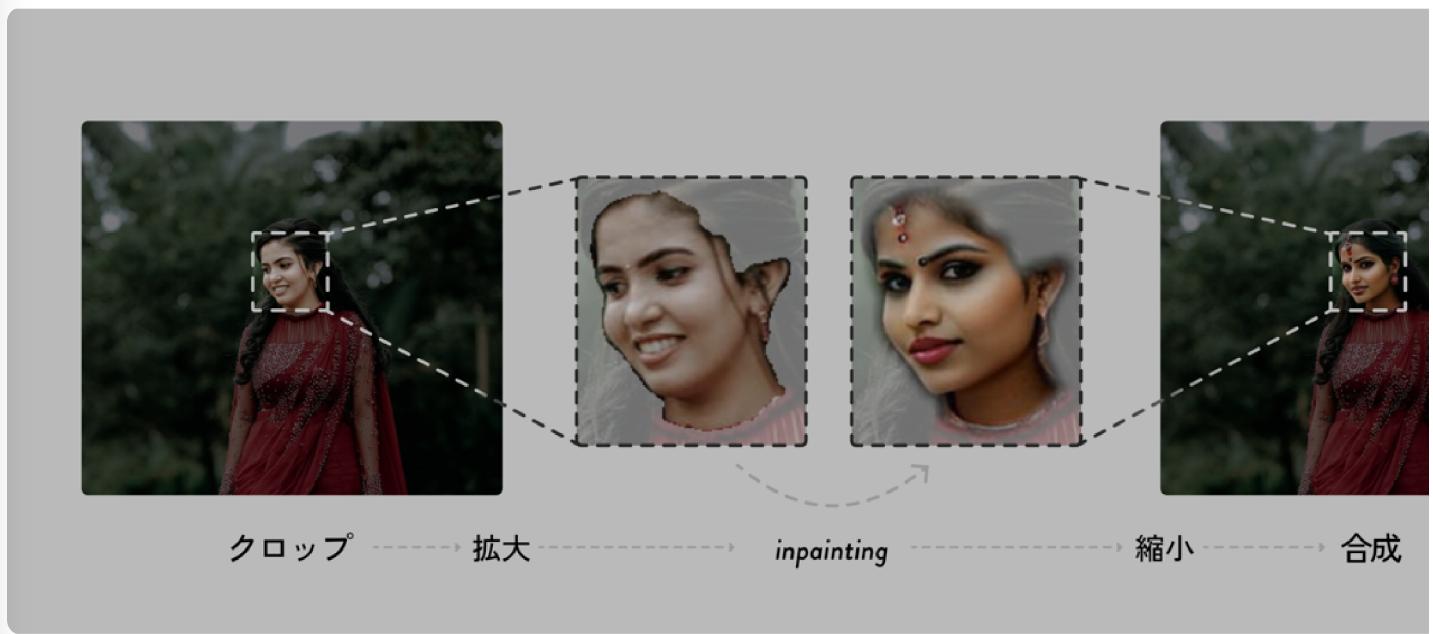


# Detailer

# Upscale & Restoration

## What is Detailer?



What Stable Diffusion 1.5 is best at is images around 512–768px. When drawing a full body at this resolution, the number of pixels available for one person's "face" is only about 30–50px.

There is also the problem of resolution, but looking from the whole screen, the face is too small, and it becomes "too detailed work" for AI. Like humans, AI is not good at detailed work.

Then, the idea came up: why not just cut out the face part, redraw it, and paste it back to the original image?

This series of processes "Crop → Resize/Upscale → Inpaint → Paste back to original" is called **Detailer**.

# Why is Detailer necessary?

There is also an idea "Why not just upscale the whole image and inpaint?".

However, the computational cost of inpainting is determined by the size of the original image, not the size of the mask. When there is a 4K image, even if you want to repaint only the face, calculation for the entire 4K image occurs, which is very inefficient.

Detailer is efficient because it cuts out only the surrounding area you want to inpaint.

---

## Custom Node

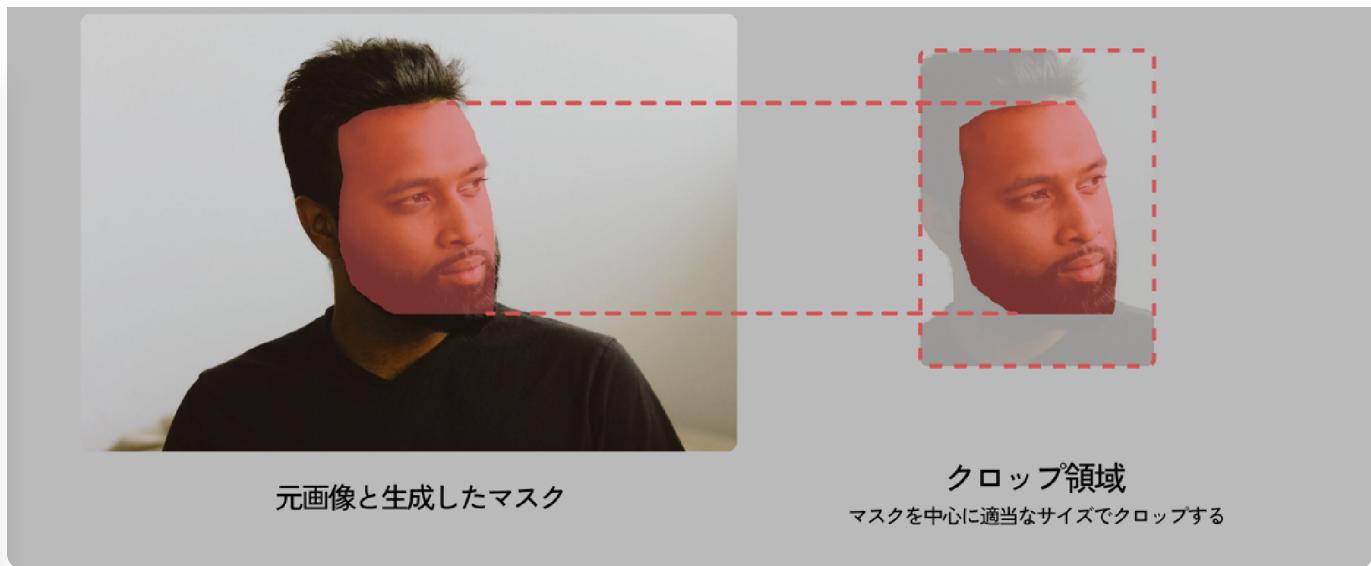
- [Iquesada/ComfyUI-Inpaint-CropAndStitch](#)

Generally, the Detailer node included in the Impact Pack is used. That one is more multifunctional because it can automate up to object detection, but since it has many unique parameters and is difficult, we will handle it separately.

---

## Mask and Crop Region

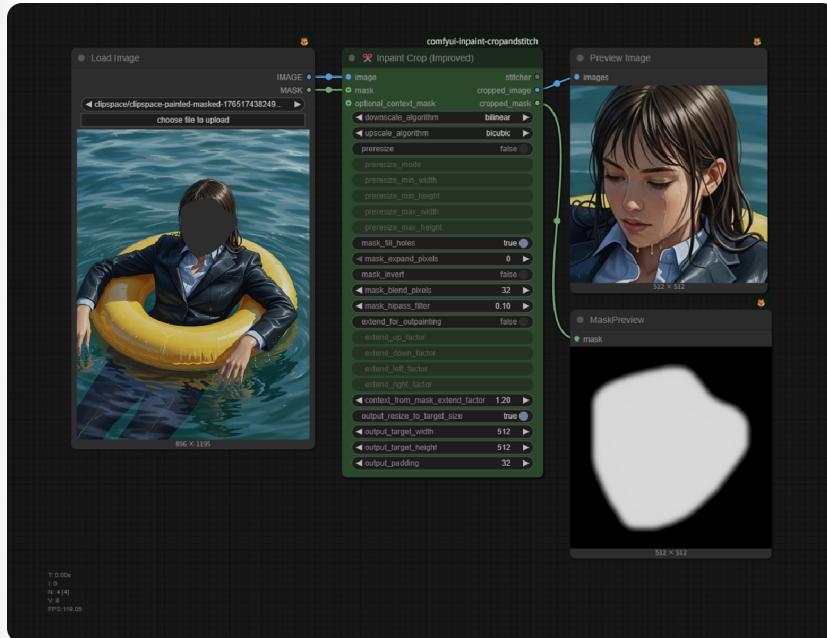
Let's check the difference between mask and crop region.



- Mask: The part you really want to rewrite (face itself, etc.)
- Crop region: "Working canvas" expanded slightly from the mask or BBOX

Detailer is inpainting performed only within this crop region.

## Inpaint Crop



Inpaint\_Crop\_(Improved).json  

As you can see from the workflow, if you pass **Mask + Original Image**, it automatically creates a crop region with a little margin added based on the mask, and resizes only that part

There are quite a few parameters, but basically, you only need to look at the ones below.

PARAMETER NAME	ROLE/MEANING
mask_fill_holes	Automatically fills small holes (missed spots) in the mask
mask_expand_pixels	Expands the boundary of the mask outward by specified pixels
mask_invert	Inverts the mask
mask_blend_pixels	Blurs the mask boundary.
🔥 context_from_mask_extend_factor	Specifies the "amount of margin" when creating a crop region from a mask by magnification
🔥 output_target_width	Output width after cropping (pixels). Specifies horizontal size of face canvas etc.
🔥 output_target_height	Output height after cropping (pixels). Specifies vertical size of face canvas etc.
output_padding	Adds margin so that resolution becomes a multiple of this value if necessary

## 🛠️ Inpaint Stitch (Improved)



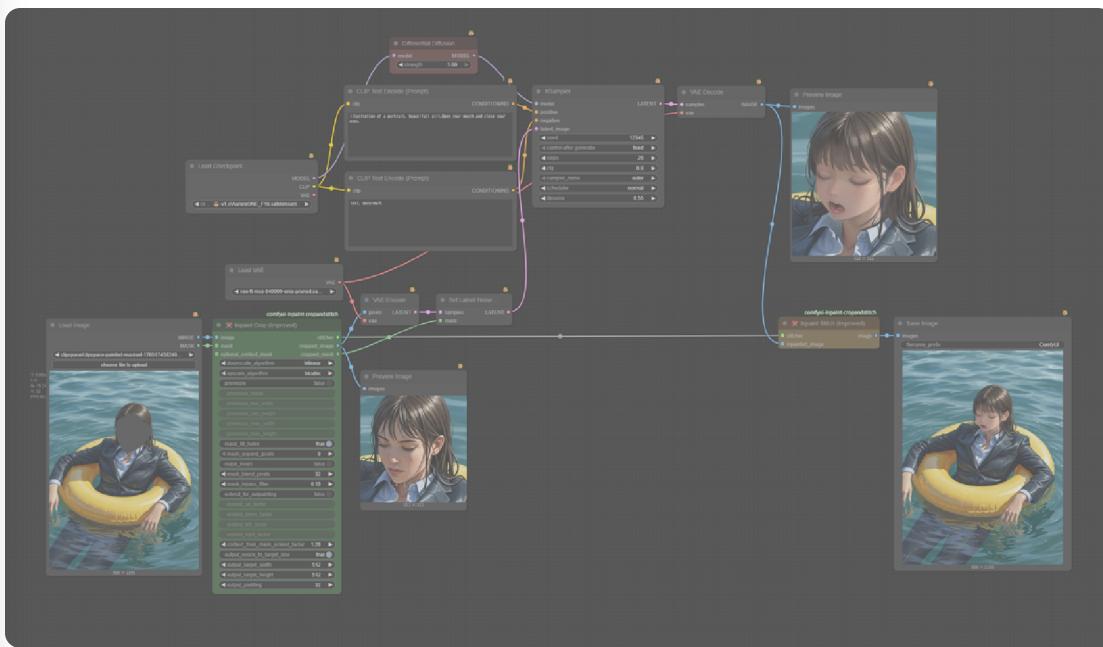
Inpaint\_Stitch\_(Improved).json

The Inpaint Stitch (Improved) node returns the modified crop image to its original position.

Only the masked part is overwritten on the original image.

## Manual Detailer with Inpaint Crop and Stitch

Now, let's try Detailer immediately. However, it's just incorporating it into the [inpainting](#) workflow.

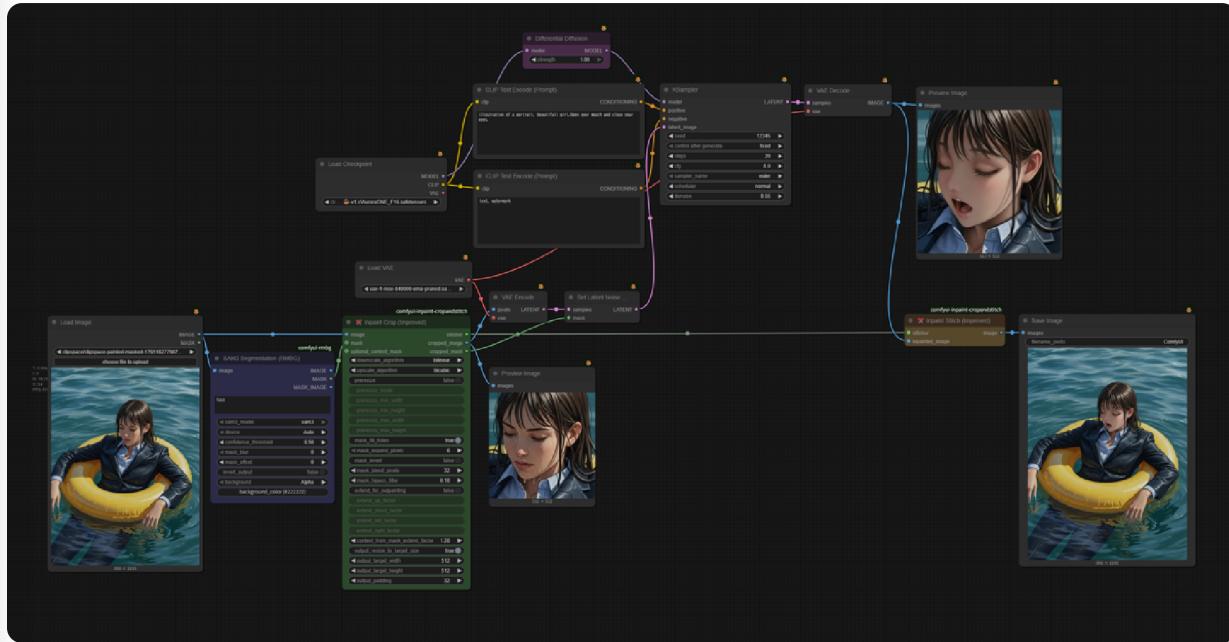


Detailer\_Inpaint\_Crop.json

- **Adjust output\_target\_width/height depending on the base model.**
  - Since it is SD1.5 this time, it is 512px.
- Although not directly related to Detailer, since Inpaint Crop outputs a "mask with blurred boundaries", it works very well with [Differential Diffusion](#) which can make use of

# Combining with Object Detection

Let's automate it a bit by automatically creating a face mask.



Detailer\_Inpaint\_Crop\_SAM3.json

- Create a face mask using SAM 3.

This is the basics of Detailer, so I think you can use it well enough. However, if you want to detect multiple people reflected in the image at once and process them all at once... you need to use ImpactPack.

→ [Detailer ImpactPack \(WIP\)](#)

## Sample Image



## Related workflows

### ESRGAN

Image upscaling and face restoration

### Ultimate SD Upscale

Super resolution upscale using Tile and ControlNet

