



# VR we are

## User Manual



### Content

Version 4.1, 1<sup>st</sup> edition

© 2025 Fortuna Cournot, <https://www.3d-gallery.org>

<b>2</b>	<b>Introduction .....</b>	<b>6</b>
2.1	Scenarios for using VR we are.....	7
2.1.1	Non-AI Creator: Convert private images and videos to side-by-side .....	7
2.1.2	For AI-Creators: Improve productivity by App and Pipeline .....	7
2.1.3	Simple for ComfyUI Experts: Just use Custom Nodes in existing ComfyUI installation.....	7
<b>3</b>	<b>Getting Started .....</b>	<b>8</b>
3.1	System Requirements.....	8
3.2	Installation Steps .....	9
3.2.1	Installation Prerequisites: Tools.....	9
3.2.2	Install Script .....	9
3.2.3	Custom Nodes Only.....	10
3.2.4	ComfyUI Workflows .....	10
3.2.4.1	External Image and Video Generators.....	10
3.2.4.2	Wan2.2 14B Pipeline-Injectors.....	11
3.2.4.3	I2I SBS-Converter .....	11
3.2.4.4	I2I SBS Depth-Analysis.....	12
3.2.4.5	I2I Watermark-Analysis .....	12
3.3	Use Case Examples.....	13
3.3.1	Convert a video to Side-By-Side for VR .....	13
3.3.2	Create a Side-By-Side image for VR .....	14
3.3.3	Create Videos from T2I and I2V workflows using the pipeline .....	14
3.3.3.1	Generate and select reference images.....	14
3.3.3.2	Generate video chunks .....	17
3.3.3.3	Build chunk and go for the next .....	18
3.3.3.4	Build video from sequence.....	18
3.3.3.5	Scene change detection.....	18
3.3.4	Other models and generators .....	18
<b>4</b>	<b>Setup of the VR headset .....</b>	<b>19</b>
4.1	Skybox VR Player .....	19
4.2	DEO VR Player .....	19
4.3	4XVR Video Player .....	20
4.4	potplayer.....	20
<b>5</b>	<b>System Overview .....</b>	<b>21</b>
5.1	User Interface.....	22
5.2	Display Toggles.....	22
5.3	Tool Pipeline.....	23
5.3.1	Customizing Pipeline .....	24
5.3.1.1	Video Stream Keys .....	25
5.3.1.2	Image Stream Keys.....	25
5.3.1.3	Audio Stream Keys .....	25
5.3.1.4	Other Keys .....	25
5.3.1.5	System Keys .....	25
5.3.1.6	Operators .....	25
5.3.1.7	Values .....	25
<b>6</b>	<b>Review and Release System.....</b>	<b>26</b>
6.1	Preparation.....	27
6.2	Edit: Crop, Trim, Snapshots.....	28
6.3	Rating .....	30
6.4	Release .....	31
<b>7</b>	<b>Stages .....</b>	<b>32</b>

7.1	<i>caption</i> .....	32
7.2	<i>scaling</i> .....	32
7.3	<i>fullsbs</i> .....	33
7.4	<i>interpolate</i> .....	33
7.5	<i>singleloop</i> .....	33
7.6	<i>dubbing/sfx</i> .....	33
7.7	<i>watermark/encrypt</i> .....	33
7.8	<i>watermark/decrypt</i> .....	34
7.9	<i>concat</i> .....	34
7.10	<i>check</i> .....	34
<b>8</b>	<b>Slideshow stages</b> .....	<b>35</b>
8.1	<i>slides</i> .....	35
8.2	<i>slideshow</i> .....	35
<b>9</b>	<b>Tasks</b> .....	<b>36</b>
9.1	<i>FFmpeg-based</i> .....	36
9.1.1	convertimage, convertvideo .....	36
9.1.2	fpslimit-30, fpslimit-15 .....	36
9.1.3	vlimit-2160p, vlimit-1080p, hlimit-2160 .....	36
9.1.4	limit-1920, limit-1600 .....	36
9.1.5	reencode .....	36
9.1.6	pad-16-9, pad-16-9-sbs .....	36
9.1.7	credit-vr-we-are, credit-no-sbs .....	36
9.1.8	desaturate, incsaturation .....	36
9.1.9	split-1m .....	37
9.1.10	createsilence, stripaudio .....	37
9.1.11	swapsides .....	37
9.2	<i>ComfyUI-workflow-based</i> .....	38
9.2.1	archive, trashbin .....	38
9.3	<i>Extra Content</i> .....	39
9.3.1	Workflow stage: recolor .....	39
<b>10</b>	<b>Task Customization</b> .....	<b>40</b>
<b>11</b>	<b>Stereoscopic - A Full Stereo Side By Side Converter</b> .....	<b>41</b>
11.1	<i>Background</i> .....	41
11.2	<i>Depth Map</i> .....	42
11.3	<i>SBS Conversion Algorithm</i> .....	43
<b>12</b>	<b>ComfyUI Custom Node Library</b> .....	<b>45</b>
12.1	<i>Convert to VR (ImageVRConverter)</i> .....	45
12.2	<i>Resolution Info</i> .....	45
12.3	<i>Encrypt Watermark</i> .....	45
12.4	<i>Decrypt Watermark</i> .....	45
12.5	<i>Strip XML</i> .....	45
12.6	<i>Save Stripped UTF-8 File</i> .....	45
12.7	<i>Scale by Factor</i> .....	45
12.8	<i>ScaleToResolution</i> .....	46

12.9	<i>Calculate Dimensions</i>	46
12.10	<i>RegexSubstitute</i>	46
12.11	<i>Strftime</i>	46
12.12	<i>Save Audio (Simple)</i>	46
12.13	<i>VRwearePause</i>	46
12.14	<i>VRwearePauseLatent</i>	46
12.15	<i>VRweareResume</i>	46
12.16	<i>Load Image Advanced</i>	47
12.17	<i>Linear Fade</i>	47
12.18	<i>Color Correct Batch</i>	47
<b>13</b>	<b>Random Prompts and LoRa Training</b>	<b>49</b>
13.1	<i>Custom Nodes</i>	49
13.1.1	Variant Prompt Builder	49
13.1.2	Join Variant Prompt Builder	49
13.1.3	Build Variant Index, Get Variant	50
13.2	<i>Variant Specification</i>	51
13.3	<i>Workflow blueprints</i>	51
13.3.1	Create_Synthetic_Testdata_Variant_BLUEPRINT	51
13.4	<i>Scripts</i>	52
13.4.1	Create Synthetic Dataset	52
13.5	<i>Tasks</i>	52
13.5.1	Stabilize-Color	52
<b>14</b>	<b>Optional Tools</b>	<b>53</b>
14.1	<i>Topaz Video AI</i>	53
<b>15</b>	<b>Troubleshooting</b>	<b>54</b>
<b>16</b>	<b>Manual Install (with your own ComfyUI)</b>	<b>55</b>
16.1	<i>ComfyUI</i>	55
16.1.1	Choose Installation Option	55
16.1.1.1	Option 1: Windows Portable Package	55
16.1.1.2	Option 2: ComfyUI Desktop (not regularly tested! Limited support.)	55
16.1.2	Install Prerequisites on ComfyUI	55
16.1.2.1	Install ComfyUI Custom Nodes	55
16.1.2.2	Download VR we are Custom nodes (comfyui_stereoscopic)	56
16.1.2.3	Fix Dependencies and Start	56
16.1.2.4	Install ComfyUI Models	57
16.1.2.4.1	Upscale Model	57
16.1.2.4.2	Depth Model	57
16.1.2.4.3	Dubbing Model	57
16.1.2.4.4	Controlnet Model <sup>V3.1</sup>	57
16.1.3	Manual Install „VR we are“	58
16.1.4	Configure „VR we are“	58
16.1.4.1	Configure for ComfyUI Desktop	58
16.1.5	Custom ComfyUI Pathes	59
<b>17</b>	<b>CLI-only Installation (without ComfyUI)</b>	<b>60</b>
17.1	<i>Side-By-Side CLI</i>	60
<b>18</b>	<b>Acknowledgements</b>	<b>61</b>
<b>19</b>	<b>Appendix: Default Pipeline</b>	<b>63</b>





Source image generated with ComfyUI and SDXL

## 2 Introduction

Welcome to „VR we are“, an immersive productivity and creativity software designed to bring the familiar multimedia content of videos and images into the world of a Virtual Reality (VR) environment, using Artificial Intelligence (AI) models on the local computer, and most of the processing can be done offline.

### Key features:

- Converter stage for 2D images and videos to 3D stereo side-by-side
- Stage concept: Stages for ffmpeg, ComfyUI and TVAI tasks like: upscaling, interpolation (double fps), dubbing, ... . User customizable tasks based on blueprint stages.
- Pipeline management: Automation of your tool chain based on stages. Customizable.
- User interface for status tracking, fast editing and rating results. Visual pipeline display.
- Windows installer for a specific minimal pre-tested set of ComfyUI portable and custom nodes.
- App with edit and rate features.

Top feature is conversion of 2D images and videos into full stereo side-by-side (SBS) left-right (LR), which can be visualized with viewers in VR headsets or glasses on 3D-capable TV displays.

Some other features can be used even without targeting a VR device or using non AI media input, such as: capture metadata, upscaling, frame interpolation, various ffmpeg tasks, dubbing or creating slideshow videos from images.

A central principle of the application is to use conventions over configuration. That said I tried to keep *required* configuration to a minimum by providing default settings wherever possible.

Installation is provided for a specific ComfyUI (portable) setup with a basic set of custom nodes.

## **2.1 Scenarios for using VR we are**

### **2.1.1 Non-AI Creator: Convert private images and videos to side-by-side**

Just want to convert images and videos to SBS? You put copies of them in folders below ...vrweare\ComfyUI\_windows\_portable\ComfyUI\input\vr\... , and pickup the results from the output folders.

Then read chapters

- 3.1, System Requirements
- 3.2.1, Installation Prerequisites: Tools
- 3.2.2, Install Script
- At install choose options:
  - Pipeline with SBS-Converter: On
  - Judging: Pipeline for FLI2V: Off
- 8, Slideshow stages
- 7.3, fullsbs
- 5.1, User Interface
- Appendix, Tool Pipeline

### **2.1.2 For AI-Creators: Improve productivity by App and Pipeline**

For AI-Creators, want to use the pipeline and app features, with or without side-by-side converter, please read chapters

- 3.1, System Requirements
- 3.2.1, Installation Prerequisites: Tools
- 3.2.2, Install Script
- At install choose options:
  - Pipeline with SBS-Converter: On or Off
  - Judging: Pipeline for FLI2V: On
- 8, Slideshow stages
- 7.3, fullsbs
- 5.1, User Interface
- Appendix, Tool Pipeline

### **2.1.3 Simple for ComfyUI Experts: Just use Custom Nodes in existing ComfyUI installation**

You just want to convert an image to side-by-side, using the custom nodes, and do not need the pipeline, daemon service or app? Then read chapters

- 16.1.2.2, Download VR we are Custom nodes (comfyui\_stereoscopic)
- 16.1.2.4.2, Depth Model
- 3.2.4.3, I2I SBS-Converter

### 3 Getting Started

This chapter explains the basic system requirements, installation steps, and setup of the VR headset.

Installation can be automatic with ComfyUI, manual with your own ComfyUI or just CLI tools.

#### 3.1 System Requirements

The AI tools require a modern computer to perform well. Nevertheless, small media files can be handled on older hardware.

- Operating System: Windows 10 or higher. (CLI tool run on Linux)
- RAM: 16 GB minimum, 64 GB or higher recommended.
- VRAM: Not required. 16 GB or higher recommended.
- GPU: Not required. CUDA-capable Nvidia GPUs supported.
- File system: 13 GB free space for installation, and another 10 GB free space is permanently required for „VR we are“ service operation.

Tested Software Versions:

- Windows 11, Windows 10. 64-bit
- 7-Zip 25.01 (must be preinstalled, required for installation)
- Git 2.51 (must be preinstalled)
- ComfyUI 0.3.62
- FFmpeg 8.0-full, 7.1.1-full,
- Exiftool 13.33, 13.34
- Topaz VideoAI 6.0.4

Supported media:

- Although the tool accept any media that FFmpeg, OpenCV or ComfyUI can handle, I mainly support
  - mp4 with h264 codec
  - webm
- Other file types may work depending on the stage, but are not consistently supported:
  - jpg
  - png
  - webp
- Media height should be even (divisible by 2).



## 3.2 Installation Steps

The installation goes in the following order:

- Install Tools
- Install ComfyUI
- Optionally Configure „VR we are“

Custom nodes packages are now cloned from github and for security checked out to a specific tag or commit. They will appear in Manager as nightly.

### 3.2.1 Installation Prerequisites: Tools

If you just want to use the ComfyUI nodes to create stereoscopic images or videos, you can skip tools and install script sections and go to chapter 17.<sup>1</sup>

Please install:

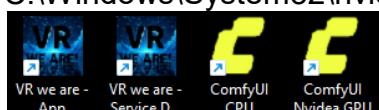
- Git 2.51 (for Windows) – you must preinstall this (accept all pages with defaults). <https://gitforwindows.org/>. I don't want to provide a auto-installer for it.
- 7-Zip 25.01<sup>2</sup> (must be preinstalled, required for installation).

If available the installation is used:

- Exiftool 13.33. Rename exe (strip -k) for usage in CLI to exiftool.exe . You must add folder to PATH environment variable. <https://exiftool.org/>
- FFmpeg 8.0. You must add folder to PATH environment variable. Otherwise it will be installed locally (+1GB). <https://ffmpeg.org/>
- Topaz Video AI (TVAI) – You need a current version (6.0.4); future versions will stop supporting CLI. Paths are detected during installation.

### 3.2.2 Install Script

To reduce conflicts, the installation of „VR we are“ it is recommend to use a separate ComfyUI installation (22 GB). Download the install script from a release (4.0 or higher) at [https://github.com/FortunaCournot/comfyui\\_stereoscopic/tree/main/installer](https://github.com/FortunaCournot/comfyui_stereoscopic/tree/main/installer), and execute it in a windows command prompt. It will guide you through a full managed installation and will take about 15 minutes. It will start ComfyUI, the „VR we are“ service and „VR we are“ App. For all of them desktop shortcuts are created. ComfyUI Nvidia shortcut only if C:\Windows\System32\nvidia-smi.exe exists:



**Sage Attention** is installed, if a local python and a Nvidia GPU is detected. Read Post<sup>3</sup> for prerequisites and troubleshooting. Remove flag from batch if necessary.

<sup>1</sup> To experiment and use SBS ComfyUI custom nodes use chapter 3.2.3.

<sup>2</sup> Security hazard for version 24.09 and below.

### 3.2.3 Custom Nodes Only

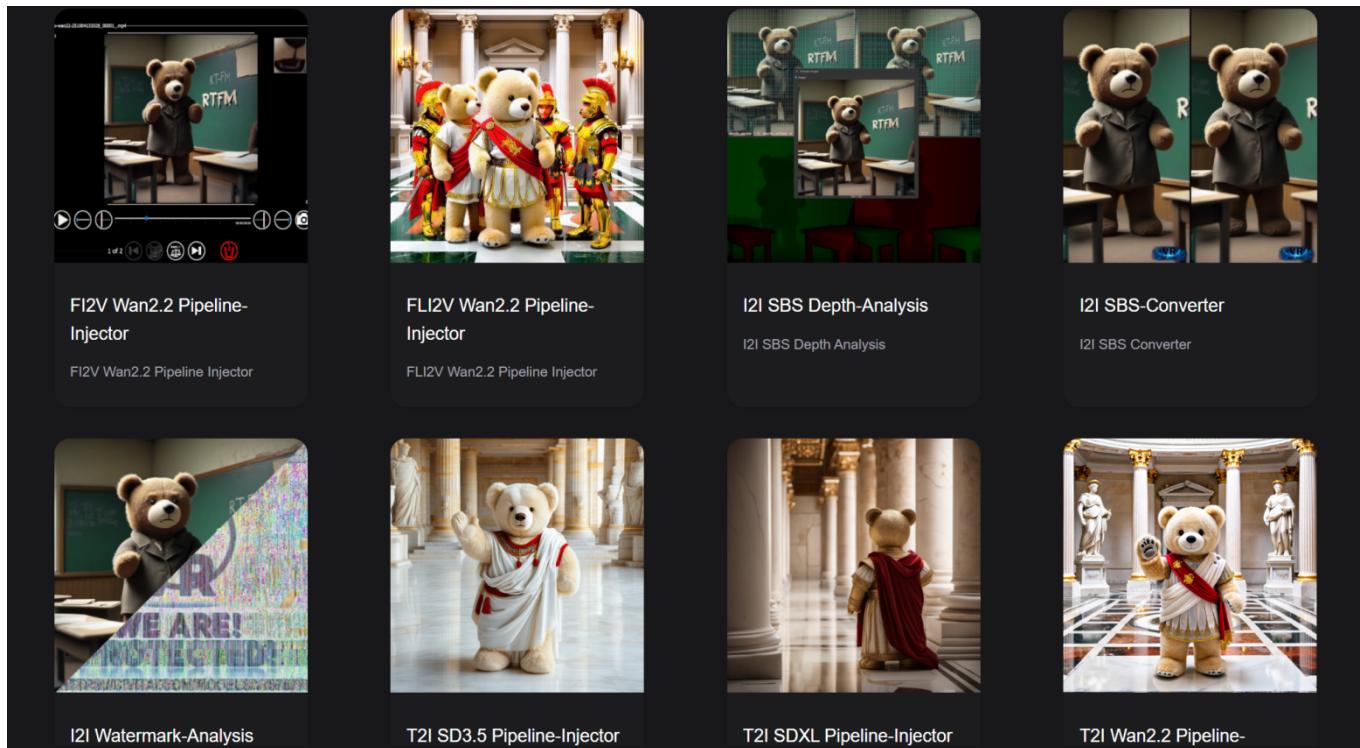
If you just want to use the ComfyUI nodes to create stereoscopic images or videos, use ComfyUI Manager to download „VR we are“ (see chapter 16.1.2.1).

### 3.2.4 ComfyUI Workflows



On the left side of ComfyUI you can open the templates using this icon:

At the bottom of the template list you see `comfyui_stereoscopic` (internal ComfyUI name of VR we are, also used on node labels), select it and you can choose a template (you must download some models manually):



#### 3.2.4.1 External Image and Video Generators

These injector workflows are made for those who wish create images and videos locally. But you can inject images and videos into the pipeline and app from any source, other ComfyUI workflows, Civitai Generator, Kling, Sora, ChatGPT, etc....

Read chapter 5 for the system overview.

To use the app to crop, trim rate resulting files, put them in  
...\\ComfyUI\\input\\vr\\check\\rate

To apply caption, scaling, convert to SBS, interpolate etc. put files in the stage's input folder.

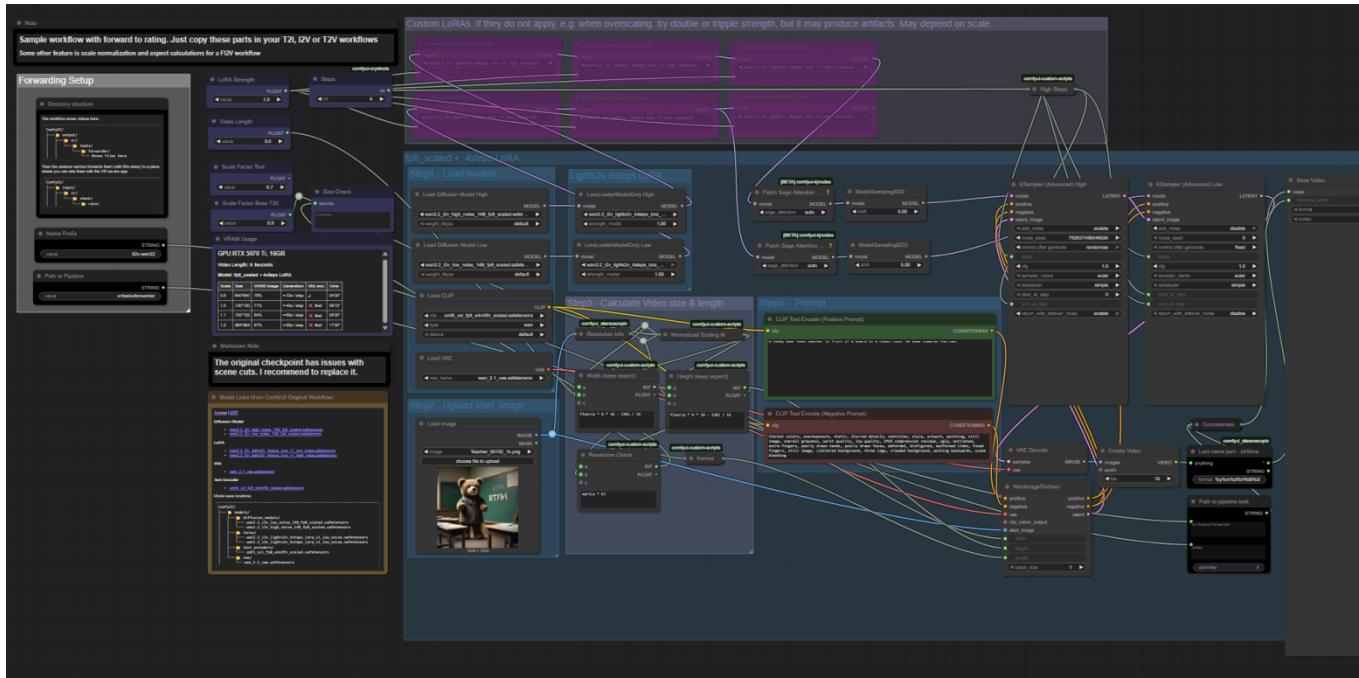
If your files have unsupported formats, use the tasks `convertimage` and `convertvideo` to let ffmpeg convert them for you (see chapter 9.1.1).

---

<sup>3</sup> [https://www.reddit.com/r/StableDiffusion/comments/1jle4re/how\\_to\\_run\\_a\\_rtx\\_5090\\_50xx\\_with\\_triton\\_and\\_sage/](https://www.reddit.com/r/StableDiffusion/comments/1jle4re/how_to_run_a_rtx_5090_50xx_with_triton_and_sage/)

### 3.2.4.2 Wan2.2 14B Pipeline-Injectors

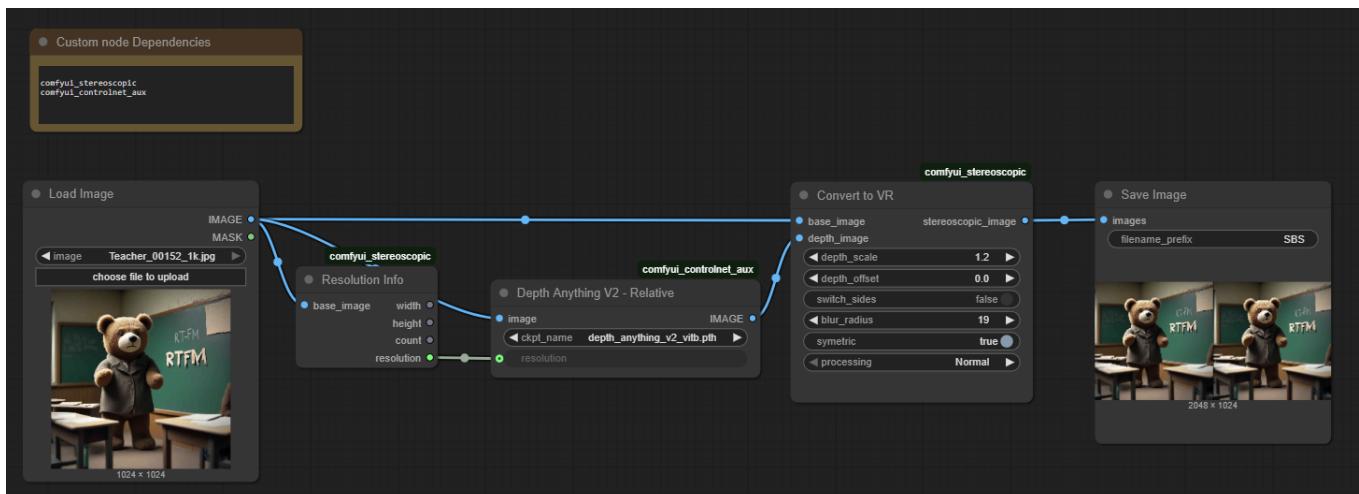
Start here to learn how to inject with your own workflows generated images and videos into the VR we are pipeline, ready to review the results, edit and rate them shortly after generation.



You can start right away, but i suggest to use different Wan 2.2 14B checkpoints for high/low. The **scope of the workflows is to illustrate how to inject results into the pipeline** and not the fabrication them self. Copy template parts into the workflows of your choice.

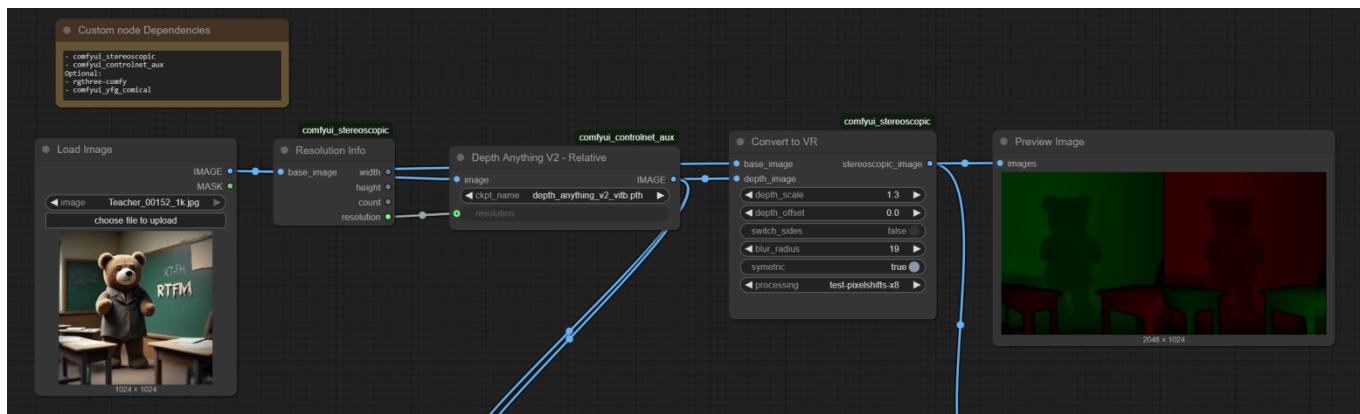
### 3.2.4.3 I2I SBS-Converter

Read more about it in chapter 11. Depth model is downloaded automatic on first usage – this may take a while (no progress bar in log!).



### 3.2.4.4 I2I SBS Depth-Analysis

For investigating the depth converter deeper you may use the workflow at ComfyUI\custom\_nodes\comfyui\_stereoscopic\examples\workflows\analysis\I2I\_SBS\_Depth\_A nalysis.json into the ComfyUI desktop screen (symbolic image of the workflow):



Read more about it in chapter 11.

### 3.2.4.5 I2I Watermark-Analysis

For investigating the watermark of an image, you need the image, the watermark image used and the secret number used in encryption.

### 3.3 Use Case Examples

The examples use Wan 2.2. If you don't want to use them (large models), look at chapter 3.3.4.

#### 3.3.1 Convert a video to Side-By-Side for VR

Ensure the pipeline is active .

Video should be mp4 or webm. For other convert by dropping in

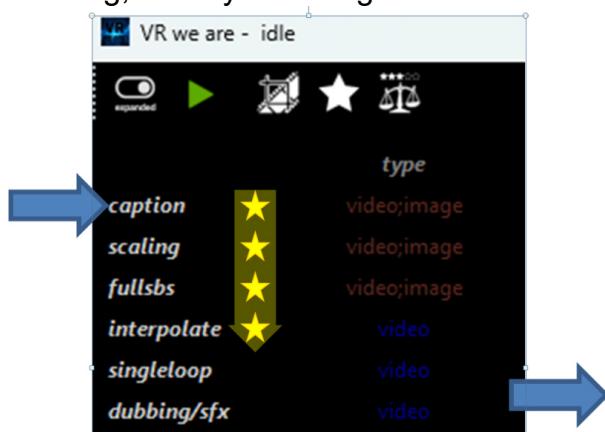
...\\ComfyUI\\input\\vr\\tasks\\convertvideo

And pick up in output.

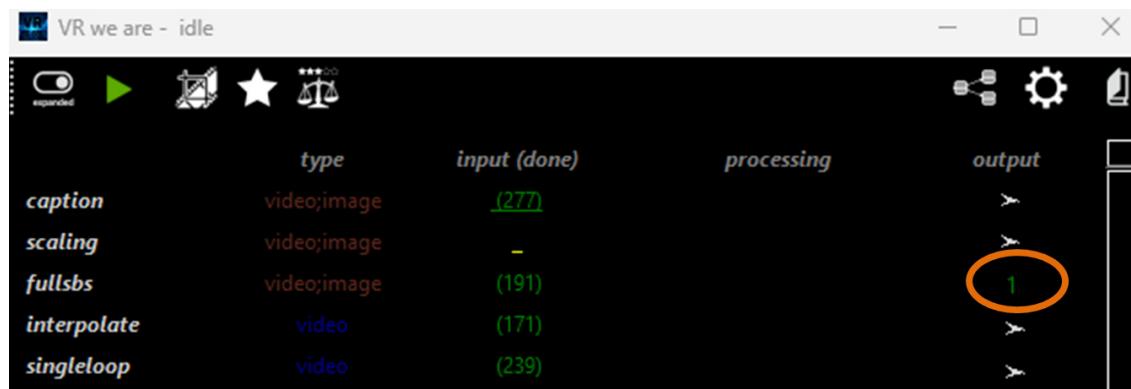
To convert a video to SBS for VR drop it into the first pipeline stage located at

...\\ComfyUI\\input\\vr\\caption

It will then go through a processing, usually including at least the following stages:



Processing will end, depending on input video at some of the stages below. Look in the output column:



	type	input (done)	processing	output
caption	video;image	(277)		>
scaling	video;image	-		>
fullsbs	video;image	(191)		
interpolate	video	(171)		>
singleloop	video	(239)		>

Click on the cell to spawn an explorer window for the folder to pick the results up.

Check out the pipeline  and chapters 7-9 for details about processing.



### 3.3.2 Create a Side-By-Side image for VR

Ensure the pipeline is active ▶.

Image should be png, jpg, jpeg or webp. For other convert by dropping in  
...\\ComfyUI\\input\\vr\\tasks\\convertimage

And pick up in output.

To convert a image to SBS for VR drop it into the first pipeline stage located at  
...\\ComfyUI\\input\\vr\\caption

The processing is similar to video, see above.

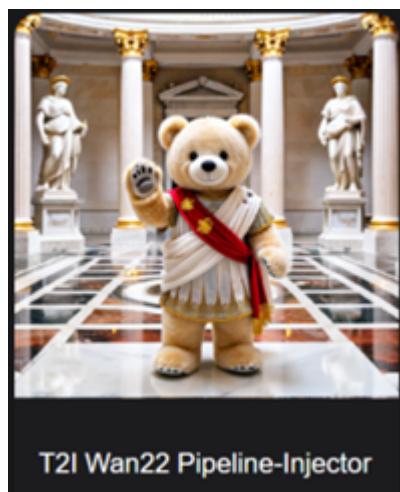
### 3.3.3 Create Videos from T2I and I2V workflows using the pipeline

Creating longer AI videos by a sequence of smaller chunks, is commonly done by a "First last image reference" approach. This refers to a feature in AI video generation where you provide a starting and ending image to guide the AI's creation of intermediate frames. This allows for greater control over the video's beginning, end, and overall motion, style, and structure, as the AI fills in the transition between the two reference points.

The reference images can be created by a txt2img (T2I) workflow, using text prompts to describe the content from scratch, while the img2vid (I2V) workflow is creating video chunks of approx. 6-10 seconds, by using the reference images and a prompt describing the transition.

#### 3.3.3.1 Generate and select reference images

In ComfyUI open Templates and navigate down to comfyui\_stereoscopic. The following template has been prepared from generating images for injection into the pipeline. Setup:

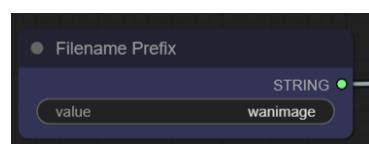


Check out the Model Links note in the workflow and install the models. I recommend downloading other diffusion models for Wan Video 2.2 T2V-A14B with High + Low Noise, that have fewer issues with scene blending. E.g.:

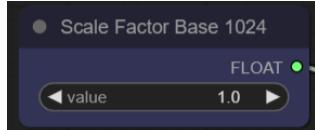
- Wan2.2-Remix-SFW T2V
- Wan2.2-Remix T2V
- Smooth Mix Wan 2.2 T2V

Use the nodes in this workflow to improve your own workflow to inject images into the pipeline.

You may want to change the file name prefix:

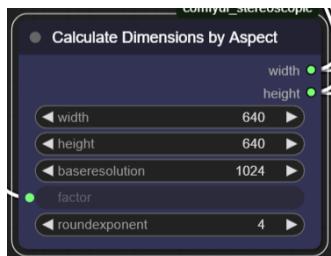


By default it generates a square 1024x1024 image. You may want to reduce the scale factor here:



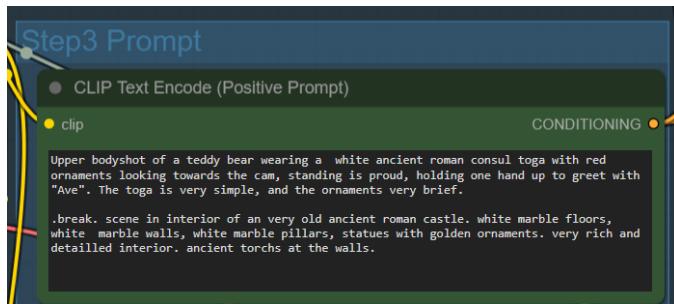
This factor is limited by your system.

To change the aspect, just enter reference width and height. The image dimension is calculated automatic based on the aspect ratio and the scale factor above:

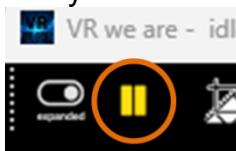


It's just the aspect. The pixel amount is already limited by the factor.

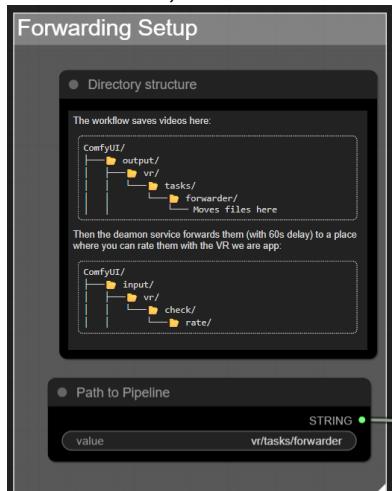
Edit the prompt and “Run” it.



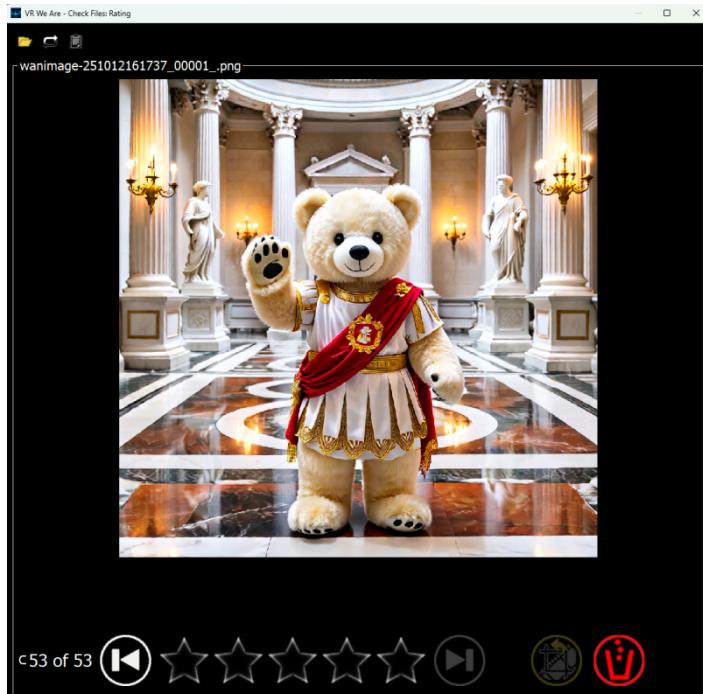
Running the pipeline is guarded by a “Pause Pipeline” custom node. This ensures the workflow is only executed while the pipeline is paused:



When done, the result is written to the forwarder task:



A minute later the image is available in the app. Cropping is no option for this option, all images should be same size. Hence, open the rate dialog  It is designed to select the best images out of a larger batch count; please read chapter 6 for more information. To choose an image directly from the app, use the “copy file path” action  from the toolbar, to copy the file path to the clipboard and use it in the next workflow.



### 3.3.3.2 Generate video chunks

For creating video chunks you can select between two options:



FI2V Wan22 Pipeline-  
Injector



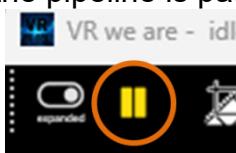
FLI2V Wan22 Pipeline-  
Injector

You can also use your own workflow.

The FI2V Workflow uses only a start reference image, while the FLI2V workflow a start and end reference image.

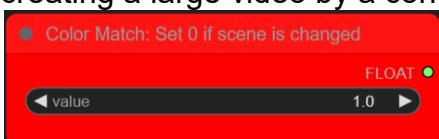
- For the FLI2V you start with using images from the T2V workflow. One for the first frame, one for the last frame. For the follow-up chunk you take the last frame as first frame.
- The FI2V Workflow is different. Either you have a start image, or create one with the T2V workflow. For the follow-up chunks you take the last (reasonable image) of the video as first image.

The setup is similar to T2I. Use some better models than the default. Running the pipeline is guarded by a “Pause Pipeline” custom node. This ensures the workflow is only executed while the pipeline is paused:



Adjust the factor fitting your system and preference. I use 1.0 or 1.1.

The prompt must describe the change. I do not go into details here because this is common stuff described elsewhere better. But if you have scene changes, maybe because of initializing the scene in the first chunk, set color match strength for this chunk to 0.0. Else, the color match strength of 1.0 ensures video frame match color if the reference image, what is important for creating a large video by a concat of chunks without color “jumps”.



Edit the prompt and “Run” it.

### 3.3.3.3 Build chunk and go for the next

After done, with a minute delay, the video is available in the app. open the Crop/Trim Dialog  from the toolbar.

When the video is displayed, you have the option to cut the first frame off. This option need only be choosen for the first chunk.



Next take a look at the video, and check for any scene changes or other stuff to trim away and select the last frame:



Then:

- Take a snapshot of the reference image  for the next run and paste the path from clipboard to your first image node in workflow.
- A modified trim button on the right side has temporarily appeared: , use it set the trim one frame before the reference image.
- Then create a trimmed copy .

Finally:

- Archive  or trash .

### 3.3.3.4 Build video from sequence

After a while you will have a sequence of a couple of trimmed video chunks. They have the same dimensions and fps rate, and by default the alphabetical order of their file names should match the time line. Now move them to ...\\ComfyUI\\input\\vr\\concat and reactivate the pipeline in the app toolbar . In the service daemon log window you will be asked to press enter to continue. Do that, and it will concat the chunks. The result can be picked up in ...\\ComfyUI\\output\\vr\\concat . You can click on the cells in the app to open an explorer window.

### 3.3.3.5 Scene change detection

Scene change detection is automatic, and comes from ffmpeg. There is nothing we can improve. It is started automatic for small videos. For larger videos you must start it manually from the toolbar (if no green bar is below it): 

## 3.3.4 Other models and generators

Of course you can use external generators or any other workflow and model with ComfyUI. In this case you use the app to control the pipeline, edit or rate media.



Source image generated with KlingAI

## 4 Setup of the VR headset

### 4.1 Skybox VR Player

Commercial application. Supports many platforms. We have tested it on Meta Quest 3.

To simply the configuration follow their guide: How to Adjust 2D/3D/VR Video Formats, e.g. by putting all videos and images under a folder with the name "fullsbs".

*Known problems:* To view images, the curved mode must be deactivated (bugged). Projection problems with height adjustment at high zoom. If curved mode is not activated, a small frame is displayed around the image, so for videos it looks better with curved mode.

### 4.2 DEO VR Player

Download for free. Supports many platforms. We have tested it on Meta Quest 3, HTC vive pro 2.

To simply the configuration follow their guide: Naming convention, e.g. by adding "\_SBS\_LR" to the end of the file name.

*Known Problems:* Projection problems with height adjustment at high zoom. No controller mapping for pitch. Subtitle track not selectable.

### **4.3 4XVR Video Player**

Another commercial application.

*Known problems:* Auto-detection of aspect ratio currently does not work (support pending). The player does not support images. Manual setup per file required to 3D FSBS if aspect unusual. No height adjustment. Subtitles not supported.

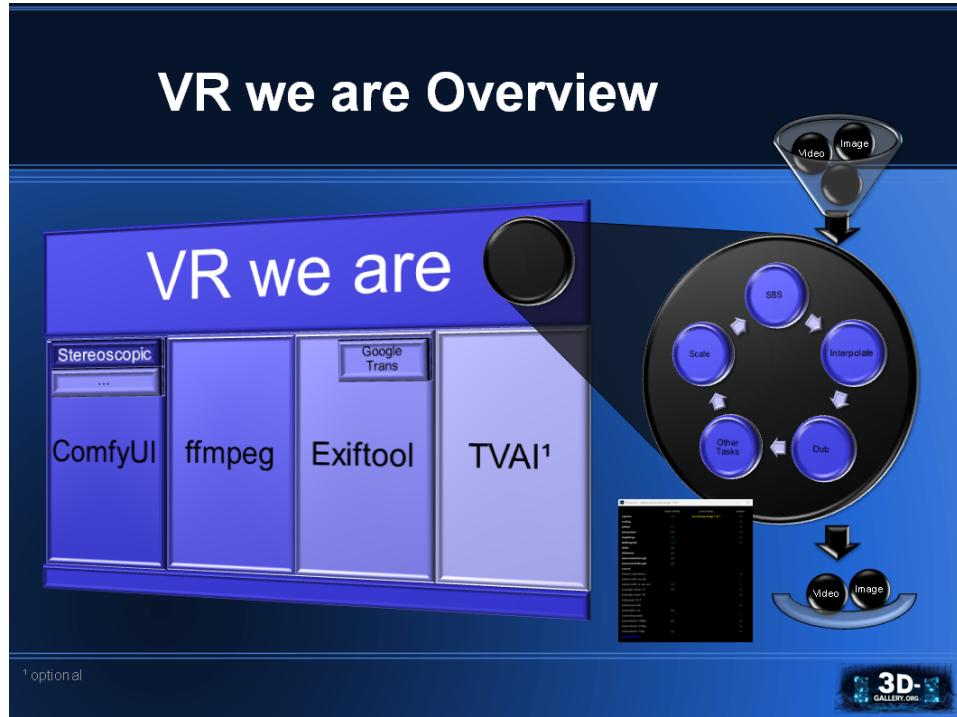
### **4.4 potplayer**

Download for free. Requires an active connection link to the computer. Can be used as video source for other VR media players that run on the Quest 3, such as Bigscreen.

*Known problems:* The player does not support images. Other video aspects than 16:9 are not suitable for SBS display, but you can use the task `pad-16-9-sbs` to counteract this problem.

## 5 System Overview

The following picture illustrates the building blocks of „VR we are“:



„VR we are“ is using other software as foundation:

- ComfyUI is building foundational open source software for the visual AI space.  
„VR we are“ uses it as distribution and execution platform
  - Stereoscopic is a custom node package for ComfyUI containing the „VR we are“ software. For the custom node I got help from iablunoshka, responsible for high performance of the SBS converter. Our first tests had been made with the nodes of SamSeen.
- FFmpeg is a command line tool providing a multimedia framework for video and image manipulation.
- Exiftool is a command line tool for reading and editing multimedia meta information.
  - Google Trans is an optional service, requiring to be online, to translate text (into the own locale).
- Topaz Video AI (TVAI) is an optional professional product, when available used. It offers a massive speed and quality boost on scaling and video interpolation (frame rate increase).
- Git Bash package, an application for Microsoft Windows environments, which provides an emulation layer for a Git command line experience, required to execute „VR we are“.

„VR we are“ waits for multimedia files to be placed in input funnels (file folders) for processing. Per default the files are processed in a non-linear pipeline, landing in output baskets (file folders). Pipelining through the stages can be customized or even completely turned off.

Important: Never place your original files there; only copies!

## 5.1 User Interface

A small graphical user interface is started with the daemon. One duty is to show the working status with a table of the stages and the file count of folders:

VR we are - fullsbs processing video 11 of 37			
	input (done)	processing	output
<i>caption</i>	8 (79)		>
<i>fullsbs</i>	27 (49)	processing video 11 of 37	10 >
<i>interpolate</i>	38		>
<i>singleloop</i>	1 (127)		>
<i>dubbing/sfx</i>			21
<i>slides</i>	(-)		
<i>tasks/convertimage</i>	(-)		

The arrow at output of stages indicates that forward rules exists that may affect the files; when the color turns green, it indicates that forward rules have been applied and the files will stay.

Use this status page to identify the output folders of the final results.

**Important feature:** Handling such many folder is hard. To make it easier for you, just open an explorer window to the folder by clicking on the stage entry for input or output, even if no text is inside. E.g. to open input folder of caption:

VR we are - idle		
	type	input (done)
<i>caption</i>	video;image	
<i>scaling</i>	video;image	
<i>fullsbs</i>	video;image	
<i>interpolate</i>	video	(79)

## 5.2 Display Toggles

 collapsed	Per default stages are collapsed. Only stages with relevant file counts or while active are displayed.
 expanded	In Expanded state all stages are displayed, and supported stage input types are shown: Video, Image or both

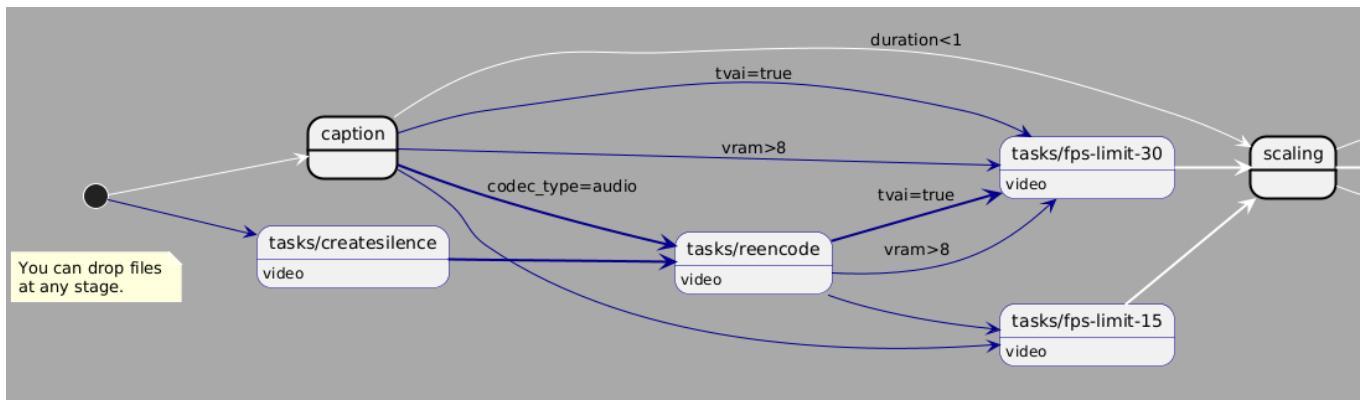
### 5.3 Tool Pipeline

To simplify things, the tool offers a video and image processing pipeline with auto-forward from stage output to another stage input. With release 3.0 it is no longer linear; and it can be customized, but with care.

You just need to place files in input folder of a stage, and pick the results up from the final output folders. No more need for common users to create own CLI command and shell scripts.

	The current pipeline can be viewed in a dialog by hitting this button.
--	------------------------------------------------------------------------

The pipeline is visualized as a UML state diagram:



See appendix for a full display of the default pipeline.

If you get confused by auto-forwarding, just disable the autoforwarding by setting in config.ini  
PIPELINE\_AUTOFORWARD=0

The pipeline status is displayed in the toolbar, and can be changed by clicking on the action.

	The pipeline is active. Click here to pause it. Pausing is necessary if you want to run custom workflows in ComfyUI.
	Pausing of the pipeline has been requested, but the pipeline is still active, and you have to wait until it reached pause state. Click here to cancel the pause request.
	The pipeline is paused. It is safe to run custom workflows in ComfyUI. Click here to resume the pipeline to be active, but ensure the custom workflow queue is empty.
	The pipeline forwarding has been deactivated. Click here to open the configuration to enable it, by setting PIPELINE_AUTOFORWARD=1

	pipeline auto-forwarding can be deactivated using this toggle.
--	----------------------------------------------------------------

### 5.3.1 Customizing Pipeline

You can deactivate the pipeline. Auto-forwarding is by default active, but you can deactivate it by setting PIPELINE\_AUTOFORWARD=0 in  
ComfyUI/user/default/comfyui\_stereoscopic/config.ini

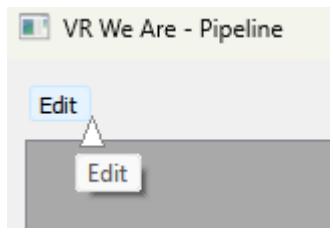
To customize the pipeline, clicking on the Edit button in the pipeline dialog. This will bring up a notepad window with the definition file, syntax see below. Any change to the file will instantly be activated. The diagram is updated (requires internet access). If text in nodes becomes invisible, please reduce the value of UML\_FONTSIZE in the config.ini

When changed while daemon is down, the pipeline is initialized every time the VR-we-are-daemon is started, by reading the configuration defined in a YAML file at  
ComfyUI\user\default\comfyui\_stereoscopic\autoforward.yaml

The file is created with defaults, if it does not exist. It contains the forwarding rules and conditions you can see in the state diagram above.

On daemon start, from this configuration the *forward.txt* files in the output stages are created. Media files that land there will be checked against these rules for transfer from output to input stage. The rules are executed in order and the first rule that matches is executed. Editing the *forward.txt files* is possible, in case you just need a temporary change while the daemon is running.

Every rule may have transition conditions. These are rule details, which must all match for the rule to be executed. Conditions are optional. There is an invisible rule that is always checked: rules are only executed if the media type (video or image) is supported by input and output stage.



You can edit the yaml file to permanently modify the pipeline. Syntax:

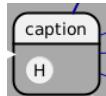
```
pipeline:
  - from: caption
    keepinput: true
    forward:
      - rule:
          to: tasks/reencode
          conditions:
            - codec_type=audio
      - rule:
          to: tasks/fps-limit-30
          conditions:
            - tvail=true
```

**YAML list items start with a dash and can be repeated.**

**Conditions are optional**

Where from and to values must be a valid stage path like scaling, tasks/fps-limit-15 or tasks/\_myjob

The key “keepinput” is used to create a .nocleanup file for the done folder of the “from” stage, so the files are excluded from cleanup.<sup>4</sup> In the pipeline a symbol is shown for media that is held in the done folder after processing, and not purged on next daemon start:



You must delete these files manually.

Rules are processed in order and must name the target “to” stage. Every rule may have conditions. The rule is executed if ALL conditions are met. Every condition has the syntax “KEY” “OPERATOR” “VALUE”, e.g. width < 3841

### 5.3.1.1 Video Stream Keys

They come from Ffmpeg / ffprobe:

bit\_rate, width, height, r\_frame\_rate, duration, nb\_frames

### 5.3.1.2 Image Stream Keys

They come from Ffmpeg / ffprobe:

width, height

### 5.3.1.3 Audio Stream Keys

They come from Ffmpeg / ffprobe. This key is special and required to detect audio stream exists (see default config).

codec\_type

### 5.3.1.4 Other Keys

calculated\_aspect: Calculated as ‘ $1000 \times \text{width} \div \text{height}$ ’ . Values < 1000 when vertical aspect.

pixel: Calculated as ‘ $\text{width} \times \text{height}$ ’

image: true if image, else false

video: true if video, else false

### 5.3.1.5 System Keys

Vram: Size in GB

tvai: true if TVAI exists, else false

sbs: true if SBS done, else false

### 5.3.1.6 Operators

= != > <

### 5.3.1.7 Values

Values must always be integer or text. floats are not supported. Numbers or expressions are trounced.

---

<sup>4</sup> Processed files in input folders are copied into a done subfolder causing a lot of file copies. These folders are cleared when the daemon is restarted. You can create and delete this .nocleanup files as you need. This system is one of the reasons you should never place your original files in input folders. Another reason: the application is renaming files. Try to avoid using filenames with non alpha-numeric characters [A-Za-z0-9], especially if they come from untrustworthy sources. I do some checks but can't guarantee it is bullet proof.

## 6 Review and Release System

The User Interface offers a review system, to integrate the txt2img and img2vid techniques into the pipeline. Images and videos from AI are usually created in batches, and must be evaluated and filtered in the meantime, before continuing in your pipeline.



Afterwards, you can then use the edit functions here to decide on the last frame of the video, trim it and generate a snapshot of the last image to run img2vid again to produce longer cuts for your videos, by concat them (see chapter 7.9).

Or you just use it for the final sentence over your media; what to keep and what to toss away.

The user interface offers a three step process, starting with files supplied in the incoming folder (see chapter 7.10):



substage check/rate:

1. Edit: Create copies of cropped images or videos. Create copies of trimmed videos. Create snapshots from videos. Delete scrap. These step are optional.
2. Rate: Categorize images or videos with a 5-star rating system as used by the Windows OS (includes tagging) and forward to next substage.

substage check/released:

3. Delete or release file sets per rating class
4. Finally forwarding them to output folder (see chapter 7.10). Per default this is capture, but in case of images you may want to redirect it to your img2vid folder.

**Input folders**, where you can also manually can **place files or pick them up**:

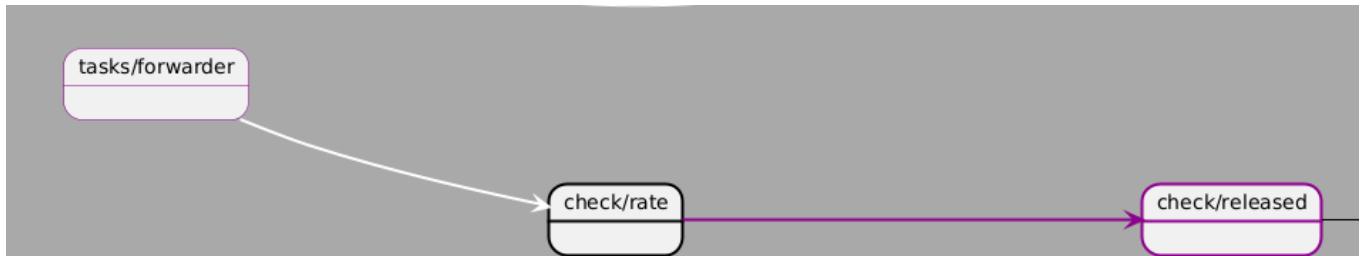
<input type="checkbox"/>	check	<u>Visible in:</u>
<input type="checkbox"/>	rate Files to edit or rate	edit, rate
<input type="checkbox"/>	edit Edited files that can be rated	rate
<input type="checkbox"/>	ready Files reviewed in edit, ready to rate.	rate
<input type="checkbox"/>	done Files archived in edit or rate.	-
<input type="checkbox"/>	released	
<input type="checkbox"/>	done Copy of forwarded files.	-

**Output folders**, where you also manually can **pick files up**:

<input type="checkbox"/>	check	<u>Visible in:</u>
<input type="checkbox"/>	rate	
<input type="checkbox"/>	1 ... 5 Rated files. Ready to judge.	judge
<input type="checkbox"/>	released Ready to forward	pipeline target and released/done

## 6.1 Preparation

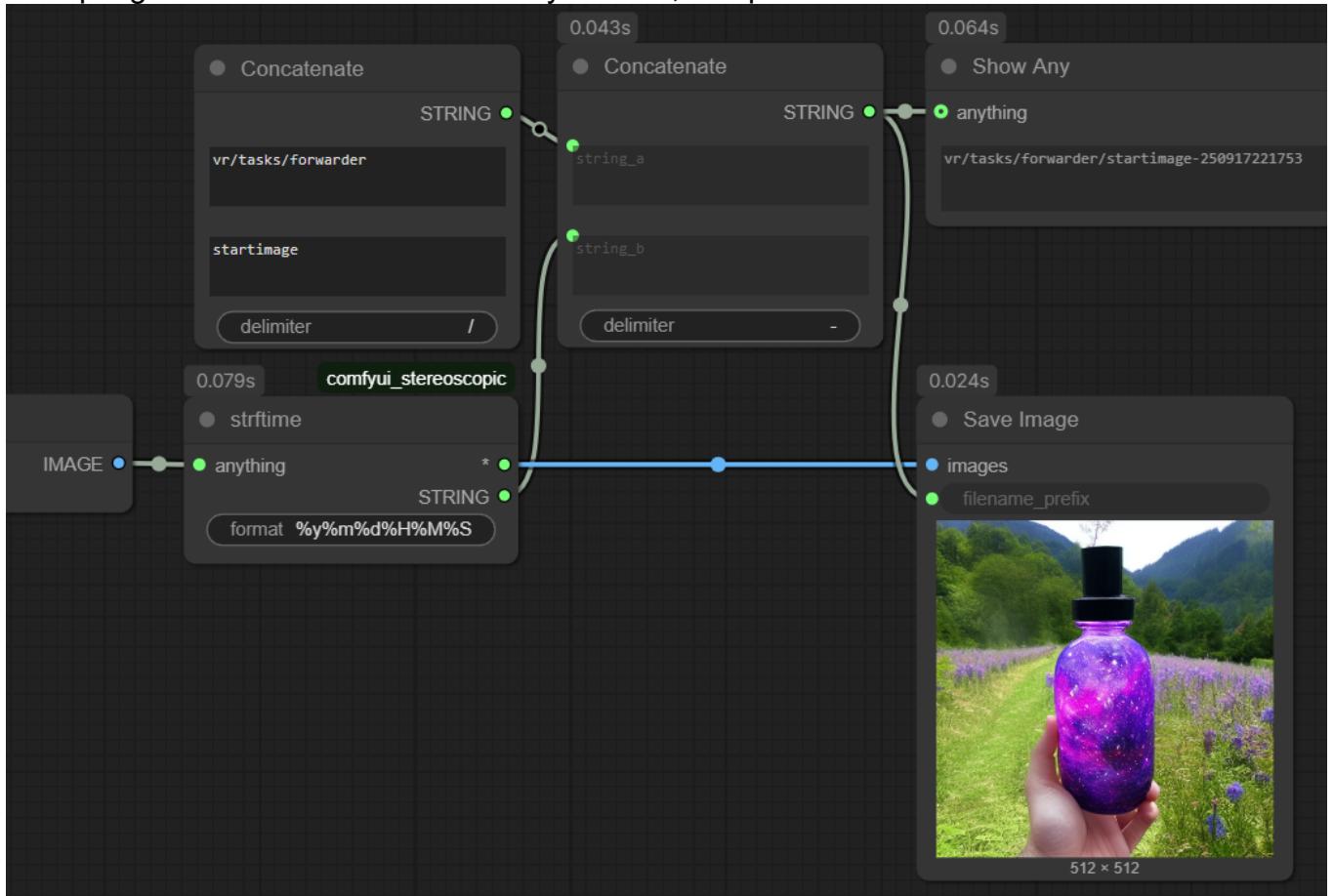
This system is designed to work with your txt2img and img2vid workflows. Save the output of your ComfyUI workflows directly to the *output/vr/tasks/forwarder* folder, if you need no backup. Or if you prefer external tools, copy your media there.



*tasks/forwarder* is a special stage prepared to forward any file to *input/vr/check/rate* that is older than 60 seconds (time of last modification). Of course you can save files directly in workflows to the input folder, but you need to use special nodes that allow this.

Ensure your files have unique names, to avoid overwriting old files and loose them!

Stereoscopic includes a node *strftime*<sup>5</sup> to generating timestamps for filenames on the fly. This example generates different name every second, independent of additional counters:



Same procedure for videos.

<sup>5</sup> <https://strftime.org/>

## 6.2 Edit: Crop, Trim, Snapshots

This is a simplified tool, for course grained cutting your videos and images, with focus on AI generated content and workflow. Use an editor of your choice for anything beyond. Crop & Trim focuses on a quick review of the files, delete crap or create edited copies. It will display images and videos from ComfyUI/input/vr/check/rate folder, but you can override the folder by using the change folder  icon in the upper left corner and select a custom directory. If override is active a green line appears below the icon and the Path of the folder:  C:\Users\User\Videos

You can open the current directory in windows explorer by using the yellow folder icon . Files are sorted by timestamp. If a video has errors (look in console), use tasks/reencode first (see chapter 9.1.5). **You can use drag and drop from local drives to change directory or media; importing media from browser this way is supported as well.** Other options: Toggle playback type by . Copy file path to clipboard with , or use drag and drop from file title. Or paste an image with  from clipboard to the rating folder.

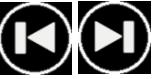
To crop use sliders



, or mouse:

a single click will erase the crop area.

The following controls allow to navigate through the files:

	Go to previous or next file in input/vr/check/rate To navigate through a lot of files use slider in lower left corner.
	Go to first or last frame of current trim. Pause video playback. This is handy to make snapshots of first or last frame, if you need reference images for workflows.

Some actions can be done with each file

	Creates a copy of the video with the selected crop and trim. <sup>6</sup> The result goes into input/vr/check/rate and is hidden in the edit view; it will be visible in the rating view.
 or 	Moves the file away for later rating or cleanup. Action switches after a copy or snapshot of the current media has created, so this raw file is no longer be subject to be rated. Folder to archived files can be opened over toolbar: 
	Deletes the current file. Allows you to get rid of crap quickly. Shows next file. If you made copies before you will see them, now.

<sup>6</sup> Video output height is always (FFmpeg issues with odd numbers).

For videos one more set of controls appear:

	Videos are continuously played in loop. This control allows pausing and continuing playing. Some operations require pausing the video.
	Here you see the video playing. While paused, you can select the frame. If trimmed, the loop is adjusted to the trim. Use PgUp/PgDown to while focused to jump +/- 1s. Use arrows to move by one frame.
	Selecting video trim at the selected frame to left or right. The trim can be applied by the  action and previewed by starting the playback. Pressing again on the same frame clears the trim. At start the icon  with "1" offsets to quickly cut first frame.
	Creates a snapshot <sup>7</sup> from video at the selected frame position. The result goes into input/vr/check/rate and is hidden in the edit view; it will be visible in the rating view. Nice for capture last frame to continue with reference image as first frame of img2vid and use concat later. Additional, the file path is copied to the <u>clipboard</u> and can be used e.g. in the “Load Image + Filename” custom node, by pasting it in the filename field in the “choose file to upload” dialog. Using snapshot of the last frame, allows to create longer videos by concat these sequences (remember to trim off the first frame later, or you’ll have it twice).



To help AI creators with trimming and faster detect unsteady parts in videos or scene intersections, short videos (<20s) are automatically analyzed and as a guess for locations of possible intersections, indicators at the estimated positions are displayed, below the video playback slider. The threshold detects only significant enough changes and position is +/- one frame.

Accelerator Keys:

S	Skip to the next scene intersection and then focus on the slider to position with arrow or page keys.
A	
D	
P	Toogle between  and
PageUp, PageDown	If Slider has no focus, it gets it and video is paused
1	

In the toolbar some filters can be set to blend out either images or videos: /   
Another feature in toolbar is sorting the media list by time or alphanumeric: / / /

---

<sup>7</sup> Currently using ffmpeg for this, which is slow and maybe inaccurate regarding the frame index. It may change in the future.

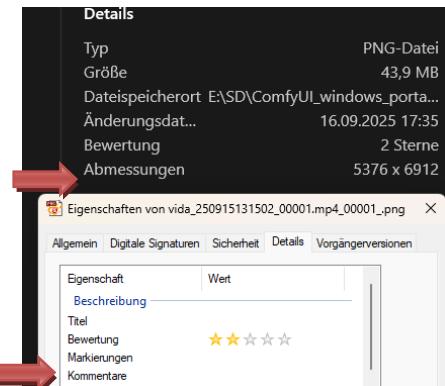
## 6.3 Rating

Rating is separated from Edit. You may want to skip editing for images – then just get here. Here goes your judgement about the quality of media, as preparation for the next stage, where these categories can be forwarded, archived or deleted in a few steps. It does not check for any existing rating.



	Rate the files between 1 and 5. They will be moved to subfolder 1 to 5, below output/vr/check/rate. If Exiftool is present the rating is applied to meta data and visible in OS Windows.
	Instead of rating, you can delete this file, or move it back to edit (if already in subfolders edit or ready).

Rated files get metadata attached if Exiftool is properly configured in config.ini. This has been tested with mp4 and png under Windows 11 and may not work with other formats.



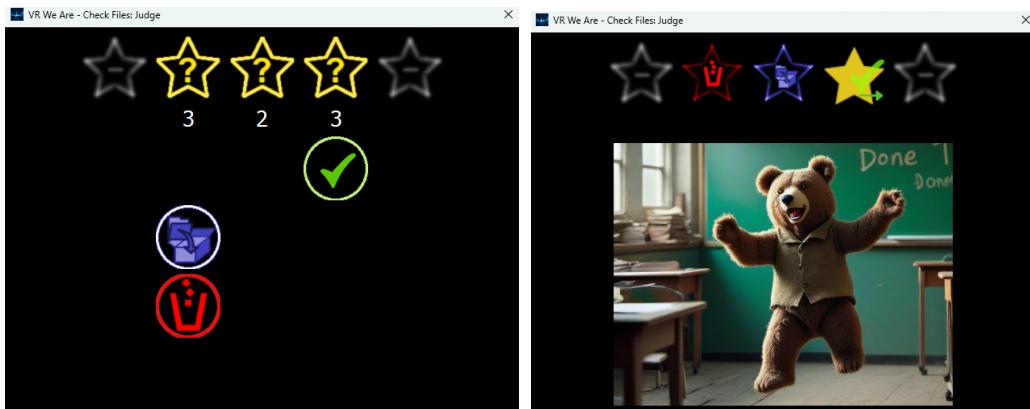
Accelerator Keys:

1 ... 5	
---------	--

Files to be edited are filtered out by default. You can deactivate the filter in the toolbar: Again, some filters can be set to blend out either images or videos: /

## 6.4 Release

Releasing allows to forward, archive or delete a bunch of files by classification category, previously done with rating. It inspects subfolder 1 to 5 at output/vr/check/rate. Click a buttons below a category:



File Categories are displayed by star icons representing on the status of each category:

	No files in this category. Nothing to do.
	Decision required, see actions below.
	Files have been archived. Action completed.
	Files have been deleted. Action completed.
	Files have been forwarded. Action completed.

Possible actions on the lowest and highest outstanding decision to be made:

	Archive all files in this category by moving them to folder <i>input/vr/check/released/done</i> Folder to archived files can be opened over toolbar:
	Deletes all files in this category.
	Forward files in this folder to <i>output/vr/check/released</i> . If there is forward.txt file, it will be further forwarded to the stages of your choice. Hint: Use first a rule with the condition [duration<1] to forward images, then a rules to forward videos to a different stage. See chapter 5.3.1 for information about how to customize the pipeline.

Hint: You may want to forward and continue processing your best files and come back later to this dialog, maybe with more rated files, so keeping the files here like a backlog for future batches. Here again, filters can be set to blend out either images or videos: /

## 7 Stages

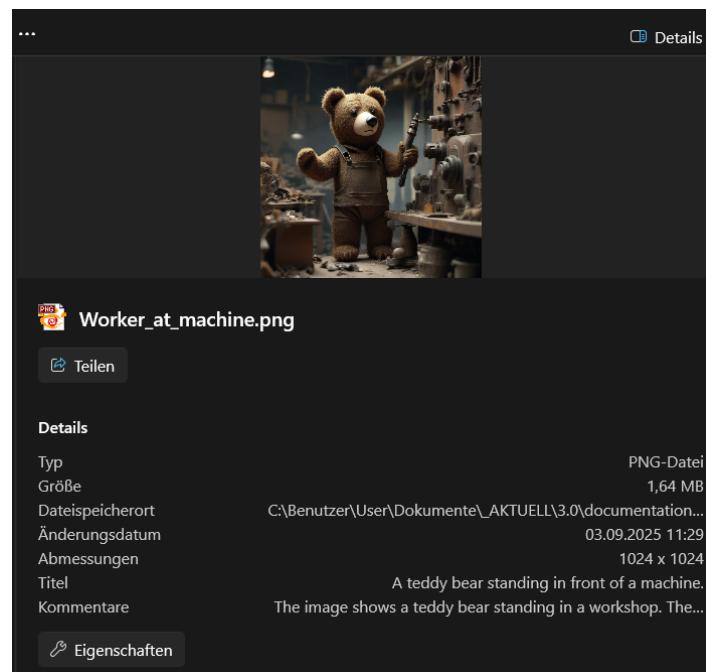
The following stages are the basic operation „VR we are“ can perform:

### 7.1 caption

Adding title, description to videos and images using analysis of first frame.<sup>8</sup> Embedded workflow and prompt are removed.

Translates descriptions if online. They are displayed and editable in the Windows Explorer in Windows 11 (only displayed in Windows 10). Forwards meta data of videos and images through stages, strips off prompts and embedded workflows. To enable, change language or turn off translation please edit the config.ini parameters EXIFTOOLBINARY, DESCRIPTION\_LOCALE, ISO\_639\_2\_CODE.

Under Windows 11, title and comments can be seen in the preview ...



... and in the property details:



### 7.2 scaling

Upscales the video and image with an AI-Model by a factor of 4 or 2 depending on the current size, some rules and the configuration, using a combination of upscale model and mixing with

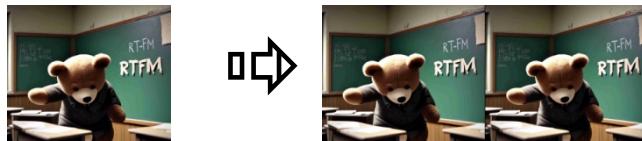
<sup>8</sup> The Windows OS requires to remove some Keys and their Metadata.

the original input. High resolution input is required by the SBS converter to produce better results at the “edges” of the depth images. Restricted use on old hardware by pipeline conditions. For low quality input and long videos I recommend using tools first. For videos it operates on frames, why fps-rate is limited in the pipeline first.

Important config.ini parameters: UPSCALEMODELx4, UPSCALEMODELx2.  
TVAI requires model configuration.

### 7.3 *fullsbs*

Convert video and image to side-by-side left right, full width to be used in a viewer on your VR display. Stereoscopic is a main feature of VR we are; please read more about it in chapter 11.



For best results input should have resolution of at least 1MP or higher. For videos it operates on frames, why fps-rate is limited in the pipeline first.

Important config.ini parameters: SBS\_DEPTH\_SCALE, SBS\_DEPTH\_OFFSET, SBS\_DEPTH\_BLUR\_RADIUS\_VIDEO, SBS\_DEPTH\_BLUR\_RADIUS\_IMAGE, DEPTH\_MODEL\_CKPT

### 7.4 *interpolate*

Doubles the video frame rate using AI model. It is a very time-consuming operation. Pipe-line default is self-looping up to 60fps.

config.ini parameters: TVAI requires model configuration.

### 7.5 *singleloop*

Appends a reverse version to a video, so it can be looped. Useful for short videos that have no forward direction in time. I recommend using this before dubbing videos.

### 7.6 *dubbing/sfx*

Designed to add sound (effects) to videos using AI model. Additional prompts can be added. Be aware, that this might be out-of-sync and contain strange parts. It is a tool for you to play with and quickly get some sound. Prompts are injected from

ComfyUI\user\default\comfyui\_stereoscopic\dubbing\_sfx\_positive.txt and  
dubbing\_sfx\_negative.txt.

Seeds are random. Repetitive calls (if auto-forward is deactivated) may produce new sound results, with ascending numbers, for the user to choose.

Long videos >20 seconds, especially slideshows, generate sound changes every 5 seconds, so every slide has its own sound.

### 7.7 *watermark/encrypt*

encode forensic watermark on images, and keep the original image in store under ComfyUI\user\default\comfyui\_stereoscopic\watermark.

Important config.ini parameters:

WATERMARK\_LABEL (maximum 17 characters) containing your name

WATERMARK\_SECRETKEY: Your personal integer used to encrypt the image. Without this key nobody can decrypt the watermark. The key is stored as folder name in the store, so it can't get lost and the key can be changed on demand, e.g. after some period of time.

File watermark\_background.png next to config, contains the watermark image where your name is printed at top:



You replace the image, but the filename is fixed.

### 7.8 *watermark/decrypt*

decode forensic watermarking from image against stored version and return the watermark image.

Returns a ghost image of the watermark that had been invisibly encoded in your images before:



### 7.9 *concat*

Can be used in combinations with tasks/split-1m or singleloop to put parts or loops, which share the same first (and last) frame, to a complete or longer video. For loops I recommend using this before dubbing.

### 7.10 *check*

The review operation is backed up by special stages under the stage name *check*, which can be integrated into the forward pipeline. Input is at *check/rate* and output at *check/released*.

## 8 Slideshow stages

Slideshow stages are outside of the normal pipeline. Turn auto-forwarding off by temporarily setting PIPELINE\_AUTOFORWARD to 0 in config.ini.

Use stages in the following order:

- scaling (optional)
- slides
- slideshow
- fullsbs, from here you may start auto-forwarding again
- interpolate (optional)
- dubbing/sfx (optional)

Finally set PIPELINE\_AUTOFORWARD to 1.

### 8.1 *slides*

Want to see your images, maybe from vacation, in VR, but are annoyed about the way your VR viewer is displaying them? Use this to create 4K slide images (scale + padding) as preparation to make a video. Take the images from the p. Then move all slides to the slideshow.

### 8.2 *slideshow*

Put slides from slides stage into a slide show, 5s each image including an 1s SBS-compatible transition.

## 9 Tasks

Tasks are standardized stages based on a blueprint. The user can duplicate and customize them. He can also replace them by modifying the pipeline.

### 9.1 FFmpeg-based

#### 9.1.1 convertimage, convertvideo

Convert any format ffmpeg can handle to mp4 or png.

#### 9.1.2 fpslimit-30, fpslimit-15

Ensures the frame rate of a video is limited for performance reasons, per default used before scaling.

#### 9.1.3 vlimit-2160p, vlimit-1080p, hlimit-2160

Ensures the height or width of a media is limited for performance reasons, per default used before sbs-converter.

#### 9.1.4 limit-1920, limit-1600

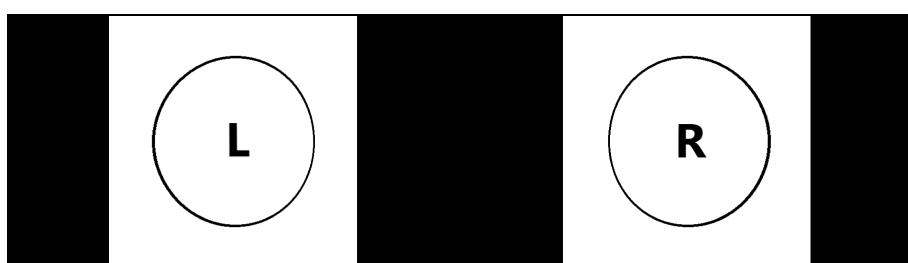
Ensures the width and height of a media is limited for performance reasons, per default used before interpolation.

#### 9.1.5 reencode

Ensures the media is well defined for processing.

#### 9.1.6 pad-16-9, pad-16-9-sbs

Pads the width of the media with black color so it gets aspect 16:9. For SBS aspect is up to 4K per side. Result:



This can be handy to display it with viewers that have problems to handle full SBS display of other aspects.

#### 9.1.7 credit-vr-we-are, credit-no-sbs

Adds credits to a video. The credit video must be loopable. This can be used a template to replace it with your own signature.



#### 9.1.8 desaturate, incsaturation

Manipulating saturation.



### 9.1.9 **split-1m**

Use this to split long videos into segments of about 1 minute length as first step. After jobs for all segments are done, use the concat stage to merge them together.

### 9.1.10 **createsilence, stripaudio**

Adds silent audio to video or strips audio off from video.

### 9.1.11 **swapsides**

Swaps left right in SBS video or image. Useful for people who want to create content for cross-eyed stereo.



## 9.2 ComfyUI-workflow-based

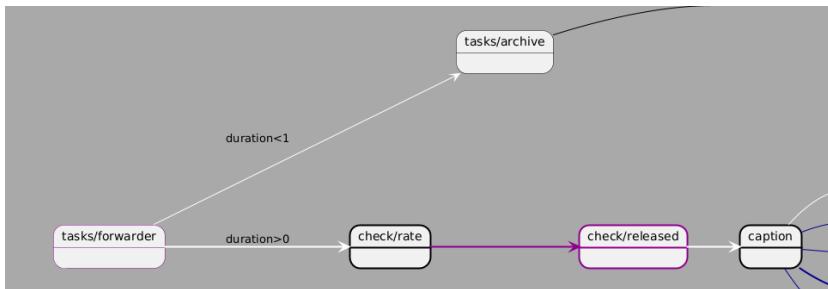
### 9.2.1 archive, trashbin

Does nothing. Prepared sinks for stuff already done, that need no further processing, ready to archive or delete later. E.g. to move reference images (first/last) from your custom ComfyUI workflows there, save them to forwarder and add the following rule to the task:

```
- rule:  
  to: tasks/archive  
  conditions:  
    - image=true
```

For videos set condition to `video=true`

Sample pipeline to later rate generated videos from your custom txt2vid workflow, while saving the first image as well at a different place, not to be rated now:



## 9.3 Extra Content

The following content require additional downloads to work.

### 9.3.1 Workflow stage: recolor

Coloring an grayscale image using AI.



Model and Custom Nodes nodes not included:

- Nodes: ComfyUI ArtVenture (#180) 1.1.1
- Model: Juggernaut-XL, update comfyui\_stereoscopic\config\tasks\workflow\api\I2I\_BP\_SXDL\_Recolor.json

Low resolution images (below ~256px) produce errors, use the scaling stage first.

## 10 Task Customization

Users can create their own tasks. For now, the blueprint ffmpeg-v2v can be used. You need to have sufficient knowledge of FFmpeg to do.

Just copy a template from ComfyUI\custom\_nodes\comfyui\_stereoscopic\config\tasks to ComfyUI\user\default\comfyui\_stereoscopic\tasks

E.g. hlimit-4k.json and give it an appropriate name.

Example: We want to scale videos down by factor 2. We name it halfsize.json.

To produce what we want we change filter in the json file, so the file looks like that:

```
{  
    "version": "1",  
    "blueprint": "ffmpeg-v2v",  
    "options": "-filter:v scale=iw/2:-2",  
    "input": "video",  
    "output": "video",  
}
```

Now restart the daemon and a new stage appears. User stages have an underscore appended to the task name, so the input folder is ComfyUI\input\vr\tasks\\_halfsize

It will also appear in the status view of the user interface:



It is not integrated in pipeline, unless you do so. Just drop files in the input folder and see how they are processed.

Additional keys in the json file:

forward\_delay      delay in seconds after the modification of the output file to be eligible to be forwarded.

extension      To enforce converting to new output format (suffix, e.g. ".mp4" or ".png")

# 11 Stereoscopic - A Full Stereo Side By Side Converter

## 11.1 Background<sup>9</sup>

Stereoscopy, also called stereoscopics or stereo imaging, is a technique for creating or enhancing the illusion of depth in an image by means of stereopsis for binocular vision. Stereoscopy creates the impression of three-dimensional depth from a pair of two-dimensional images. The left image is presented to the left eye and the right image is presented to the right eye. When viewed, the human brain perceives the images as a single 3D view, giving the viewer the perception of 3D depth. However, the 3D effect lacks proper focal depth, which gives rise to the vergence-accommodation conflict.

Human vision, including the perception of depth, is a complex process, which only begins with the acquisition of visual information taken in through the eyes; much processing ensues within the brain, as it strives to make sense of the raw information. One of the functions that occur within the brain as it interprets what the eyes see is assessing the relative distances of objects from the viewer, and the depth dimension of those objects.

Although the term "3D" is ubiquitously used, the presentation of dual 2D images is distinctly different from displaying an image in three full dimensions. The most notable difference is that, in the case of "3D" displays, the observer's head and eye movement do not change the information received about the 3-dimensional objects being viewed. Holographic displays and volumetric display do not have this limitation. Just as it is not possible to recreate a full 3-dimensional sound field with just two stereophonic speakers, it is an overstatement to call dual 2D images "3D". The accurate term "stereoscopic" is more cumbersome than the common misnomer "3D", which has been entrenched by many decades of unquestioned misuse.

Although most stereoscopic displays do not qualify as real 3D display, all real 3D displays are also stereoscopic displays because they meet the lower criteria also.

The arrangement of the images created in with the nodes here, is left/right (side by side – short: SBS) with full width-resolution.

---

<sup>9</sup> Extract from Wikipedia

## 11.2 Depth Map

Depth information is taken from an AI model, generating depth map for a 2D image:



The quality of the depth map is important for correct depth feeling and improves with model quality. However, large model require significant longer time to process and in dynamic videos it is of less importance. Since videos need to be processed frame by frame it is usually sufficient and more efficient to use a small depth map generation model.

We apply some blurring to the depth map image to reduce artifacts. Can be changed by keys SBS\_DEPTH\_BLUR\_RADIUS\_VIDEO and SBS\_DEPTH\_BLUR\_RADIUS\_IMAGE in config.ini.

Depth model comes over ComfyUI custom nodes from comfyui\_controlnet\_aux. The model can be configured in config.ini by the key DEPTH\_MODEL\_CKPT. It defaults to depthAnything\_v2\_vitb.pth

The processing time of this model is about the same as the SBS converting algorithm. The giant variant must be installed manually.

### 11.3 SBS Conversion Algorithm

Left and right side of the image are generating by shifting pixels of the original image to the left or right, depending their value in the depth map. This tricks the eyes to let them show you the 3D effect. However, you can of course not see around objects, so at value “jumps” in the depth map, it can generate irritations to the eye. To overcome this situation, the depth “effect” scale should not be too big, it should be divided on both images equally and the resolution should be high.

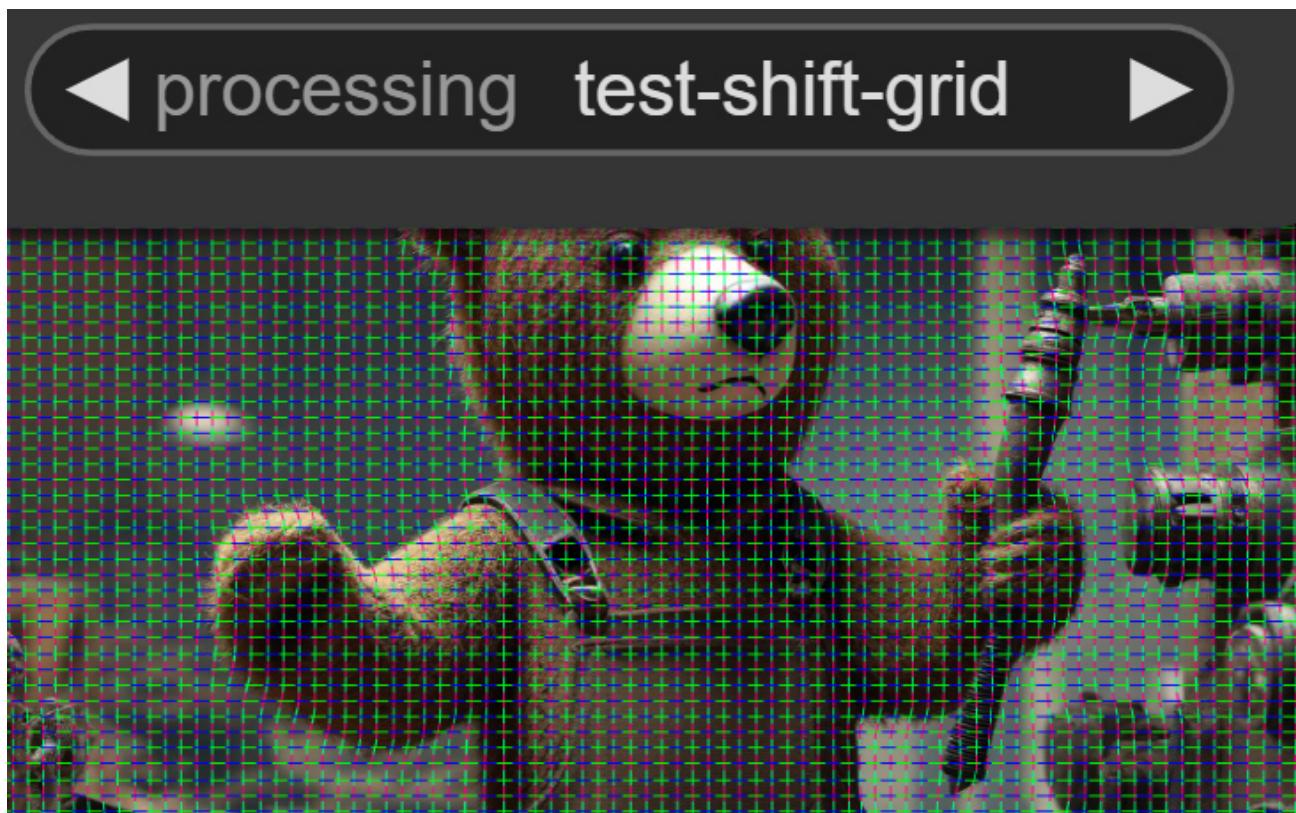
The following illustrations are made with an analysis workflow you can find under  
custom\_nodes/comfyui\_stereoscopic/examples/workflows/analysis/I2I\_SBS\_Depth\_Analysis.js  
on

So first the shift information is calculated, using the parameter `depth_scale` and `depth_offset`:



In config.ini they can be changed by the keys `SBS_DEPTH_SCALE` (default 1.0) and `SBS_DEPTH_OFFSET` (default 0.0). These values are normalized, half of the shift is to the back, the other half to the front and depth scale is balanced out to some natural feeling value. Higher `depth_scale` makes the depth effect stronger, and higher `depth_offset` brings the image more to the front. Symetric is set to true for quality in my workflow; `switch_sides` can't be adjusted though configuration, but for cross-eye fans there is a task called `swapsides` (see chapter 9.1.11).

Next the shift offsets are applied to image. You can see with another processing mode how the shift is applied to a grid laid over the image:



Finally we black-out some pixels at left and right side of the image to hide pixel artifacts generated due to lack of information in this area. To keep this area relative low, the resolution should be as high as possible.

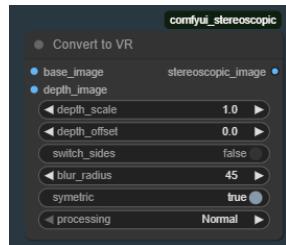
The final result is a side-by-side image:



## 12 ComfyUI Custom Node Library

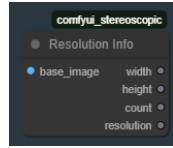
In `comfyui_stereoscopic` several custom nodes are defined.

### 12.1 Convert to VR (*ImageVRConverter*)



Explained in chapter 11.3.

### 12.2 Resolution Info



Helper Nodes to get basic information from an image batch.

### 12.3 Encrypt Watermark



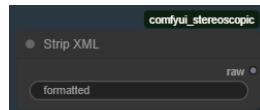
Explained in chapter 7.7.

### 12.4 Decrypt Watermark



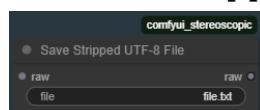
Explained in chapter 7.8.

### 12.5 Strip XML



Internally used node to strip XML from string (Florence 2 problem).

### 12.6 Save Stripped UTF-8 File



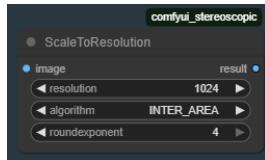
Internally used node to save raw text data.

### 12.7 Scale by Factor



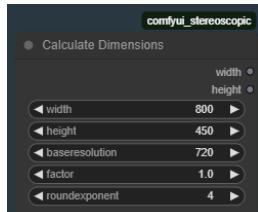
Scale up or down by factor and algorithm. Do nothing at factor 1.0.

## 12.8 ScaleToResolution



Internally used node for creating depth maps, scales to specific resolution.

## 12.9 Calculate Dimensions

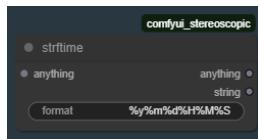


Node that helps to calculate dimensions of an image, keeping aspect and pixel amount (square of product of baseresolution and factor).

## 12.10 RegexSubstitute

Deprecated. Please use ComfyUI core node Regex Replace.

## 12.11 Strftime



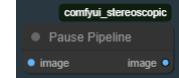
creates a formatted timestamp, guarded by the moment “anything” is done.

## 12.12 Save Audio (Simple)



Saves audio to a flac file.

## 12.13 VRwearePause



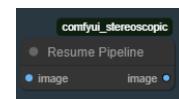
Blocks the workflow progress for image until pipeline is paused.

## 12.14 VRwearePauseLatent



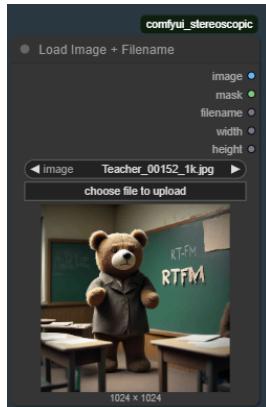
Blocks the workflow progress for latent image until pipeline is paused.

## 12.15 VRweareResume



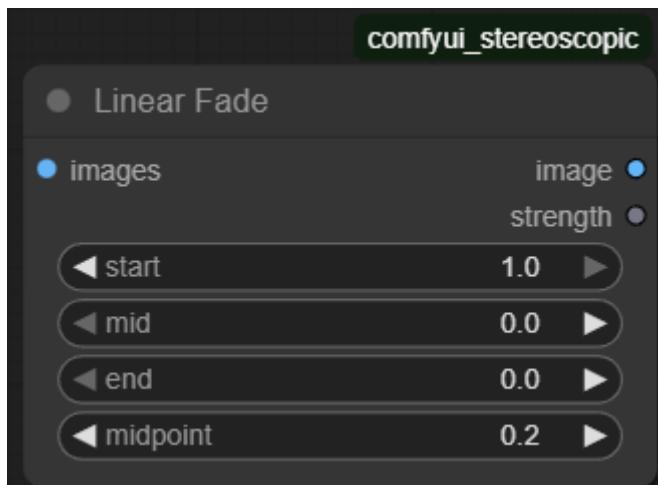
Resumes the pipeline. (Not used yet)

## 12.16 Load Image Advanced

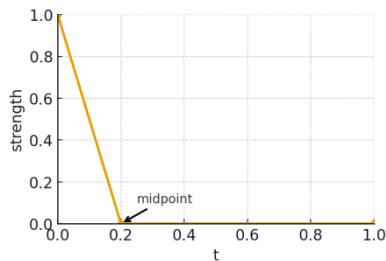


Loads an image from absolute filepath or selection of input image, showing preview, and give information about basename of image file, width and height.

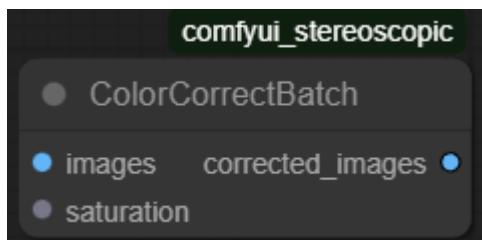
## 12.17 Linear Fade



Control color correction strength of video over time, to avoid hard color cuts when extending a video sequence by a new chunk, usually appearing in First Image To Video Workflows. E.g. to be used after VAE Decode and before kjnodels Color Match hm-mvgd-hm, or Color Correct Batch. If you have scene changes you might want to set start to 0.0. To enforce keeping Colors set start and end to 1.0.



## 12.18 Color Correct Batch



Color correct does image manipulation over time (for video). Currently only saturation is supported. It is intended to be used after Linear Fade.

# 13 Random Prompts and LoRa Training

This is experimental work in progress. Overview of scope:

First step is to create variants of images for training.



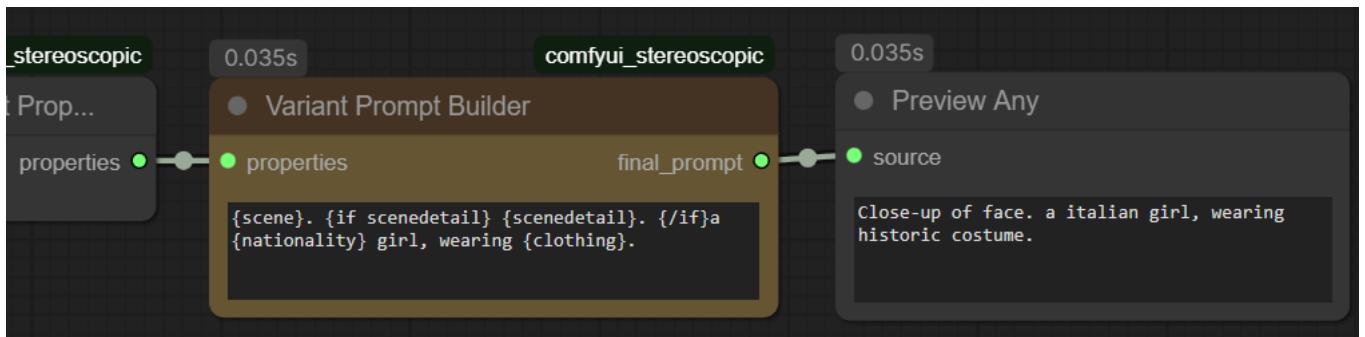
For I2V LoRAs, next step is to create training videos from synthetic images. Then apply the Stabilize-Color task to the videos before using them as test data.

## 13.1 Custom Nodes

These custom nodes can be used directly by a simple install of VR we are in ComfyUI.

### 13.1.1 Variant Prompt Builder

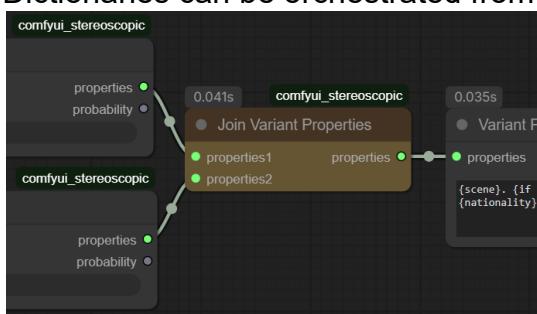
The Variant Prompt Builder node allows to create a prompt from a pattern, using the dictionary provided, to fill the slots defined by {key} in the pattern:



Optional parts (non-cascading) can be defined using a {if key} .... {/if} pattern.

### 13.1.2 Join Variant Prompt Builder

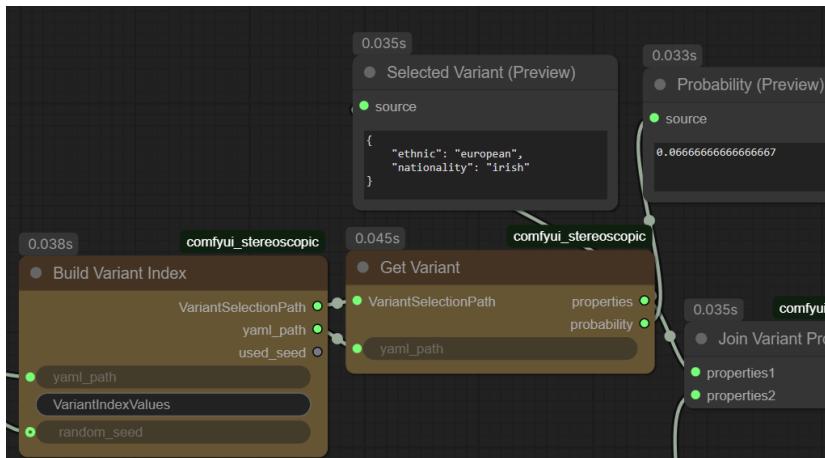
Dictionaries can be orchestrated from multiple sources into a single dictionary.



The keys (scope) must be different.

### 13.1.3 Build Variant Index, Get Variant

A random variant from a specification file is selected by using two nodes.



The path to the YAML file, containing the specification, should be absolute.

## 13.2 Variant Specification

The variant specification is defining a tree graph of variants. The file has the format shown below. Under every variant of the variations a new variations level can be defined, building a multiple-level hierarchy. Weights are normalized per variants (on same level). Props are listing key/value pairs to be chosen for the dictionary in the ComfyUI workflow.

```
1 profile:
2   name: human_scene
3   variations:
4     - variant:
5       weight: 2
6       props:
7         scene: Upper body shot
8         variations:
9           - variant:
10             weight: 3
11             props:
12               clothing: historic costume
13             - variant:
14               weight: 1
15               props:
16                 clothing: casual clothing
17             - variant:
18               weight: 1
19               props:
20                 scene: full body shot
21                 variations:
22                   - variant:
23                     weight: 3
24                     props:
25                       clothing: historic costume
26                   - variant:
27                     weight: 1
28                     props:
29                       clothing: casual clothing
```

## 13.3 Workflow blueprints

Workflow blueprints can be used directly, but to use their API exports for the script, it requires a full install of VR we are. Blueprint workflows can be dropped to ComfyUI from the folder `cli\lora\files`. The workflow format is with sub graph, available with the **newest ComfyUI** versions.

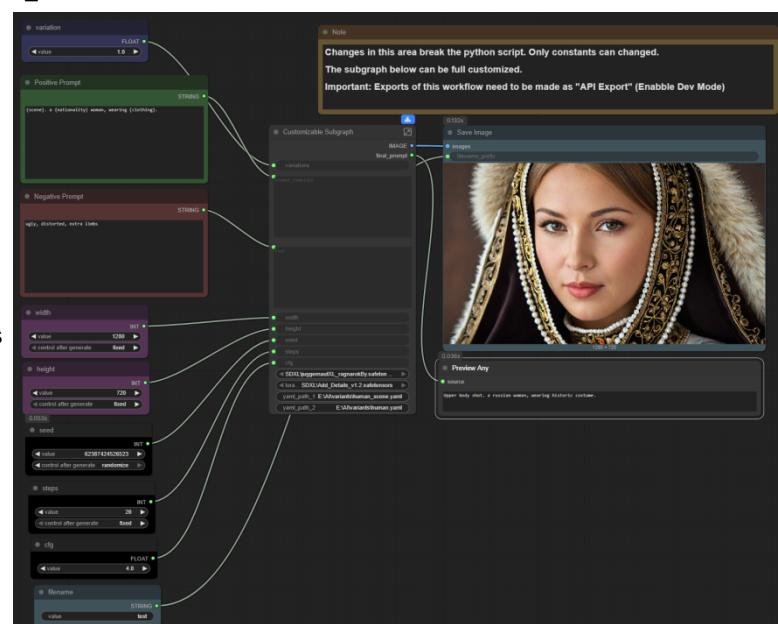
Workflow blueprints on the top level outer graph contain nodes that are required. If planned to export as API workflow for the script, deleting nodes or their links on this top level would break the script.

### 13.3.1 Create\_Synthetic\_Testdata\_Variant\_BLUEPRINT

This is a `txt2img` workflow. Inputs:

Variant	Float to control variation over a series images, e.g. frames of a video.
Positive Prompt	Prompt with variant pattern
Negative Prompt	
Width	
Height	
Seed	The seed. It will be constant for frames of the same video (using script)
Steps	Steps for sampler
Cfg	Cfg for sampler
Filename	Filename prefix of image

The subgraph is customizable. This sample includes the variant handling nodes explained before.



## 13.4 Scripts

The scripts require a full install of VR we are.

### 13.4.1 Create Synthetic Dataset

To create videos to train an img2vid LoRa, under `cli\lora` a bash shell script is provided, that will print a sample of the commands to be executed. You can modify and execute it, e.g.:

```
export COMFYUIHOST=127.0.0.1
export COMFYUIPORT=8188
"/e/SD/vrware/ComfyUI_windows_portable/python_embeded/python" "/e/AI/my synthetic testdata variant.py" --workflow
"/e/SD/vrware/ComfyUI_windows_portable/ComfyUI/user/default/workflows/Create Synthetic Testdata SD1.5 API.json" --tests 1 --prompt_text
"(scene). a (nationality) woman, wearing (clothing)." --target_length 17 --iterations 4 --startvalue 0.0 --endvalue 1.0 --output_dir
"/e/SD/vrware/ComfyUI_windows_portable/ComfyUI/output/vr/tasks/forwarder"
```

In this case we are using a sample workflow, based on the blueprint explained before.

To print a help page, just execute the python script with just `--help` to print this output (a little bit briefer):

```
usage: create_synthetic_dataset.py [-help] [--workflow WORKFLOW] [--tests TESTS] [--prompt_text PROMPT_TEXT]
                                   [--negative_prompt_text NEGATIVE_PROMPT_TEXT] [--iterations ITERATIONS] [--target_length TARGET_LENGTH]
                                   [--startvalue STARTVALUE] [--endvalue ENDVALUE] [--steps STEPS] [--cfg CFG] [--width WIDTH]
                                   [--height HEIGHT] [--fps FPS] [--codec CODEC] [--output_dir OUTPUT_DIR]

Create synthetic dataset via per-frame queueing to ComfyUI.

options:
  --help            show this help message and exit
  --workflow WORKFLOW      Path to ComfyUI workflow JSON, required.
  --tests TESTS          Number of test videos to create, required.
  --prompt_text PROMPT_TEXT      Prompt text (can be multiline), required.
  --negative_prompt_text NEGATIVE_PROMPT_TEXT      Negative prompt (optional, multiline), default exists
  --iterations ITERATIONS      Number of generated frames per test (real iterations), default: 4
  --target_length TARGET_LENGTH      Final length of video in frames (>= iterations), default 17
  --startvalue STARTVALUE      Start value for variation, default: 0.0
  --endvalue ENDVALUE      End value for variation, default: 1.0. Will be applied to the last target_length-iterations+1 frames.
  --steps STEPS          Sampler steps injected, default: 20
  --cfg CFG              CFG scale injected, default: 4.0
  --width WIDTH          Render width (int), default: 720
  --height HEIGHT         Render height (int), default: 1280
  --fps FPS              FPS for the output video, default: 16.0
  --codec CODEC          ffmpeg codec, default: libx264
  --output_dir OUTPUT_DIR      Directory to store videos (frames are temporary), default: ./output/vr/tasks/forwarder
```

Example for Style LoRa settings: `--target_length 17 --iterations 4`

Example for Motion LoRa settings: `--target_length 17 --iterations 17`

Example for porting your LoRa to another model archetype: `--startvalue 0.0 --endvalue 1.0`

## 13.5 Tasks

The tasks require a full install of VR we are.

### 13.5.1 Stabilize-Color

This will apply the color match of kjnodels, to do a color transfer from the first frame of a video to the other frames of the video to avoid unwanted side effects during model training. Use this after creating synthetic test data on each video before adding the videos to the LoRa training.

## 14 Optional Tools

### 14.1 Topaz Video AI

Using this software for upscaling and interpolation instead of using ComfyUI models and workflows archives much better results; it is at least 10 times faster and you can, in the application itself, perfectly adjust parameters in a preview mode, that may help to improve settings in the „VR we are“ workflow.

Currently tested with v6.0.4, models: prob-4, iris-2, chf-3

Topaz Video AI requires a valid login, that need to be renewed online after some time. The login procedure is not covered by this VR we are. Future versions of Topaz Video AI will not offer CLI support; Hence, new customers need to ensure they can still get a version of Topaz Video AI that supports it.

The product warranty is just one year, and has to be repaid. In the US, there is no legally mandated minimum duration for software updates as part of the warranty, as the country does not have a statutory warranty system like Germany or the EU.

Known Bugs:

- Sometimes the ffmpeg task runs infinite. You can detect this when time stamp has no progress. In this case, kill the process with the TaskManger.

```
frame= 31 fps=0.0 q=37.0 size= 1kib time=00:00:00.64 bitrate= 14.7kbit/s speed=0.000739x
```

- Sometimes the ffmpeg task pops up with a GPU error. In this case, try to execute the step again, by moving the files from the `error` subfolder back into the input folder of the stage that failed.

Topaz Video AI Configuration options can be displayed by calling:

- ...Topaz Video AI\ffmpeg.exe -hide\_banner -h filter=tvai\_up
- ...Topaz Video AI\ffmpeg.exe -hide\_banner -h filter=tvai\_fi

There is a regular token check and automatic login procedure, to prevent files get watermark.





## 15 Troubleshooting

Problems or questions? Contact the Discord Support! Activation Link:  
<https://discord.gg/ZegT6Cc8FG>

- If an error occurs, the daemon.bat window may close or the error not invisible. Please start daemon.sh in a bash shell instead, to see the error. For just the user interface the script is gui/restart.sh. If there are errors you can report them on discord.

Some tips:

- For long videos: Use task *split-1m* first, at the end use stage *concat*.
- Some tasks may fail randomly. Try again.
- Some media raise errors and are not displayed. You need to reencode them.

# 16 Manual Install (with your own ComfyUI)

Manual install is no longer necessary and not recommended. Dependency management is a complex task. Please use the provided install script that is provided. The following steps are just hints for experts.

## 16.1 ComfyUI

### 16.1.1 Choose Installation Option

#### 16.1.1.1 Option 1: Windows Portable Package

Download a 7z file from release 0.3.58 to 0.3.59 on GitHub:

<https://github.com/comfyanonymous/ComfyUI>

It includes Python. Execute run\_cpu.bat or the nvidia ones if you have that GPU.

#### 16.1.1.2 Option 2: ComfyUI Desktop (not regularly tested! Limited support.)

You need to install Python 3.12+ and ComfyUI Desktop:

<https://www.python.org/downloads/>

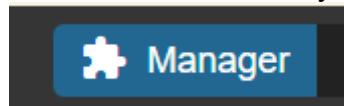
<https://www.comfy.org/>

### 16.1.2 Install Prerequisites on ComfyUI

#### 16.1.2.1 Install ComfyUI Custom Nodes

In ComfyUI, if not already there, install the Manager 3.35 from <https://github.com/ComfyOrg/ComfyUI-Manager>. (Download "Code" ZIP) and move folder into ComfyUI\_windows\_portable\ComfyUI\custom\_nodes.

Then restart the ComfyUI server and you will see this icon:



Now wait for the Manager to update the ComfyRegistry (watch logs), so the custom node database is up to date.

Now, you can use the ComfyUI Manager:

Custom Nodes Manager

To install other custom node packs. It will download most of them from GitHub<sup>10</sup>.

Please pick the right version number, at least minimum.

- ComfyUI-Custom-Scripts 1.2.5
- comfy-mtb dev/0.6.0
- ComfyUI-Crystools 1.27.3
- ComfyUI-Florence2 1.0.6
- ComfyUI-VideoHelperSuite 1.7.4
- ComfyUI-Frame-Interpolation 1.0.7

---

<sup>10</sup> Remark: In automatic install, custom nodes are all redistributed in unmodified form through a fork, since the original repository do not offer releases, or no release for the required snapshot, which may cause stability issues.

- comfyui-depthanythingv2 1.0.1<sup>11</sup>
- ComfyUI-MMAudio. This is not registered at ComfyUI and you have to install it manually:

Download from <https://github.com/kijai/ComfyUI-MMAudio>, and install to ComfyUI/custom\_nodes folder, next execute in the bash shell:

```
# Locate and switch to installation folder:
cd ComfyUI_windows_portable
# Execute install script
./python_embeded/python.exe -m pip install -r ComfyUI/custom_nodes/ComfyUI-MMAudio/requirements.txt
# create model directory
mkdir -p ComfyUI/models/mmaudio
```

### 16.1.2.2 Download VR we are Custom nodes (comfyui\_stereoscopic)

Use the ComfyUI Manager or Github<sup>12</sup> to download and install the current release of VR we are. The folder name is comfyui\_stereoscopic.

### 16.1.2.3 Fix Dependencies and Start

Dependencies are hard to manage in this environment. The following commands are samples how to fix them. Look into the installer script for the last list.

```
# Fix missing or wrong dependencies
.\python_embeded\python -m pip install -I decorator
.\python_embeded\python -m pip install -I platformdirs
.\python_embeded\python -m pip install -I piexif
.\python_embeded\python -m pip install -I wcwidth
.\python_embeded\python -m pip install -I matplotlib
.\python_embeded\python -m pip install -I pynvml
```

When starting ComfyUI with one of the run scripts, the custom nodes should load without error (symbolic image of a log):

```
[comfyui_stereoscopic] Successfully imported ImageVRConverter
[comfyui_stereoscopic] Successfully imported GetResolutionForVR
[comfyui_stereoscopic] Successfully imported EncryptWatermark
[comfyui_stereoscopic] Successfully imported DecryptWatermark
[comfyui_stereoscopic] Successfully imported StripXML
[comfyui_stereoscopic] Successfully imported SaveStrippedUTF8File
[comfyui_stereoscopic] Successfully imported ScaleByFactor
[comfyui_stereoscopic] Successfully imported RegexSubstitute
[comfyui_stereoscopic] Successfully imported strftime
[comfyui_stereoscopic] OK.

Import times for custom nodes:
0.0 seconds: E:\SD\vrware\ComfyUI_windows_portable\ComfyUI\custom_nodes\websocket_image_save.py
0.0 seconds: E:\SD\vrware\ComfyUI_windows_portable\ComfyUI\custom_nodes\comfyui-custom-scripts
0.0 seconds: E:\SD\vrware\ComfyUI_windows_portable\ComfyUI\custom_nodes\comfyui_stereoscopic
0.0 seconds: E:\SD\vrware\ComfyUI_windows_portable\ComfyUI\custom_nodes\comfyui-frame-interpolation
0.0 seconds: E:\SD\vrware\ComfyUI_windows_portable\ComfyUI\custom_nodes\comfyui_controlnet_aux
0.1 seconds: E:\SD\vrware\ComfyUI_windows_portable\ComfyUI\custom_nodes\comfy-mtb
0.1 seconds: E:\SD\vrware\ComfyUI_windows_portable\ComfyUI\custom_nodes\comfyui-crystools
0.1 seconds: E:\SD\vrware\ComfyUI_windows_portable\ComfyUI\custom_nodes\comfyui-videohelpersuite
0.1 seconds: E:\SD\vrware\ComfyUI_windows_portable\ComfyUI\custom_nodes\comfyui-manager
0.6 seconds: E:\SD\vrware\ComfyUI_windows_portable\ComfyUI\custom_nodes\comfyui-florence2
1.0 seconds: E:\SD\vrware\ComfyUI_windows_portable\ComfyUI\custom_nodes\comfyui-mmaudio
```

When ComfyUI is up, You can start the VR we are service using daemon.bat or daemon.sh. It will bring up the UI, you can restart it later with restart\_gui.bat or gui/restart.sh

---

<sup>11</sup> Replaced comfyui\_controlnet\_aux 1.1.2 due to errors with cpu only.

<sup>12</sup> [https://github.com/FortunaCournot/comfyui\\_stereoscopic/releases](https://github.com/FortunaCournot/comfyui_stereoscopic/releases)

## 16.1.2.4 Install ComfyUI Models

### 16.1.2.4.1 Upscale Model

Download RealESRGAN\_x2plus and RealESRGAN\_x4plus manually from <https://openmodeldb.info/users/xinntao> and place them at ComfyUI\models\upscale\_models  
Or use and configure other upscale models in config.ini later.

### 16.1.2.4.2 Depth Model

With version 3.0 depth\_anything\_v2\_vitb.pth is preconfigured, since the giant model is a performance killer. It is downloaded automatic on first usage – this may take a while (no progress bar in log!).

If you want to use the giant model, download Depth-Anything-V2-Giant manually from <https://huggingface.co/likeabruh/depth Anything v2 vitg/tree/main> and place it at ComfyUI\custom\_nodes\comfyui\_controlnet\_aux\ckpts\depth-anything\Depth-Anything-V2-Giant\depthAnything\_v2\_vitg.pth and configure it in config.ini

### 16.1.2.4.3 Dubbing Model

Please follow this Guide on Github to install MMAudio models:

<https://github.com/kijai/ComfyUI-MMAudio?tab=readme-ov-file#installation>

In short:

Download the fp16 variants of the models from huggingface (4 files):

[https://huggingface.co/Kijai/MMAudio\\_safetensors/tree/main](https://huggingface.co/Kijai/MMAudio_safetensors/tree/main)

and move them to ComfyUI\models\mmaudio

### 16.1.2.4.4 ControlNet Model<sup>V3.1</sup>

For the recolor task the following model (from ComfyUI model manager) is required:  
stabilityai/control-lora-recolor-rank256.safetensors

### 16.1.3 Manual Install „VR we are“

Use ComfyUI Manager to install the latest version of „VR we are“ (= `comfyui_stereoscopic`). When restarting the ComfyUI server, watch the log files for any errors<sup>13</sup>.

From the ComfyUI/custom\_nodes/comfyui\_stereoscopic folder you can create desktop shortcuts from the two batch files, by dragging them with the right(!) mouse key on the desktop. They will call shell scripts. Instead you can still call the shell scripts by hand from git bash; this can be necessary to investigate errors. The desktop shortcuts can be configured with icon and you can choose to minimize the window shell.

Now call `demon.bat` or `daemon.sh` and the installation will complete and be tested (takes up to 3 minutes). You will notice a small guy window to popup as well. If the tests went well, you can do the configuration. Close the daemon by pressing **CTRL + C** in the bash shell.

To call the script manually, open a new Git Bash Shell, and drag the script there or enter:

```
# Locate and switch to installation folder:  
cd ComfyUI_windows_portable/ComfyUI  
# Execute background task  
./custom_nodes/comfyui_stereoscopic/daemon.sh
```

### 16.1.4 Configure „VR we are“

The properties are located at `ComfyUI\user\default\comfyui_stereoscopic\config.ini`

I recommend to check and complete the values for the following keys:  
`EXIFTOOLBINARY`, `DESCRIPTION_LOCALE`, `ISO_639_2_CODE`, `WATERMARK_LABEL`

If TVAI is installed provide `TVAI_BIN_DIR`, `TVAI_MODEL_DATA_DIR`, `TVAI_MODEL_DIR` (request CLI instructions Video AI – Nvidia from support if necessary).

If you don't want to the pipeline, e.g. it is confusing you; disable auto forwarding by setting `PIPELINE_AUTOFORWARD` to 0. Auto forwarding is a powerful feature; you can edit the pipeline in `ComfyUI\user\default\comfyui_stereoscopic\rebuild_autoforward.sh` which is executed every time the daemon is started.

Many more properties can be modified there. If the tool fails, delete the config file and the generator will generate a new one with default values.

You need to restart `daemon.sh` after changing the configuration. You do not need to restart the ComfyUI server.

#### 16.1.4.1 Configure for ComfyUI Desktop

For ComfyUI Desktop, the port needs to be changed, since the scripts are using the default port of ComfyUI Portable. Go either to Settings->Server Config, and set the value for the port to 8188, or change the value of the property `COMFYUIPORT` in `config.ini` to the ComfyUI Desktop port (8000, look into Settings in ComfyUI Window). Sometimes when you close the window, the new instance will allocate a different port, like 8002, because the old process has not terminated.

---

<sup>13</sup> The latest ComfyUI requires `opencv-python>=4.10.0.84` and raise an import error for `cv2` package, in case you try to use it with version 3.0 of `comfyui_stereoscopic` module, which can be changed in `requirements.txt` file.

### **16.1.5 Custom ComfyUI Pathes**

Custom pathes are not supported. Three directories are expected to be next to `custom_nodes`: `input`, `output` and `user`

## 17 CLI-only Installation (without ComfyUI)

The CLI Tools include their own Setup and can be installed without ComfyUI. They can be used under Linux as well.

First get Python and “VR we are”:

1. Install Python 3.13.6. (at least 3.12) from <https://www.python.org/>
2. Download and unpack the latest “VR we are” Release from  
[https://github.com/FortunaCournot/comfyui\\_stereoscopic](https://github.com/FortunaCournot/comfyui_stereoscopic)

Create a Virtual Environment and activate it:

```
python -m venv venv  
venv\Scripts\activate under Linux: source venv/bin/activate
```

### 17.1 Side-By-Side CLI

Then install the Side by Side CLI:

4. Go to “VR we are” folder cli/sbs
5. Run the Setup Script: `python setup_env.py`

The tool can be run now:

Video mode:

```
python main.py -i input.mp4 -o output_sbs.mp4 --preset balance
```

Folder with frames :

```
python main.py -i frames -o output --input-type folder --preset balance
```

Image-to-Image (single or folder):

```
python main.py -i frame.png -o frame_sbs.png --input-type i2i
```



## 18 Acknowledgements

The software was developed by me, Fortuna, with lot of help from lablunoshka, who is responsible for the insane speed of the SBS algorithm. We did this in the hope, that we soon see more VR content, from AI here or classic somewhere else.

Additional help come from some artists, namely (in alphabetic order): Robotka, schmede, VisionaryAI\_Studio and z1000k. You will see the result of this tool in the many examples in the gallery on Civitai. Please visit and rate their original work (links included in posts), instead of

the stereoscopic variant. I want to say "thank you!" to them for the opportunity to propagate the potential of this tool through their artwork.

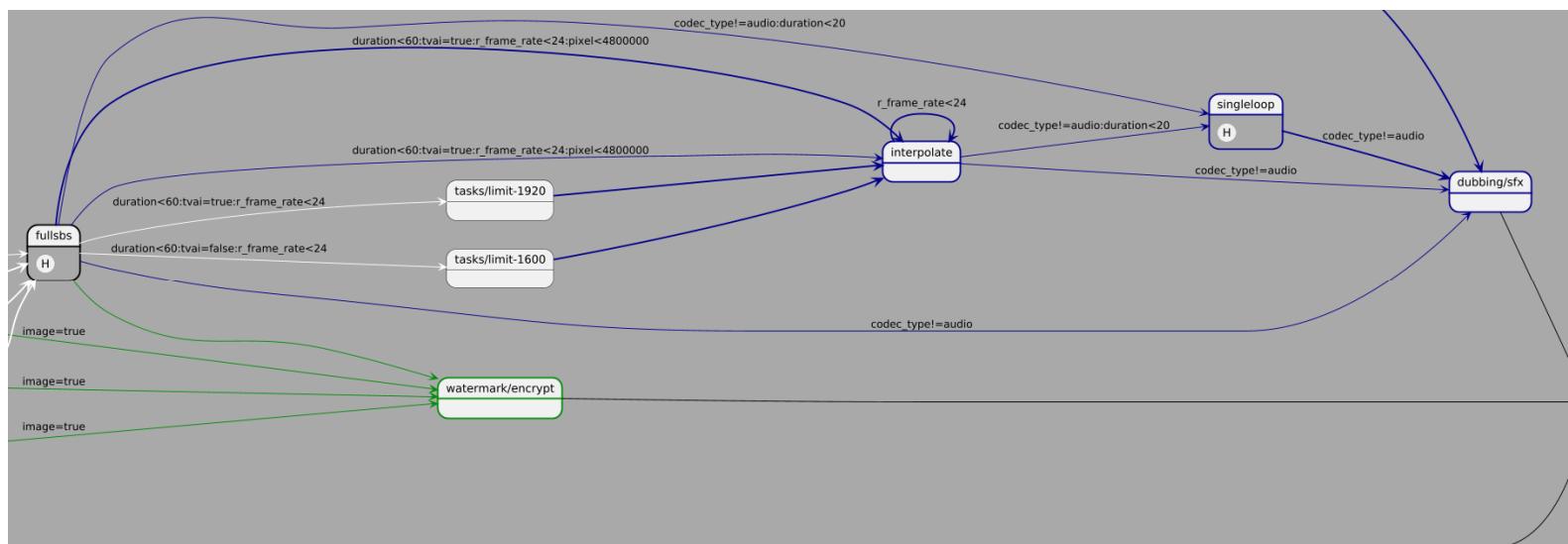
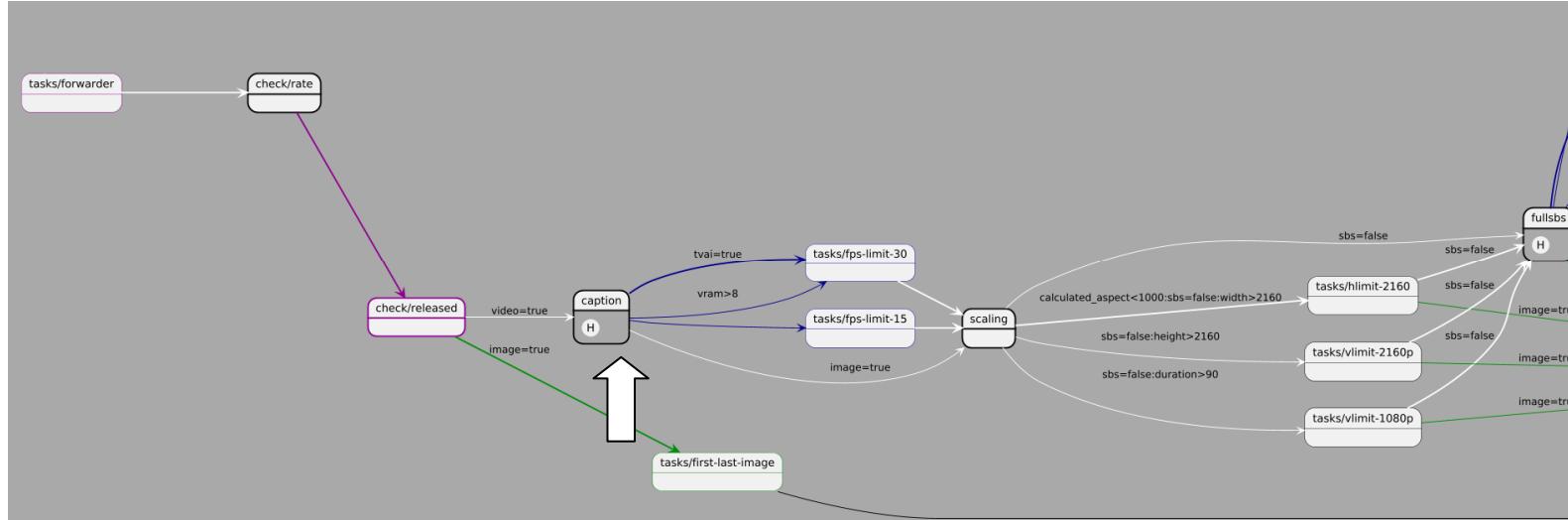
## 19 Appendix: Default Pipeline

Best practice: Put your files at input folder of stages, recommended are those with the H symbol, that keep your input files.

They are located under ComfyUI/input/vr/

You can use the UI app status page to open the windows explorer directory by clicking on a cell.

The following workflow of pipeline depends on your installation choices and may vary. Use the app to see your active pipeline.



## 20 Appendix: Custom Node Packs Compatibility Tests

The following Custom Node Packs did install with ComfyUI 0.3.62. This list is not frequently updated.  
The installer may use newer versions.

1. rgthree-comfy 1.0.2509092031, <https://github.com/rgthree/rgthree-comfy>
2. comfyui-easy-use 1.3.3, <https://github.com/yolain/ComfyUI-Easy-Use>
3. ComfyUI-GGUF, 1.1.4, <https://github.com/city96/ComfyUI-GGUF>
4. comfyui-inspire-pack, 1.22.0, <https://github.com/ltdrdata/ComfyUI-Inspire-Pack>
5. comfyui\_controlnet\_aux, 1.1.2, [https://github.com/Fannovel16/comfyui\\_controlnet\\_aux](https://github.com/Fannovel16/comfyui_controlnet_aux)
6. comfyui-kjnodes, 1.1.7, <https://github.com/kijai/ComfyUI-KJNodes>
7. comfyui-mxtoolkit, 0.9.92, <https://github.com/Smirnov75/ComfyUI-mxToolkit>
8. comfyui\_yfg\_comical, 1.2.3, [https://github.com/gonzalu/ComfyUI\\_YFG\\_Comical](https://github.com/gonzalu/ComfyUI_YFG_Comical)

Nodes that come with the installer are currently downloaded from a fork. Updating through ComfyUI may currently not work; you have to replace them first with a git checkout from the links above.