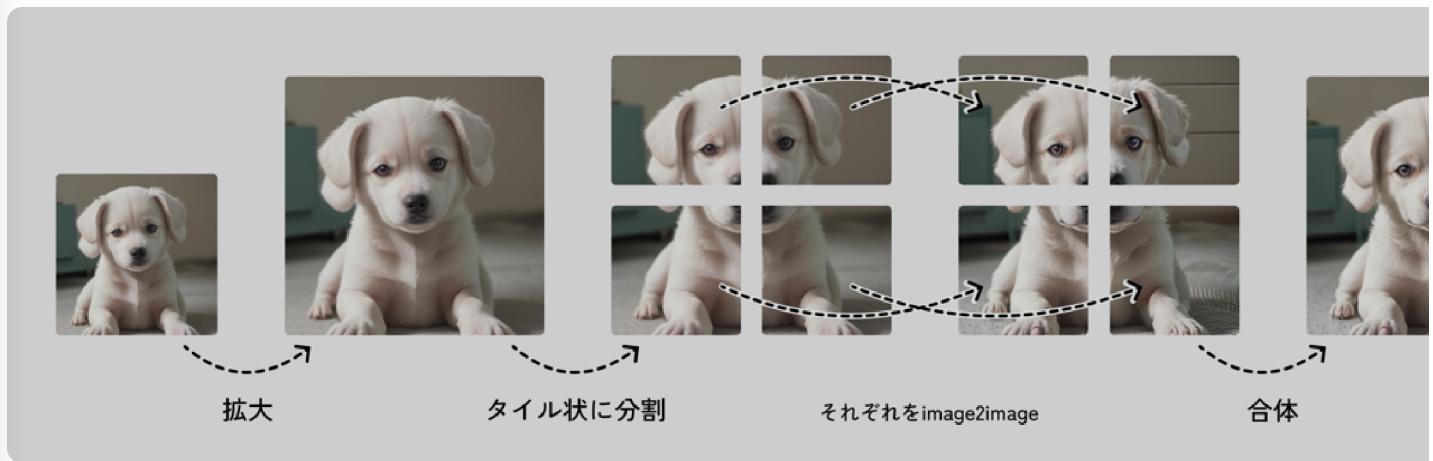


Ultimate SD Upscale

Upscale & Restoration # ControlNet Family

What is Ultimate SD upscale?



There were two reasons why Stable Diffusion could not generate large images: one was that it was not trained on large images, and another simple reason was the problem of computational cost.

Trying to generate a super high-resolution image like 4K or 8K in one go is quite severe in terms of VRAM and computation time.

Therefore, the idea was born not to make it all at once, but to divide the image and perform Hires.fix on each of them.

- 1. Enlarge the image
- 2. Divide into tiles
- 3. image2image for each tile individually
- 4. Connect the tiles at the end

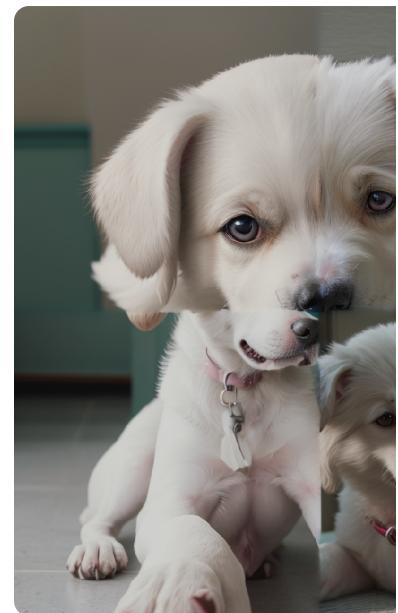
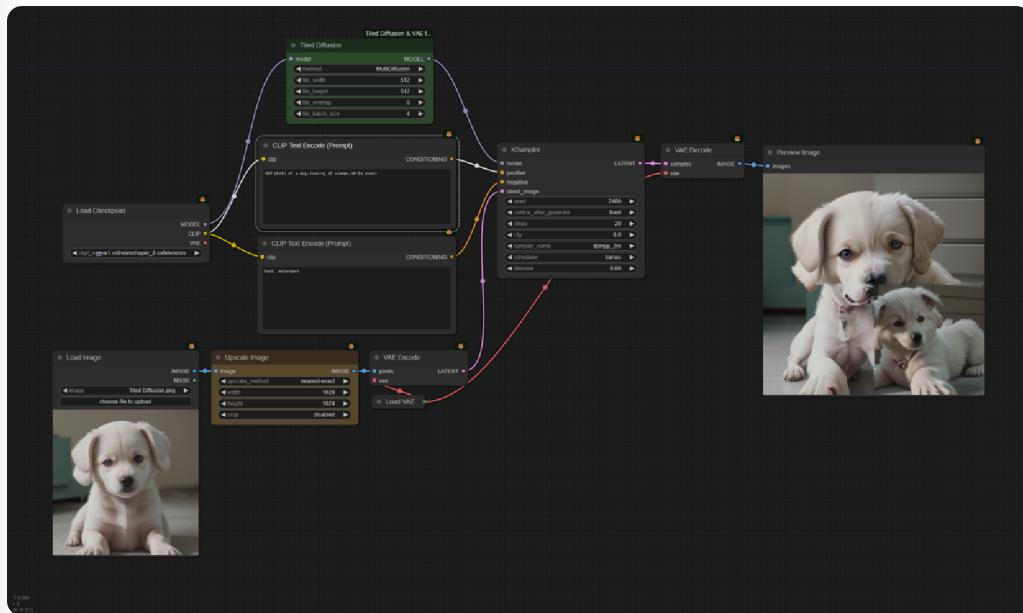
Custom Nodes

– [shiimizu/ComfyUI-TiledDiffusion](#) ↗

There is also a node exactly named [ssitu/ComfyUI_UltimateSDUpscale](#) ↗, but since we want to follow the principle this time, we will use the simple node above.

Weakness of Tile: Seams

First, let's look at the basic behavior of Tile. Here we will explain using the Tiled Diffusion node as an example, but the node can be anything as long as you understand the concept.



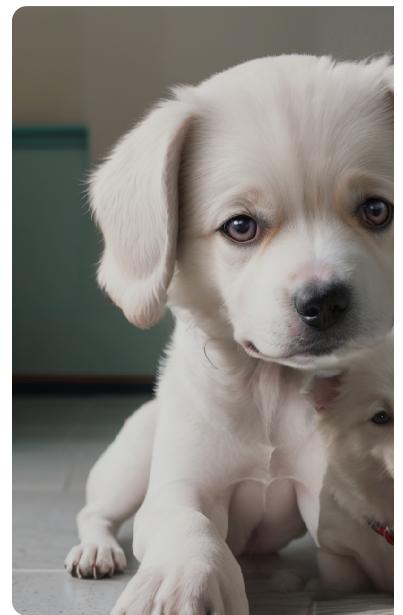
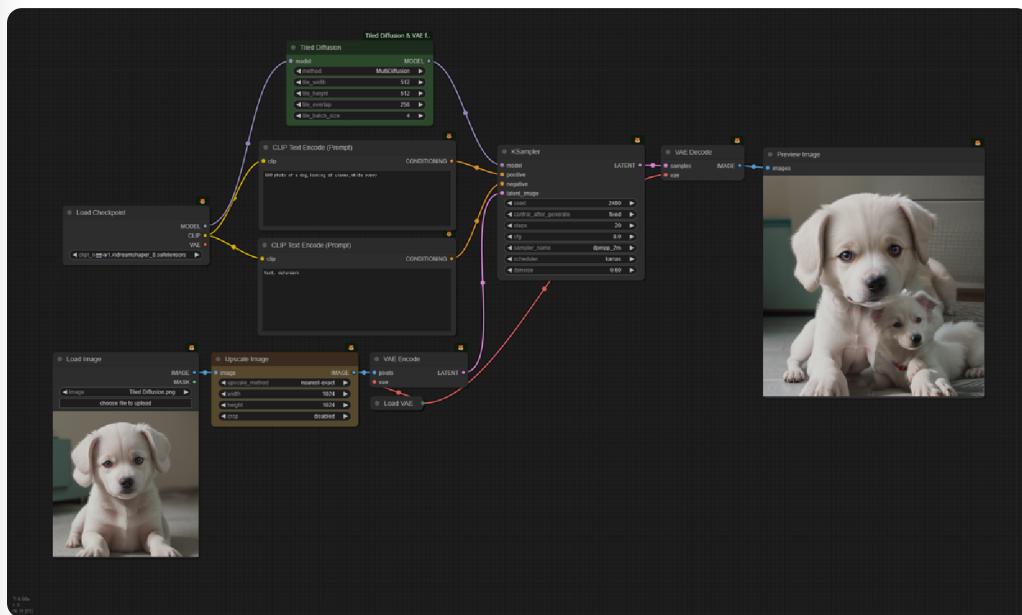
Tiled_Diffusion_overlap0.json

- Resize input image to 1024 × 1024 px
- Set tile size to 512 × 512 px

As you can see, the boundaries of the tiles are clearly visible, and the unity of the screen as a whole is weak. This is the **first weakness** of Tile.

Blending seams with overlap

If you are concerned about boundaries, you can arrange the tiles with a slight overlap. This is `tile_overlap`.



Tiled_Diffusion_overlap256.json [Open](#) [Download](#)

- Set `tile_overlap` to 256px
- Instead of lining up tiles neatly, imagine arranging them intentionally overlapping by about half.



The overlapped part acts like a cushion where adjacent tiles share information, so the boundary blends in as sampling progresses, making the seams of the tiles less noticeable.

However, increasing the overlap means sampling the same area multiple times, so the generation time increases.

Another weakness of Tile: Prompt

Tile has another major weakness. Since the same prompt is used for all tiles, unnecessary things are generated in unexpected places.

Let's try generating with settings like `tile_overlap = 0 / denoise = 1` in the previous workflow, and writing only `a dog` in the prompt. Then, as shown in the image, many dogs appear in one image.

Since it tries to generate one dog in each of the top left, top right, bottom left, and bottom right tiles, four dogs are drawn as a whole. This is the **second weakness of Tile**.

Idea of changing prompt for each tile

In theory, you could consider writing a separate prompt for each tile.

- Top left tile: `dog right ear, right eye`
- Top right tile: `dog left ear, left eye`
- Bottom left tile: `dog front leg`
- Bottom right tile: `dog back leg`

If you do this, every tile should understand "I only need to be responsible for the ear".

However, in practice, this is rarely used.

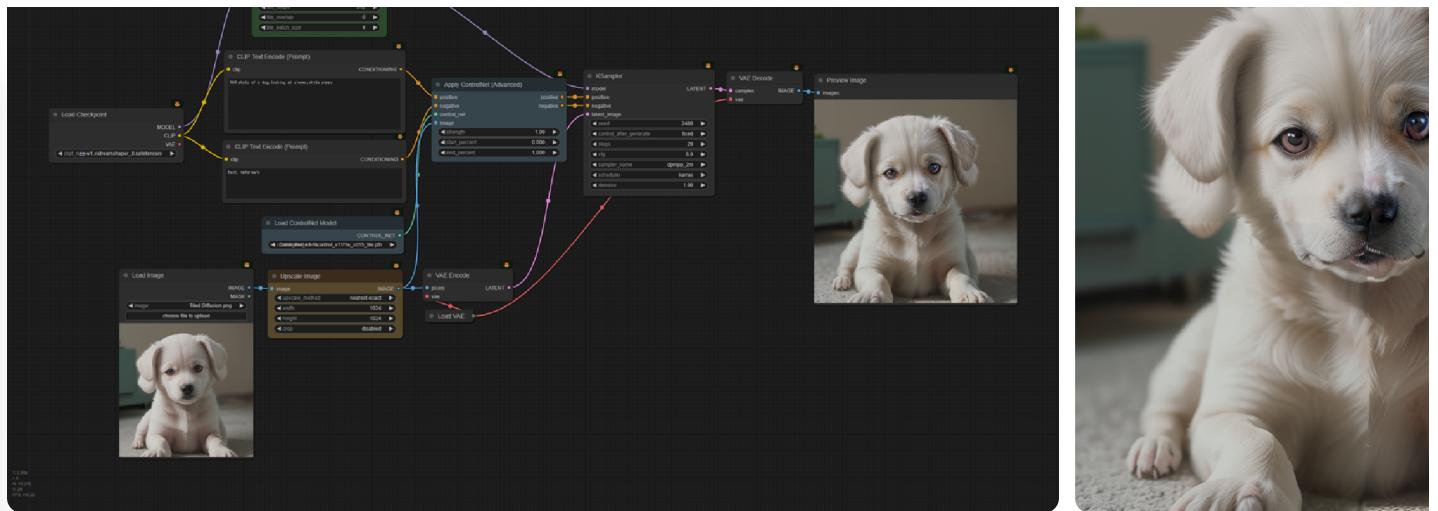
Writing prompts for the number of tiles is not realistic, and above all, Stable Diffusion cannot understand and differentiate prompts like "only the top right quarter of a dog's face".

Fixing structure with ControlNet Tile

This is where **ControlNet Tile** comes in.

ControlNet Tile is a ControlNet that generates a new image while **maintaining the structure of the input image quite strongly**.

It does not copy pixels as they are, but behaves like repainting textures and details while maintaining the **rough shape and positional relationship of objects**.



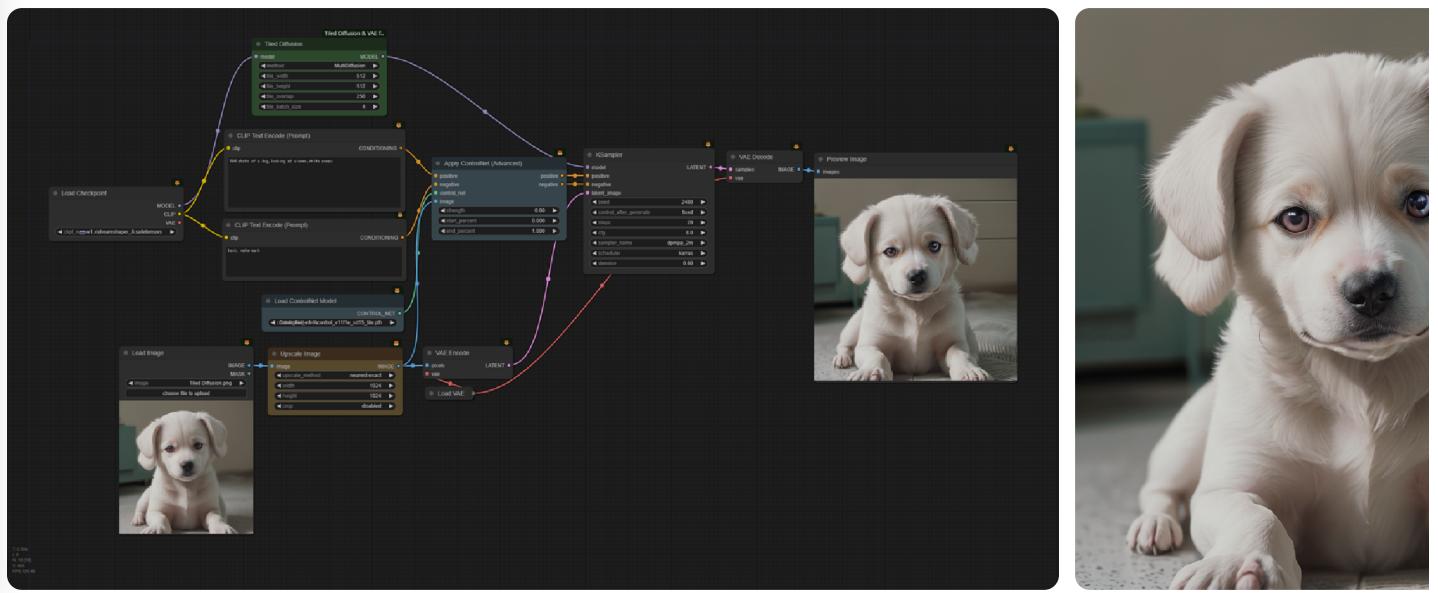
TiledDiffusion_ControlNet_Tile.json [🔗](#) [⬇️](#)

In this workflow, we dare to set `tile_overlap = 0`, `denoise = 1`, which makes the weakness of Tile most visible.

Still, you can see that by passing through ControlNet Tile, upscaling is possible while keeping the composition of the original image to a considerable extent.

Finishing with overlap x ControlNet Tile

Combining the elements so far, practical Tile upscale form comes into view.



-  overlap 256px
-  Controlnet strength 0.6

It has become a much more natural finish.

Summary: The concept of Ultimate SD upscale

The essence of Ultimate SD upscale is the following three pillars.

- 1. **Tile (Tile Division)** Instead of handling large images as they are, divide them into tiles and perform image2image to aim for super-resolution while suppressing VRAM and computation time load.
- 2. **overlap (Tile Overlap)** Arrange tiles slightly overlapping each other and blend boundaries during the sampling process to make seams unnoticeable.
- 3. **ControlNet Tile (fixing structure)** By upscaling tiles while strongly maintaining the structure of the input image, suppress the "dog inside dog" problem and the problem that the whole becomes disjointed.

Actual Ultimate SD upscale nodes and presets are just packaging this concept into a single node.

By the way, a similar idea can be applied to video generation, where frames are divided. It's like taking 20 frames at a time from a 100-frame video and overlapping 5 frames. We won't cover details here, but the point of dividing finely to reduce calculation cost is exactly the same.

Related workflows

ControlNet Preprocessor

Creating auxiliary images for use with ControlNet

Detailer

A mechanism to cut out only small faces or details and inpaint them

ESRGAN

Image upscaling and face restoration

Flux.1 Tools

How to use Flux.1 Tools

ControlNet

Controlling image generation using poses and line drawings
