FROGS

Fine Resolution Optimization and Gradient Smoothing

David Hunter, Ethan Mandel, Harrison Sun

Background and Motivation

- When searching for high resolution backgrounds for abstract screen aspect ratios, a plethora of backgrounds exist, though they exist behind “paywalls”

- Diffusion based models, if prompted correctly, can serve to generate backgrounds, though existing services hosting these models do not run “on-device”

Introduction

- We propose an implementation of a pre-trained stable diffusion model based pipeline that serves to generate high resolution images “on-device” on memory constricted devices (**under 8Gb VRAM**)

- By combining **super-resolution DNN models** with a set of image **post-processing techniques**, we have created a stable diffusion pipeline that generates high resolution images using a **fraction** of the VRAM (~1Gb) required to natively generate high resolution images

Pre-Trained Models

- For this project, the pretrained diffusion model in use is:
  - Lykon's Dreamshaper V7
  - (Lykon/dreamshaper-7 · Hugging Face)

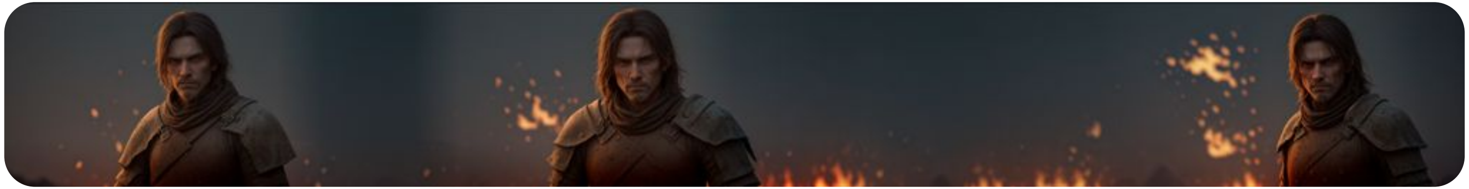
- The Latent Consistency LORA model utilized is:
  - Latent Consistency Model (LCM) LoRA: SDv1-5
  - (https://huggingface.co/latent-consistency/lcm-lora-sdv1-5)

- The pretrained super-resolution model in use is:
  - Latent Diffusion Model (LDM) for super-resolution
  - (CompVis/ldm-super-resolution-4x-openimages · Hugging Face)

Existing Solutions

- Generate a single high resolution image using a Stable Diffusion model “on-device”
  - Pros:
    - Native image is generated **without the need for post-processing**
    - Hugging Face API allows for a short code-base
  - Cons:
    - **High memory usage** (upwards of 20Gb for very high resolution images, beyond 4k)
    - **Poor image quality** (for both high resolution and atypical aspect ratios)
- Photoshop
  - Pros: Able to leverage Adobe Suite **- works off the shelf**
  - Cons: Requires **external application** and **manual adjustments**
- Use DALLE-2 for Image Generation
  - Limited resolutions supported
  - Requires large model

Problems



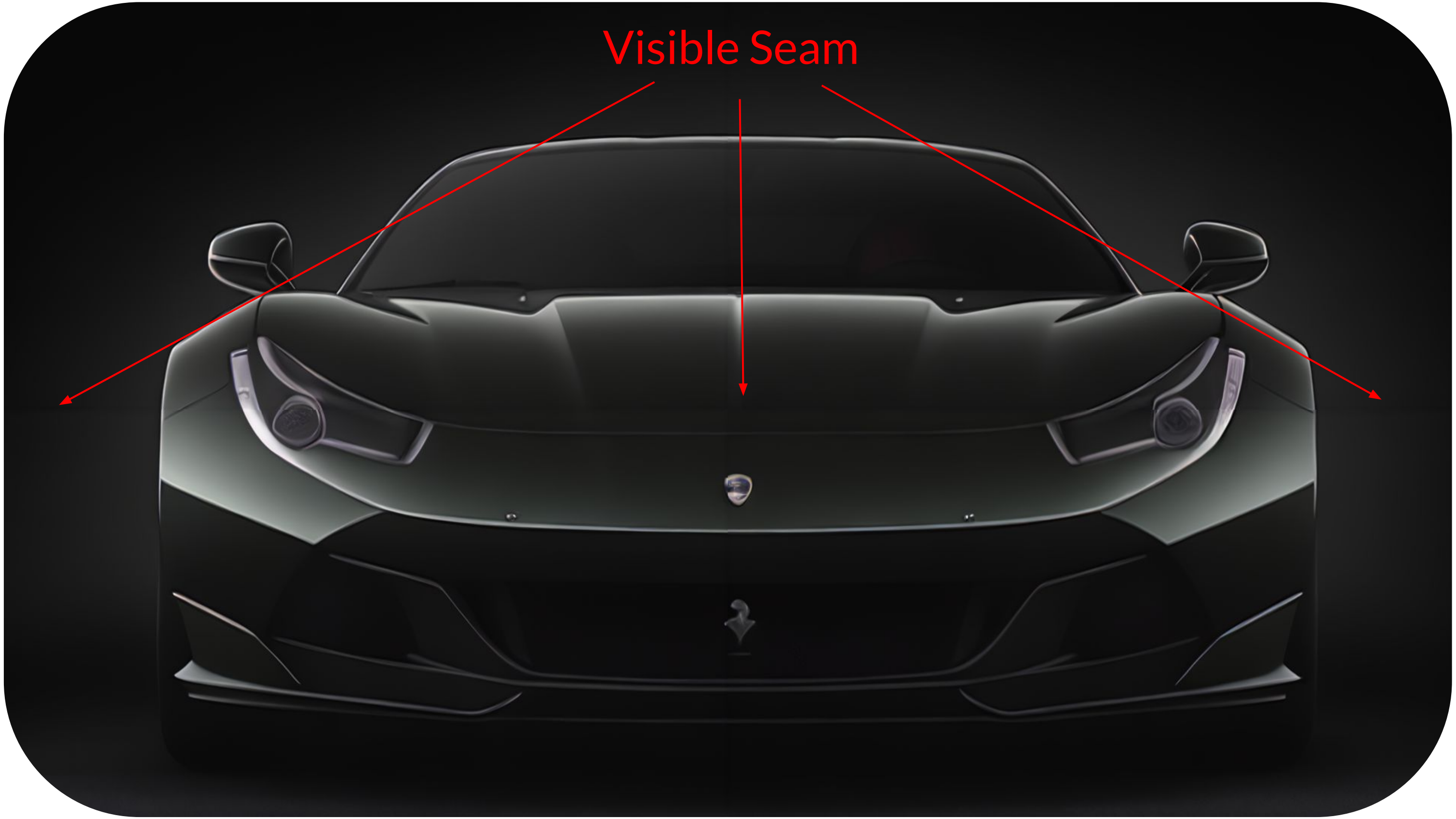
Single battle torn ancient soldier staring at camera, fire filled background, 8k

- Generates multiple subjects per image, even when prompted not to
- Single generated subject is identical throughout the expanded image
- Areas between each generated subject may become discontinuous

Improvements

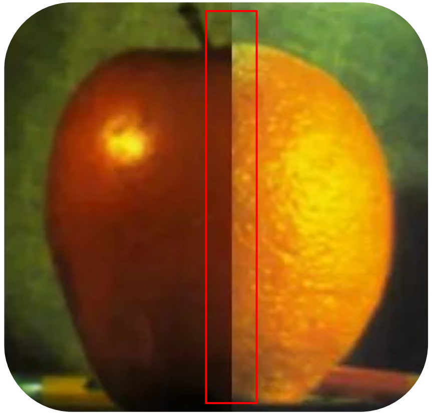
- Reduce Memory Profile
  - Port image generation capabilities to mobile devices
  - Rewrite code in Swift to utilize Apple specific hardware acceleration
  - Implement TensorRT
- Reduce artifacts generated in high resolution images
- Generate images of arbitrary resolution (multiple of 8)
  - Blend multiple generated images to create non-standard resolution images
- Improve parallelizability
  - Super Resolution

Output of From Image Quadrant Merge



Introduction to Alpha Blending

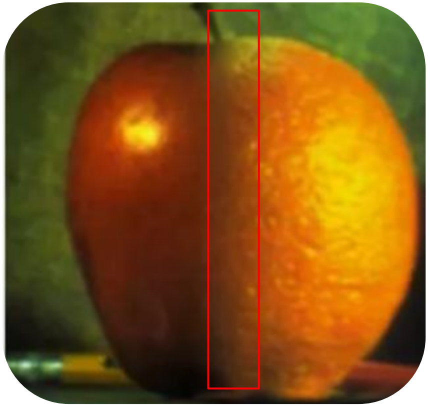
- Used for creating transparent and semi-transparent effects
- Common in image processing, video games, and GUIs
- Alpha = transparency level 0 (transparent) to 1 (opaque)
- $C_{result} = \alpha * C_{foreground} + (1 - \alpha) * C_{background}$
- Applied per pixel in the overlapping region



Unblended image

Apple Image

Orange Image



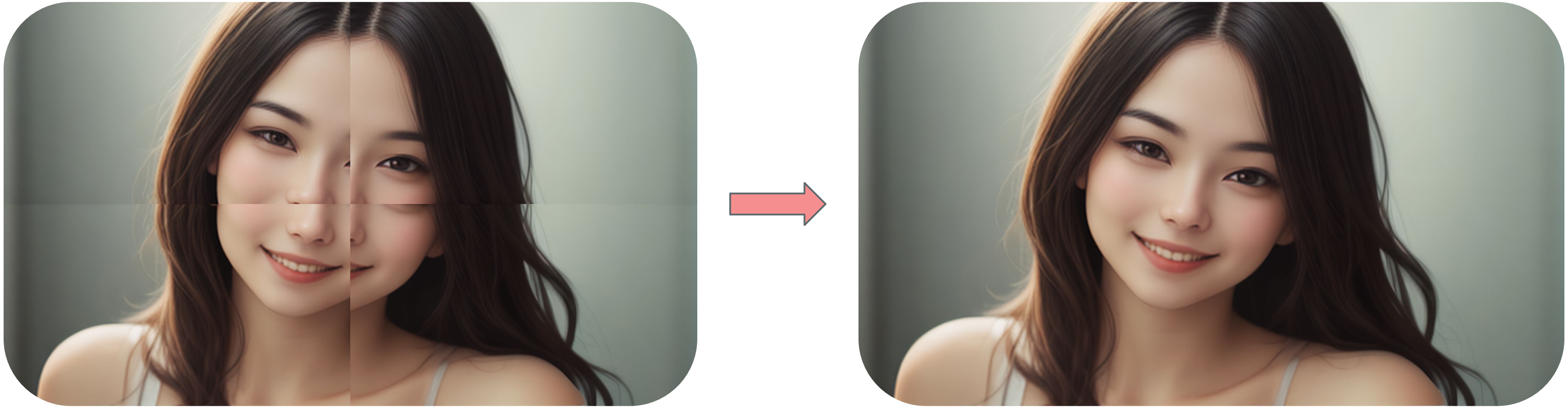
Blended Image

Alpha Blending For Seam Removal

- Desired Output = 2560 x 1440
- Initial Image Size = 640 x 360
- Overlap = 30 pixels (helps with blending and 30 pixels works best)
- Generated Quadrants = (1280 + 30) x (720 + 30)
- Blend Region is 4*overlap
  - Experimentally determined best blend ratio to balance seam removal and image sharpness near the seam
- Final Output = 2560 x 1440

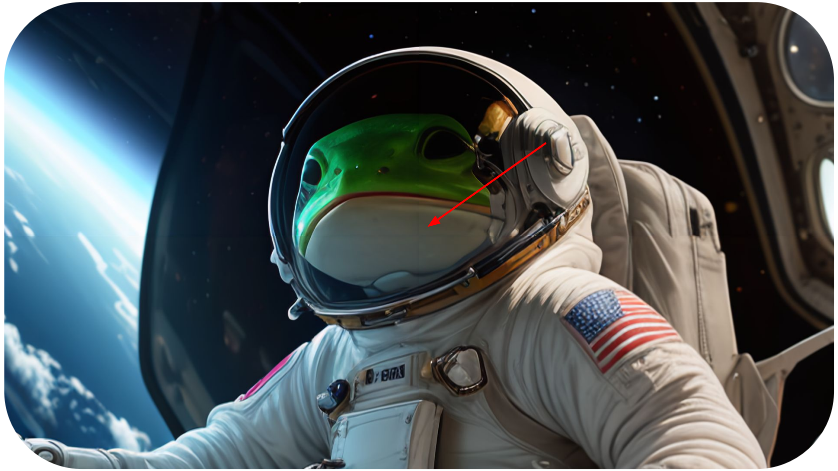
Image Smoothing

- Generate overlap between images for improved smoothing quality
- Separating the images into chunks reduces memory requirements
- Can be done in a parallel fashion

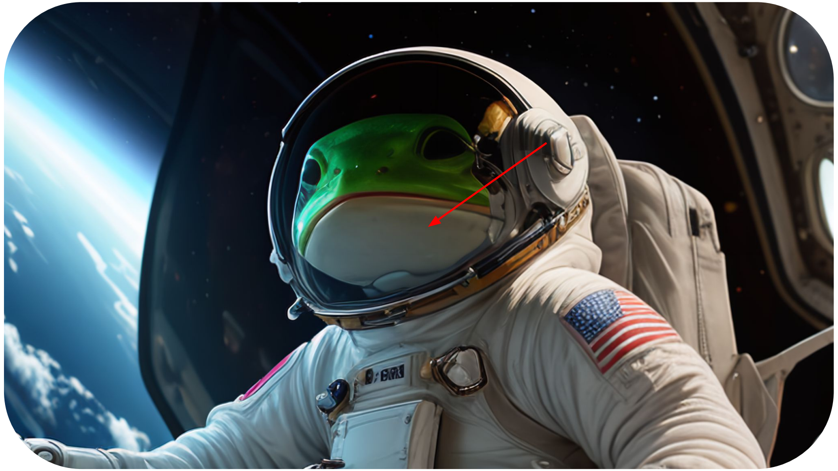


person smiling at the camera

Image Comparison Before and After Seam Removal



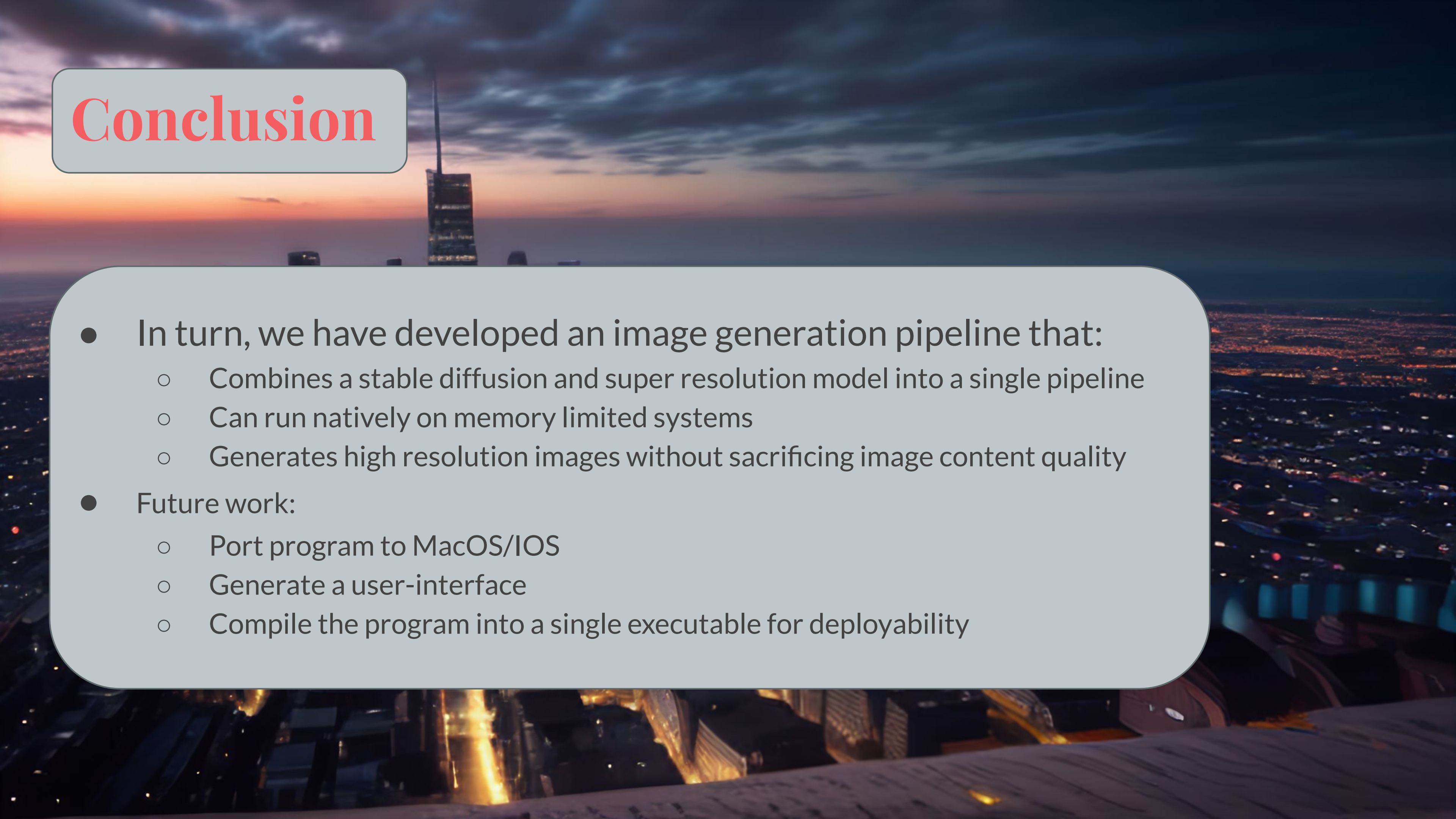
Seam



Seam removed

Performance on NVIDIA GeForce RTX 3070

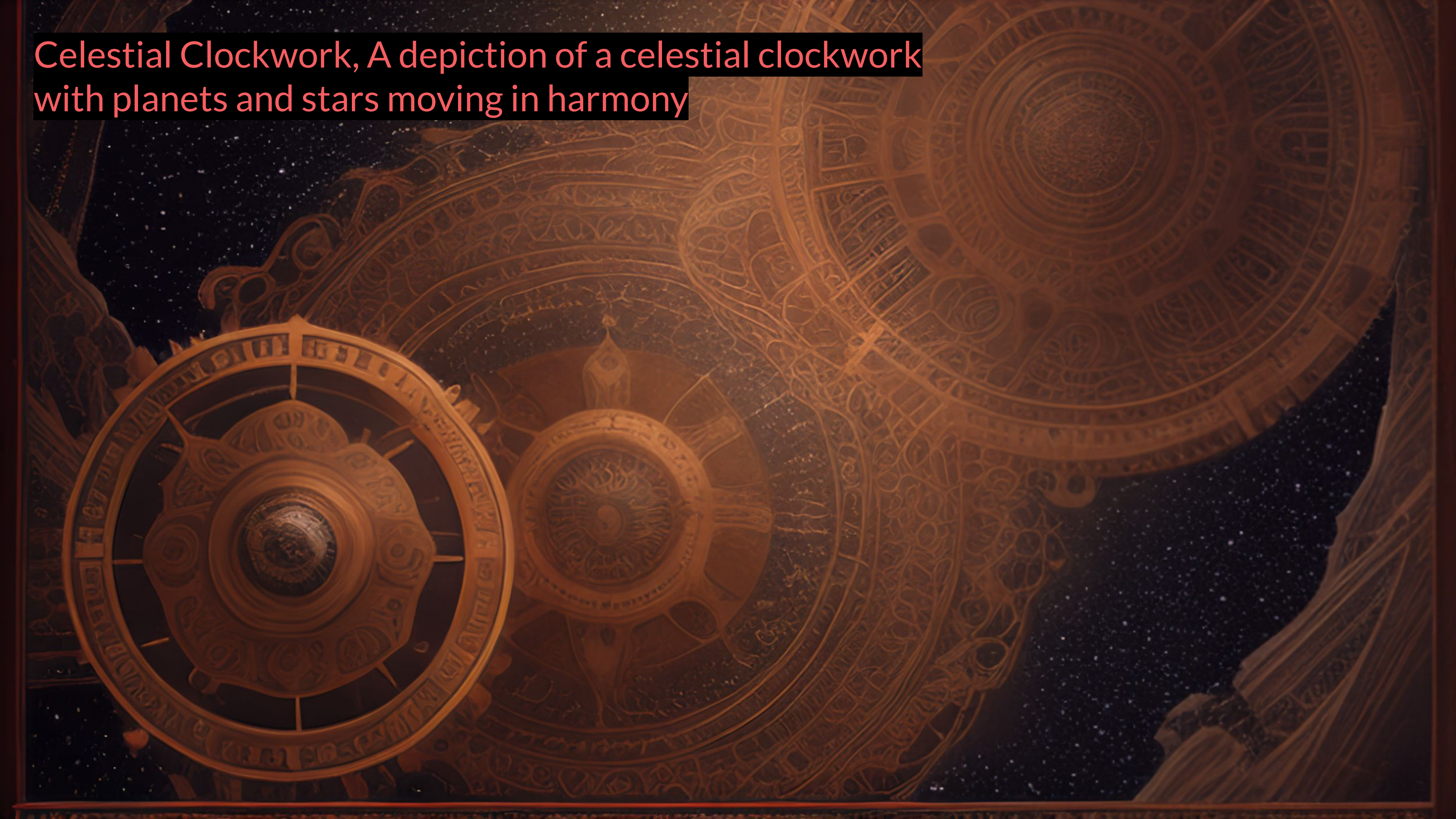
- Memory
  - Image generation: 1.5 Gb
  - Super resolution: 7.7 Gb
    - (Can be reduced by performing sequential CPU offloading of the image) estimated load of ~ 2.0Gb
- Time
  - Image generation: ~7 seconds
  - Super resolution: ~15 seconds



Conclusion

- In turn, we have developed an image generation pipeline that:
  - Combines a stable diffusion and super resolution model into a single pipeline
  - Can run natively on memory limited systems
  - Generates high resolution images without sacrificing image content quality
- Future work:
  - Port program to MacOS/IOS
  - Generate a user-interface
  - Compile the program into a single executable for deployability

Demonstration



Celestial Clockwork, A depiction of a celestial clockwork with planets and stars moving in harmony

Japanese Cherry Blossoms, A serene Japanese garden in spring, with cherry blossoms in full bloom

Winter Wonderland, A magical snowy landscape with a cozy cabin with a penguin

Space Odyssey, A stunning view of a distant galaxy from the window of a spaceship, with planets, stars

Aurora Over Mountains, Northern Lights dancing over snow-capped mountains under a starry sky

Redwood forest national park, photo realistic

Source Code