CS302P_SE_Lab_2024

Release-1 Report

Ewe’s Eyes presents: ReVamp UI generator

LINKS

GitHub:- https://github.com/Aeromaster213/ewes_eyes

Canva :- Presentation

https://www.canva.com/design/DAF8MfViEtQ/npblsiihPOkLwV17v3d2MQ/edit?utm_content=DAF8MfViEtQ&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton (copy link if above is broken)

Introduction

Welcome to the release report for our project, where we present the key features and techniques employed in Release 1. Our project aims to revolutionize UI design by generating energy-efficient color palettes for user interfaces. Through the integration of advanced technologies such as React Framework for frontend development and Python Backend for server-side processing, we've created a platform that allows users to upload images and receive energy-efficient UI alternatives. Leveraging machine learning models and Stable Diffusion 2.1, we ensure the generation of visually appealing color palettes while adhering to energy efficiency standards. In this report, we delve into the features, techniques, and advancements achieved in this initial release.

Features

- **Image Energy Efficiency Evaluation**: Our system evaluates the energy efficiency of uploaded images and provides recommendations on areas for improvement.
- **Text Prompt-Based UI Generation**: Users can input text prompts to generate UI designs with similar purposes. The system provides comments on the energy values of the newly generated UI.
- **Energy-Efficient GUI Image Generation**: We offer the capability to generate energy-efficient GUI images, ensuring reduced energy consumption on display.

- **Customizations Based on User Input**: Users can customize generated options based on both uploaded images and text prompts, allowing for tailored solutions.
- **Multiple Color Palette Alternatives**: The system generates multiple energy-efficient color palette alternatives, providing users with a range of choices.
- **Online Accessibility**: Our platform is easily accessible online, eliminating the need for users to download or wait for repetitive retraining processes.
- **Dockerized and Modularized Architecture**: The system is dockerized, well abstracted, and modularized, facilitating quick and easy adaptation to changing demands.
- **Specially Optimized Pipelines:** We use an optimised ML inference pipeline with quantized weights to keep our model small, memory-efficient and fast.
- **Reliability and Security**: Our system follows industry standards for secure image generation, ensuring that any generation of violent or harmful images is avoided.

Methodology

We connected the thought of energy efficiency in UIs to something numerable: the UI's brightness and the displayed color palette.

So, for generating the Images, we are using the ML model of stable diffusion alongside a novel architecture in order to condition the stable diffusion to generate realistic and energy-efficient UIs, with a primary focus on Mobile UIs due to the ubiquitous and increasing importance of handheld devices [1]

Luma Intensity of a given color is calculated using the below equation,

$$Y' = 0.2126R' + 0.7152G' + 0.0722B',$$

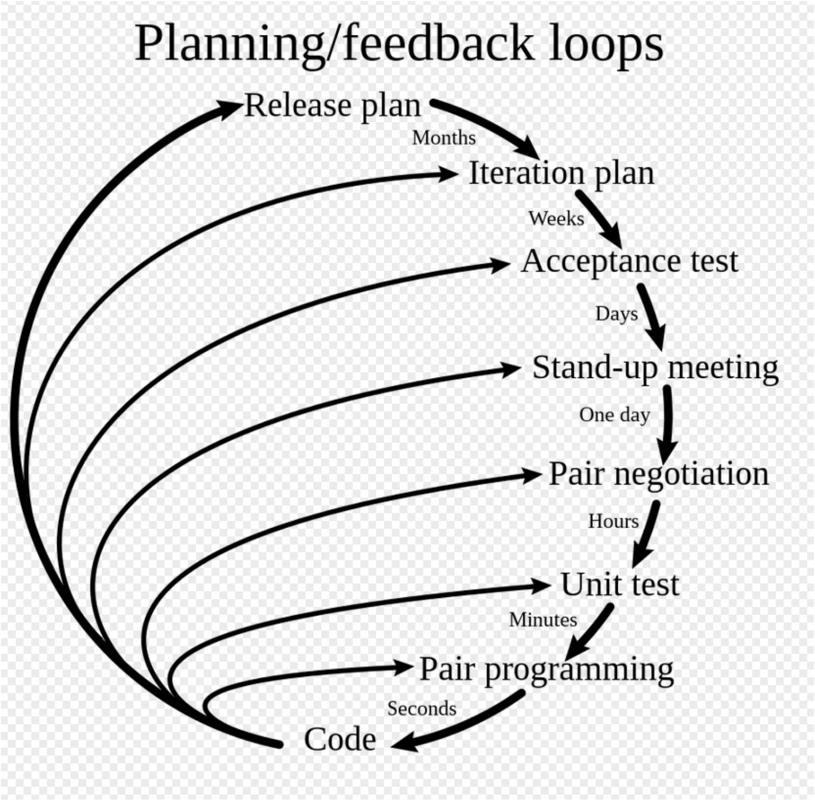
Followed by weighted average over the area of the specific, to give the Luma intensity.

For developing the deliverables to the user, the team has decided to host it online using a web application developed on react framework, with FastAPI as the backend. We have also developed a dockerized inference pipeline to streamline things. This helps us to run the tool even on machines with low GPU rates.

For starting the GUI design here is the initial FIGMA DESIGN which we adapted during the development.

https://www.figma.com/file/WcRYuTOvYRVwIBB6AsRaXn/Revamp?type=design&node-id=2%3A474&mode=design&t=mTwcwVYdcWaYtlJP-1https://www.figma.com/file/WcRYuTOvYRVwIBB6AsRaXn/Revamp?type=design&node-id=2%3A474&mode=design&t=mTwcwVYdcWaYtlJP-1 (copy link if above is broken)

We are following the “Extreme Programming” Model, as our Software lifecycle model.



The testing and the feedback are presently local to the team and unit testing and we plan to collectively test the software for the metric moreover release two.

For ensuring the right results we need to test the GUI on isolated cores in some system. This methodology to check the efficiency is under consideration and subject to change if we are able to find some research paper.

Techniques

React Framework: We developed the user interface (UI) for users to interact with the application using the React Framework. This framework allows for the creation of dynamic and responsive UI components, enhancing the user experience.

Python Backend: At the server-side, we employ Python-FastAPI for backend development. Python serves as the backend language for collecting user inputs and facilitating communication with various components of the system, including the image generator, Energy Efficiency Calculator, and Color Palette Matcher.

Energy Efficiency Calculator: The Energy Efficiency Calculator utilizes predefined metrics to evaluate the energy rating of uploaded images. This component provides comments on the energy efficiency of images based on the defined metrics.

Color Palette Matcher: The Color Palette Matcher matches images to color palettes based on their energy ratings. This component ensures that images are matched with color palettes that align with their energy efficiency ratings.

ML Model :We leverage machine learning (ML) models to generate images in different color palettes, incorporating both text prompts and uploaded images. [2] This approach allows for the generation of customized images tailored to user preferences and input.

Stable Diffusion 2.1 :Initially, we worked with the GUIGAN framework for image generation. However, due to limitations in the quality and cleanliness of the code shared in the open-source community, we transitioned to Stable Diffusion 2.1. This decision was motivated by the superior performance and cleanliness of Stable Diffusion. Thus, we proceeded with Stable Diffusion as our primary image generation technique.

The Team:

- **Arvind Srinivasan**
  - Handled configuring, training, testing, and fitting backend models.
  - Designed the ML architecture and specified backend requirements.
  - Integrated stable diffusion techniques into the backend processes for generating reliable results.

- **Abhinav Gupta**

  - Coding the frontend website user interface, translating design concepts into functional code.
  - Configured the middleware to manage user input, interaction with the model, and displaying the output.
  - Co-ordinated with everyone to integrate the working modules and debugged it.

- **Keshav Kumar Manjhi**
  - Designed the website's user interface, ensuring it is visually appealing and user-friendly.
  - Integrated middleware components, enabling communication between different software applications.
  - Implemented pipelines, optimizing data processing and task execution for the project.

- **Arpit Gupta**
  - Research on project topics, providing valuable insights and informing decision-making processes.
  - Contributed to formulating energy efficiency metrics, generation and selection of color palettes, ensuring they meet energy efficiency standards.
  - Debugging the graphical user interface (GUI), resolving issues and ensuring functionality.

- **Arup Biswas**
  - Designing the project documentation.
  - Debugging the generative models, helping to identify and resolve issues efficiently.
  - Research on energy-efficient metrics for User Interfaces and its applicability to our project.

References

[1] L. Cruz and R. Abreu, "Performance-Based Guidelines for Energy Efficient Mobile Applications," 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft), Buenos Aires, Argentina, 2017, pp. 46-57, doi: 10.1109/MOBILESoft.2017.19. keywords: {Androids;Humanoid robots;Energy consumption;Mobile applications;Mobile handsets;Mobile communication;Performance evaluation;Green Computing;Mobile Computing;Anti patterns},

[2] arXiv:2306.06233v1 **[cs.SE]** “Boosting GUI Prototyping with Diffusion Models”, 9 June 2023, Jialiang Wei, Anne-Lise Courbis, Thomas Lambolais, Binbin Xu, Pierre Louis Bernard, Gérard Dray

[3] Wei-Chung Cheng, Yu Hou and Massoud Pedram, "Power minimization in a backlit TFT-LCD display by concurrent brightness and contrast scaling," Proceedings Design, Automation and Test in Europe Conference and Exhibition, Paris, France, 2004, pp. 252-257 Vol.1, doi: 10.1109/DATE.2004.1268857. keywords: {Brightness;Lighting;Energy consumption;Cathodes;Fluorescent lamps;Liquid crystal displays;Photometry;Videoconference;Image quality;Propagation losses},

[4] Mian Dong, Yung-Seok Kevin Choi, and Lin Zhong. 2009. Power modeling of graphical user interfaces on OLED displays. In Proceedings of the 46th Annual Design Automation Conference (DAC '09). Association for Computing Machinery, New York, NY, USA, 652–657. Power modeling of graphical user interfaces on OLED displays | Proceedings of the 46th Annual Design Automation Conference

[5] Mistry, M., Apte, A., Ghodake, V., Mane, S.B. (2020). Machine Learning Based User Interface Generation. In: Pandian, A., Ntalianis, K., Palanisamy, R. (eds) Intelligent Computing, Information and Control Systems. ICICCS 2019. Advances in Intelligent Systems and Computing, vol 1039. Springer, Cham. https://doi.org/10.1007/978-3-030-30465-2_50

[6] Generative Models by Stability AI

[7] https://developer.apple.com/design/human-interface-guidelines/color

[8] The 60–30–10 Rule: A Foolproof Way to Choose Colors for Your UI Design | by Judith Lopez | UX Planet

[9] https://m3.material.io/styles/color/system/how-the-system-works

[10] Bhawna Suri, Shweta Taneja, Isha Bhanot, Himanshi Sharma, and Aanchal Raj. 2023. Cross-Platform Empirical Analysis of Mobile Application Development frameworks: Kotlin, React Native and Flutter. In Proceedings of the 4th International Conference on Information Management &amp; Machine Intelligence (ICIMMI '22). Association for Computing Machinery, New York, NY, USA, Article 60, 1–6. https://doi.org/10.1145/3590837.3590897

[11] http://resolver.tudelft.nl/uuid:8708c82f-1100-4607-ba28-eaebae8b03ce

[12] Cerny, T., Donahoo, M.J. On energy impact of web user interface approaches. Cluster Comput 19, 1853–1863 (2016). https://doi.org/10.1007/s10586-016-0665-7