PictographAI: Interactive Generation of Stylized Pictographs for Presentations

Sarah Makarem
Karlsruhe Institute of Technology
Karlsruhe, Germany
sarah.makarem@kit.edu

Tobias Röddiger
Karlsruhe Institute of Technology
Karlsruhe, Germany
tobias.roeddiger@kit.edu

Till Riedel
Karlsruhe Institute of Technology
Karlsruhe, Germany
till.riedel@kit.edu

Michael Beigl
Karlsruhe Institute of Technology
Karlsruhe, Germany
michael.beigl@kit.edu

Abstract

In today's data-driven world, effective data visualization is crucial for communication. Recent studies have shown that meaningful and relevant visual embellishments and decorations can significantly enhance data visualization memorability and comprehension. Hence, we introduce PictographAI, a generative tool integrated into presentation software to transform traditional bar charts into Pictographic visualizations. Utilizing a multimodal AI pipeline, PictographAI processes text, images and raw data from presentation slides to automatically generate contextually appropriate pictographs. Our pipeline uses an Large language model agent, a text-guided image-inpainting model, and algorithmic post-processing to make sense of the slide contents and generate pictographs. As users update their presentation slides, the AI pipeline automates the generation of new pictographs that represent the respective contents. In this work, we demonstrate the concept and working principle that motivate the system architecture and the generative AI pipeline on a bar chart generation use case that integrates into a presentation slide creation workflow.

CCS Concepts

• **Human-centered computing** → **Interactive systems and tools**; **Visualization systems and tools**.

Keywords

pictographs, generative AI, LLM agent, inpainting, visulization, image generation, presentations

1 Introduction and Related Work

The term "Chart Junk" was introduced by visualization expert Edward Tufte in his 1983 work [14] to describe unnecessary chart annotations and decorations that clutter the data presentation. Tufte advocated for presenting the data as clearly as possible, without any extra embellishments. In support of this view, some research has shown that plain, minimalist visualizations are typically easier to understand [5]. However, other research highlights the benefits of visual embellishments, including improved memorability [1, 4], recognition [3], and even better comprehension [10], particularly in contexts where viewers' attention could be divided, such as meetings and lectures [2].

We introduce PictographAI, a multimodal generative AI pipeline that integrates into the user's presentation creation workflow. This integration enables the automatic extraction of content from presentation slides as they are created. Using the extracted semantic information, PictographAI generates meaningful, topic-relevant pictographs with little effort.

Using generative AI, our demo aims to demonstrate that automatically generated visual embellishments can make presentations more effective, an effect that was previously shown for human-generated embellishments. Research by Borkin et al. [4] involving more than 2000 visualizations found that the inclusion of Human Recognizable objects significantly increases a visualization's memorability. Not only do the pictographs need to be recognizable, but also relevant. Pictographic visualizations studied by [7] that are relevant to the data visualization topic have been shown to improve data recall and engagement compared to conventional bar charts. Such embellishments, when they represent data metaphorically, do not qualify as chart junk [6].

Given the limited attention spans and cognitive resources available during presentations, pictographic visualizations offer an opportunity to enhance memorability and grab the viewer's attention. Despite their effectiveness, creating these pictographic representations required professional design skills, considerable time, and multiple iterations [8]. With PictographAI we demonstrate how generative AI technologies can bridge the skill gap, allowing presenters access to visual aids to deliver content more effectively, without any experience in graphic design.

With this demo, we seek to showcase how generative AI can ease graphics design by customizing contents of a presentation through pictographs that are potentially more engaging and memorable than conventional bar charts.

2 PictographAI

In the following sections, we first introduce the general concept and working principle of PictographAI. Then, we explain in details how the underlying AI pipeline generates recognizable, topic-relevant pictographs from user input.

2.1 Concept and Working Principle

PictographAI leverages large language as well as image generation stable diffusion models to process multimodal input in the form of: (i) user text input from the presentation slide; (ii) descriptive metadata of the raw input data provided; and (iii) image input of

Mensch und Computer 2024 – Workshopband, Gesellschaft für Informatik e.V., 01.-04. September 2024, Karlsruhe, Germany

https://doi.org/10.18420/muc2024-mci-demo-305

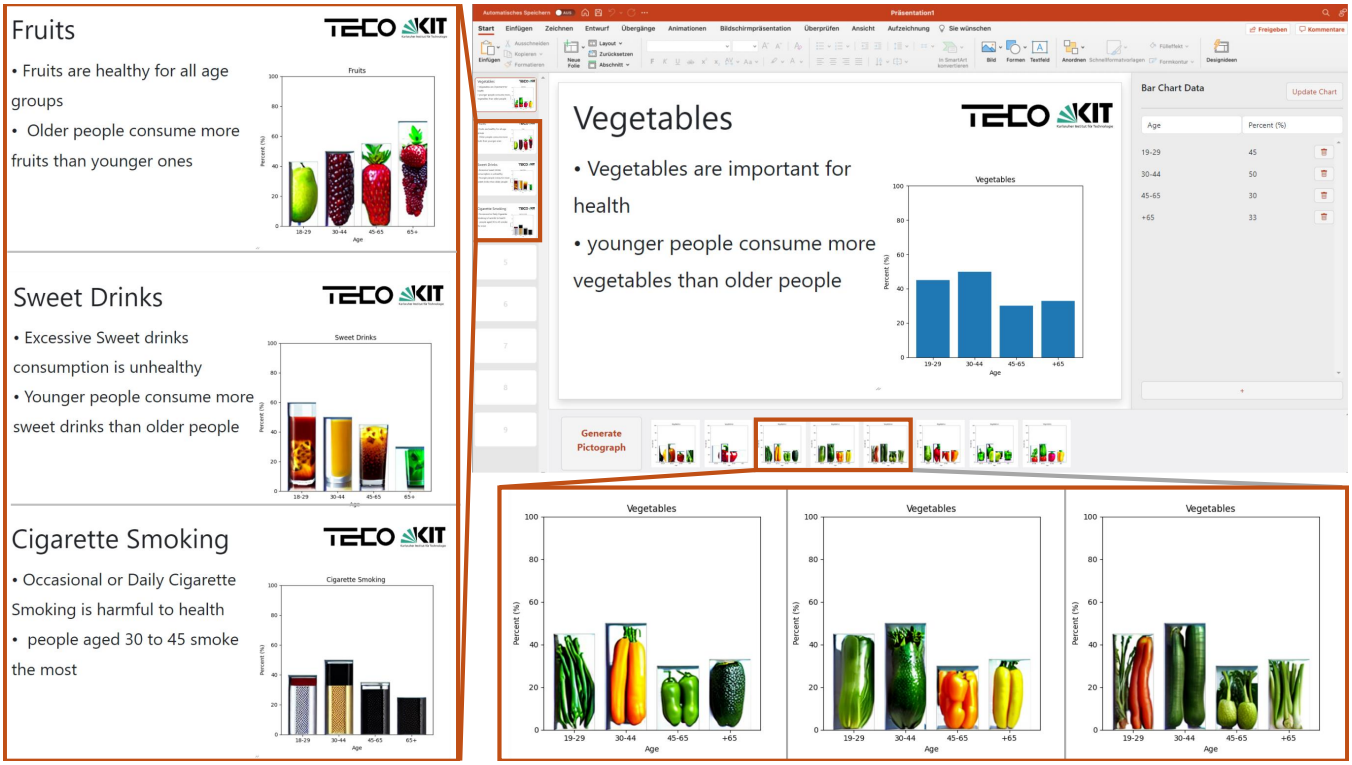

Figure 1: PictographAI user interface. Users can edit contents such as title and bullet list of their slides and the chart data. They can generate new pictographs by clicking the "Generate Pictograph" button and select an output to display in their presentation slide to replace the regular bar chart.

a conventional bar chart. This multimodal information provides a rich semantic context, outlining the intended message and insight the presentation aims to convey.

With PictographAI integrated into the presentation tool, the user begins by creating a slide. They input or upload raw data for a simple bar chart and provide slide text that provides context for the data. This is shown in Figure 1. Upon clicking 'Update Chart, a simple bar plot is shown next to the text, visualizing the data in its basic form. When the user clicks on 'Generate Pictograph', a relevant Pictographic visualization is generated, representing the bar chart in terms of geometrical layout, axes, and numerical values. In addition, incorporating a meaningful visual representation relevant to the slide's text content. Since stable diffusion image generation can sometimes result in unexpected or less relevant images (often referred to as "hallucinations"), the user can generate multiple pictographs and select the best one.

2.2 System Architecture

The system architecture of PictographAI is shown in Figure 2. In the figure, we break down the different parts of the pipeline and describe how they were implemented.

Slide Content. The User starts the interaction with the system by providing text content in the form of slide title and bullet points, as well as the data for a simple bar chart visualization and descriptive metadata such as axes labels. The text content is then automatically extracted from the slide and transferred to a Python Flask backend [12] where it will be further processed by the PictographAI pipeline.

Prompt Generation. On the backend server, a LangChain Large Language Model Agent is hosted [11]. The Large Language Model Agent from the LangChain framework serves as a reasoning engine to determine which visualization object to generate based on the given input. With OpenAI's chatGPT 3.5, the Large Language Model Agent is prompted to extract a suitable Human recognizable Object (HRO) from the provided slide content to guide the pictographic representation of the data. This ensures that the generated visualization will be both relevant and visually appealing.

Image Generation. The extracted HRO, mostly a single word, is then used as a text prompt for image generation. On the backend server, a text-guided image-to-image inpainting stable diffusion pipeline from the Hugging Face framework [9] is running. The stable diffusion model is initialized with the weights of Stable-Diffusion 1.5 [13]. To perform inpainting, the model takes an initial reference image, a mask outlining the areas to be inpainted, and textual prompts describing the desired image subject. The model uses the input image of the conventional bar chart as a reference for generation. The denoising strength, which is the extent to which the reference image is transformed, is set to 0.8. The input prompts are the HRO extracted in the previous step, as well as a generic negative prompt: "text, blurry, disfigured". The two text prompts guide the

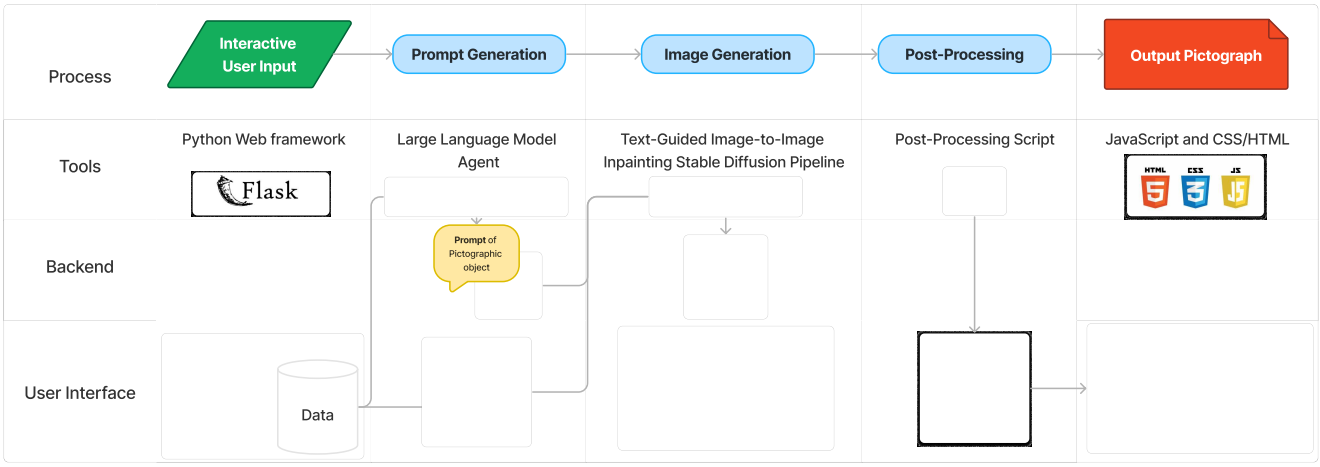

Figure 2: The pipeline of PictographAI: (1) The user inputs slide title, text and bar chart data (2) PictographAI extracts a prompt of a Human Recognizable Object from the slide text content; (3) the stylized Pictograph is generated; (4) post processing is applied to cut off irrelevant parts of the stylized graph and add the axis accordingly; (5) The generated pictograph is returned to the presentation tool in which the charts can be selected to replace the original bar charts with a pictograph.

image generation, while the guidance scale parameter is set to 20, as a higher guidance scale value encourages the model to generate images closely linked to the textual prompts. The generation seed is set to random so that each pictograph is regenerated with new seeds to provide alternative outputs from the same slide input and account for potential model hallucinations.

Post-Processing. After the pictographic image is generated in the backend, some post processing operations are performed to cut off inaccurate parts of the pictograph if there are any and overlay the pictoraph on the original chart axes, maintaining the original geometry of the visualization. This ensures that pictographs are stylized and still accurate in terms of the data they represent.

Output Pictograph. The generated pictograph is returned to the frontend. The user can view and select the generated pictograph, and generate another one with a new random seed until the user is satisfied with the result. PictographAI works on a per-slide level, so users can generate different pictographs for different slides.

3 Conclusion

We present PictographAI as an add-on to presentation tools that uses large language and image generation models to automate the generation of recognizable, topic-relevant, and data-faithful pictographic visualizations for bar charts. Using generative AI, PictographAI helps improve presentations with minimal effort and offers the potential to make data more engaging and memorable.

Acknowledgements

This research was partially funded by the Baden-Württemberg Ministry of Economy and Tourism as part of the AI Alliance initiative (grant no. WM31-43-99/140).

References

[1] Scott Bateman, Regan L. Mandryk, Carl Gutwin, Aaron Genest, David McDine, and Christopher Brooks. 2010. Useful junk? the effects of visual embellishment on comprehension and memorability of charts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 2573–2582. https://doi.org/10.1145/1753326.1753716

[2] Rita Borgo, Alfie Abdul-Rahman, Farhan Mohamed, Philip W. Grant, Irene Reppa, Luciano Floridi, and Min Chen. 2012. An Empirical Study on Using Visual Embellishments in Visualization. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2759–2768. https://doi.org/10.1109/TVCG.2012.197

[3] Michelle A. Borkin, Zoya Bylinskii, Nam Wook Kim, Constance May Bainbridge, Chelsea S. Yeh, Daniel Borkin, Hanspeter Pfister, and Aude Oliva. 2016. Beyond Memorability: Visualization Recognition and Recall. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 519–528. https://doi.org/10.1109/TVCG.2015.2467732

[4] Michelle A. Borkin, Azalea A. Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. 2013. What Makes a Visualization Memorable? *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2306–2315. https://doi.org/10.1109/TVCG.2013.234

[5] William S. Cleveland and Robert McGill. 1984. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *J. Amer. Statist. Assoc.* 79, 387 (1984), 531–554. http://www.jstor.org/stable/2288400

[6] Stephen Few. 2011. The chartjunk debate: A close examination of recent findings.

[7] Steve Haroz, Robert Kosara, and Steven L. Franconeri. 2015. ISOTYPE Visualization: Working Memory, Performance, and Engagement with Pictographs. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1191–1200. https://doi.org/10.1145/2702123.2702275

[8] Scarlett R. Herring, Chia-Chen Chang, Jesse Krantzler, and Brian P. Bailey. 2009. Getting inspired! understanding how and why examples are used in creative design practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 87–96. https://doi.org/10.1145/1518701.1518717

[9] Hugging Face. 2024. Inpainting. https://huggingface.co/docs/diffusers/v0.29.2/en/using-diffusers/inpaint. Accessed: 2024-07-05.

[10] Jessica Hullman, Eytan Adar, and Priti Shah. 2011. Benefitting InfoVis with Visual Difficulties. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2213–2222. https://doi.org/10.1109/TVCG.2011.175

[11] LangChain. 2024. Using Language Models. https://python.langchain.com/v0.2/docs/tutorials/agents/#using-language-models. Accessed: 2024-07-05.

[12] Pallets. 2023. Flask. https://flask.palletsprojects.com/en/3.0.x/ Accessed: insert date of access.

[13] RunwayML. 2024. Stable Diffusion v1.5. https://huggingface.co/runwayml/stable-diffusion-v1-5. Accessed: 2024-07-05.

[14] Edward R. Tufte. 2001. *The visual display of quantitative information.* Graphics Press, Cheshire, Conn.