DeepLab [10], to encoder-decoder architectures [11] and, more recently, transformer-based models. The SegFormer architecture [2] introduced a hierarchical transformer encoder paired with a lightweight multilayer perceptron (MLP) decoder, achieving strong performance across multiple benchmarks while remaining computationally efficient. Several domain-specific adaptations of SegFormer have been published on Hugging Face [4], including models fine-tuned for face parsing [5] and clothing segmentation [6].

Mask2Former [7] advanced the state of the art by reformulating segmentation as a mask classification problem using a masked-attention Transformer decoder. Operating on multi-scale feature maps, Mask2Former achieves a mean Intersection over Union (mIoU) of 57.7 on ADE20K for semantic segmentation, a substantial improvement over the 37.4 mIoU reported for SegFormer-B0 [2], while simultaneously supporting instance and panoptic segmentation within a unified architecture.

Face parsing, the fine-grained labeling of facial regions, such as eyes, nose, mouth, and hair, has been studied using methods ranging from convolutional neural network (CNN)-based approaches [12] to graph-based models and, recently, transformer fine-tunes. Clothing segmentation has similarly progressed with the release of large datasets, such as Crowd Instance-level Human Parsing (CIHP) [13], enabling models to distinguish among dozens of garment and body-part classes.

B. Model Evaluation and Benchmarking Tools

While standard benchmarks, such as ADE20K [14], Cityscapes [15], and COCO-Stuff [16], provide quantitative leaderboards, they do not support interactive, image-level qualitative comparison. Gradio [17] and Streamlit are widely used for building model demos, but are designed primarily as single-model interfaces: side-by-side multi-model comparison, structured dataset organization, and persistent inference caching are not provided out of the box and require substantial custom implementation effort. FiftyOne [18] offers more advanced dataset management and quantitative model evaluation capabilities, including support for multiple prediction fields and dataset-level metrics, but targets dataset curation and quantitative evaluation workflows rather than interactive multi-model comparison in a ready-to-use web environment. Table I summarizes the key differences between these tools and the platform presented in this work.

The platform presented here addresses the identified gaps by combining multi-model side-by-side comparison, interactive segment inspection, a structured dataset taxonomy, and persistent inference caching in a self-hostable web environment.

III. System Architecture

The platform follows a three-tier architecture comprising a web frontend, a middleware layer, and a backend inference service, with a shared database for caching and metadata storage. Figure 2 illustrates the high-level system design: the Angular frontend communicates exclusively with the Node.js middleware, which routes inference requests to either the built-in face-parsing handler or the separate Python service, and the cache layer intercepts repeated requests before they reach the model runtimes. The architecture was designed to decouple model inference from presentation, ensuring that changes to either the model registry or the frontend impose no coupling

TABLE I. COMPARISON OF VISUALIZATION AND EVALUATION TOOLS

Feature	Gradio	Streamlit	FiftyOne	This work
Multi-model side-by-side	No	No	No	Yes
Pixel-level hover inspection	No	No	No	Yes
Structured dataset taxonomy	No	No	No	Yes
Persistent result caching	No	No	Yes	Yes
GPU-free runtime	No	No	Yes	Yes
Custom model integration	Yes	Yes	Yes	Yes
Managed cloud hosting	Yes	Yes	No	No
Domain-agnostic use	Yes	Yes	Yes	No

constraints on the other – a requirement for a platform intended to evolve alongside the model landscape. Consequently, integrating new segmentation architectures requires updates only to the Python service, leaving the middleware and user interface completely unmodified.

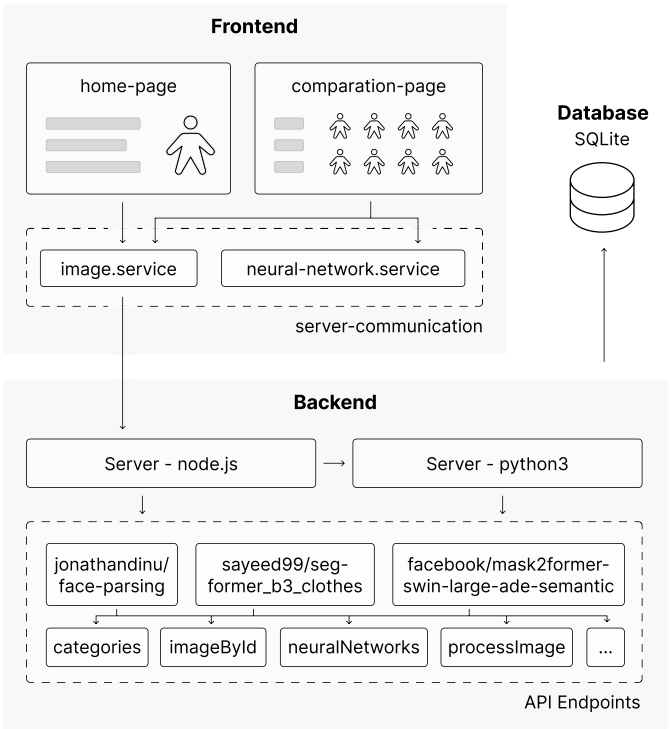

Figure 2. High-level system architecture. The frontend communicates with the backend through dedicated services. The backend routes inference requests to the appropriate model runtime and caches results.

A. Backend

The backend exposes a REST API covering image management, category retrieval, model listing, and inference. When a segmentation result is requested, the server first checks the cache; if a result already exists, it is returned immediately without re-running inference. Otherwise, the request is dispatched to the appropriate model handler. A concurrency guard prevents duplicate inference jobs from being launched for the same image and model simultaneously.

To reduce loading times when a user browses a category, the backend provides a batch endpoint that returns all cached segmentation data for every image in the category and every registered model in a single request. This reduces the number of required frontend requests from one per image-model pair to just two per category switch, regardless of how many images the category contains.

B. Inference Services

The face-parsing model runs directly within the Node.js backend process, requiring no separate service. The clothing and scene segmentation models are handled by a dedicated Python inference server, which loads both models at startup and keeps them in memory for fast repeated use. Each model returns per-segment results including the class label, a confidence score, and a colored mask overlay.

C. Offline Pre-computation

The platform includes a pre-computation script that processes all images in the dataset with all registered models and stores the results in the cache. Once this has been run, no inference is performed during normal use: all pages load results directly from the database, making the platform fast and suitable for deployment without GPU access at runtime.

D. Extensibility

Adding a new segmentation model requires registering it in the database and providing a handler that returns results in the standard format. No changes to the frontend or database schema are needed, keeping the integration cost low.

IV. INTEGRATED SEGMENTATION MODELS

This section presents the segmentation models integrated into the platform. Table II summarizes their key characteristics.

A. Face-Parsing Model

The `jonathandinu/face-parsing` [5] model is a SegFormer [2] fine-tuned for facial region segmentation. It produces 19 semantic classes covering detailed facial anatomy: background, skin, nose, left and right eyes, left and right eyebrows, left and right ears, mouth, upper lip, lower lip, hair, hat, left earring, right earring, necklace, neck, and clothing. This granularity makes it well suited to tasks requiring precise facial structure analysis, such as identity verification, face editing detection, and makeup transfer.

TABLE II. INTEGRATED SEGMENTATION MODELS

	jonathandinu/ face-parsing [5]	**sayeed99/seg-former_b3 _clothes** [6]	**facebook/mask 2former-swin -large-ade** [7]
Backbone	SegFormer	SegFormer-B3	Swin-L + Mask2Former
Domain	Facial anatomy	Clothing, body	General scenes
Classes	19	18	150

B. Clothing Segmentation Model

The `sayeed99/segformer_b3_clothes` [6] model uses a SegFormer-B3 [2] backbone fine-tuned for human parsing and clothing segmentation. Its 18 output classes include background, hat, hair, sunglasses, upper clothes, skirt, pants, dress, belt, left and right shoes, face, left and right legs, left and right arms, bag, and scarf. On the ATR dataset [19], the model achieves an mIoU of 69 [6]. This model is relevant for applications involving fashion analysis, person re-identification, and clothing-aware image retrieval.

C. General Scene Segmentation Model

The `facebook/mask2former-swin-large-ade-semantic` [7] model implements the Mask2Former architecture with a Swin-Large backbone [20], fine-tuned on the ADE20K dataset [14] covering 150 semantic categories. Mask2Former reformulates segmentation as a mask classification problem using a masked-attention Transformer decoder operating on multi-scale feature maps. On ADE20K, the model achieves 57.7 mIoU [7]. Its broad categorical coverage, spanning structural elements (wall, floor, ceiling), natural objects (tree, plant, water), and human-made objects (chair, table, car), makes it useful for understanding scene context surrounding subjects, complementing the face- and clothing-specific models.

V. FRONTEND AND VISUALIZATION

The frontend is implemented in Angular 17 [21] and organized into a comparison page (Figure 1) and a landing page. The comparison page is the primary workspace, comprising an image and category selector, a comparison section, and a floating segment information panel.

A. Canvas-Based Rendering

Segmentation results are rendered on Hyperext Markup Language 5 (HTML5) canvas elements using a layered compositing approach: the original image is drawn first, followed by semi-transparent segment masks overlaid in sequence, allowing the original image texture to remain visible.

Original images and segment overlays are loaded concurrently: the original image begins loading immediately when the component is initialized, in parallel with the asynchronous segment data fetch. This ensures that the base image is displayed as soon as it arrives from the network, without waiting for inference results.

B. Optimized Hover Interaction

Efficient real-time segment inspection is central to the platform's utility as an evaluation tool, as it allows researchers to directly probe model behavior at the pixel level without interruption. The key usability feature enabling this is segment-level hover highlighting: as the cursor moves over the canvas, the segment under the pointer is isolated and displayed with full opacity while all other segments are hidden. Implementing this interactively requires determining which segment covers each canvas pixel.

To support interactive frame rates, the system precomputes a *label map* – a downscaled integer array encoding the topmost segment index at each pixel – once per model result. On each `mousemove` event, the segment under the pointer is resolved in O(1) time from this map rather than by querying the rendered canvas, keeping hover interaction smooth even for the 150-class ADE20K model. Redundant redraws are suppressed when the cursor remains within the same segment, and canvas updates are deferred to the browser's display refresh rate.

Upon identifying the hovered segment, the component redraws the canvas by compositing only the original image and the highlighted mask, and updates the floating information panel with the segment's class label and confidence score.

C. Category-Level Data Loading

When the user switches to a new category, the frontend issues two parallel requests: one for the image list and one for all pre-computed segment data for that category in a single batch response. This replaces the $N \times M$ per-image-per-model fetch pattern with just two requests per category switch, regardless of dataset size.

D. Image and Category Management

The platform supports uploading new images via a multi-part form. Figure 3 shows the first step of this process, where the user provides the image Uniform Resource Locator (URL) and a display name. In subsequent steps, the image can be assigned to one or more undercategories in the hierarchy, and metadata including image origin, tags, and notes can be attached. Tags are intended to support a future search feature that will allow users to filter images across categories by arbitrary keywords. Existing images can be re-categorized or deleted. The category browser on the comparison page exposes the full two-level taxonomy, allowing users to filter the image gallery to a specific demographic or contextual subset before selecting images for comparison.

E. Homepage Animation

The landing page features an animated carousel that cycles through randomly selected segments from a pre-designated image using fade transitions. Each segment is displayed for a fixed interval while the next is preloaded, providing an illustrative preview of the platform's segmentation capabilities without requiring user interaction.

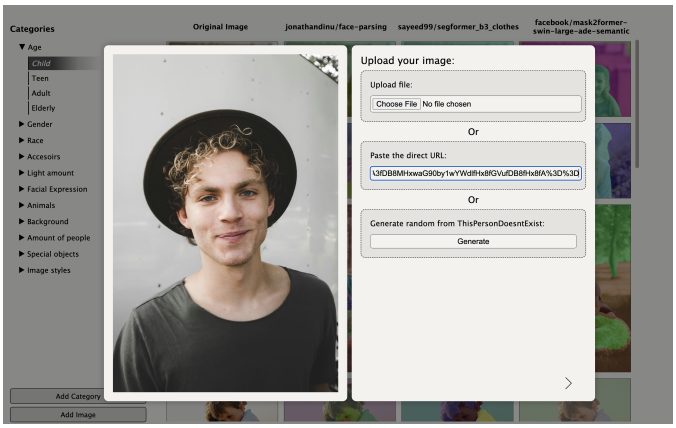

Figure 3. First step of the image upload dialog. After providing the image URL and name, the user proceeds to assign the image to categories.

VI. Image Dataset and Taxonomy

The platform ships with a curated set of approximately 120 images pre-loaded from Unsplash [22], a source of openly licensed high-resolution photographs. The images were selected to cover a diverse range of subjects relevant to face and body segmentation tasks, including variation in age, gender, ethnicity, lighting conditions, accessories and backgrounds.

Images are organized under an 11-category, two-level taxonomy. The top-level categories and selected undercategories are shown in Table III. Each image can be associated with multiple undercategories, enabling cross-cutting queries, such as retrieving images depicting elderly subjects wearing glasses.

TABLE III. Image Taxonomy (Top-Level Categories and Selected Undercategories)

Category	Selected Subcategories
Age	Child, Teen, Adult, Elderly
Gender	Male, Female, Other
Ethnicity	Asian, Black, Latino/Hispanic, White
Accessories	Hat, Glasses, Scarf, Medical mask
Lighting	Natural, Indoor, Low light, Backlit
Facial Expression	Neutral, Smiling, Angry, Surprised
Animals	Dog, Cat, Bird, Other
Background	Indoor, Outdoor, Plain, Blurred
Number of People	Single, Pair, Group
Special Objects	Phone, Weapon, Sign, Vehicle
Image Style	Photo, Painting, Illustration, Statue

This taxonomy serves two purposes. First, it allows structured, controlled comparison of model behavior across demographic subgroups: for example, assessing whether the face-parsing model consistently segments ear regions across different skin tones. Second, it enables evaluation of model robustness to domain shift, such as testing the scene segmentation model on paintings versus photographs. Such controlled, variable-specific evaluation conditions are a standard methodological requirement in empirical machine learning research, yet are rarely provided by existing model comparison tools out of the box.

Category labels were assigned by a single annotator based on the visual content of each image. Importantly, labels describe perceived image characteristics rather than verified attributes of the individuals depicted: an image is categorized, not a person. For example, an image assigned to the *Elderly* undercategory reflects the apparent age of the subject as visible in the photograph, not any verified demographic information. Images where the relevant attribute was visually ambiguous were excluded where possible; in edge cases, assignment was based on best judgment. This approach is consistent with the illustrative purpose of the dataset, which is intended to provide varied examples for model comparison rather than to serve as a ground-truth demographic resource.

VII. Case Studies

This section illustrates two representative use cases of the platform: one grounded in an ongoing research project, and one describing a researcher integrating and evaluating a custom model.

A. Model Selection for Identity Verification (ADRIAN Project)

Within the ADRIAN project, the platform was used to select a segmentation model suitable for identity consistency verification across image pairs. Researchers loaded a set of face images spanning multiple demographic categories — varying in age, lighting, and accessories — and compared the outputs of all three integrated models side by side. The face-parsing model was identified as the most appropriate choice: its 19-class facial anatomy segmentation provided the structural granularity needed for comparing facial regions across image pairs, while the clothing segmentation model complemented it for body-level features. The Mask2Former model, despite its broad coverage, was found less suitable for fine-grained face-level analysis. The platform's demographic taxonomy enabled verification that the selected model produced consistent results across subgroups, supporting confidence in the selection before integration into the ADRIAN pipeline. A conventional workflow — running inference scripts locally and comparing static output images manually — would not have supported this structured cross-subgroup inspection without significant additional scripting effort. The interactive hover-based exploration further allowed researchers to inspect ambiguous boundary regions between adjacent facial segments directly in the browser, an insight that would have been difficult to obtain from static result images alone.

B. Benchmarking a Custom Segmentation Model

A second representative use case is a researcher developing a custom domain-specific segmentation model who wishes to situate its performance within the existing model landscape. The researcher uploads a set of domain-relevant

images through the platform's image management interface and registers the custom model by providing a handler that returns results in the standard format. Without any further changes to the frontend or database schema, the custom model appears alongside the existing models for direct side-by-side comparison. Hover-based segment inspection allows the researcher to visually examine regions where the custom model's output differs from the baselines — for example, areas that appear over-segmented or assigned to visually implausible classes — enabling targeted qualitative diagnosis. The structured taxonomy supports controlled evaluation across image variables, such as lighting or background, replacing a previously manual, script-based comparison workflow. An additional practical benefit arises in presentation contexts: by running the offline pre-computation script before the presentation, all inference results are stored in the cache, allowing the researcher to navigate smoothly between images and models during a live demo without any loading delays or dependency on GPU availability at the time of the presentation.

VIII. Discussion and Limitations

This section discusses performance characteristics, limitations of the platform, and directions for future work. The observations reflect broader challenges inherent to building unified evaluation environments for heterogeneous neural network models and are intended to inform research in this area.

A. Performance Considerations

Inference latency varies substantially across the three models and depends on hardware availability. The face-parsing model, executed via Open Neural Network Exchange (ONNX) within the Node.js process, typically completes in under 2 seconds on a consumer central processing unit (CPU). The Python-based models benefit from GPU acceleration when available; on CPU alone, inference for the Mask2Former model can take 10-30 seconds for a standard image. The database caching layer and offline pre-computation script mitigate this for repeated queries: once all images have been pre-processed, the platform loads results directly from the database with no model inference at runtime. Planned future work includes controlled latency measurements across a range of hardware configurations (CPU-only, GPU-accelerated, and cloud-hosted deployments) and scalability tests under concurrent multi-user load, to provide a more rigorous characterization of the platform's performance envelope.

B. Model Coverage

The current release integrates three models covering face, clothing, and scene domains. Important segmentation domains, such as medical imaging, satellite imagery, and autonomous driving, are not yet represented. The extensible architecture was designed to make adding new models straightforward, and the model registry is planned to be expanded.

C. Quantitative Evaluation

The platform currently focuses on qualitative, side-by-side visual comparison. It does not include ground-truth segmentation masks for the pre-loaded dataset and therefore cannot compute standard segmentation metrics, such as mIoU or per-class accuracy within the interface. This absence does limit the platform's usefulness as a rigorous standalone benchmark tool: it cannot replace formal evaluation on established benchmarks, such as ADE20K or Cityscapes, where ground-truth annotations and standardized metrics exist. However, rigorous benchmarking is not the platform's primary purpose. Its intended role is to support exploratory, domain-specific model selection — helping practitioners identify which model produces visually appropriate outputs on their target data before committing to a deployment choice. For this use case, qualitative comparison is both sufficient and practical, since standard benchmark scores often do not transfer reliably to application-specific image distributions. Incorporating a ground-truth annotation mode is a planned enhancement that would combine both functions, enabling quantitative metrics alongside visual inspection within the same interface.

Beyond model-level metrics, the platform has not yet been evaluated in terms of workflow efficiency: no user study has been conducted to measure whether it reduces the time or effort required for model selection compared to conventional script-based approaches. A controlled user study is planned as future work to quantify this benefit.

D. Scalability

SQLite is appropriate for single-user and small-team deployments but may become a bottleneck under concurrent multi-user load. Migrating the storage backend to a more scalable solution would be straightforward given the existing database abstraction layer.

E. Model Integration Heterogeneity

A practical challenge in building a unified comparison platform is that different segmentation models produce outputs in incompatible formats: they differ in how segments are represented, how confidence is reported, and whether masks are returned as dense label maps or as individual binary masks. The platform addresses this through a per-model adapter layer that normalizes all outputs to a common segment format before storage and display. This normalization cost is a recurring consideration when integrating new models and highlights a broader challenge in the field: the absence of a standardized output format for segmentation inference.

F. Ethical and Privacy Considerations

The image dataset is organized in part by demographic attributes, such as age, gender, and ethnicity. While this taxonomy enables structured evaluation of model behavior across subgroups (for example, assessing whether a face-parsing model performs consistently across different skin tones), it also raises important ethical questions. Categorizing people by such attributes requires careful consideration of

potential bias, fairness implications, and the risk of reinforcing stereotypes. All images in the platform are sourced from openly licensed repositories, and no personal data is stored. However, any extension of the dataset or use of the platform in contexts involving real user data should be conducted in accordance with applicable data protection regulations and ethical guidelines for research involving human subjects.

G. Dataset Scope and Limitations

The image dataset included in the platform comprises approximately 120 images and is intended as an illustrative collection to demonstrate the comparison workflow and provide ready-to-use examples across a range of image categories. At this scale, the dataset is sufficient for its intended purpose: populating the interface with varied examples that allow the comparison features to be exercised across demographic and contextual variables. It is, however, too small for comprehensive quantitative evaluation, where statistical reliability across subgroups typically requires substantially larger and more balanced samples. It is not designed or validated as a standalone benchmark dataset. As described in Section VI, labels were assigned by a single annotator based on perceived visual content, without formal annotation criteria or inter-annotator validation. The subjectivity inherent in categories, such as age, ethnicity, and gender – where assignment is based on appearance rather than verified identity – means that the labels carry uncertainty that is not quantified. This is an acknowledged limitation of the current dataset. The dataset can be used to compare model performance across the included images, but its focus on human subjects and faces limits the representativeness of comparisons for models targeting non-human or general-scene domains. For such cases, the platform's image upload functionality allows users to extend the dataset with domain-relevant images of their own.

IX. Conclusion and Future Work

A web-based platform for interactive, side-by-side comparison of semantic image segmentation models has been presented. The system integrates three transformer-based segmentation models spanning facial anatomy, clothing, and general scene understanding, and provides an intuitive canvas-based interface with real-time hover highlighting enabled by an efficient downscaled label map. A hierarchical image taxonomy supports structured evaluation across demographic and contextual variables, and a database caching layer with a batch retrieval endpoint ensures that inference results are reused across sessions and that category switches require only two HTTP requests regardless of dataset size.

The platform has been applied within the ADRIAN project as a practical tool for segmentation model selection in AI-manipulated image detection research. It is publicly available and designed with extensibility as a first-class concern, requiring minimal effort to integrate additional models.

By lowering the barrier to comparative segmentation evaluation – eliminating the need for local model installations and providing immediate visual feedback – the platform aims to accelerate the iterative model selection process in applied computer vision research. Future work will add ground-truth annotation support for quantitative mIoU evaluation, extend the model registry with additional domain-specific models, and explore real-time video segmentation as an interactive demonstration mode. Further planned enhancements include full-text search across categories and tags to improve dataset navigation, and the integration of AI-assisted image categorization to automate the assignment of demographic and contextual labels when new images are added.

Acknowledgment

This research is funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr. dtec.bw is funded by the European Union – NextGenerationEU.

References

[1] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy.

[2] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *Advances in Neural Information Processing Systems*, 2021. [Online]. Available: https://arxiv.org/abs/2105.15203.

[3] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3523–3542, 2022. DOI: 10.1109/TPAMI.2021.3059968.

[4] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020. DOI: 10.18653/v1/2020.emnlp-demos.6. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.6.

[5] J. Dinu, *Face parsing*, HuggingFace Model Hub, https://huggingface.co/jonathandinu/face-parsing [retrieved: May, 2026], 2023.

[6] S. Afridi, *Segformer b3 fine-tuned for clothes segmentation*, HuggingFace Model Hub, https://huggingface.co/sayeed99/segformer_b3_clothes [retrieved: May, 2026], 2023.

[7] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. DOI: 10.1109/CVPR52688.2022.00135. [Online]. Available: https://arxiv.org/abs/2112.01527.

[8] F. Bäumer, S. Denisov, M. Geierhos, and Y. S. Lee, "Towards authority-dependent risk identification and analysis in online networks," in *Proceedings of Artificial Intelligence, Machine Learning and Big Data for Hybrid Military Operations (AI4HMO)*, 2021. DOI: 10.14339/STO-MP-IST-190. [Online]. Available: https://www.hsbi.de/publikationsserver/record/2717.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. DOI: 10.1109/CVPR.2015.7298965. [Online]. Available: https://arxiv.org/abs/1411.4038.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018. DOI: 10.1109/TPAMI.2017.2699184.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015. DOI: 10.1007/978-3-319-24574-4_28. [Online]. Available: https://arxiv.org/abs/1505.04597.

[12] J. Lin, H. Yang, D. Chen, M. Zeng, F. Wen, and L. Yuan, "Face parsing with RoI tanh-warping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. DOI: 10.1109/CVPR.2019.00580. [Online]. Available: https://arxiv.org/abs/1906.01342.

[13] K. Gong, X. Liang, Y. Li, Y. Chen, M. Yang, and L. Lin, "Instance-level human parsing via part grouping network," in *Proceedings of the European Conference on Computer Vision*, 2018. DOI: 10.1007/978-3-030-01225-0_47. [Online]. Available: https://arxiv.org/abs/1808.00157.

[14] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5122–5130, 2017. DOI: 10.1109/CVPR.2017.544. [Online]. Available: https://api.semanticscholar.org/CorpusID:5636055.

[15] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. DOI: 10.1109/CVPR.2016.350. [Online]. Available: https://arxiv.org/abs/1604.01685.

[16] H. Caesar, J. Uijlings, and V. Ferrari, "COCO-Stuff: Thing and stuff classes in context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. DOI: 10.1109/CVPR.2018.00132. [Online]. Available: https://arxiv.org/abs/1612.03716.

[17] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, and J. Zou, *Gradio: Hassle-free sharing and testing of ml models in the wild*, [retrieved: May, 2026], 2019. DOI: 10.48550/arXiv.1906.02569. arXiv: 1906.02569 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1906.02569.

[18] Voxel51, *FiftyOne: The open-source tool for building high-quality datasets and computer vision models*, https://github.com/voxel51/fiftyone, [retrieved: May, 2026], 2020.

[19] X. Liang et al., "Human parsing with contextualized convolutional neural network," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015. DOI: 10.1109/ICCV.2015.163. [Online]. Available: https://ieeexplore.ieee.org/document/7410520.

[20] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. DOI: 10.1109/ICCV48922.2021.00986. [Online]. Available: https://arxiv.org/abs/2103.14030.

[21] Google, *Angular: The modern web developer's platform*, https://angular.io, [retrieved: May, 2026], 2023.

[22] Unsplash, *Unsplash: Beautiful free images & pictures*, https://unsplash.com, [retrieved: May, 2026], 2024.