COMPUTER VISION AND IMAGE PROCESSING FINAL PROJECT REPORT

- NIKHIL TATIKONDA (50604863)

Title : SophieVision: Smart Object Detection for Visually Impaired. (The title is based on a fictional character who suffers from monochromatic blindness)

Project Overview:

Sophie is an application which is esigned to assist users in understanding their surroundings through real-time object detection, color recognition, and audio response in multiple languages. Sophie combines several modern computer vision and natural language processing techniques to deliver an interactive, accessible system.

The system works by capturing live video through a webcam, identifying objects in the frame using a YOLOv8 model, analyzing the dominant colors of the detected objects, translating the descriptive sentences into a user-selected language (English, Chinese, Spanish, or French), and finally reading these descriptions out loud using text-to-speech synthesis. Sophie also includes several display modes, such as a color-blind-friendly mode, edge detection mode, and a blur mode, which help make the system accessible to a wider range of users. Language switching can be done by jusr through voice commands.

When I looked at the state of the art, I found there are many separate tools that handle individual tasks, such as YOLO object detection demos, Google Translate, or text-to-speech apps. However, very few systems integrate all these functionalities into one coherent, real-time pipeline that runs locally on a user's machine and provides both visual and auditory feedback. Sophie bridges that gap, creating an interactive and adaptable experience.

Inputs and Outputs:

- **Input:** Live video stream from a webcam; user selection of language (via keyboard or voice command)
- **Output:** Annotated live video feed with object labels and dominant color; audio description spoken in the selected language; optional saved annotated video log

Summary of My Contributions:

- Developed a fully integrated pipeline combining object detection, color analysis, translation, and text-to-speech
- Built and tested multiple display modes to enhance accessibility
- Designed a flexible voice-based language switching mechanism
- Wrote modular, well-documented Python code that can be extended for future improvements
- Conducted detailed qualitative testing on system performance.

Approach : For this project, I combined computer vision, clustering, translation, and text-to-speech techniques to build a real-time multilingual object description system.

Algorithms used:
I used YOLOv8n for real-time object detection, KMeans clustering to extract dominant object colors, MarianMT for translation, and gTTS for speech synthesis. I also used the SpeechRecognition library to implement voice-based language switching.

What I coded on my own:
I wrote the complete system pipeline, including frame capture, integrating YOLO outputs with color analysis, and building the color matching logic using Euclidean distance. I managed the translation and speech pipelines, wrote code for handling device placement (CPU/GPU), and developed the system's display modes (color-blind, edge, blur) with real-time keyboard controls. The voice command system, temporary audio management, and UI overlays were also my own implementations.

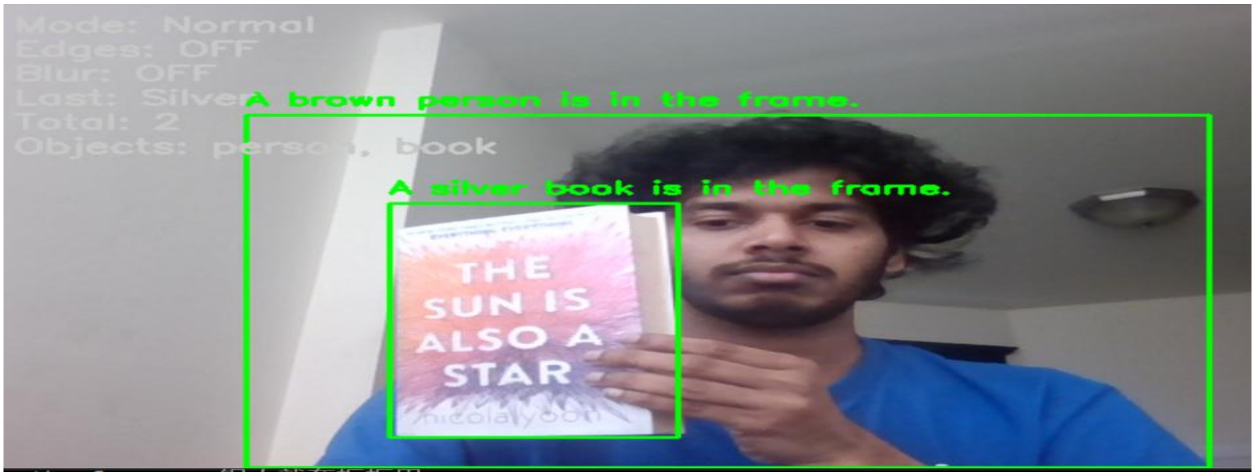
What I used from online resources:
I used pre-trained YOLOv8n (Ultralytics), MarianMT (Hugging Face), gTTS, SpeechRecognition, and KMeans (scikit-learn).

Experimented Protocol: Since my project runs entirely on live webcam input, I did not use a fixed dataset. Instead, I tested the system across a range of real-world conditions to ensure it worked reliably. I evaluated the system qualitatively by running it on different objects (such as bottles, laptops, plants, and furniture), in varying lighting conditions (natural light, indoor lighting, dim settings), and against both simple and cluttered backgrounds.

To evaluate success, I checked whether the system could correctly detect objects, identify dominant colors, translate descriptions into the target language, and deliver clear and well-timed audio feedback. I also tested user interactions, such as the responsiveness of display modes and the accuracy of the voice-based language switch.

For compute resources, I used a personal laptop equipped with an Intel i7 CPU, 16GB RAM, and an NVIDIA RTX 3060 GPU. While the GPU accelerated YOLO and translation tasks, I also tested the system on CPU-only mode to ensure it could still function at acceptable speeds.

Results : Below are the images for results I got.

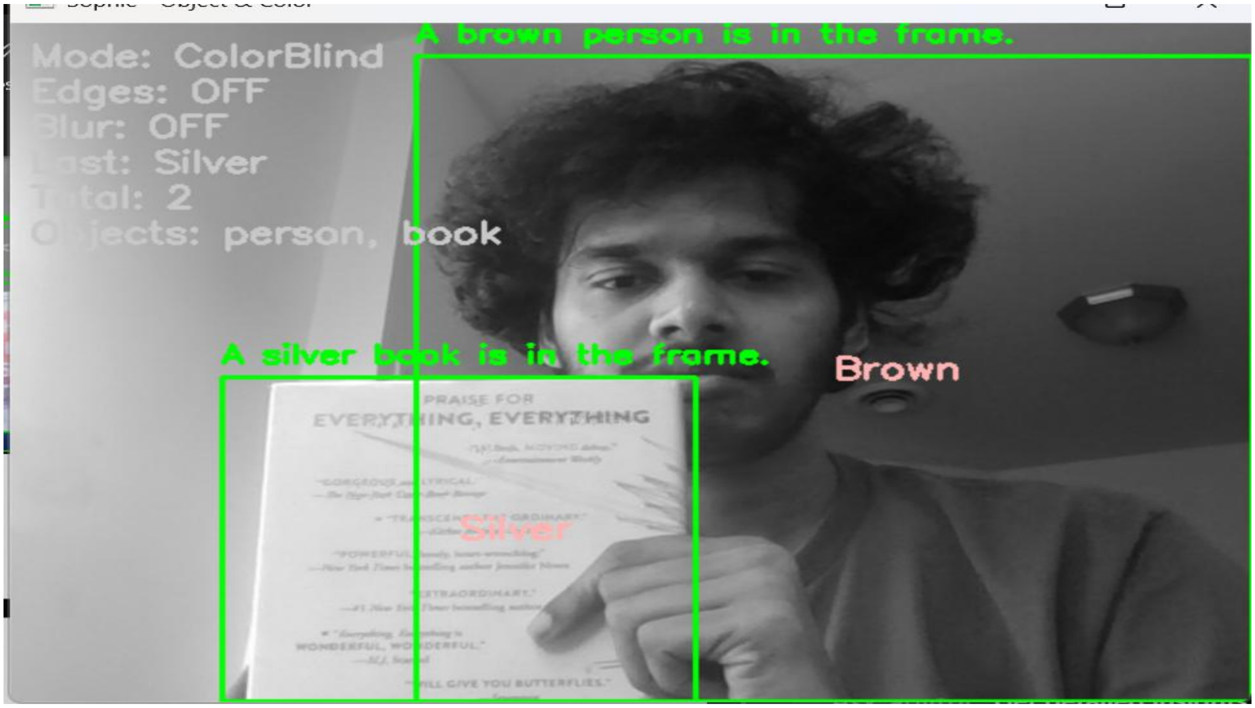


```
Mode: Normal
Edges: OFF
Blur: OFF
Last: Silve
Total: 2
Objects: person, book
```

A brown person is in the frame.

A silver book is in the frame.

Lets look at the same image in colorblind mode.

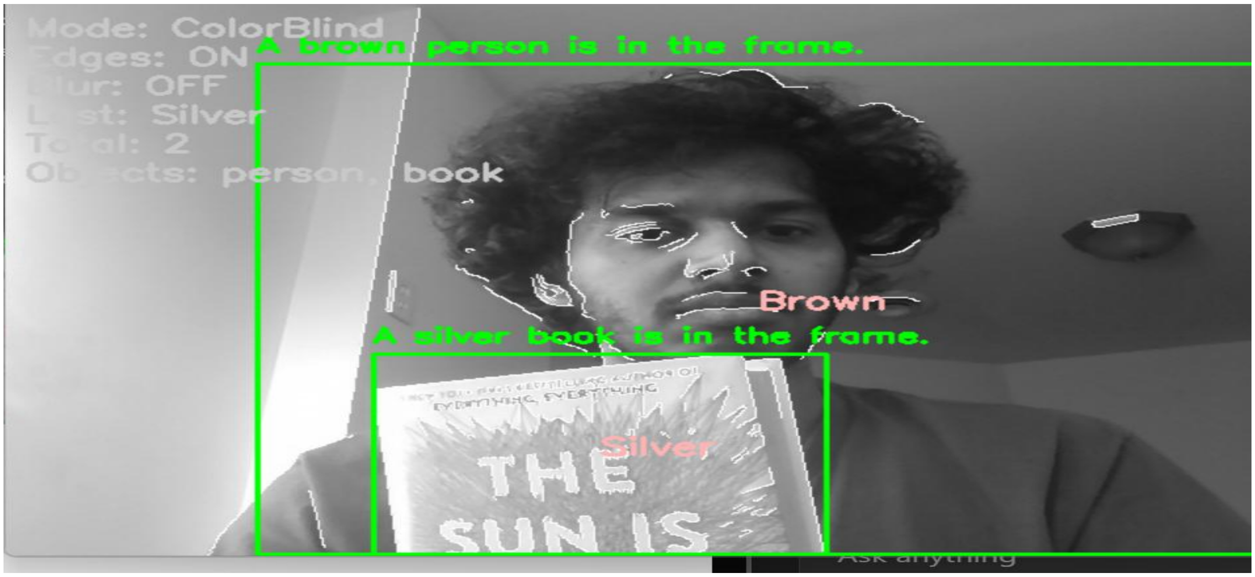


Mode: ColorBlind
Edges: OFF
Blur: OFF
Last: Silver
Total: 2
Objects: person, book

A brown person is in the frame.

A silver book is in the frame.

Lets look at Edge detection mode. (Canny edge detection which is taught in class)

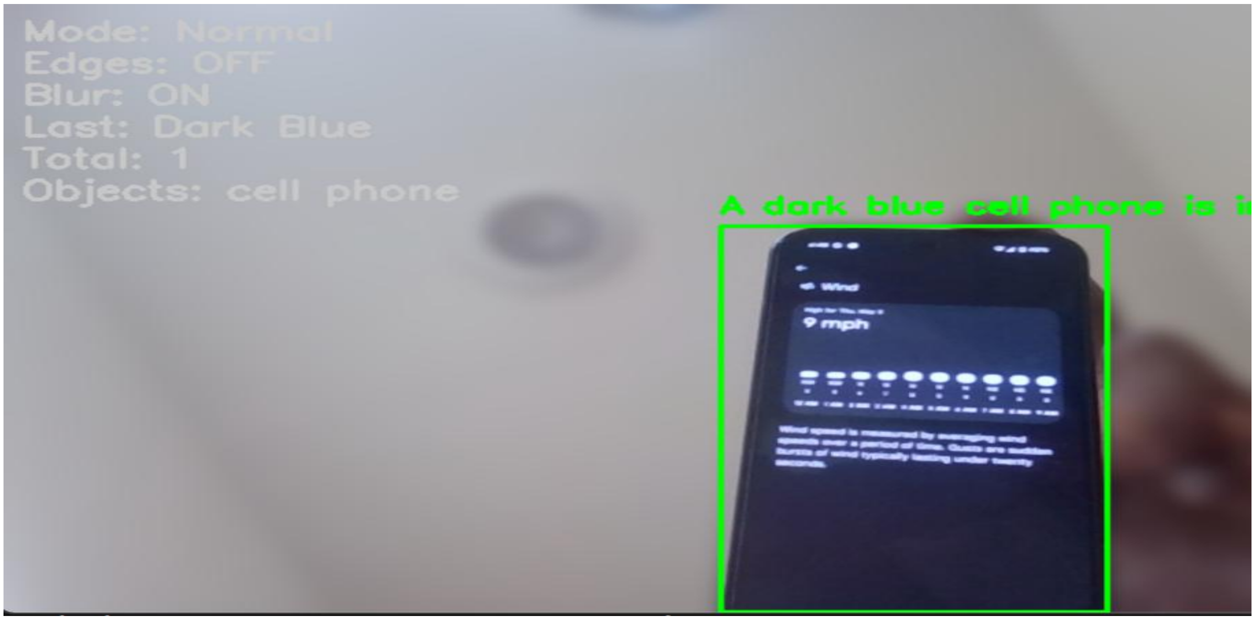


Mode: ColorBlind
Edges: ON
Blur: OFF
Last: Silver
Total: 2
Objects: person, book

A brown person is in the frame.

A silver book is in the frame.

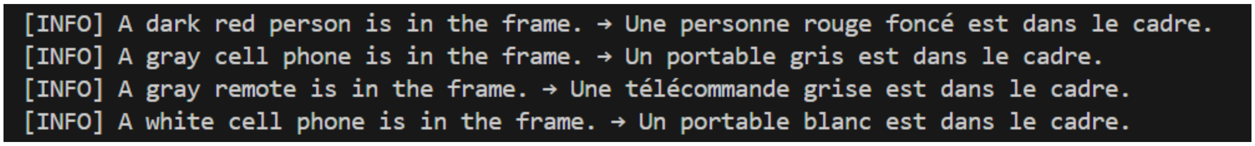
Now lets look at Blur mode results:



Mode: Normal
Edges: OFF
Blur: ON
Last: Dark Blue
Total: 1
Objects: cell phone

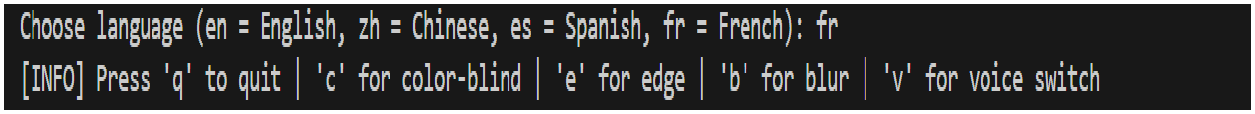
A dark blue cell phone is i

Lets look at the language translations:



```
[INFO] A dark red person is in the frame. → Une personne rouge foncé est dans le cadre.
[INFO] A gray cell phone is in the frame. → Un portable gris est dans le cadre.
[INFO] A gray remote is in the frame. → Une télécommande grise est dans le cadre.
[INFO] A white cell phone is in the frame. → Un portable blanc est dans le cadre.
```

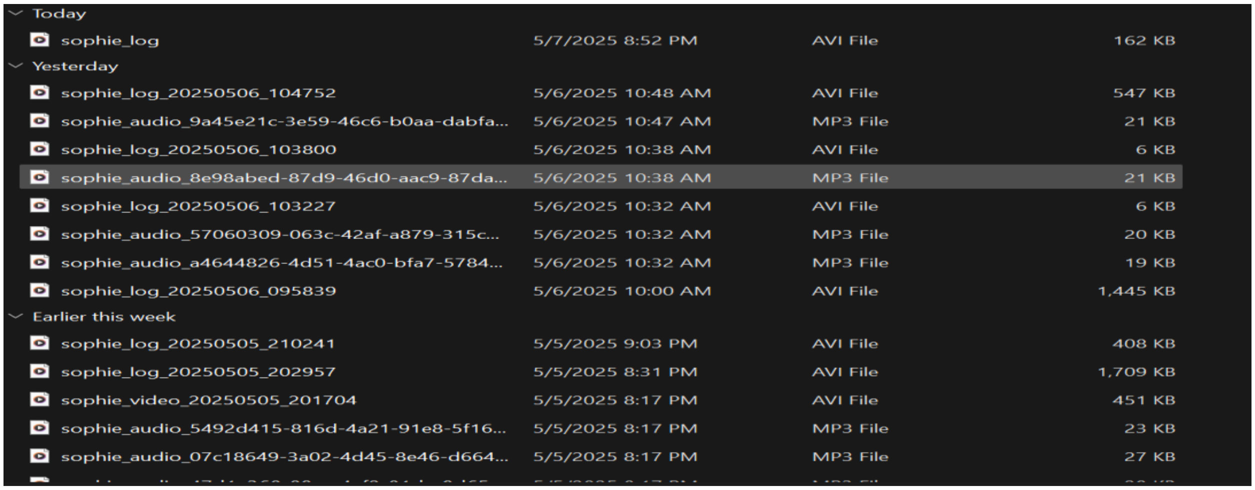
The above results are also spoken out by voice assitance using Google's text to speech translation. I cannot demonstrate it on the document.



```
Choose language (en = English, zh = Chinese, es = Spanish, fr = French): fr
[INFO] Press 'q' to quit | 'c' for color-blind | 'e' for edge | 'b' for blur | 'v' for voice switch
```

We can change the voice translation by voice command. For example to change to spanish, click V and say out "Spanish."

Logs folder:



```
Today
  sophie_log                                      5/7/2025 8:52 PM     AVI File      162 KB
Yesterday
  sophie_log_20250506_104752                      5/6/2025 10:48 AM    AVI File      547 KB
  sophie_audio_9a45e21c-3e59-46c6-b0aa-dabfa...   5/6/2025 10:47 AM    MP3 File       21 KB
  sophie_log_20250506_103800                      5/6/2025 10:38 AM    AVI File        6 KB
  sophie_audio_8e98abed-87d9-46d0-aac9-87da...    5/6/2025 10:38 AM    MP3 File       21 KB
  sophie_log_20250506_103227                      5/6/2025 10:32 AM    AVI File        6 KB
  sophie_audio_57060309-063c-42af-a879-315c...    5/6/2025 10:32 AM    MP3 File       20 KB
  sophie_audio_a4644826-4d51-4ac0-bfa7-5784...    5/6/2025 10:32 AM    MP3 File       19 KB
  sophie_log_20250506_095839                      5/6/2025 10:00 AM    AVI File    1,445 KB
Earlier this week
  sophie_log_20250505_210241                      5/5/2025 9:03 PM     AVI File      408 KB
  sophie_log_20250505_202957                      5/5/2025 8:31 PM     AVI File    1,709 KB
  sophie_video_20250505_201704                    5/5/2025 8:17 PM     AVI File      451 KB
  sophie_audio_5492d415-816d-4a21-91e8-5f16...    5/5/2025 8:17 PM     MP3 File       23 KB
  sophie_audio_07c18649-3a02-4d45-8e46-d664...    5/5/2025 8:17 PM     MP3 File       27 KB
```

Analysis :

Advantages:

- Modular code makes it easy to upgrade models or add features
- Real-time performance enables immediate user feedback
- Blur mode and Edge detection mode helps Visually impaired people to focus on objects.
- Multilingual audio makes sophie to be accessible without language barriers.
- Color blind mode shows the video in B/W while mentioning the dominating colors of the objects so that Colorblinded people can understand the world better.

Limitations:

- Voice recognition struggles in noisy settings
- Translation models occasionally output awkward phrasing, especially for complex sentences
- YOLOv8n, chosen for speed, sometimes misses small or overlapping objects.
- Dominant colors are not always the true colors of the object. Factors like lighting, background clutter and position may impact the dominance of colors.

Effect of Input Difficulty:

- Dim lighting reduced color detection accuracy.
- Complex backgrounds made some object detections fail or overlap.
- Similar-colored objects and backgrounds were harder to segment.

Discussion and Lessons Learned:

Working on Sophie taught me a great deal about integrating multiple AI models into a real-time pipeline, handling resource constraints, and designing systems for accessibility. I learned to balance model accuracy and speed, manage GPU and CPU workloads, and troubleshoot issues across computer vision, translation, and audio domains.

Going forward, I believe this project will help me in future roles that require building practical AI applications, especially those aimed at inclusivity and real-world usability.

Future Directions:

- Add object tracking to improve stability over video frames
- Incorporate larger YOLO models for higher accuracy
- Allow users to teach Sophie new objects or colors
- Add offline translation capabilities
- Expand to additional languages (e.g., Telugu, Tamil, Hindi)

Bibliography:

1. Ultralytics YOLOv8: https://github.com/ultralytics/ultralytics
2. Hugging Face MarianMT: https://huggingface.co/Helsinki-NLP
3. Google Text-to-Speech (gTTS): https://pypi.org/project/gTTS/
4. SpeechRecognition library: https://pypi.org/project/SpeechRecognition/
5. Scikit-learn KMeans: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
6. OpenCV library: https://opencv.org/

THANK YOU.