
Real-CUGAN ncnn Vulkan

:exclamation: :exclamation: :exclamation: This software is in the early development stage, it may bite your cat



ncnn implementation of Real-CUGAN converter. Runs fast on Intel / AMD / Nvidia / Apple-Silicon with Vulkan API.

realcugan-ncnn-vulkan uses ncnn project as the universal neural network inference framework.

Download

Download Windows/Linux/MacOS Executable for Intel/AMD/Nvidia/Apple-Silicon GPU

<https://github.com/nihui/realcugan-ncnn-vulkan/releases>

This package includes all the binaries and models required. It is portable, so no CUDA or PyTorch runtime environment is needed :)

About Real-CUGAN

Real-CUGAN (Real Cascade U-Nets for Anime Image Super Resolution)

<https://github.com/bilibili/ailab/tree/main/Real-CUGAN>

Usages

Example Command

```
1 realcugan-ncnn-vulkan.exe -i input.jpg -o output.png
```

Full Usages

```
1 Usage: realcugan-ncnn-vulkan -i infile -o outfile [options]...
2
3 -h                show this help
4 -v                verbose output
5 -i input-path     input image path (jpg/png/webp) or directory
6 -o output-path    output image path (jpg/png/webp) or directory
7 -n noise-level    denoise level (-1/0/1/2/3, default=-1)
```

```
8 -s scale          upscale ratio (1/2/3/4, default=2)
9 -t tile-size     tile size (>=32/0=auto, default=0) can be 0,0,0
    for multi-gpu
10 -c syncgap-mode  sync gap mode (0/1/2/3, default=3)
11 -m model-path   realcugan model path (default=models-se)
12 -g gpu-id       gpu device to use (-1=cpu, default=auto) can be
    0,1,2 for multi-gpu
13 -j load:proc:save thread count for load/proc/save (default=1:2:2)
    can be 1:2,2,2:2 for multi-gpu
14 -x             enable tta mode
15 -f format       output image format (jpg/png/webp, default=ext/
    png)
```

- `input-path` and `output-path` accept either file path or directory path
- `noise-level` = noise level, large value means strong denoise effect, -1 = no effect
- `scale` = scale level, 1 = no scaling, 2 = upscale 2x
- `tile-size` = tile size, use smaller value to reduce GPU memory usage, default selects automatically
- `syncgap-mode` = sync gap mode, 0 = no sync, 1 = accurate sync, 2 = rough sync, 3 = very rough sync
- `load:proc:save` = thread count for the three stages (image decoding + realcugan upscaling + image encoding), using larger values may increase GPU usage and consume more GPU memory. You can tune this configuration with “4:4:4” for many small-size images, and “2:2:2” for large-size images. The default setting usually works fine for most situations. If you find that your GPU is hungry, try increasing thread count to achieve faster processing.
- `format` = the format of the image to be output, png is better supported, however webp generally yields smaller file sizes, both are losslessly encoded

If you encounter a crash or error, try upgrading your GPU driver:

- Intel: <https://downloadcenter.intel.com/product/80939/Graphics-Drivers>
- AMD: <https://www.amd.com/en/support>
- NVIDIA: <https://www.nvidia.com/Download/index.aspx>

Build from Source

1. Download and setup the Vulkan SDK from <https://vulkan.lunarg.com/>
 - For Linux distributions, you can either get the essential build requirements from package manager

```
1 dnf install vulkan-headers vulkan-loader-devel
```

```
1 apt-get install libvulkan-dev
```

```
1 pacman -S vulkan-headers vulkan-icd-loader
```

2. Clone this project with all submodules

```
1 git clone https://github.com/nihui/realcugan-ncnn-vulkan.git
2 cd realcugan-ncnn-vulkan
3 git submodule update --init --recursive
```

3. Build with CMake

- You can pass `-DUSE_STATIC_MOLTENVK=ON` option to avoid linking the vulkan loader library on MacOS

```
1 mkdir build
2 cd build
3 cmake ../src
4 cmake --build . -j 4
```

Sample Images

Original Image



Upscale 2x with ImageMagick

```
1 convert origin.jpg -resize 200% output.png
```



Upscale 2x with ImageMagick Lanczo4 Filter

```
1 convert origin.jpg -filter Lanczos -resize 200% output.png
```



Upscale 2x with Real-CUGAN

```
1 realcugan-ncnn-vulkan.exe -i origin.jpg -o output.png -s 2 -n 1 -x
```



Original Real-CUGAN Project

- <https://github.com/bilibili/ailab/tree/main/Real-CUGAN>

Other Open-Source Code Used

- <https://github.com/Tencent/ncnn> for fast neural network inference on ALL PLATFORMS
- <https://github.com/webmproject/libwebp> for encoding and decoding Webp images on ALL PLATFORMS
- <https://github.com/nothings/stb> for decoding and encoding image on Linux / MacOS
- <https://github.com/tronkko/dirent> for listing files in directory on Windows