Journal

Qi Chen
22016834

Week1 6th July

Project Idea

As a seasoned international traveler who has visited over 50 countries, I have often pondered on ways to enhance people's travel experiences. One area that I have found lacking is the ability to discover local music while on the road. Most people end up listening to their own playlists, and it's tough to get a sense of the local music scene. I believe that a music app that generates local music based on location would greatly enhance the audio-visual experience of travelers, while also providing insights into the local music culture.

Idea 1
Generate new music by subdividing countries' places and re-categorizing local music based on location.
Analysis: This application is like a traffic radio station, listening to local radio music wherever you go but generating music without copyright issues. However, this application is solely an output of music and lacks interactive interest. This application requires collating a huge dataset, starting with music from a particular country.

Automatic Music Generation using Deep Learning
https://medium.com/@jdchauhan703/automatic-music-generation-using-deep-learning-68179914f30e

Idea 2
Take photos to identify scenery or objects based on positioning, and convert the images into music.
Analysis: This application becomes more than limited to the tourism scene, it has a broader range of environments and blurs the attributes of tourism positioning. But this application is more interesting and mobilizes both visual and auditory senses.

Generating music using images with Hugging Face's new package diffusers
https://medium.com/mlearning-ai/generating-music-using-images-13bdf1c78437

Q: I don't know exactly where to get the music dataset from, Spotify? Besides solving the machine learning model problem, I need to finish the UX design part, which I have some experience in, though the procedure to make an app is unknown to me. I need to make flowcharts and sort out the logic of the program.
How will the music generated in real time accumulate a lot of data and what will be done with it? How will the generated music be repeatedly played by users?

Tutorial: I don't need to make an app; I need to express the concept.

Idea1:

-Music can be played and accessed through Spotify's APIs https://developer.spotify.com/documentation/web-api

- Creating a dataset of artists' songs in each location

Retrieve the artist's city of residence using Wikipedia's API. Perform sorted lookups through geodatabases that store extensive latitude and longitude coordinates data.

- How to use generative AI to generate local style music

1）Train a musical AI vocalizer from scratch, manually collecting a list dataset of musicians from 5 locations.

2）Fine-tune an existing model.

3）Use a text-to-music model.

Final presentation: an interactive site, e.g., showing a map, clicking on locations, and hearing locally generated music.

Idea2:

Initially, I was obsessed with the location of the image. After some questioning from my tutor, I realized I was more interested in the music generated from the image's content. Next week, I will look at existing models of music generation and what information on the image corresponds to what style of music.

Week 2 19th July

Of the two initial ideas, I chose the second option.

I found some music generation models.

-Image to Music Gen:

Clip interrogator: Converting images to text

https://colab.research.google.com/github/pharmapsychotic/clip-interrogator/blob/main/clip_interrogator.ipynb

Music Gen Converting words to music

https://github.com/facebookresearch/audiocraft

https://huggingface.co/docs/transformers/main/en/model_doc/musicgen

-Mubert-text-to-music

In the process of sound selection for generation, input cues and Mubert API labels are encoded as potential space vectors for the Transformer neural network. The system identifies the closest label vector for each cue, and the corresponding label is then sent to our API for music generation. Notably, all sounds, including distinct loops for bass, lead, etc., are crafted by musicians and sound designers. It's important to clarify that the neural network itself does not synthesize these sounds.

How Mubert API works?

Step 1: Musicians create sample packs.

Step 2: Mubert buys out all the rights to them.

Step 3: AI arranges samples into the composition.

Step 4: Finally, the song is created.

https://github.com/MubertAI/Mubert-Text-to-Music

https://colab.research.google.com/github/ferluht/Mubert-Text-to-Music/blob/main/Mubert_Text_to_Music.ipynb#scrollTo=a4ACdvWLRJ5U

-Music ML

https://google-research.github.io/seanet/musiclm/examples/

https://arxiv.org/abs/2301.11325

After testing, the music generated by Musicgen is more compatible with text input.

Image recognition

-Objects and their location

The YOLO model can detect objects in an image in real-time. Multiple objects in an image can be identified and localized with information about their category and bounding box coordinates. This information is then mapped for use in a music generation system.

-color

Use Open CV to get information, such as the most prominent colors in an image. Create a color palette based on the image to show the three most prominent colors in the image, write

code to view these colors, and generate text for use in making music.

What other elements in the image can affect the music and how will they affect the music?
As well as being based on objects and colors, I can also influence the music based on the characters' expressions or the style of the buildings and the lines of the shapes in the image. For example, gothic architecture corresponds to a solid heavy metal style, and modern architecture corresponds to soothing music.
I also found tools to recognise architectural styles. architectural-style-recognition.ipynb, the recognition corresponds to the corresponding music.

In thinking about what elements in an image can influence the music, I needed to think about the types of things photographed the most while traveling from the perspective of the person taking a photograph and then start with one type first. But in the process, I found that the elements in the images corresponded to music that was just syllables, and it was challenging to form a melody. In addition, everyone's understanding of images and music is different and too subjective, and I needed clarification about how to do this to achieve my ideal description of music to image.

Week 3 25th September

New Ideas Emerge

One morning this week, I checked my email and found a ransom email from a hacker. The content was that during your last visit to a pornographic website. My spyware was activated on your computer system and eventually triggered your webcam, recording an eye-catching video of you masturbating". The hacker threatened to share the video with my email contacts within 24 hours unless I paid $2,000 in Bitcoin. When my friend received a similar email, it confirmed that this was a crazy attempt at "sextortion." Although I knew it was a scam and no video had gone viral, the fear and paranoia persisted until I became numb to the hacker's non-stop email bombardment.

Meanwhile, I found news on the algorithm-watching website https://algorithmwatch.org/en/spain-schoolboys-create-fake-nudes-ai/ about Spanish students using generative modeling to create fake nudes. Since 2017, generative models such as deepfake have been used to strip women from photographs and produce convincing pornographic videos artificially. 2018 has seen an explosive proliferation of deepfake videos, almost all of which are pornographic, with a cumulative total of over 134 million views. Instead of being curbed, tools for such purposes have become more accessible. In June 2019, an app called DeepNude allowed users to remove clothing from photos of women for $50. https://www.eldiario.es/tecnologia/negocio-lista- espera-app-usada-desnudar-menores-badajoz-cobra-9-euros-25-fotos_1_10522989.html The anonymous creators of DeepNude claim it is based on pix2pix, an open-source algorithm developed by researchers at the University of California, Berkeley in 2017.Pix2pix uses Generative Adversarial Networks (GANs) to train algorithms on large image datasets (in the case of DeepNude, the programmers claim to have over 10,000 photos of nude women) and then attempts to improve itself. This algorithm resembles those used to "imagine" road scenes in deeply faked videos and self-driving cars. The creators mention that the algorithm only works on women because pictures of naked women are easier to find online. The app was quickly removed after the Vice revelations, but a year later, a bot appeared on Telegram that enabled users to send pictures of girls and women in exchange for fake nude photos. https://www.zdnet.com/article/deepfakes-for-now-women-are-the-main -victim-not-democracy/

These examples show that women are the primary victims of "deep fakery."
Image-Based Sexual Abuse: Online Gender-Sexual Violations The paper states that
Private photographs are shared without the consent of the victim, which is defined as Image-Based Sexual Abuse (IBSA). While these behaviors have detrimental effects on the bodies of both the victim and the perpetrator, the harm is not in the body but in the gender power, control, and intent to harm. In situations of gender-sexual power imbalance, the perpetrator often remains anonymous and out of the picture, but the victim is affected by a constant sense of threat.

However, regulatory platforms and laws must still catch up with technological advances. Legal

scholars such as Danielle Keats Citron and Mary Anne Franks have called for a more robust legal framework to combat non-consensual pornography. In works such as their book The Fight for Privacy: Protecting Dignity, Identity, and Love in the Digital Age, they argue for a more nuanced understanding of digital privacy and consent. https://www.penguin.co.uk/books/446475/the-fight-for-privacy-by-citron-danielle-keats/9781784744847 However, some digital law experts have stated that "if the person affected is an adult and has not shared the image with anyone, then using AI to strip them naked is not a punishable act." I'm afraid I have to disagree with this statement. However, it further illustrates that deepfake technology puts everyone at potential risk. However, as long as people know the potential of deep forgeries, their spread can break this state of affairs.

As one of the victims, I have researched deeply into the community culture of fake nudity, as well as models of deepfake nudity (https://github.com/yuanxiaosc/DeepNude-an-Image-to-Image-technology/tree/master) , thereby critically examining the role of artistic interventions in revealing and address the gendered harm caused by AI deepfake technology.

My research found that Online subculture communities are groups of people who share similar interests and identities. These communities thrive in the digital realm through communal engagement(Digital Youth, Innovation, and the Unexpected - Xenos, Michael A.). They have their own social norms and cultural expressions that are unique to them. The internet serves as a fertile ground for the creation and evolution of subcultural content. These virtual spaces enable a participatory culture, facilitating the exchange of ideas and the creation of complex media landscapes where traditional and new media intersected(Convergence Culture Where Old and New Media Collide - Jenkins, Henry).

This interconnectedness of online subcultures has inevitably led to the emergence of contentious issues, notably the proliferation of deepfake pornography. The rise of deepfake technology has been mainly accelerated by subculture groups such as Reddit and 4chan, as well as other deep subculture communities that generate and view deepfake content. The increasing accessibility and user-friendliness of media manipulation tools have lowered the barrier to entry into this realm.(Politics and porn: how news media characterizes problems presented by deepfakes - Gosse, Chandell, Burkell, Jacquelyn).

An example and one of my inspirations for my project comes from the furry community. The furry fandom exemplifies a subculture characterized by its members' appreciation and engagement with anthropomorphic animal representations. This community has a rich history of artistic expression, from literature and visual arts to performance in the form of costume play (fursuiting) and role-playing. However, the creativity within this subculture is not confined to innocent manifestations; it has also been extended to adult-oriented content, a trend observable on platforms like DeviantArt, where there is a notable presence of furry content, including that of a controversial and even harmonious nature.

In the form of expression, I was influenced by an art intervention project on racial and gender discrimination due to algorithmic bias.

Joy Buolamwini directly challenged racial and sexual discrimination because of algorithmic bias while developing Aspire Mirror("Aspire Mirror,"). http://www.aspiremirror.com/ . Philip K. Dick's sci-fi classic inspired the development of this project, "Do Bionics Dream of Electric Sheep?". (1968) and the influence of emotional devices such as the empathy box and the emotion organ, as well as Anansi the Spider, a legendary creature from Ghanaian shapeshifting lore. We can generate more empathy by seeing ourselves as another person and changing our thoughts. Gazing into the Aspire Mirror, the onlooker can see reflected on her face, animals, quotes, symbols, or anything else we can encode into the system. https://artsandculture.google.com/story/BQWBaNKAVWQPJg
The Aspire Mirror allows people to see themselves become different entities, encouraging a shift in perspective. This transformative experience is designed to resonate with people by changing how they see themselves and others.

Another piece is Karen Palmer's perception i0. Participants will take on the role of a police officer and watch an interactive training video of an escalating volatile situation. They will experience the interaction from the point of view of the police officer's body-worn camera and engage (respectively) with the black and white protagonists. Each protagonist will either play the role of a criminal or someone with mental health issues. The Perception iO system will track participants' facial expressions. Their emotional responses to the scenes will impact the characters("Perception iO,"). https://ars.electronica.art/keplersgardens/en/perception/
This work explores the viewer's perception of reality and subconscious behavior, allowing them to examine their implicit racial or gender bias(Nast). https://www.wired.co.uk/article/karen-palmer-racist-bias

In both cases, the artists invited the audience to transform into participants by creating immersive and transformative experiences, using technology to stimulate emotions and reactions in the audience and influence the participants' perceptions. Through this artistic intervention, people are provoked to explore and renew their understanding of themselves and others, promoting a broader awareness of AI technology's ethical issues.

I plan to use ethical hacking practices for this project as an artistic intervention. I will reveal the pervasive harm done to women by AI-generated explicit content by making newspaper-style posters of their news stories. As people view the posters, they unknowingly have their images captured by a camera and transformed into nude photographs of animals for printing, enabling people to use AI properly.

Regarding the choice of animals, I considered using animals endangered or threatened by human activity, such as pandas, tigers, and elephants. This will show how vulnerable and powerless women are in front of Deepfake. Consideration was also given to using animals that symbolize strength and beauty, such as lions, eagles, and horses. This will show how tough and proud women are in the face of deepfake. The anthropomorphic characterization of the lion serves a dual purpose, projecting female resilience onto this symbol of strength and providing the viewer a way to connect the lion to the world and providing the viewer a way to connect with the issue emotionally.

Week 4 2nd October
Dataset

I need to collect relevant datasets to experiment with various machine learning algorithms to move forward with my new idea. I have briefly considered a few combinations of datasets to train with, hoping that they will produce the intended results I aim for.

Idea 1
Animal Face and Human Face Dataset

Objective: To collect and categorize various animal and human faces, focusing on capturing the diversity in facial features and expressions.

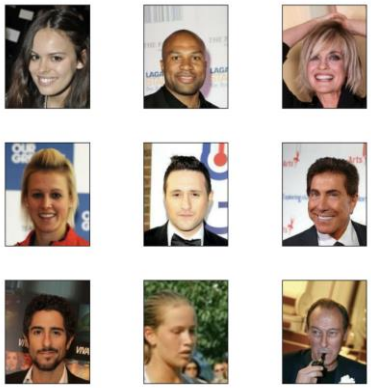
I downloaded the animal face database from Animal FacesHQ (AFHQ). The dataset includes 15,000 high-quality images at 512x512 resolution, with 5000 images for each of the cat, dog, and wildlife domains. All images are aligned horizontally and vertically to center the eyes. The dataset has already been modified to meet the requirements for data training purposes. You can find the link to the dataset at

https://github.com/clovaai/stargan-v2

AFHQ Animal Face Dataset

For human faces, I collected the images from CelebFaces, a widely used dataset featuring 202,599 face images of various celebrities. The images have been pre-processed for training purposes. You can find the link to this dataset at

https://www.kaggle.com/datasets/jessicali9530/celeba-dataset



CelebA Dataset
Expectation: I hope that the animal-like texture of our faces can be achieved by transferring the fur feature of various animal faces to human faces through training.

Idea 2
Lion Face and Human Face Dataset

Objective: A more focused dataset comparing the facial features of lions with humans.As In this project, I have envisioned the lion as my chosen animal avatar.

Data Source: The human faces are still using the CelebFaces dataset, the lion face I collected from the website: https://images.cv/dataset/lion-image-classification-dataset . There are many lion face images, but not many full-body images, which may need to be filtered manually.

Expectation: To transfer the lion fur feature more specifically to the human face.

Idea 3
Fursuit Human Pic and Human Pics Dataset

Objective: To compare images of humans in fursuits with regular human imagery, I researched the furry community as a potential dataset.

Data Source: The images of fursuits that I discovered come from a website known as [Fursuit Database](https://db.fursuit.me/), a dedicated community platform where members upload their fursuit photos to online albums. Since the site doesn't offer a bulk download feature, I resorted to web crawling to retrieve the pictures. For this task, I used [HTTrack](https://www.httrack.com/), a tool that efficiently downloads entire websites, including embedded images, through recursive fetching. However, this process necessitates substantial effort afterward to sift through and organize the images.

Fursuit Pic Crawled from Fursuit db

Idea 4

Sexually Lion Fursuit and Human Dataset
Objective: To create a dataset for comparing lion-themed sexually content fursuits with human imagery.

Data Source: I discovered a lot of adult content with this fur or anthropomorphism lion concept on a website called DeviantArt, a platform known for its diverse range of creative works, including animalized adult content. To gather specific imagery related to lion fursuits, I utilized a tool named gallery-dl. This Python-based command line tool is highly efficient for downloading images based on specific search keywords. It automates the process of fetching search result images. While gallery-dl is adept at bulk downloading, it necessitates a thorough review and curation process to ensure the dataset aligns.

** Pornographic content is indexed and tagged by category through the raddit nsfw411 adult content index. This website is not for research purposes but is now being used for research purposes for a pornography content classifier.

https://www.reddit.com/r/NSFW411/wiki/index/

Image Pre Processing
I integrated a new tool into my workflow to prepare images for training datasets. ImageMagick, an open-source image editing software accessible via the command line.

These are the steps I learn how to use ImageMagick:

1. Installation of ImageMagick via Homebrew:
-I installed Homebrew, a package manager for macOS, from https://brew.sh/.
brew install imagemagick

2. Batch Resizing of Images:
-Using the `mogrify` command from ImageMagick, I resized all `.jpg` images in a folder to the uniform dimensions of 256x256 pixels:
mogrify -resize 256x256 *.jpg

3. Resizing and Cropping a Specific Image:
- For individual images, I needed to first resize them to dimensions slightly larger than my target size and then crop them centrally. The command used was:
sudo mogrify -resize 256x256^ -gravity center -crop 256x256+0+0 image_name.jpg

- Finally, to apply the resizing and cropping process to all `.jpg` images in the batch, I ran:
sudo mogrify -resize 256x256^ -gravity center -crop 256x256+0+0 *.jpg

Week 5 12th October

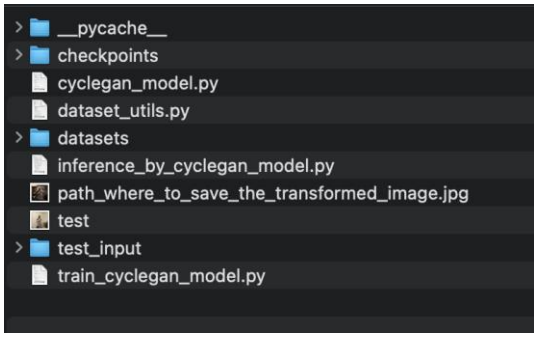
CycleGAN

This week, I experimented with the CycleGAN training algorithm. CycleGAN is a type of Generative Adversarial Network (GAN). The core innovation of CycleGAN lies in its framework, which uses two sets of GANs – each consisting of a generator and a discriminator. The generators learn to transform an image from one domain to another (and vice versa), while the discriminators evaluate the authenticity of the generated images. Specifically, it can convert an image from one domain to another without needing paired examples.

To test CycleGAN, I modified the TensorFlow implementation code (https://www.tensorflow.org/tutorials/generative/cyclegan) and migrated it to a normal Python environment, rather than using Jupyter notebook. However, my MacBook was not powerful enough to handle the training process, so I opted to rent a cloud GPU from vast.ai. I decided to rent a GPU from Vast.ai instead of using Google Colab because I needed to ensure continuous running of the training process. Although I tried subscribing to Google Colab Pro, which would allow me to use a powerful A100 GPU, the notebook shuts down every 6 hours, causing me to restart the training process all over again. It's difficult to train the model under these circumstances, especially since I set the epoch to 200 times. If I need to use Google Colab, I have to save the checkpoints frequently.

VAST AI Price List

I decided to rent a 4090 for testing purposes, and then I uploaded the training code and the dataset to the cloud server.



Training code File Structure

Dataset A

Dataset B

Results during the Training Process

Generated Results from the Final Training Model

Throughout the experiment, it became apparent that the CycleGAN algorithm may not be the best choice for transferring animal features to human bodies in my project. This may be due to the algorithm's inherent nature, which, while effective for specific image-to-image translations, may not be suitable for handling the complexity and subtleties required for this specific application. Additionally, the limitations observed could also result from the dataset used. The dataset may have been too small or not properly aligned, which is crucial for achieving optimal training results in deep learning models like CycleGAN. This experience highlights the importance of selecting the appropriate tool and dataset for specific machine-learning tasks, emphasizing the need for thorough research and analysis.

Week 6 18th October

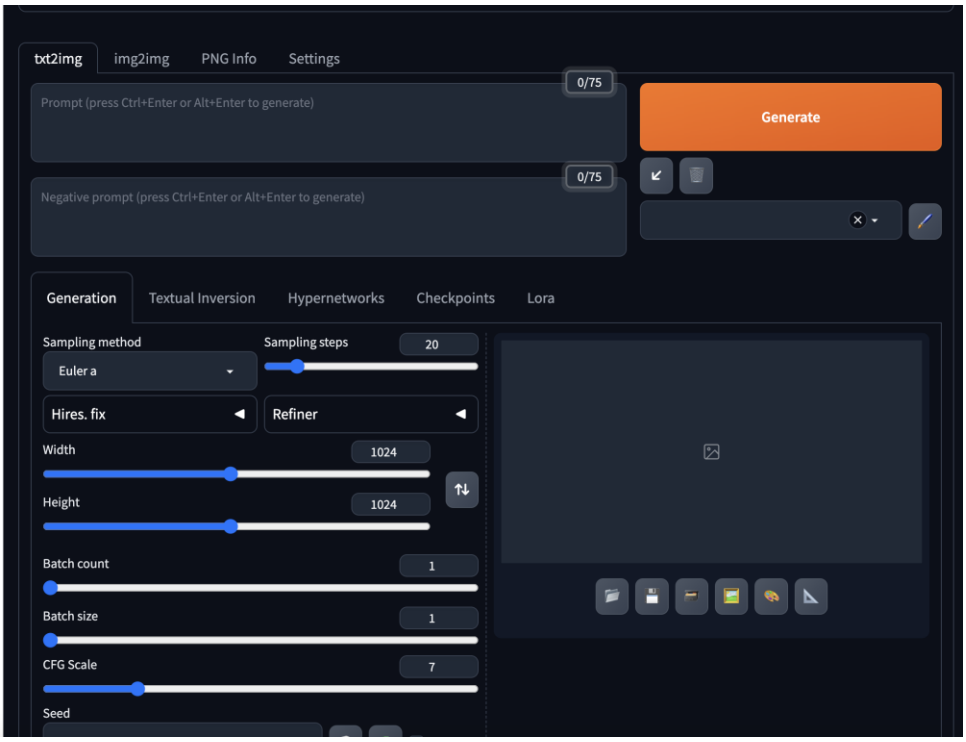
Stable Diffusion

This week, I shifted my focus from the CycleGAN experiments to exploring the capabilities of the Stable Diffusion WebUI, specifically the Automatic1111 version. After the challenges encountered with CycleGAN in transferring animal features to human bodies, I was intrigued by the potential of a different, more versatile image generation tool.

Automatic1111 Stable Diffusion WebUI

https://github.com/AUTOMATIC1111/stable-diffusion-webui

The user interface is based on the Stable Diffusion model, a powerful deep learning model that generates precise images from written descriptions. The Automatic1111 WebUI is renowned for its user-friendly interface that provides greater control over the image generation process by adjusting various parameters. Additionally, it offers a high degree of flexibility in customizing its functions by integrating community plugins.



Stable Diffusion WebUI Interface

In my exploration, I focused on several key parameters:

Steps: The number of iterations the model undergoes to refine the image. More steps generally result in more detailed images, though they increase generation time. Also I find if the steps are too high, the relational image to the original image is less obvious.

Diffusion Noise Strength: This parameter dictates the level of randomness in the image. Higher noise strength can lead to more creative outputs, while a lower setting often results in more realistic images.

Prompt Strength: This setting controls the degree to which the generated image follows the input prompt. Higher values enforce greater adherence to the input, while lower values allow for more abstract interpretations. This aspect often requires a lot of tweaking because the intended input prompt and the prompt that the model understands can be quite different from what was originally thought.

Seed Value: The seed value determines the level of randomness in the initial stages of the process. When the same seed value is used with identical settings, it will consistently produce the same image. This feature is useful for saving seed values that generate good output, allowing me to use them in future installations to produce similar output that matches my envisioned result.

My Original Photo

Nudity Photo, with anime enhanced, but the relation to the original human figure is lost.

With fewer steps, the images are closer to the original human body figure.

Result:
The process of tweaking these parameters was more intuitive and yielded results that were closer to my project's goals compared to my experience with CycleGAN. The images generated were impressively detailed and coherent, blending animal features with human forms per my requirements.

Conclusion:
After evaluating the results, I have made the decision to incorporate Stable Diffusion into my project. For the next phase, I plan to improve the prompts and test different parameter settings to enhance the output quality.

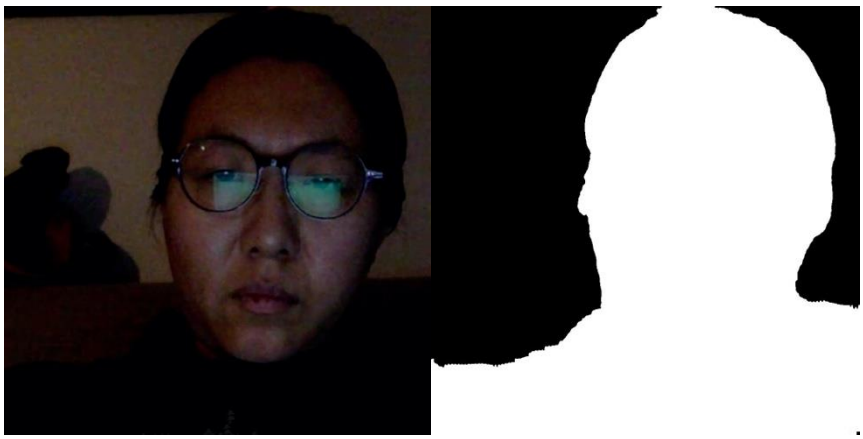
Week 7 26th October

OpenCV / MediaPipe Segmentation/ YOLO Tracking

This week's focus was on setting up a system that seamlessly captures audience images, processes them through various pipelines, including MediaPipe segmentation, OpenCV video capture, and YOLO tracking, and finally outputs them for further printing.

MediaPipe Segmentation

MediaPipe is an open-source framework developed by Google that offers efficient and easy-to-use machine learning solutions for media processing. I used its segmentation feature to prepare images for the image-to-image inpainting function. This step is crucial for creating masks that help separate subjects from their backgrounds, which is essential for the following stages of image processing.



Tested Segmentation Function from My Selfie

OpenCV

I used OpenCV, a powerful tool for computer vision, to capture live images of the audience. This tool fetched live feed from the webcam and served as the starting point for various processing pipelines. The image was cropped and sent through the MediaPipe segmentation, then the Stable Diffusion model, and finally to the printing stage.

YOLO

Another key aspect was implementing YOLO (You Only Look Once) for real-time audience tracking. YOLO is an efficient algorithm for object detection and is known for its speed and accuracy. In the context of my project, it's used to detect and track audience members as they interact with the installation.

Tracking the Person's Bounding Box

I programmed YOLO to track the audience and detect when someone stands in front of the installation for a certain period. This is done by detecting the bounding box around the person and monitoring the center movement. If the movement falls under a defined threshold, the person is considered still, triggering the system to capture their image. To mitigate data jittering – the minor, random fluctuations in the detected position – I implemented a running median filter. This filter smooths out the input data, ensuring that the decision to capture an image is made based on stable and reliable tracking information.

```
Median distance over last 5 frames: 0.2801992404566163
Median distance over last 5 frames: 0.28464767463870305
Median distance over last 5 frames: 0.3617780387731494
Median distance over last 5 frames: 0.43790912258381276
Median distance over last 5 frames: 0.43790912258381276
Median distance over last 5 frames: 0.43790912258381276
Median distance over last 5 frames: 0.3617780387731494
Median distance over last 5 frames: 0.3957874639865147
Median distance over last 5 frames: 0.3395454807871655
Median distance over last 5 frames: 0.3395454807871655
Median distance over last 5 frames: 0.3234907851721851
Median distance over last 5 frames: 0.3957874639865147
Median distance over last 5 frames: 0.24669215745811376
Median distance over last 5 frames: 0.24669215745811376
Median distance over last 5 frames: 0.19447321871206563
Median distance over last 5 frames: 0.19447321871206563
```

Tracking the median distance change per frame

Challenges

There was an issue with the system where it would occasionally detect non-human objects or wrongly identify them. To resolve this problem, I have come up with a plan to experiment with the confidence score threshold in the YOLO model. By increasing the threshold, the model will only detect objects it is more certain about, which will reduce false positives. However, it will require careful calibration to avoid missing genuine human detections. To

determine whether a person is at an optimal distance from the camera, I suggest using the size of the bounding box as an indicator. A larger box would indicate closer proximity, making it ideal for capturing a detailed image. And I need to conduct further experiments in the following weeks to tackle these challenges.

Remarks:

The conclusion of the week was marked by progress in the development of the system, which now integrates MediaPipe segmentation, OpenCV video capture, and YOLO tracking in the automatic pipeline. With these components in place, the system is capable of capturing an audience member's image when they stand still before the installation, processing it through the various pipelines, and preparing it for printing (although the printing functionality has not yet been implemented).

Week 8 31th October
Finalizing Printing Setup

This week, my project progressed into a crucial phase - choosing the correct printer and paper size for the final output of my generated images. Given that Stable Diffusion's output is in a 1:1 format (1024x1024), the choice of printing paper and printer was important to ensure quality and aesthetic appeal.

I took my first step by selecting a printer that can handle high-resolution images efficiently. After conducting thorough research, I finally settled on a second-hand model from a friend. This printer is capable of printing normal paper as well as glossy/matte photo paper. One of the reasons I chose this printer is due to its cost-effectiveness. The ink cost for this model is also not too expensive, and it can run for up to three days without any issues. Moreover, the printer is compatible with various paper sizes and types, which was a significant factor in my decision-making process.

In the field of photography, paper size and type significantly influence the visual impact of an image. Common sizes include 4x6 inches (standard photo size), 5x7 inches, and 8x10 inches. The square format is favored in Polaroid images, which resonates with the 1:1 aspect ratio of my project's images.

Initially, I experimented with printing the 1:1 format images directly onto standard paper and then cut the image to 1:1. However, this approach left the images feeling cramped. After several tests, I found that printing the 1:1 photo on a 4x6 paper, with white bars at the top and bottom, significantly enhanced its aesthetic appeal. This framing technique, reminiscent of the classic Polaroid style, added an elegant touch while ensuring the integrity of the image's dimensions.

Integrating Automatic Printing with CUPS:

Web Interface of CUPS

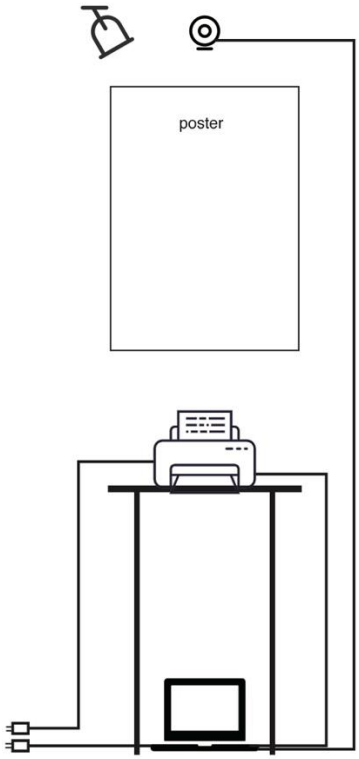
To automate the printing process, I explored CUPS (Common Unix Printing System), an open-

source printing system for macOS and other Unix-like operating systems. CUPS allows for detailed control over the printing process, including parameters like paper size, print quality, and color management.

It supports a wide range of printers and has a well-documented API, which was crucial for my needs. I delved into the Python implementation of CUPS, which allowed me to integrate printing capabilities into my existing codebase seamlessly.

With the printing function integrated and the final output format determined, my project is now at a stage where the generated images can be automatically printed with the desired aesthetic quality. Combining the 1:1 format image with the Polaroid-style framing on 4x6 paper offers a visually pleasing presentation of the output.

Week 9 5th November
Testing



✖

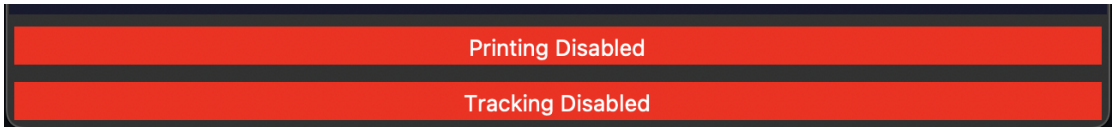
Based on the previous concept, I created a preliminary sketch. The poster is affixed to the wall, with the camera positioned above to capture full-body shots of the audience. However, the initial camera we tested had a too short cable when assembled, so I had to borrow another camera from the school. Additionally, I need two adapters since my computer has tap ports while the camera and printer have USB ports. The printer is situated below the poster, and my computer is concealed underneath, where it is not easily visible to the audience.

Issue 1: Lighting
During testing, the lighting only illuminated the poster. It did not consider the audience, causing the camera to occasionally struggle to recognize people's shapes, resulting in longer image generation times than expected and an inconsistent experience. Though I adjusted the tracking confidence value and increased the tracking sensitivity, there was only a slight improvement.

Issue 2: Continuous Recognition

As the camera constantly recognized individuals, it captured images of viewers who needed to be more actively engaged in the project from a distance, leading to a significant waste of photo paper. I added manual switches in the Python code to bypass tracking and printing logic using a bool variable.

```python
# Tkinter window setup
tabnine: test | explain | document | ask
def setup_tkinter_window():
    global print_toggle_button, tracking_toggle_button, root

    root = Tk()
    root.title("Interactive Control Panel")
    root.protocol(
        "WM_DELETE_WINDOW",
        on_closing,
    )

    # Image display label
    image_label = Label(root)
    image_label.pack()

    # Custom toggle buttons using labels
    print_toggle_button = Label(root, bg="green", fg="white")
    print_toggle_button.pack(fill="x", padx=5, pady=5)
    print_toggle_button.bind("<Button-1>", toggle_printing)

    tracking_toggle_button = Label(root, bg="green", fg="white")
    tracking_toggle_button.pack(fill="x", padx=5, pady=5)
    tracking_toggle_button.bind("<Button-1>", toggle_tracking)

    # Initialize button text
    update_button_appearance(print_toggle_button, "Printing", print_enabled)
    update_button_appearance(tracking_toggle_button, "Tracking", tracking_enabled)

    # Start the video capture in a new thread
    threading.Thread(target=lambda: cam_capture_loop(image_label), daemon=True).start()
    print("Program started")
    root.mainloop()
```



The Python code of the GUI interface with two manual buttons

Issue 3: Printer

The printer encountered occasional paper jams and ran out of color ink during testing. When I attempted to switch to black-and-white printing after converting the code to grayscale, the test photo produced no image. I promptly purchased ink for the printer, only to discover that it required color and black ink.

Given these issues, the project would have been challenging easier to run as expected with my on-site supervision.

Week 10 12th November

Expectation

The installation's development presented a series of technical challenges, prompting a continuous process of evaluation and refinement:

- The precision in tracking participant stillness using YOLO v8, although practical, reveals potential areas for enhancement. Future iterations could explore advanced data filtering techniques, such as Kalman filters, or shift the focus to tracking specific key points on the person rather than the entire bounding box. Custom training of the model to identify standing poses could refine the tracking to target the intended audience more accurately.
- Image preprocessing in OpenCV presents opportunities for incorporating enhancement techniques that could refine image quality before segmentation, optimizing them for better results in subsequent processing stages.
- The potential for custom training the Stable Diffusion model is vast. Tailoring the model with targeted prompts could offer more controlled artistic outputs, balancing the creative elements of stochastic variation with thematic consistency.
- Printer functionality is critical to the installation's success, and incorporating fallback mechanisms or automatic recovery scripts could significantly improve reliability, addressing issues like over-printing or print job failures.
- When setting up the installation, the lighting requirements were essential to capture the portraits with open CV, and the subsequent exhibition will project two stations of spotlights. One projected towards the news posters and one towards where the audience will be standing.

Testing showed that the project stimulated audience participation. Acceptance lies in the nudity of facial features. Artistic nude photos are generally acceptable if done tastefully. In addition, the danger of nude photos to people is not the generated model but whether it is distributed. More boldly, it is assumed that these artistically manipulated photos will be used for distribution, requiring viewers to purchase their copyrights for a minimum amount and giving viewers the right to delete or retain the originals. How will people choose?