VERSION 02: USING DIFFUSION-BASED RESIZING

TABLE OF CONTENTS

1. Why

- Using scaling methods such as **INTER_LANCZOS**, the main subject in the image becomes **larger than its original size**, which violates the requirement of preserving visual integrity.

=> To address this issue, using **diffusion-based resizing** helps the **original image remains unchanged**, and the additional area is generated based on contextual understanding instead of geometric scaling.

2. The big picture

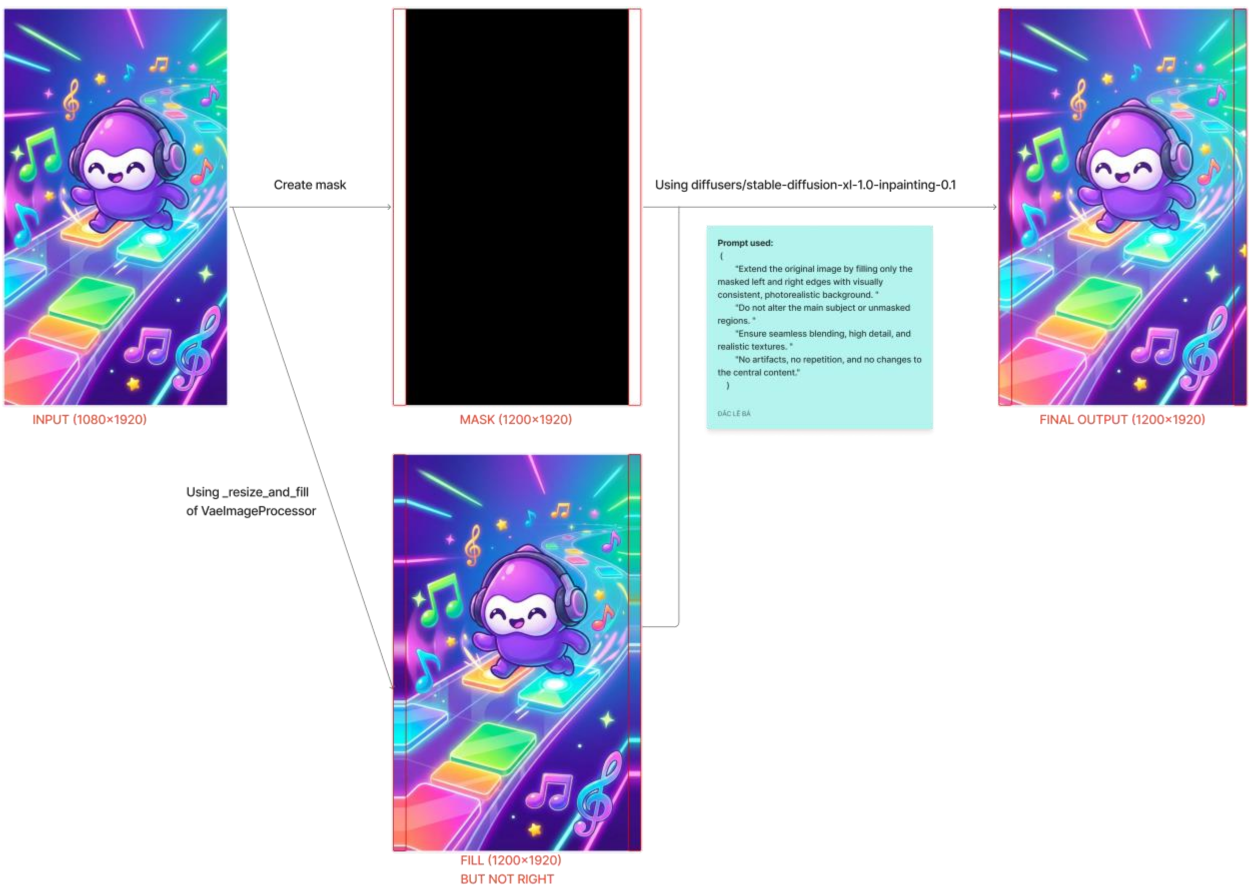

Figure 1: The big picture when using Diffusion

Step 1: Create a mask by placing the original image in the center.

- The **center area** (original image) is filled with **black pixels**
- The **left and right sides** are filled with **white pixels** to indicate regions to be generated

Step 2**:** From the original image, use _resize_and_fill to extend the image on both sides while keeping the main characters in the center unchanged.

```
# Resize image using VaeImageProcessor for best quality
processor = VaeImageProcessor()
resized_image = processor._resize_and_fill(image, width=target_w, height=target_h)
```

Step 3: Apply the diffusion model to fill only the masked areas (the left and right sides) of the resized image.

```
prompt = (
    "Extend the original image by filling only the masked left and right edges with visually cor
    "Do not alter the main subject or unmasked regions. "
    "Ensure seamless blending, high detail, and realistic textures. "
    "No artifacts, no repetition, and no changes to the central content."
)

result = pipeline(
    prompt=prompt,
    image=resized_image,
    mask_image=mask,
    height=target_h,
    width=target_w,
    strength=0.99
).images[0]
end = time.time()
logging.info(f"Resized with AI fill in {end - now:.2f}s")
return result
```

3. Results

3.1. The time inference and the GPU MEMORY

- Inference time is nearly 100× higher compared to Lanczos4

```
INFO: Resized with AI fill in 97.19s
```

```
INFO: Resized 5.png with Lanczos4 in 0.01s
```

- Consumes **more GPU memory**:

```
GPU-MEM
=========
11116MiB
```

=> Although computationally expensive, this approach provides superior visual quality and content preservation.

3.2. Compare with the best methods of version 1

Figure 2: Compare with LANCZOS4

➔ **INTER_LACZOS**: The main image is scaled up, causing characters and objects to appear larger than the original, which may distort proportions.

➔ **Diffusion-based resizing:** The main characters remain unchanged, and the newly generated regions are semantically related to the original scenario, resulting in a more natural and visually coherent output.

4. Trade-off

There is a clear trade-off between **computational efficiency** and **visual quality**:

- Interpolation methods are fast and lightweight but may distort content
- Diffusion-based methods preserve content fidelity at the cost of higher computation and GPU requirements

5. Limitations

- PSNR and SSIM metrics were not applied to AI-based filling because they are not suitable for evaluating generated content.
- The diffusion-based method uses a lot of GPU memory.
- The inference process takes much more time compared to interpolation methods.

6. Conclusions

- For this problem, where the width increase is relatively small (1080 → 1200 pixels), INTER_LANCZOS is sufficient and efficient.
- However, for scenarios requiring larger image expansion or stricter preservation of visual semantics, diffusion-based resizing becomes necessary and requires a GPU-enabled environment.

7. Future consideration

- Use more detailed prompts to better guide the diffusion model when generating new image areas.
- Optimize inference time by applying model quantization and knowledge distillation.
- Reduce GPU memory usage by using lighter models or lower-precision inference.