Lens Pro Max

Folder Structure

```
.
|-- LICENSE
|-- README.md
|-- requirements.txt
`-- src
        `-- Lens_pro_max_finalproj.ipynb
```

How to Setup

1) **Open the Notebook:**
   - Access **Google Colab** and open the file named
   - `Lens_pro_max_finalproj.ipynb`.

2) **Configure GPU:**
   - In Colab, navigate to **Runtime > Change runtime type**.
   - Set the **Hardware accelerator** to **T4 GPU** to enable GPU support.

3) **Generate Access Token for Llama-3.2-3B-Instruct Model:**
   - Log in to your **Hugging Face** account.
   - Search for **Llama-3.2-3B-Instruct** in the model hub.
   - Input your contact information on the model page to request access and generate an **access token**.

4) **Update the Notebook:**
   - In the **fifth code block** of the Colab notebook, locate the `access_token` variable.
   - Replace the placeholder value with the **access token** you generated from Hugging Face.

5) **Execute the Notebook:**
   - Run all the code blocks in the notebook **sequentially**.
   - Note that the **last code block will remain running continuously** and will not terminate automatically.

6) **Access the Application:**
   - Open the following link: https://anchored-harmless-swan.anvil.app/

- On the app interface, **input text and images** as required and click **Submit** to process the data.

Methods Tried

The following methods were tried to generate appropriate web query from the user's query and image:-

1) We tried to set up an image-text to text multi-modal to do this task but the we faced the following problems:-
   a) Models which were able to perform this task accurately were trained on billions of parameters (`Molmo-7B-D-0924)` and hence could not be used on T4 GPUs.
   b) Such models which we were able to run on the T4 GPUs were unable to generate appropriate web queries. (`moondream2`).

2) We set up an image-text to text model for giving a descriptive caption of the image based on the user's query. The user's query along with the caption generated were passed to an LLM which was tasked with generating an accurate web query.This method was tried on following pairs of models:-
   a) Image-text to text: `Qwen2-VL-2B-Instruct` , LLM: `Llama-3.2-3B-Instruct`
   b) Image-text to text: `InternVL2-1B` , LLM: `Llama-3.2-3B-Instruct`

Due to difference in inference times, we selected a) from the above two pairs.

Final Approach: We settled with the second method, compromising a bit on the inference time for a considerable increase in accuracy.

Code Overview

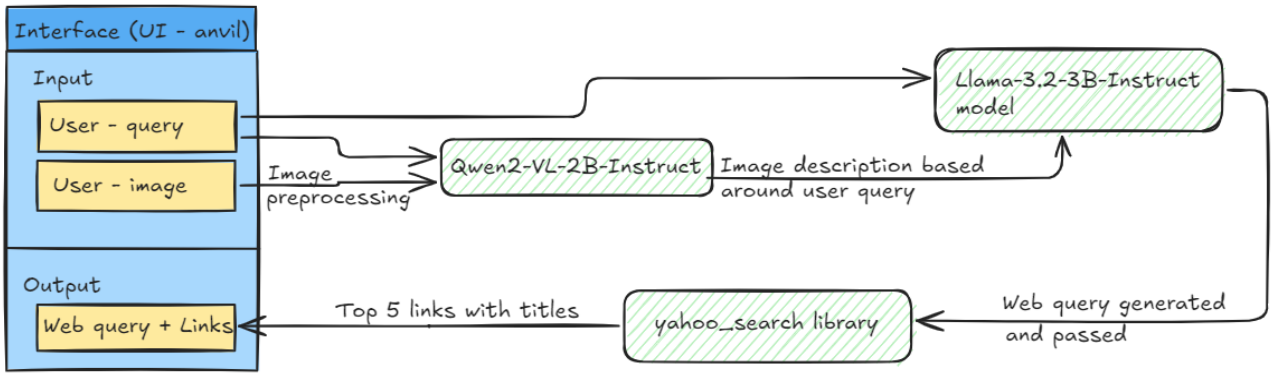
1. **Invocation of the `result` Function:**
   - When the **Submit** button is pressed in the web application, the `result` function in the Google Colab notebook is triggered, with the **input text and image** passed as Arguments.
2. **Image Preprocessing:**
   - The **image** is first converted into **bytes** and passed to the `process_data` function.
   - Within this function, the image is converted to **RGB format** and, along with the user query, is fed into the **Qwen2-VL-2B-Instruct model** to generate a descriptive caption.

3. **Generating the Web Query:**

- The **user query** and the **generated caption** are then passed to the **Llama-3.2-3B-Instruct model**, which produces an accurate **web query**.

4. **Performing Web Search:**
   - The generated web query is passed to the `get_web_results` function, which uses the **yahoo_search library** to perform the search and returns the relevant search results.
5. **Returning the Results:**
   - Finally, the **output links** along with the **web query** are sent back to the **user interface** for display.

Flow Chart



Scope for future improvements:

1. The Anvil server setup might need to be replaced with a different web framework for production use.
2. Fine-tune an existing image-text to text model to generate accurate web queries and use less computation power.
3. Performing quick and efficient web scraping based on the user query to display images, text and videos to the user rather than just links.