

Location/Time based wallpaper generator

Group 11, 15/11/2023

Introduction

After a short brainstorming session on what our project could be, we ended up with an idea to make an application that generates desktop wallpapers based on different factors such as location, weather, and time of day to closely match the wallpaper to the conditions outside your window.

For this project we would have to explore different techniques that we're not all too familiar with, like working with API's for both the generative models, but also getting necessary data like the weather at your current location.

We wanted to work with OpenAI's API, first to generate a descriptive prompt based on our weather/location data, and then to generate a wallpaper based on that prompt. We discussed if it was necessary to generate a prompt, or if we could define some standard prompt to use with the data when generating the picture, but came to the conclusion that a generated prompt for each picture would make a more interesting and unique result for each one.

Methodology

For the application we used two models from OpenAI. GPT 3.5 Turbo and Dalle 3. GPT 3.5 was used to generate a descriptive prompt based on our input. We wanted to avoid using a standard prompt to mix it up a little between the different wallpapers we generated. The Dall-E 3 model was used to generate the photos. The reason we chose these models is because we are most familiar with OpenAI's models, and their API seems to be the most well documented. The Dall-E 3 model also supports generating photos with 1792x1024 resolution, which is not too far off a normal 1080p screen. A model with worse resolution would not work our application.

Application Design and Development

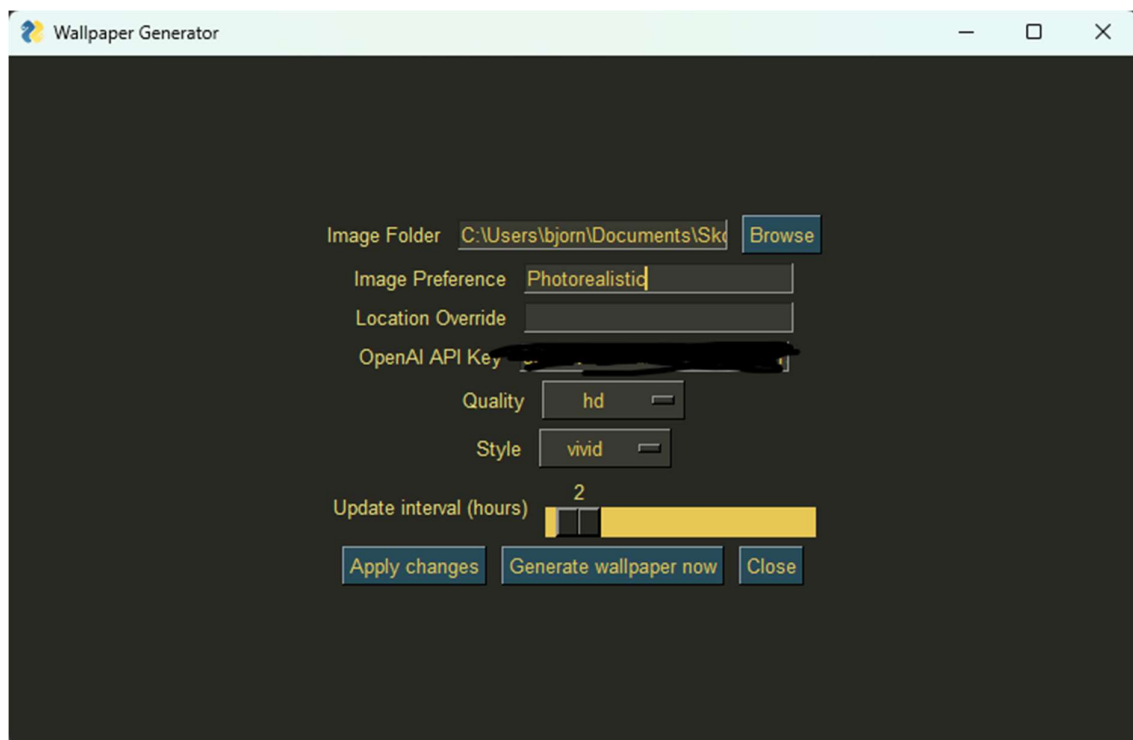
The application uses PySimpleGUI for a simple graphical user interface where the user can edit preferences for the generated images. These preferences are saved in a file locally using the pickle package. There is also a button for generating a wallpaper immediately, instead of scheduled generations throughout the day.

When generating a wallpaper, we first retrieve the users geolocation using the geocoder python package, we then use the MET Weather API to get a description of the current weather at this location, this is done with the requests python package. This information,

combined with an optional user preference, is used as input when generating a prompt using the OpenAI API, more specifically the GPT 3.5 Turbo model with a short system definition. This prompt is then used as input when generating an image using the Dall-E 3 model. The image generated is downloaded using a http GET request, and then saved to file locally. The last step is setting the image as desktop wallpaper through a system call using the ctypes python package. Currently the application only works for windows, but it should be easily expanded to work on other operating systems. The system type is found by using the platform python package.

Here is an example of what the process is currently like:

The user opens the application where they can set some preferences and insert the API key, unless they already have OPENAI_API_KEY defined as an environment variable. Currently the generation is not scheduled throughout the day, so the user has to press a button to generate a wallpaper.



Here's an example of the data retrieved from the weather API, this is passed on to a GPT model as a "user". There is also a system message passed on to define the task.

User location: Bergen, NO
Time: 2023-11-15T23:00:00Z
Temperature: 1.6
Cloud area fraction: 99.3
Relative humidity: 93.0
Wind speed: 0.2
Description: cloud

The GPT model sends back a response:

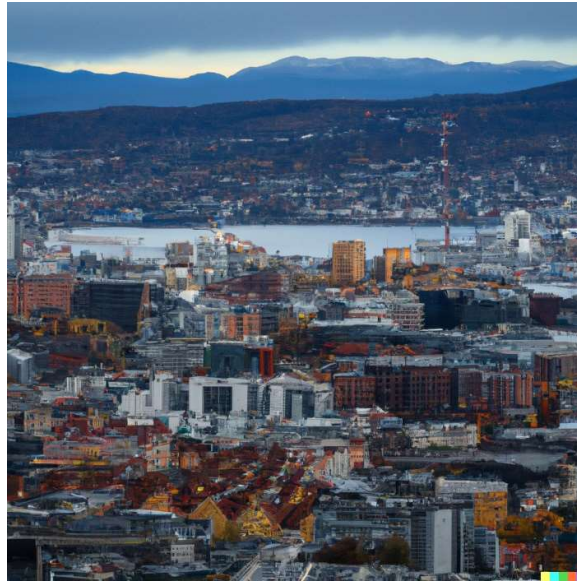
“The atmosphere in Bergen, NO, on November 15th, 2023, at 23:00 UTC, is cloud-covered with a temperature of 1.6°C. The air is moist, and a gentle breeze blows through. Create a photorealistic image of a cityscape at night with fog-covered streets and dimly lit buildings.”

And then the prompt is passed on to the Dall-E 3 model, which generates a wallpaper:



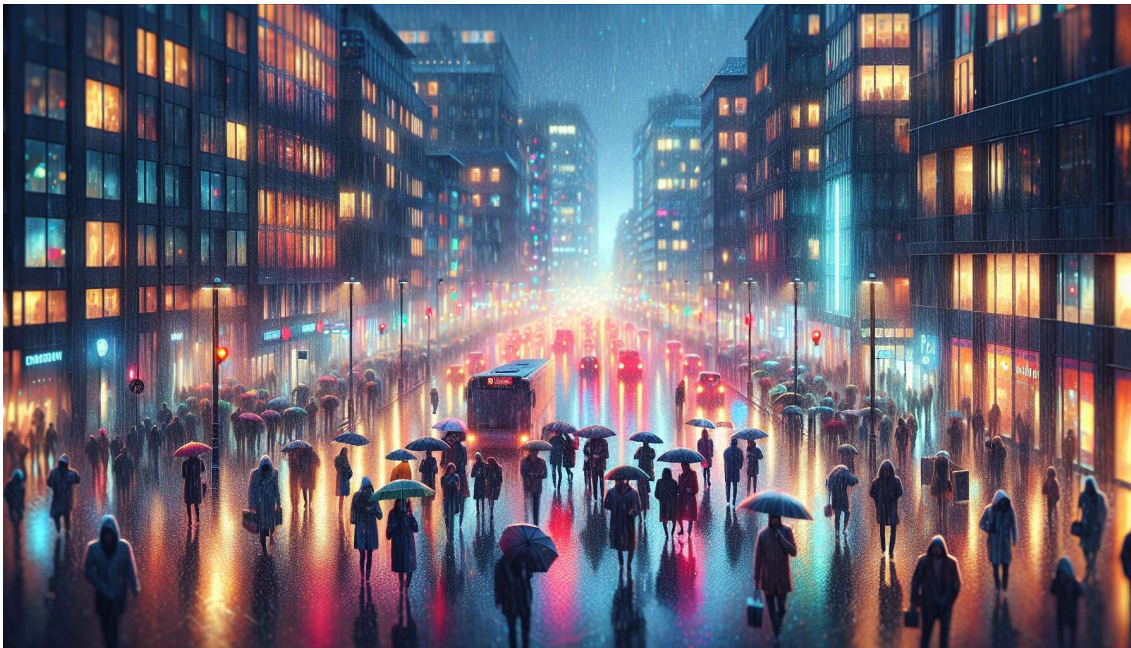
Experiments and Results

In the beginning of the project, we did some experimenting with the MET API and geolocation to retrieve the relevant information using Postman. We used this info to generate some prompts and images manually to see if the Dall-E model would interpret the data in a way that generated suitable desktop wallpapers. Using the Chat-GPT model interface directly didn't exactly prove very useful as it was hard to define the right instructions for it to generate good prompts. This was because this model is supposed to have natural conversations with the user, so you have to do some work to make it do exactly what you want without all the extra unnecessary stuff. This generated image of “Oslo” on a cloudy afternoon a couple of days ago is an example of the images we generated during experimentation. You can see the resemblance to the location, but it is not exactly suitable as a desktop wallpaper.



After adding some funds to an OpenAI wallet, we could finally start using the Dall-E 3 model for generating much better wallpapers. We also found that the GPT model we ended up using through the API was much more concise and to the point, and fit perfectly for our use.

Here is a wallpaper based on weather and location data in Oslo:



More examples can be found in the “examples” folder in the [repository](#).

Discussion

There are a few things to consider for this tool. Privacy could be a concern since we send the users location to OpenAI, where the prompts will be stored in their databases. The cost, in terms of money and energy must be considered since new images might be generated throughout the day, to keep the wallpaper synced to weather and light levels. This might be a big problem if the tool has many users.

Reproducibility

Currently to use the application the user needs their own OpenAI API key. This can be generated on the [OpenAI web page](#) after registering an account. There might also be a bug when generating a wallpaper where the weather API gets an error, but this could be worked around by first sending an API request with your coordinates via an app like Postman. After doing this it should work fine. This is a strange bug, and we're not sure why the API request through the requests package acts differently than through Postman.

Conclusion

The application was a fun little project that we could see ourselves finishing for our own use, and even share with others as an open source application for those that are interested. The market for dynamic wallpapers is there, as we can see with the popularity of applications like Wallpaper Engine. Working on the project was a good experience, especially when it comes to working with API's, and using the data

Future Work

There are a few things that could be worked on in the future. The tool should be supported on more operating systems. It could be more settings in the GUI to get more personalized wallpapers. The tool could automatically update the wallpaper throughout the day, to keep the wallpaper synced to the weather and light levels. Other types of data could be used to generate the prompt. Wallpapers could be stored locally or on a server in a way that wallpapers could be reused when data is similar. These are some ideas that could be expanded upon to further improve the results or reduce costs.