大規模言語モデル(LLM)を用いた柔軟なユーザーインターフェースの開発

愛知県立一宮高等学校 物化部
1 年 船橋一汰

概要

近年、生成 AI は大きな進歩を遂げているが、インターフェースとしての AI の使われ方は発展途上にあるように感じる。今回はこの LLM の応用方法として、自然言語ベースの UI を搭載した画像編集ソフトウェアを開発した。

課題 / 解決方法

Bad UI は宿命なのか？？

エディタ系のアプリケーションでは、特に画面が複雑になる

メニューUI が二重の Bad UI

UI (User Interface) という根本の概念から解決できないか?

LLM による自然言語ベースの UI

「自然言語の指示」が複雑な機能をすべて置き換える

直感的でありながら高度な操作も可能→広いユーザー層に利点

目指すもの

以上のコンセプトより、AI をインターフェース一部として備えた、高い柔軟性をもつ画像編集ソフトウェア Gen5 を開発した

デモ

元画像 「中心から外側にかけてぼかしを強くして」 「ゲームボーイ風の液晶風」 「LED 電光掲示板風にして」

方法

パラメータ UI

AI を組み込んだアプリケーションの多くは、固定的な成果物をそのまま出力する

「自然言語 UI」だけでは…

- 微細な修正でも再生成する必要がある
- 自然言葉の限界に対処しなければならない

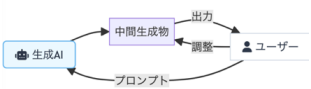
「自然言語 UI」と「GUI」を融合する

GUI としてパラメータ UI を切り出し、両方の利点を取り込む

LLM が画像を直接出力
（既存のアプリケーション）

些細な調整でも再生成が必要

LLM がプログラムを出力
（Gen5）



AI の再生成なしで微調整が可能

パラメータ UI

GPU 画像処理

シェーダー言語 (GLSL)

はじめは JavaScript 言語を LLM に生成をさせていたが、負荷が高く、パラメータ変更に対するリアルタイム性が失われてしまった

- WebGL API を通じて Web ブラウザからユーザーの **GPU** を使う
- LLM に **GLSL 言語**でプログラムを書かせる

```
void main() {
    vec4 color = texture(u_image, v_texCoord);
    vec3 brightened = color.rgb + vec3(u_brightness);
    brightened = clamp(brightened, 0.0, 1.0);
    fragColor = vec4(brightened, color.a);
}
```

プロンプト「明度を上げて」に対して LLM が書いたプログラム

スライダーを操作後、即座に変更が適用

ユーザー体験の最適化

(1) **LLM に「優しい」タスク**
JSON 形式…出力を解析しやすくなる
JSON 内にプログラムを書かせたところ LLM に高い負荷がかかり、JSON 解析エラーが頻発
→ プログラム部分を JSON から分離し、プログラムとメタデータを Markdown 形式で分けて出力

(2) **出力を順次表示**
生成途中でパラメータ UI をストリーミングにより順次表示

(3) **AI にクリアなコンテキスト**
AI が操作対象の画像がどんなであるか判断するために、マルチモーダル LLM を用いて画像を解釈
画像アップロード時に画像の説明文を自動生成
→ エフェクト生成時に参考資料として LLM に与える

使用技術

- LLM モデル: OpenAI o1, OpenAI o1-mini
- プログラミング言語: TypeScript
- 自宅サーバーで運用

作成したソフトウェア

https://b3zfR4Yg-beta.gen5.app

Gen5

参考文献

Apple ・ Guiding Instruction-based Image Editing via Multimodal Large Language Models
https://machinelearning.apple.com/research/mgie