

Guideline

Document name

Content production guidelines:
AI-Based Super Resolution Upscaling



Project Acronym:	IMMERSIFY
Grant Agreement number:	762079
Project Title:	Audiovisual Technologies for Next Generation Immersive Media

Revision:	0.5
Authors:	Ali Nkrang, Johannes Pöll
Support and review	Diego Felix de Souza, Mauricio Alvarez Mesa
Delivery date:	March 31 2020
Dissemination level (Public / Confidential)	Public

Abstract

This guideline aims to give an overview of the state of the art software-implementations for video and image upscaling. The guideline focuses on the methods that use AI in the background and are available as standalone applications or plug-ins for frequently used image and video processing software. The target group of the guideline is artists without any previous experiences in the field of artificial intelligence.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 762079.



TABLE OF CONTENTS

Introduction	3
Using Machine Learning for Upscaling	4
General Process	4
Install FFmpeg	5
Extract the Frames and the Audio Stream	5
Reassemble the Frames and Audio Stream After Upscaling	5
Upscaling with Waifu2x	6
Further Information	8
Upscaling with Adobe CC	10
Basic Image Upscale using Adobe Photoshop CC	10
Basic Video Upscales using Adobe CC After Effects	15
Upscaling with Topaz Video Enhance AI	23
Topaz Video Enhance AI workflow	23
Conclusion	28



1 Introduction

Upscaling of digital pictures (such as video frames) refers to increasing the resolution of images using algorithmic methods. This can be very useful when for example, artists work with low-resolution materials (such as old video footages) that need to be used in a high-resolution output. In the scope of this guideline, we will only focus on methods that are based on artificial intelligence (or more precisely deep learning). The target group of this guide is artists who have no previous experience in upscaling with machine learning algorithms and want to get an overview of available systems and how they can be used for their work with as little effort as possible.

Recent researches in the field of AI-based upscaling technologies have shown promising results. The source code of the main part of this research is published as Open-Source and can be used by researchers, engineers and artists. However, running the source code of a research project in the field of machine learning usually requires particular hardware (e.g., powerful CPU, latest NVIDIA GPU) and a special development environment setup (e.g., a Linux distribution with CUDA, Python, TensorFlow, PyTorch, besides others). It can be a challenging and time-consuming task for people without previous experiences in the field. Nonetheless, there are several implementations of these upscaling technologies as standalone applications. Additionally, some models have been published as plugins for commonly used image processing applications such as Adobe CC. Image upscaling models can also be used for videos, where the upscaling algorithm is applied to each frame separately. Consequently, they do not take dependencies between video frames into consideration. In this document, we only focus on image-based algorithms. To our knowledge, there is no standalone application or plugin available that also considers the visual dependencies between the video frames.

It should be noted that the type of image data could also influence the quality of the final results: deep learning models are trained with a large number of examples coming from a so-called training-dataset. Datasets could contain pictures of different types of content or be specialized for a certain kind of content (e.g., anime content). Consequently, models perform better for the special type of content they are trained with, but less for content types that are highly different from the training dataset. For instance, assuming a model that is trained with pictures of natural landscapes; The upscaling quality of such a model would be best for natural pictures, but it can suffer for other types of content such as anime.

Therefore, for the best possible result for a special type of data, the model should be trained by the user. On the downside, this requires the availability of the data, as well as proper hardware or expensive cloud computing services for training.

In most cases, however, users could use pre-trained models. For instance, standalone applications like Topaz Gigapixel AI or Waifu2x-Caffe come with pre-trained models and provide an easy way to use advanced deep learning technologies to upscale videos and images.

In this guideline, we will explain three AI-based models for video upscaling that can be used by artists without great effort. In a step-by-step guide, we will explain the progress of upscaling a video within a standard software like Adobe CC or with a standalone application like Topaz Gigapixel or Waifu2x.



2 Using Machine Learning for Upscaling

This section describes the basic idea behind using deep neural networks for upscaling images and videos.

Convolutional neural networks (CNN) are a type of neural networks in the field of deep learning technology and are widely applied for learning particular features of visual data. A detailed description of the inner working of convolutional neural networks is out of the scope of this guideline. Nevertheless, we will briefly describe the main idea behind using CNN for upscaling since different types of CNN are very often used in AI-driven upscaling algorithms.

Like other deep neural networks, the basic architecture of a CNN consists of an input and an output layer with several successive hidden layers in between. Each layer learns to recognize special features of its input and passes the information to the next layer. While the first layers of the network learn very basic local features of the data, the subsequent layers learn features that are more complex since they combine the features learned in the previous layers. Consequently, patterns learned in the network's deeper layers have much more statistical dependencies on context than the layers at the beginning of the network. In other words, the further a layer is from the input layer, the more contextual dependencies it can learn.

Deep neural networks are trained with a dataset of examples to do a specific task. In image-based upscaling models, the task is usually to predict (or generate) a higher version of an input image. Since a lower version of an image contains fewer pixels than its corresponding high-resolution version, an upscaling system has to generate new pixels based on the context. A very simple upscaling algorithm would only take the neighbor pixels as a context into consideration and generate new pixels by applying a type of interpolation between them. AI-based models use the statistical dependencies between pixels and consider their local and global context and generate the missing pixels in a way that statistically conforms to the training data set. In other words, an AI-driven upscaling system generates new visual data that is more consistent with the visual context of the input image. These features are learned during the training and are characteristic of the corresponding training set. For example, imagine two AI-driven upscaling systems; the first system was trained with anime images and the second with natural landscape images. The anime system would attempt to upscale the image by using (generating) visual features that appear in the anime data set, such as sharp lines or curves with high contrast to the background. The landscape system, on the other hand, might try to create smoother transitions between different types of natural terrains, such as water, meadows, and trees.

3 General Process

For the seek of flexibility and for the best compression results, we recommend the following steps:

- 1) Extract the video frames from the bitstream as a PNG sequence
- 2) Extract the audio from the bitstream
- 3) Upscale the PNG files using one of the methods mentioned in this document
- 4) Recombine the upscale frames and the audio stream and create a new bitstream with higher video resolution



For this progress, we will use FFmpeg¹. FFmpeg is an open-source application containing several libraries, programs that can be used for handling numerous types of multimedia content. We will explain the steps required for extracting and combining video frames using FFmpeg in this section.

3.1 Install FFmpeg

First, we need to download FFmpeg executables for Windows.

- 1) Go to <https://ffmpeg.zeranoe.com/builds/win64/static/> and download the latest version of FFmpeg for Windows (ffmpeg-latest-win64-static.zip) and extract the content on your Windows machine.
- 2) For faster access within the operating system, we can add the path of FFmpeg executable to the environment variable. The advantage of adding the path to the environment variable is that we can use FFmpeg from anywhere in the operating system and it is not necessary to remember the exact path each time we want to use it.

In order to add the FFmpeg to your environment variable, the following steps are required:

- a) Open cmd.exe as administrator
- b) Run `setx /M PATH "path\to\ffmpeg\bin;%PATH%"`²

Example: `setx /M PATH "c:\\programs\\ffmpeg\\bin;%PATH%"`

Now, FFmpeg should be ready for use within the operating system. You can use FFmpeg also without adding it to the environment variable. In that case, you should always use the full path to the FFmpeg executable (for instance `c:\\programs\\ffmpeg\\ffmpeg.exe`)

3.2 Extract the Frames and the Audio Stream

- 3) To achieve the best possible quality, we use PNG for extracted video frames, as it is a lossless compression format that is supported by all upscaling applications mentioned in this guideline. However, it may require a large amount of disk space on the hard drive compared to the original video file.
 - a) First, create a folder named "output"
 - b) Open cmd.exe, navigate (cd) to the folder containing "output"
 - c) Run `%ffmpeg% -i input-video.mp4 -r 60 output/pic%05d.png`
 - d) Run `%ffmpeg% -i input-video.mp4 audio-of-input-video.aac`

The above command extracts all frames in the `input-video.mp4` and gives them an increasing number as part of their names.

The argument after `-i` specifies the name of the input file (in our case `input-video.mp4`) and the argument after `-r` specifies the frame rate of the video (in our case, 60).

In the example above, the audio stream is extracted in aac format, but other audio formats can also be used.

3.3 Reassemble the Frames and Audio Stream After Upscaling

Once we have upscaled the frames, we can create the target video by combining the frames. However, the target video should also be compressed and properly encoded. In order to ensure

¹ <https://ffmpeg.zeranoe.com>

² You can also use the System Properties dialog box (Progress tab) to update the environment variable.



high-quality results, it is crucial to use the right encoder and encoding options. For instance, for all videos beyond 4K, we recommend using the x265 (or hevc265) codec. Please note that high-resolution videos contain much more information per frame than their low-resolution versions, so they require a higher bit rate to keep the same visual quality. The final bitrate depends on several factors, such as encoder, resolution and even the type of content. For example, point cloud content should be encoded at a higher bit rate than camera content since point cloud content has lower spatial redundancy to be exploited by video encoder algorithms.

FFmpeg supports hevc265 encoding³. The following command creates a video (upscaled-video.mp4) from the images (pic%05d.png) stored in the "output" folder and combines it with the audio stream extracted from the input file (audio-of-input-video.aac). The codec used here is libx265 and the quality is set by constant rate factor (crf=25). The crf can be set to a number between 0-51, which refers to the compression intensity. That means, lower numbers result in better quality (less compression). Using this option is recommended as FFmpeg will use the optimal bitrate to achieve a certain (constant) quality.

```
%ffmpeg% -r 60 -i output/pic%05d.png -i audio-of-input-video.aac -vcodec libx265 -crf 25 -pix_fmt yuv420p -acodec copy upscaled-video.mp4
```

FFmpeg comes with a wide range of encoding options. For more information about the encoding options and the use of x265, see <https://ffmpeg.org/ffmpeg.html> and <https://trac.ffmpeg.org/wiki/Encode/H.265>.

4 Upscaling with Waifu2x

Waifu2x is a famous super-resolution model for anime-style images. Originally it was designed for upscaling anime style images (known as Waifu). However, Waifu2x can also be used for photos and other types of data according to the available pre-trained models that are trained with different types of content. Another nice feature of Waifu2x is the ability to reduce noise within the images. Since Waifu2x is open source, the model can be trained with custom data that includes even other types of visual content besides anime.

According to the official description on GitHub⁴, it is inspired by Super-Resolution Convolutional Neural Network (SRCNN)⁵, a convolutional neural network based model for image super resolution. Image super resolution refers to a class of techniques that can be used to increase the resolution of an image. Due to its popularity, there are several standalone implementations of Waifu2x for Windows, Linux and macOS. With these implementations, it is quite straightforward to use it in almost any software and hardware environment. For example, these applications can use either the CPU or the GPU for computation, and for the latter there are implementations that are not limited to

³ However, in the scope of the Immersify project, instead of FFmpeg we use an optimized and high-performance HEVC encoder developed by Immersify partner Spin Digital named as Spin Enc. The Spin Enc is a command-line software that can be used to create videos from sequences of images by encoding them according to the HEVC standard. For more information about Spin Enc please check https://spin-digital.com/products/spin_enc.

⁴ <https://github.com/nagadomi/waifu2x>

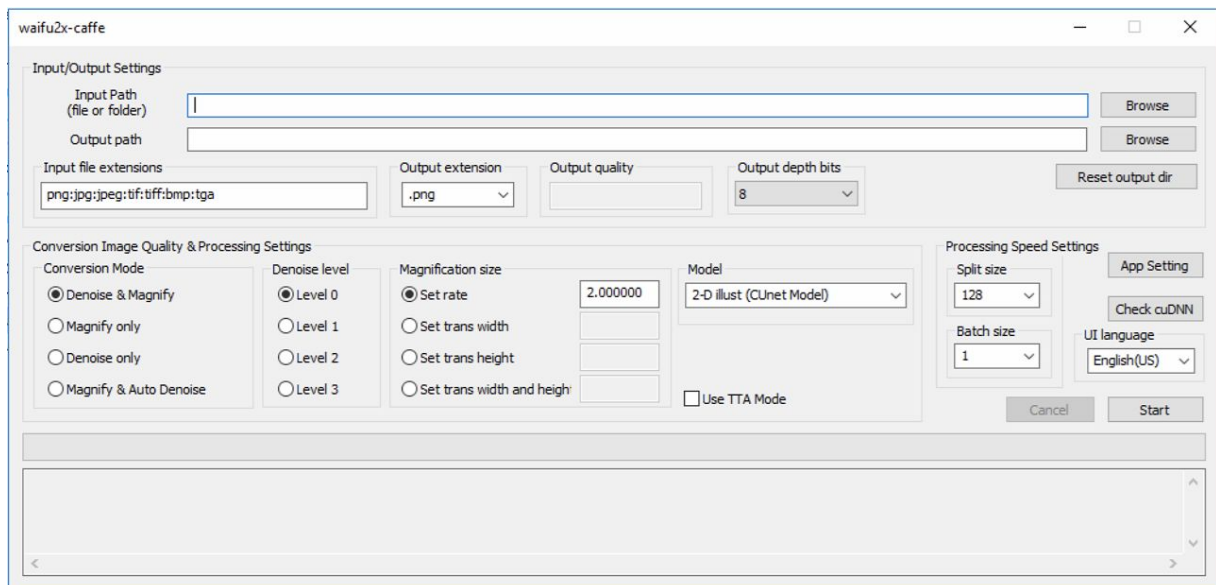
⁵ <https://arxiv.org/pdf/1501.00092.pdf>



CUDA (which is a NVIDIA framework and requires a NVIDIA GPU) and also support OpenCL⁶. OpenCL (Open Computing Language) is currently supported on all main GPU manufacturers (NVIDIA, AMD and Intel).

In this guideline, we will focus on Waifu2x-caffe⁷, which is a highly rated and recommended implementation of Waifu2x. The installation of Waifu2x-Caffe is uncomplicated:

1. Go to the GitHub repository of Waifu2x-caffe (<https://github.com/litcgjie/waifu2x-caffe>) and click on the tab “releases” and download the executable zip-file (waifu2x-caffe.zip) from there (At the time of writing this document the latest version is 1.2.0.2)
2. Unzip the content of the zip file
3. Run “waifu2x-caffe-cui.exe”



The user interface of Waifu2x-Caffe

Waifu2x-caffe-cui.exe comes with a quite self-explanatory interface (showing in the picture above) that contains the following options:

- **check cuDNN:** It verifies if cuDNN⁸ is supported on the machine. cuDNN is an NVIDIA library for GPU-accelerated deep neural networks. Using cuDNN required an NVIDIA GPU. If cuDNN is available on the system, it will be used during the upscaling process. Otherwise, the CPU will be used instead.
- **Input and output Path:** Select the folder containing all input images and define a folder where the output images should be stored.
- **Conversion Mode:** Waifu2x-Caffe can be used for magnifying the size of images as well as for reducing the compression artifacts of images. Users can choose between magnification only, noise reduction only, or both magnifying and denoising of input images.
- **Denoise level:** The denoising level can be set here. 0 means low noise reduction and 3 means the highest noise reduction. The denoising options refer to JPEG-typical artefacts that occur as a result of compression. The model has been trained with different noise levels⁹. We recommend using this option if the input images are compressed with JPEG. Use noise level 0

⁶ <https://www.khronos.org/opencl/>

⁷ <https://github.com/litcgjie/waifu2x-caffe>

⁸ <https://developer.nvidia.com/cudnn>

⁹ <https://github.com/nagadomi/waifu2x/issues/320>



for input images compressed with JPEG quality 85-95 and noise level 1 for JPEG quality 65-85. The application will decide whether or not to apply a noise reduction model (based on the input format) when “Magnify & Auto Denoise” is selected.

- **Magnification size:** The original Waifu2x model only supports 2x upscaling. All scale values above that are realized by rerunning the upscaling. For example, upscaling a 480x270 to 1920x1080 means that the upscaling is first performed to 960x540 and then again from 960x540 to 1920x1080. Waifu2x-Caffe supports arbitrary sizes too. However, it is done by downscaling the image after scaling up once or several times to a size larger than the target size. Because of this limitation, it is recommended to use 2x (2.00000) upscaling rather than arbitrary values, as the best result can be achieved if we use the exact scaling value the model has been trained with.
- **Model:** Waifu2x-Caffe comes with several pre-trained models that can be divided into two main categories: photo and illustration. The choice of the pre-trained model depends strongly on the input content. For art-like content (such as anime), an illustration model is recommended, while a photo model is for camera contents. However, it is not possible to make a statement about which model will work best without an empirical test. Therefore, it is strongly recommended to take a short (e.g., 10 seconds) representative content of the input video and try it with different versions of photo or illustration models to find out which model gives the best results.
- **TTS:** If this option is activated, the application uses a technique called “test-time augmentation” to improve the quality of the result. It applies data augmentation techniques (such as rotation, shift, crop or flip) and creates eight versions of the input picture in the first step and averages the upscaling results in the second step. Using this technique Waifu2x-Caffe is able to remove more artefacts from the input picture. On the downside, it is 8x slower.

4.1 Further Information

The original repository of Waifu2x contains several scripts to use Waifu2x for upscaling and training the model. By using these scripts, it is possible to train the model with custom images. For more information, please check: <https://github.com/nagadomi/waifu2x>.

There are several re-implementations of Waifu2x beside Waifu2x-Caffe. For instance, Waifu2x-converter-cpp¹⁰, which uses OpenCV instead of CUDA for GPU-accelerated calculation and thus does not require an NVIDIA GPU, or Waifu2x-ncvv-vulkan, that re-implements Waifu2x in ncnn¹¹ and runs on Intel/AMD/NVIDIA devices using Vulkan API.

¹⁰ <https://github.com/DeadSix27/waifu2x-converter-cpp>

¹¹ <https://github.com/Tencent/ncnn>



Figure 1. Comparing original HD frame¹² area to 4K and 8K upscaling via Waifu2x-Caffe

¹² All example frames in this guideline are taken from Sung Rok's animation "Scroll Down Journey"



Figure 2. Comparing HD scaled to 400% (nearest neighbor) with 200% 4K upscale & 100% 8K upscale via Waifu2x-caffe (using 2-D illust, CUnet Model, TTA activated, Denoising & Magnifying)

5 Upscaling with Adobe CC¹³

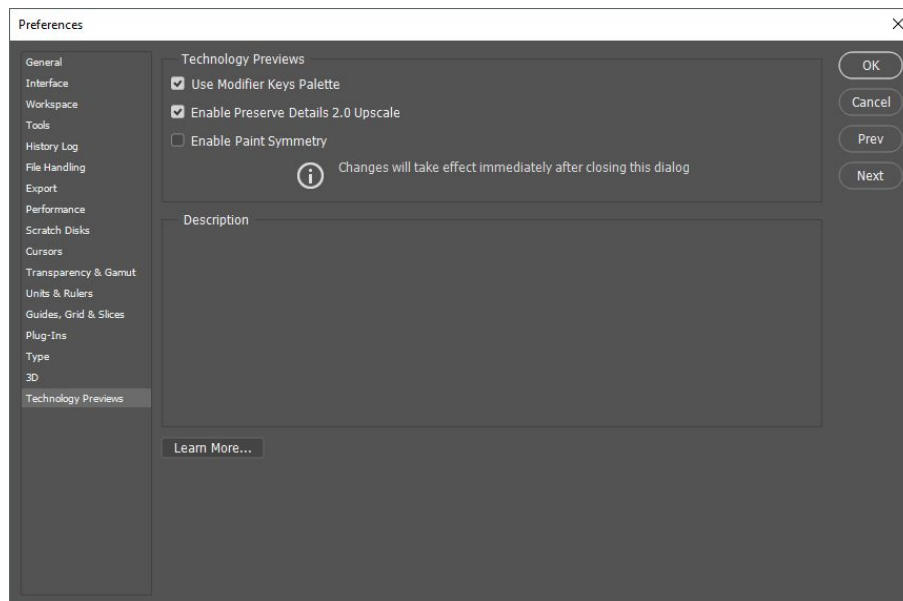
In this section, we use the Preserve Details 2.0 resampling method for upscaling. Preserve Details 2.0 resampling is an upscaling algorithm introduced in Adobe Photoshop CC 2018. It is Adobe's most advanced upscaling technology used in a variety of Adobe CC apps like Photoshop or After Effects (Detail-preserving Upscale). It utilizes artificial intelligence to interpolate more effectively. Adobe CC can be considered an industry standard for image/video workflows. Thus it is used as a control group to compare results.

5.1 Basic Image Upscale using Adobe Photoshop CC

The interface of Preserve Details 2.0 Resampling is quite simple and self-explanatory:

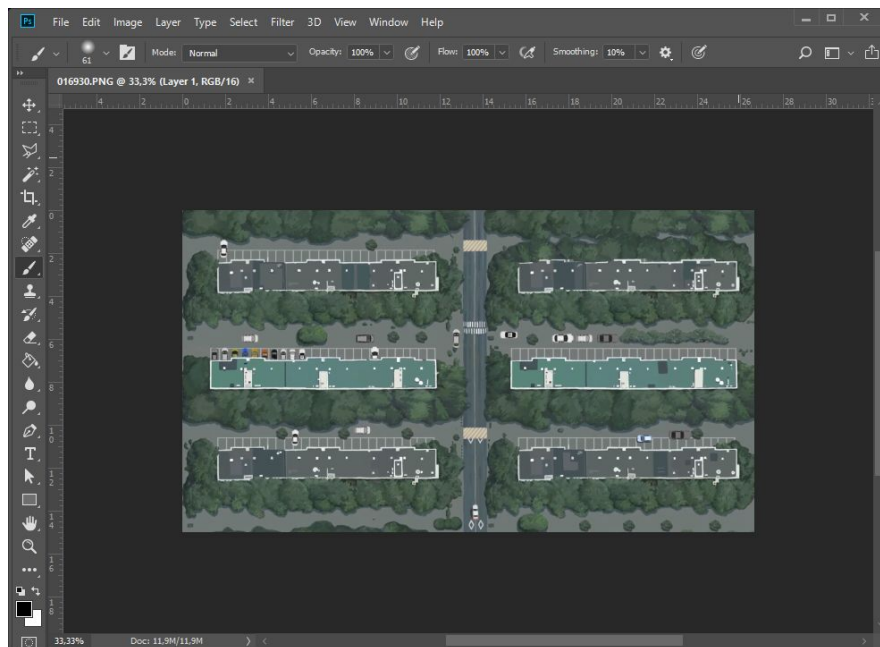
- (1) Use Adobe Photoshop CC 2018 or newer. If you are using Adobe Photoshop CC 2018 you have to enable Preserve Details 2.0 upscaling in **Preferences/Technology Previews**.

¹³ <https://helpx.adobe.com/photoshop/using/technology-previews.html>



This feature is already enabled in more recent versions of Adobe Photoshop.

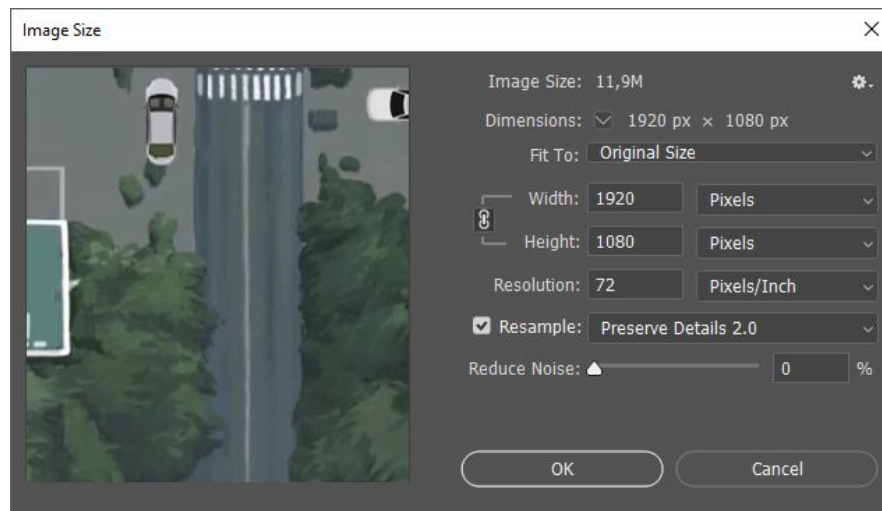
- (2) Open your original image file in Adobe Photoshop CC (we used a Frame of Sung Rok's animation "Scroll Down Journey").



(2)

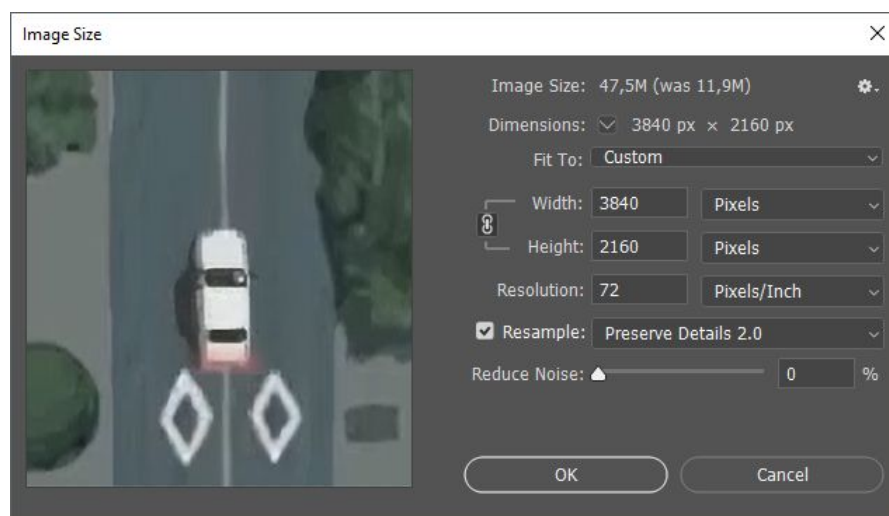


- (3) Go to **Image/Image Size...** context menu (key combo $\text{⌘}+\text{I}$ on MAC or **Alt+Ctrl+I** on PC) and set it to **Preserve Details 2.0** next to the Resample tickbox.

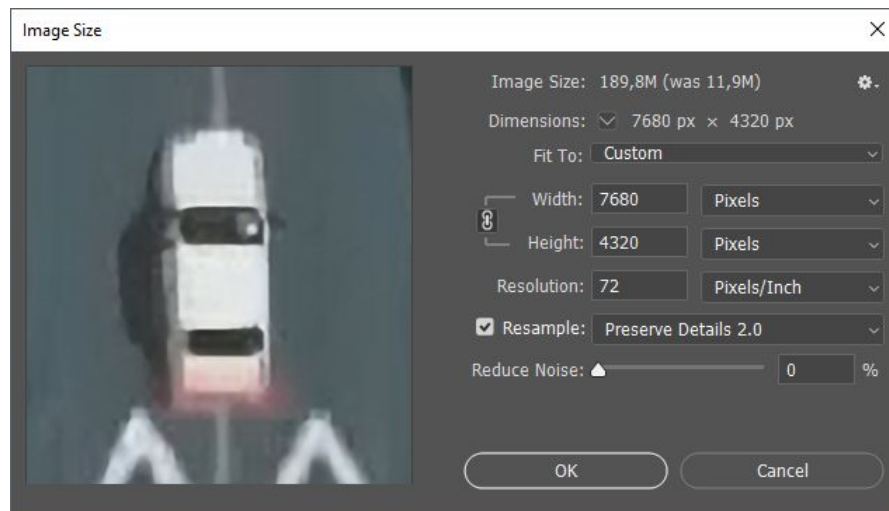


(3)

- (4) Keep width & height locked together (to preserve image proportions) and change pixel size of height or width to desired higher resolution (1920x1080px = Full HD, 3840x2160px = 4K, 7680x4320px = 8K). The “Reduce Noise” option can be used to get better results (should be tested individually for different qualities of input images).



(4) scaling to 4K



(4) scaling to 8K

- (5) Hitting OK will upscale your image inside Photoshop. You can then save the upscaled version as e.g., PNG file (theoretically saving the above procedure/steps as an “Adobe Photoshop action” allows for image sequence batch processing). Also see: <https://helpx.adobe.com/photoshop/using/creating-actions.html>.

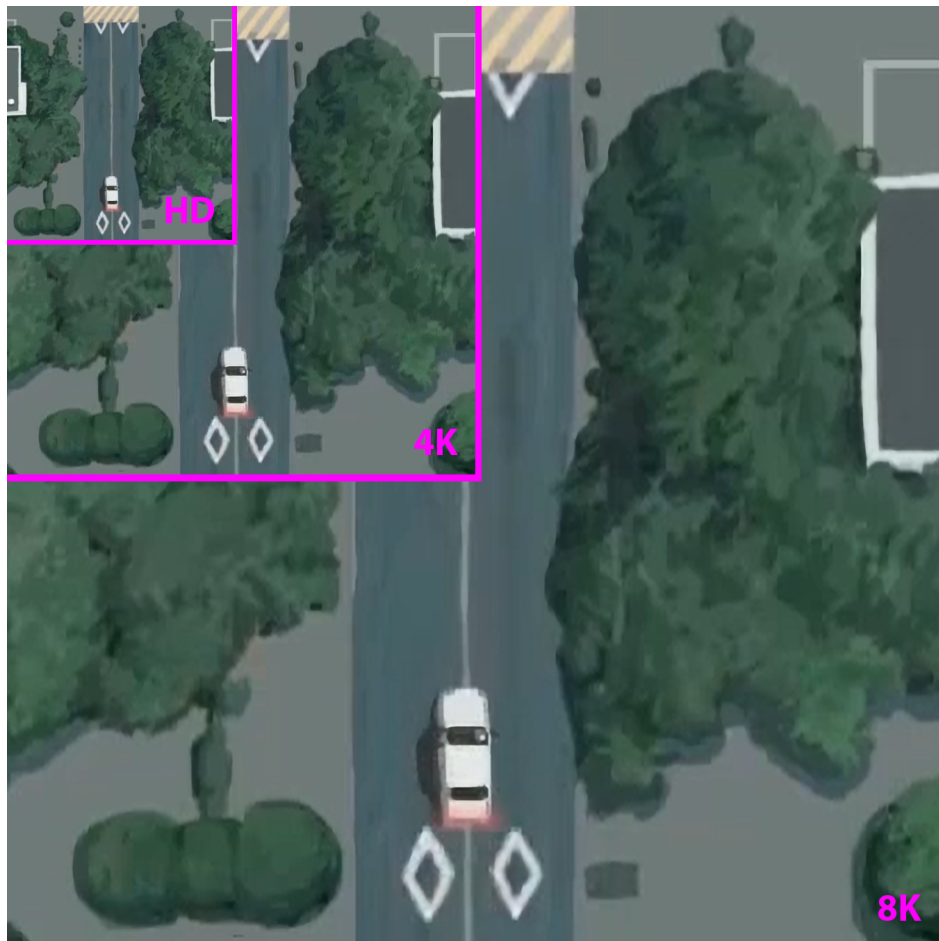


Figure 3. Comparing original HD frame area to 4K and 8K upscaling via Adobe Photoshop

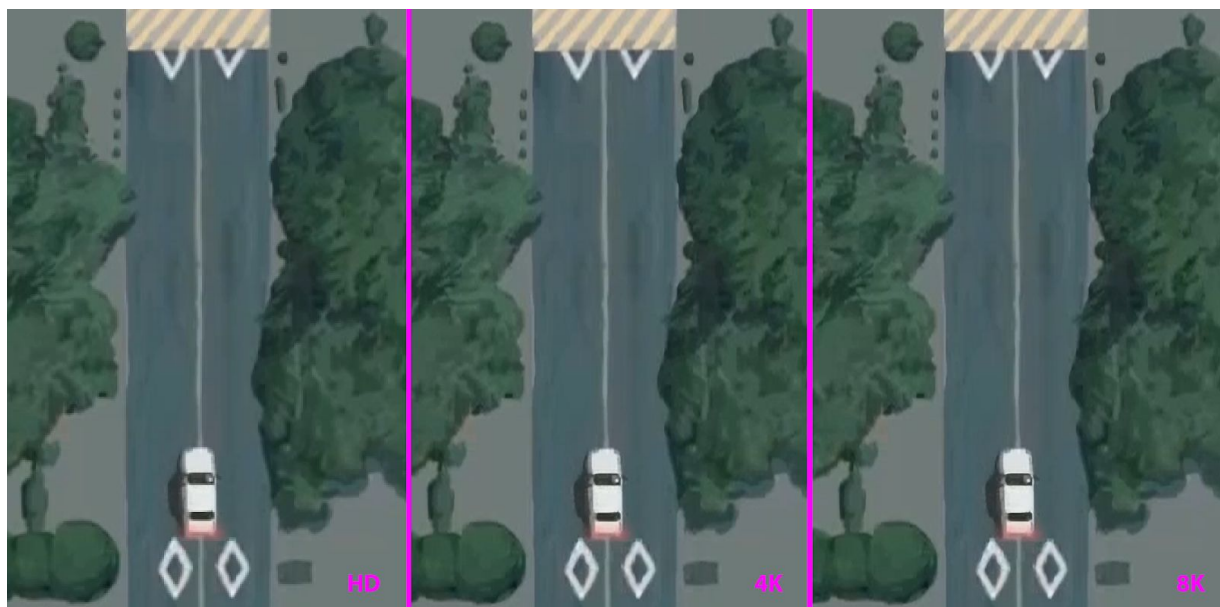


Figure 4. Comparing HD scaled to 400% (nearest neighbor) with 200% 4K upscale & 100% 8K upscale via Adobe Photoshop (Preserve Details 2.0)

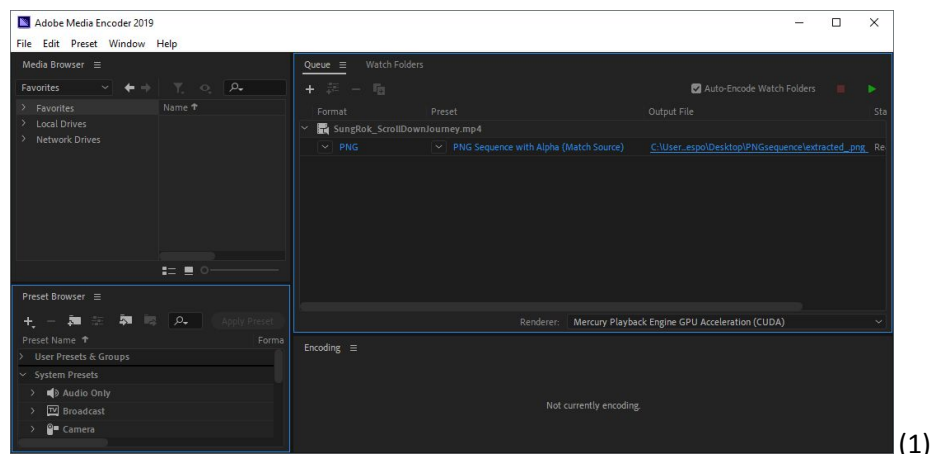


5.2 Basic Video Upscales using Adobe CC After Effects

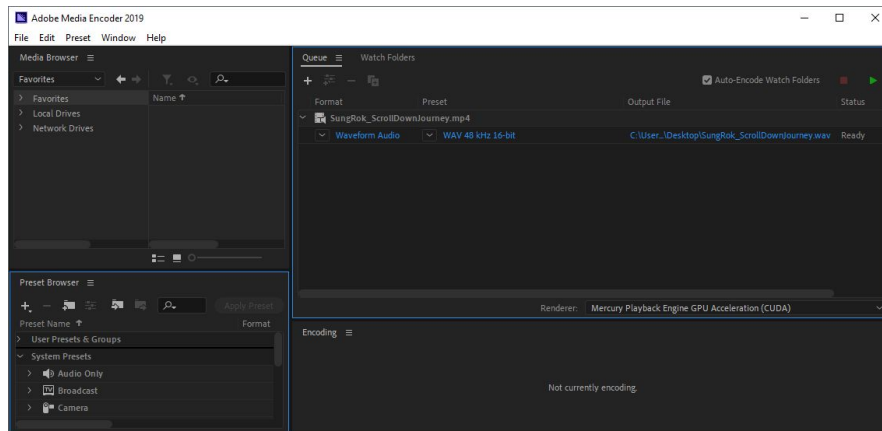
The direct upscaling of video files can be tricky as their original encoding yields uncertainty for the upsizing process. Therefore, upsizing directly from an encoded video file is not advised. Encoding/decoding video files and upsizing them should be different processes. For this reason, we render source videos into image sequences beforehand, as well as audio as a separate sound file. If converting into an image sequence is not feasible: analyse your original video file, especially for **encoding method & framerate** and be comfortable to test and run your upsizing method with different settings again until you get satisfactory results. Since it is possible to extract and recombine the frames within Adobe CC After Effects, there is no need to install and to use FFmpeg. However, you can still use FFmpeg for extracting and encoding the output video as it was described in [General Process](#). It can be particularly helpful if you want to use a video codec for the result that is not supported by Adobe CC.

Similar to the single image workflow, we use applications from Adobe CC as a control group here. In particular we are using After Effects CC 2018 (v15) or newer and Adobe Media Encoder CC in the case of image sequences or video files.

- (1) Render image sequence from original video file with Adobe Media Encoder CC as, e.g., PNG files. PNG (Match Source) should interpret the video files frame rate correctly; otherwise, you can adjust the FPS there. Make sure to name the Output File, e.g., “extracted_.PNG”. Ending in an underscore is used for readability as Adobe Media Encoder CC will take care of numbering the PNG file names correctly according to the video frames.

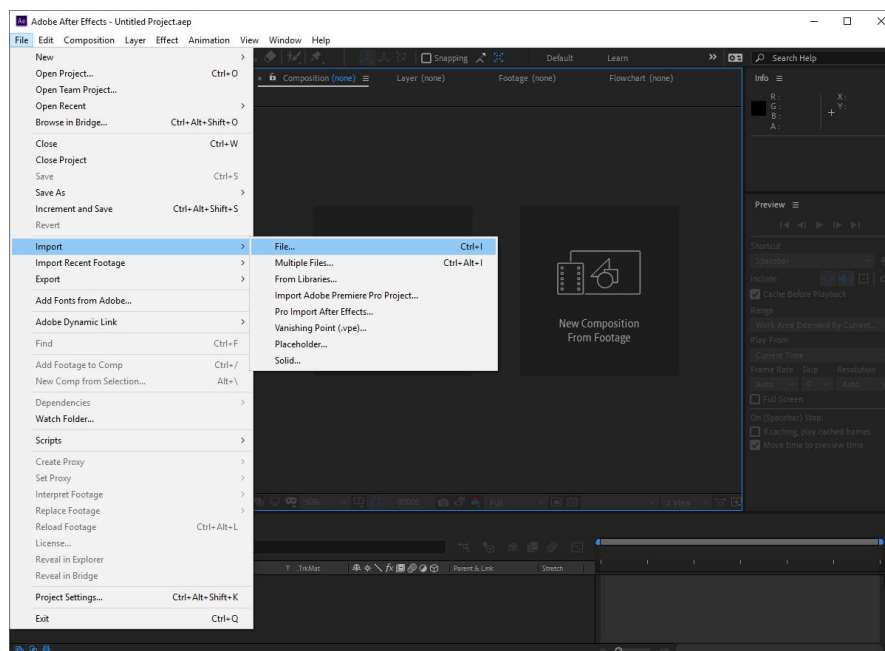


- (2) Render audio file from original video file with Adobe Media Encoder CC as Waveform Audio 48kHz 16 bit.



(2)

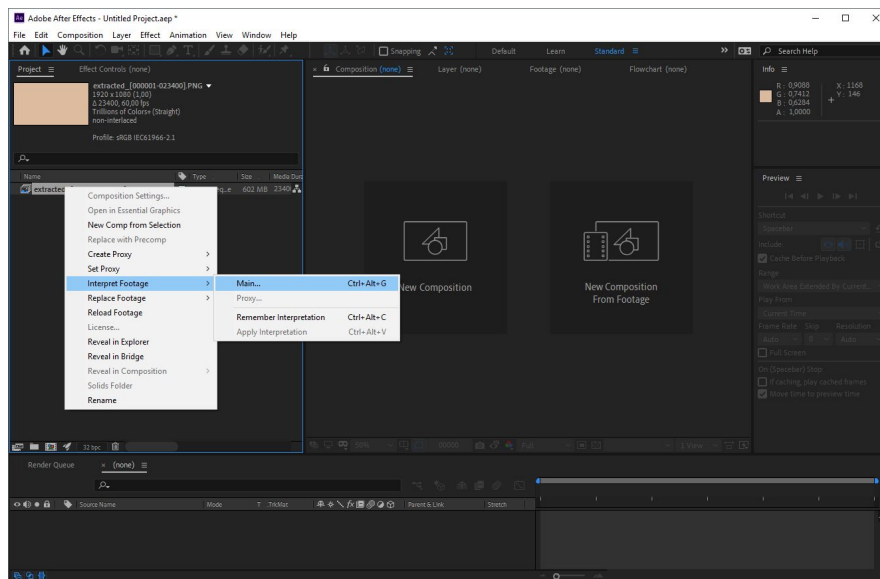
- (3) Import the image sequence in a new Adobe After Effects CC project via **File/Import/File...**



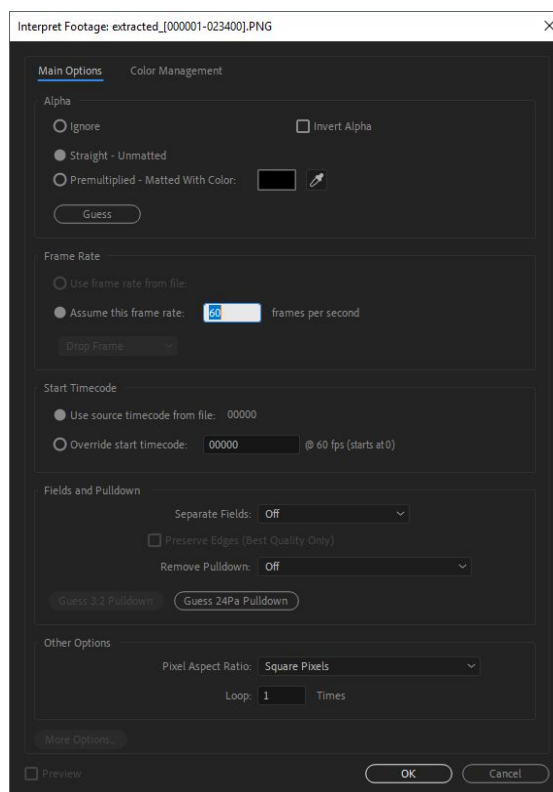
(3)

- (4) Navigate into the folder of the image sequence and highlight the first image. **Import As: Footage** should be set. Do not use Create Composition under Import Settings as it should be left unchecked. After clicking “Import” the image sequence should be accessible inside the Project Explorer of Adobe After Effects CC.

- (5) Adobe After Effects CC does internal guessing on frame rates when importing image sequences. Nevertheless, the frame rate of your imported image sequence can be changed via right-click on it: **Interpret Footage/Main...**

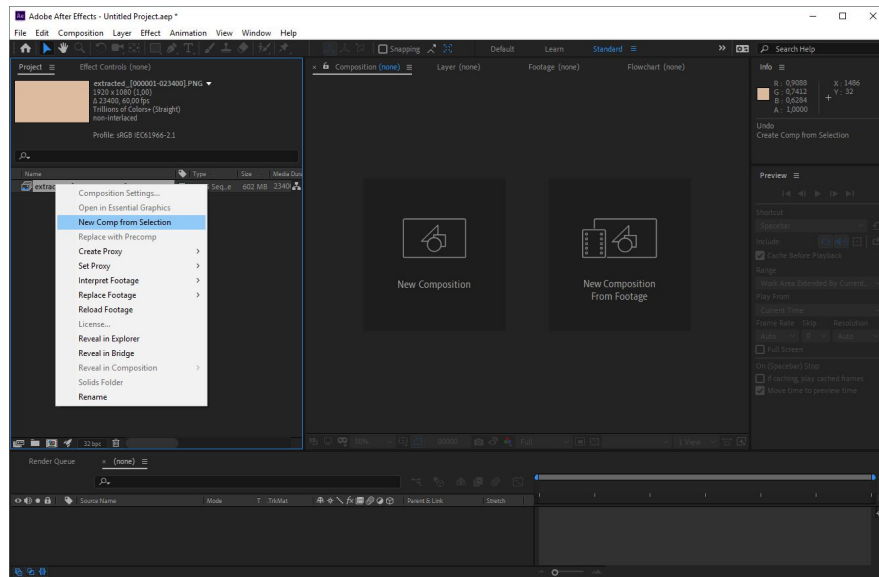


(5a)



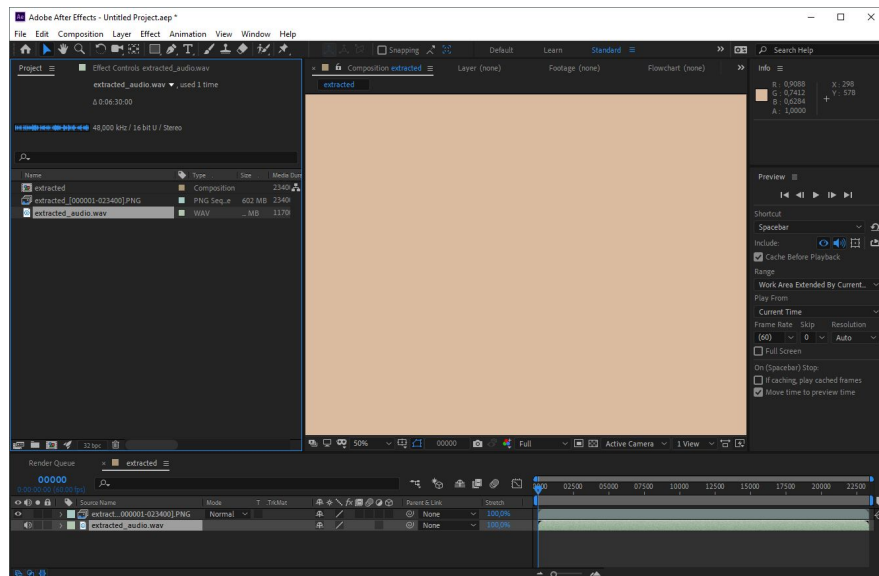
(5b)

(6) Create a new Composition from the image sequence footage by right click on it: **New Comp from Selection.**



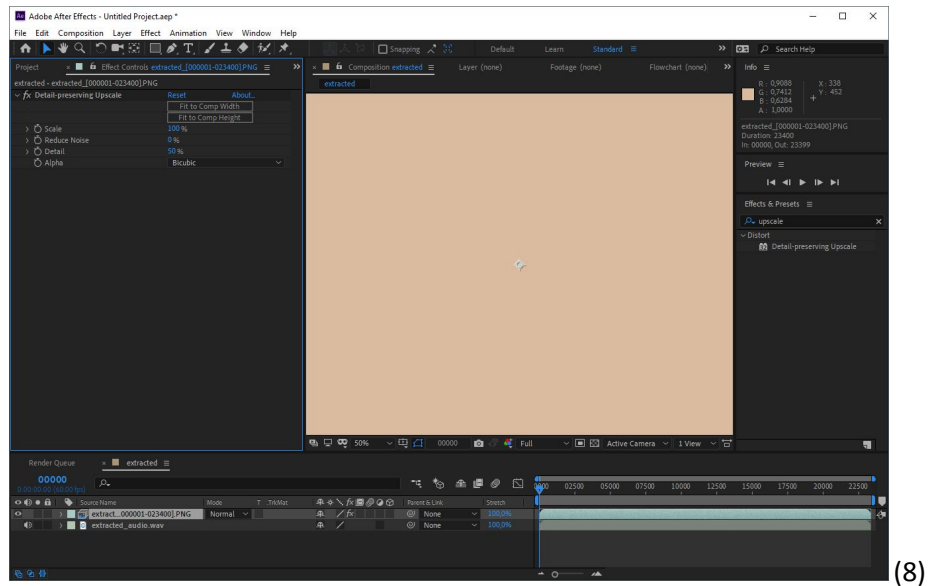
(6)

- (7) Import the extracted audio into Adobe After Effects CC via **File/Import/File...** drag the imported audio file from the Project Explorer into the composition timeline and drop it underneath the image sequence. The image sequence and audio file should have exactly the same length assuming the frame rate of the image sequence is set correctly.

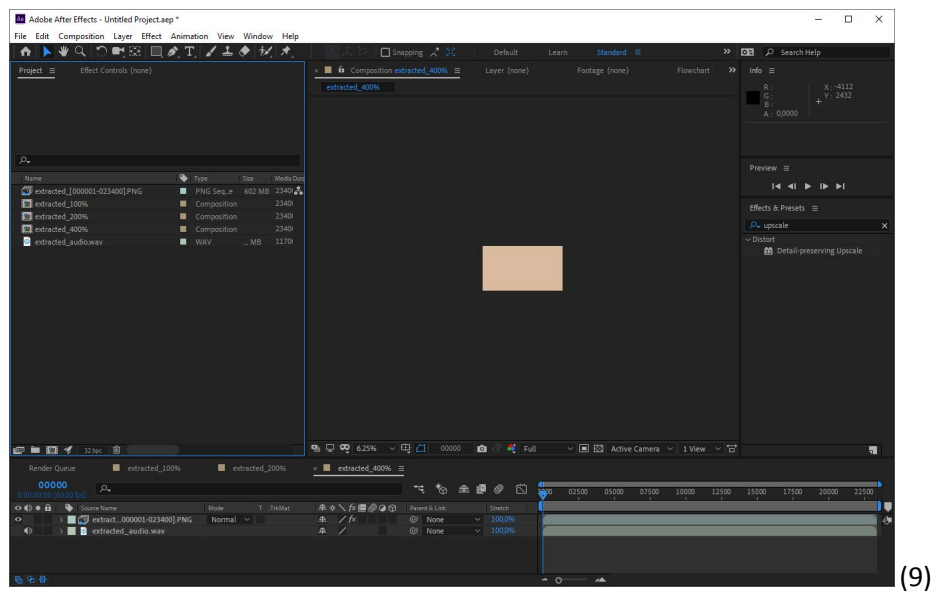


(7)

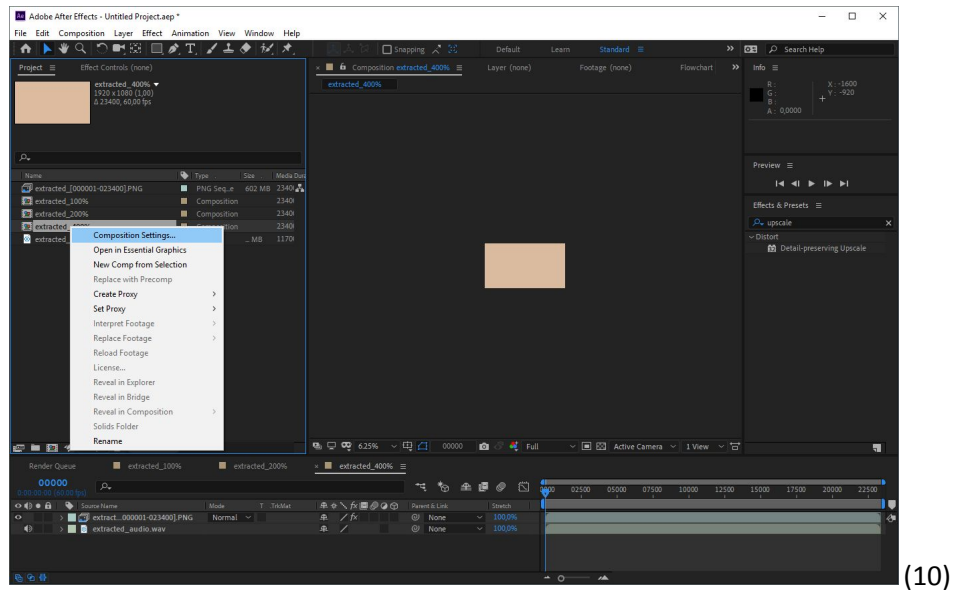
- (8) Select the image sequence in your composition timeline, then go to **Effect/Distort/Detail-preserving Upscale** or use the Effects & Presets Browser to simply for "upscale". After, drag and drop the Detail-preserving Upscale Effect onto the image sequence inside the composition. Do not change any of the values yet.



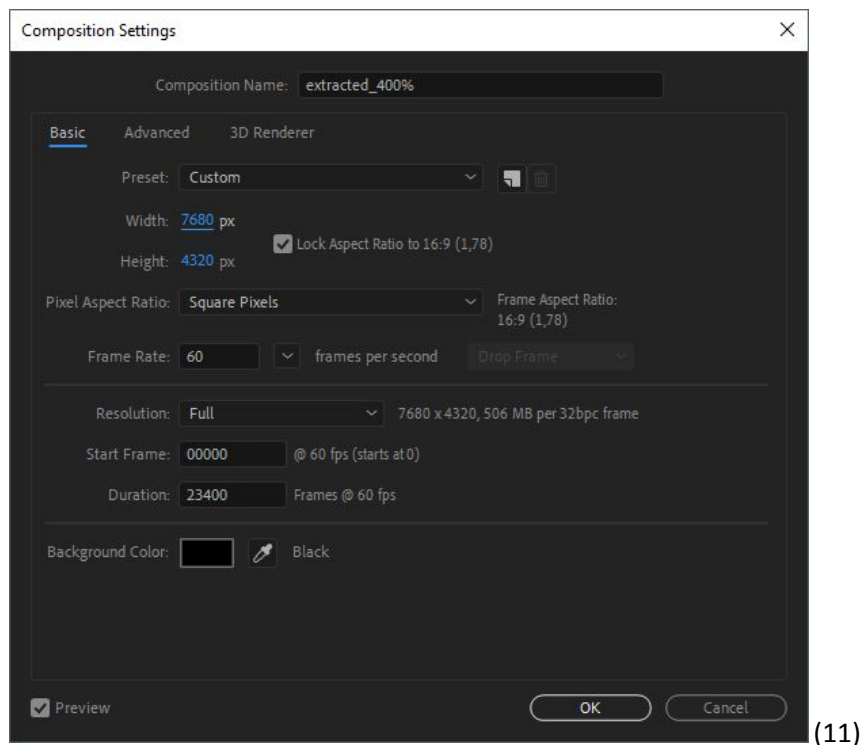
(9) Go back to the Project Explorer (click on Project Tab). **Copy / Paste** the composition inside the Project Explorer to get as many versions you need for upscaling (e.g., original, 4K version, 8K version, ...) and rename them accordingly.



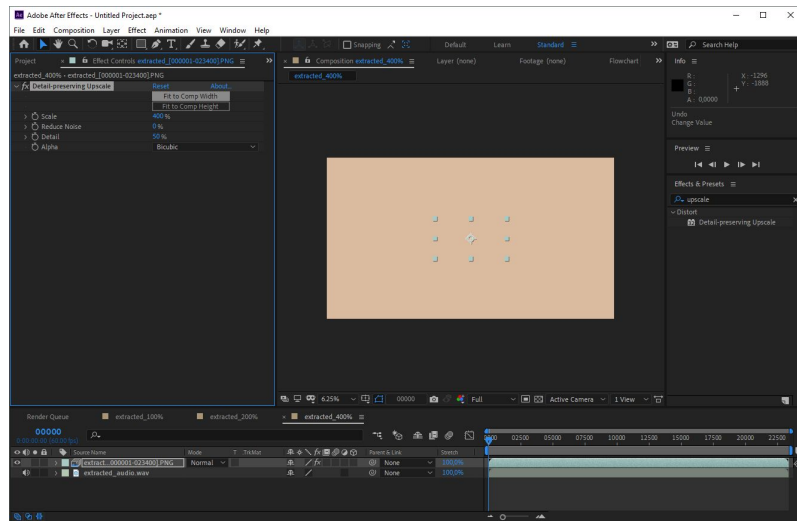
(10) Still in the Project Explorer, right-click on the composition of the e.g., 8K (400% upscale if your original file was Full HD) version and choose **Composition Settings...**



(11) Inside the Composition, Settings Menu changes the Width value of the composition to the desired upscaled value e.g, 7680 px for 8K (16:9 Aspect Ratio locked).

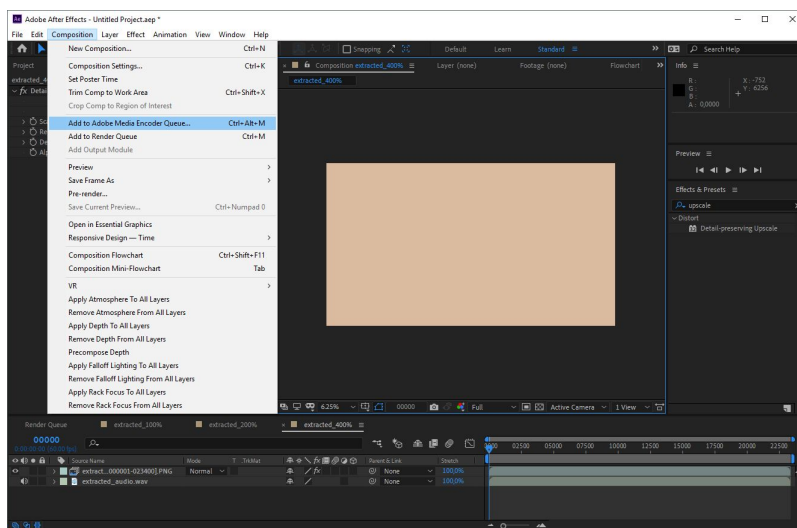


(12) Open the composition by double-clicking on it in the Project Explorer. Select the image sequence inside the composition timeline, then click on the Effects Controls Tab of the image sequence, which is next to the Project Tab (Project Explorer). Click on **Fit to Comp Width** (the Scale value should adjust automatically e.g., to 400% in our case of upscaling Full HD to 8K).



(12)

- (13) While the upscaled composition is still in focus on the timeline view, go to **Composition/Add to Adobe Media Encoder Queue...** Adobe Media Encoder CC will start up with the selected comp in its queue.



(13)

- (14) Use HEVC (H.265) for 8K here. Video and audio encoding settings (e.g., bitrate) can be adjusted via the Preset links. Be comfortable to test and run your render with different settings again until you get satisfactory results. To start your render queue, go to **File/Start Queue** or just hit enter on your keyboard.

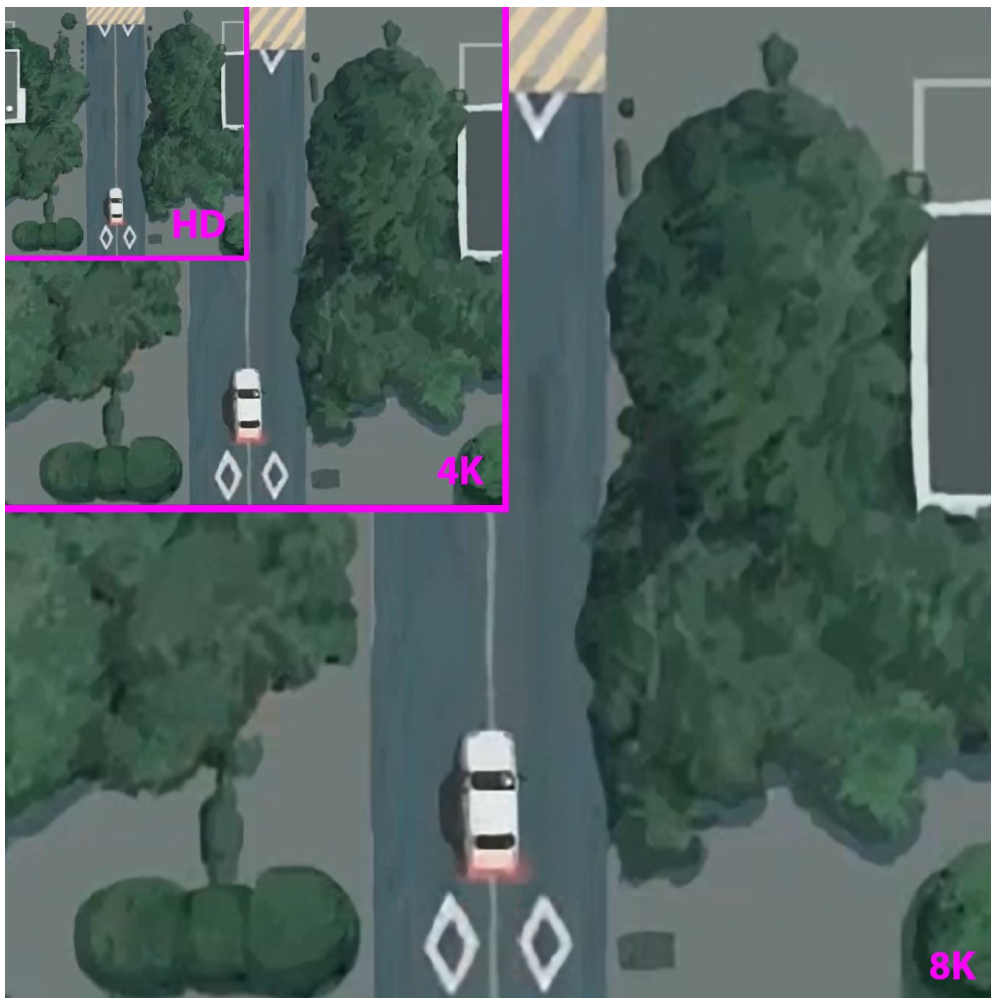
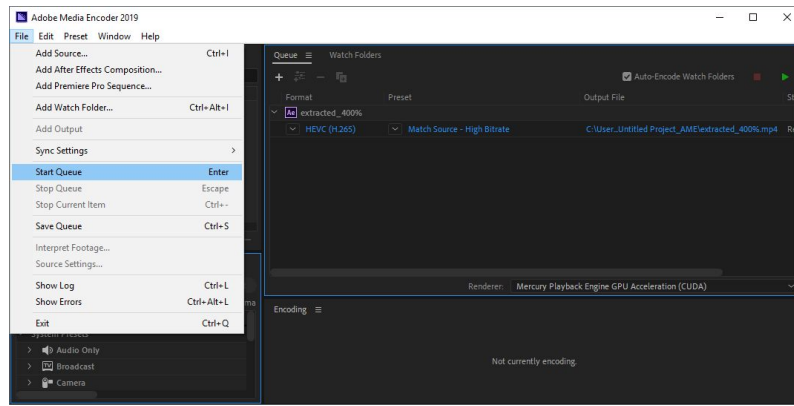


Figure 5. Comparing original HD frame area to 4K and 8K upscaling via Adobe After Effects

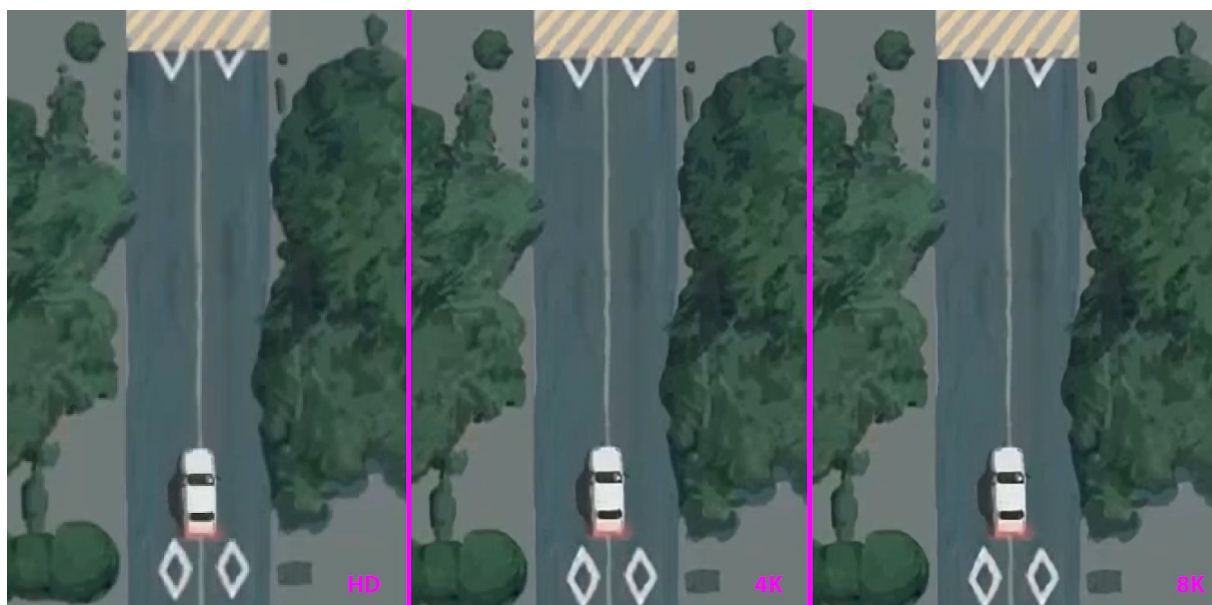


Figure 6. Comparing HD scaled to 400% (nearest neighbor) with 200% 4K upscale & 100% 8K upscale via Adobe After Effects (Detail-preserving upscale)

6 Upscaling with Topaz Video Enhance AI

To our knowledge, the only out-of-the-box AI-driven upscaling desktop solution available on the market is **Topaz Gigapixel AI**¹⁴ for single image upscaling and **Topaz Video Enhance AI**¹⁵ for video upscaling. We keep our focus on Topaz Video Enhance AI. Currently, it is a Windows system-only application and requires, at minimum, an NVIDIA GTX 1050 graphics card. However, if no viable GPU is present, the application will fallback to the system's CPU, which will mostly result in a tenfold increase of render times or will send your computer crashing.

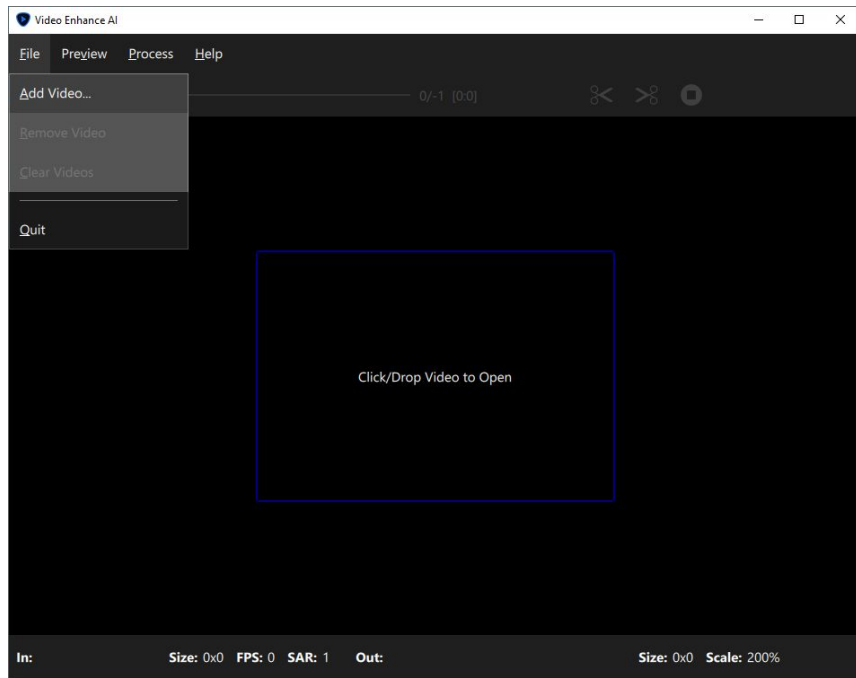
6.1 Topaz Video Enhance AI workflow

Although Topaz Video Enhance AI supports direct upscaling from an encoded video file, we again recommend upsizing from image sequences for better results. Hence the first step should always be to render original source videos into image sequences beforehand, as well as audio as a separate sound file. This can be done with FFmpeg or Adobe Media Encoder CC (see steps 1 & 2 of [Basic Video Upscale](#)) or using [FFmpeg](#).

Import the original image sequence into Topaz Video Enhance AI by **File/Add Video...** Make sure to switch to image files in the explorer dialog and navigate into your original image sequence folder. Highlight the first image of your sequence and hit open. Topaz Video Enhance AI will load your entire image sequence.

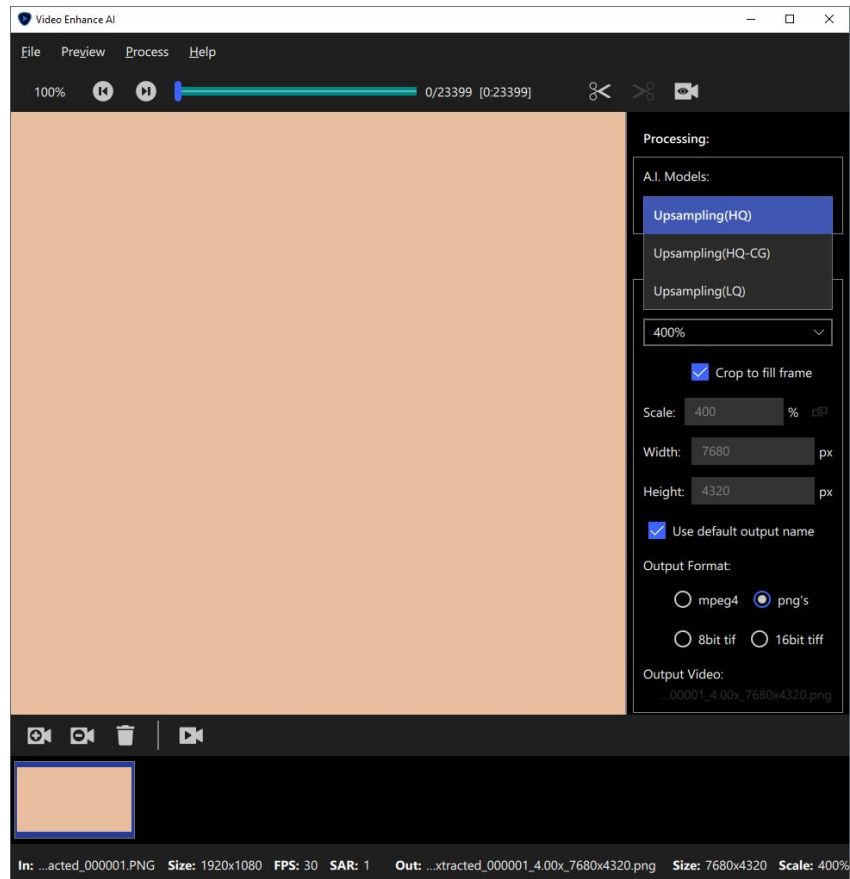
¹⁴ <https://topazlabs.com/gigapixel-ai/>

¹⁵ <https://topazlabs.com/video-enhance-ai/>



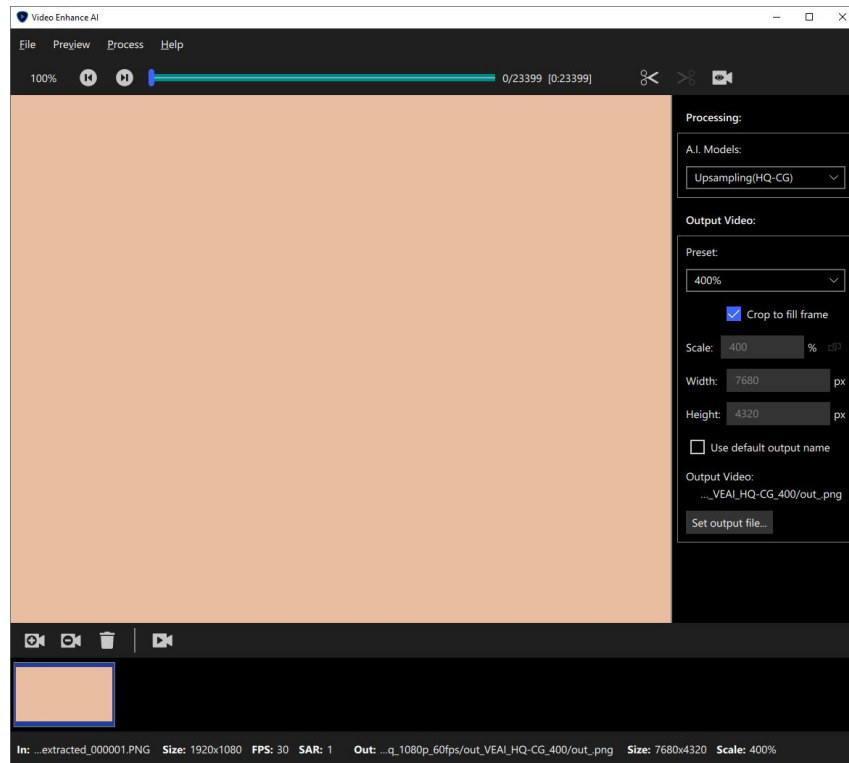
(1)

(1) The Processing menu on the right lets you choose between 3 AI Models for Upsampling according to the nature of your original source footage.

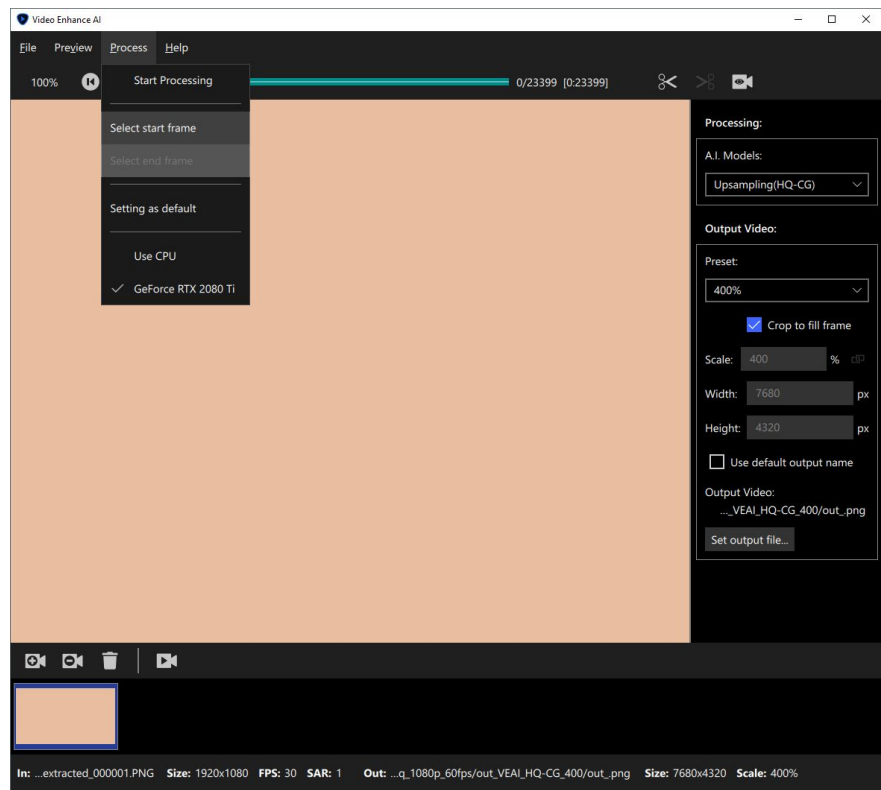




- (2) The amount of upscaling (e.g., via percentage) can be adjusted in the Output Video Menu. Uncheck “Use default output name” and use **Set output file...** to specify an output folder for your upscaled image sequence as well as their filetype (e.g., PNG, TIFF, ...)



- (3) Use **Process/Start Processing** to begin the rendering of the upscaled image sequence. Due to the nature of your source video e.g., the number of frames, the upscaling method and your systems GPU/CPU, the rendering can be a very long process.



(4)

- (4) After the upscaling is complete, you can encode the new upscaled image sequence with the original audio file as a video file with Adobe After Effects: i) import; ii) put together in the same composition; and iii) added to Adobe Media Encoder CC Queue (similar to [Basic Video Upscale](#)) or using other tools like FFmpeg as it was mentioned in [General Process](#).

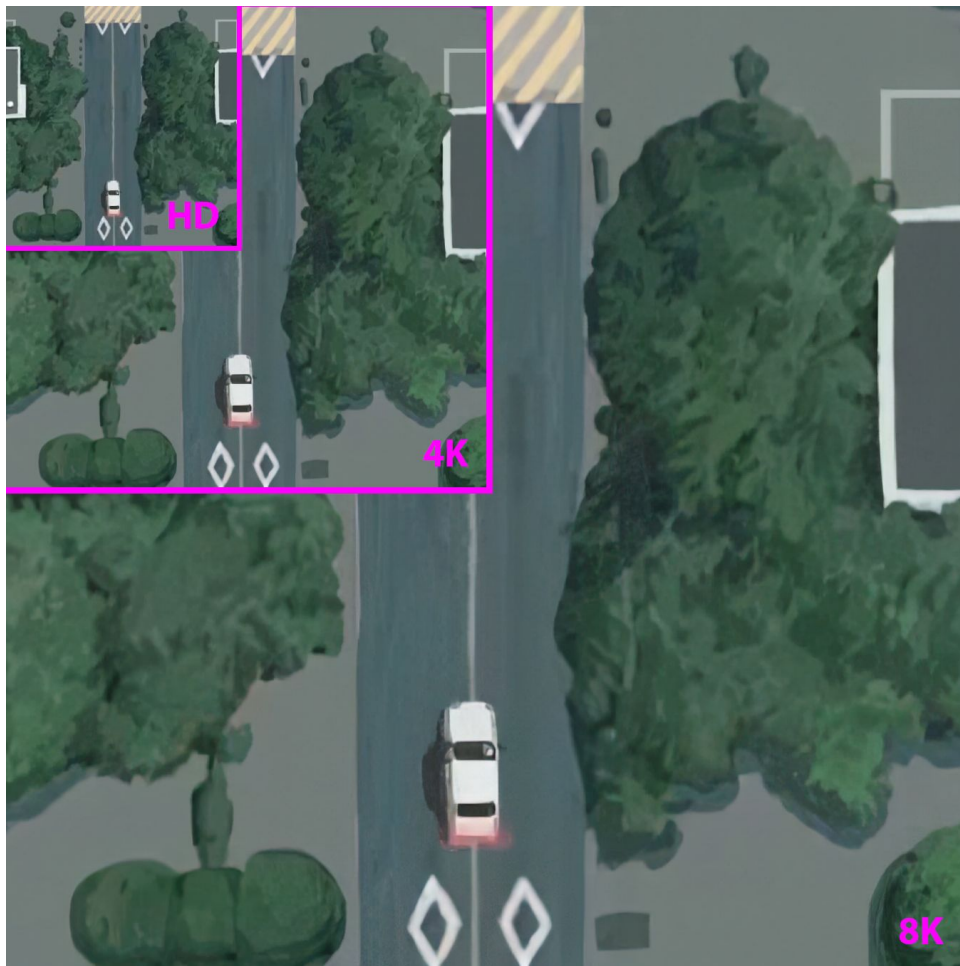


Figure 7. Comparing original HD frame area to 4K and 8K upscaling via Topaz Video Enhance AI

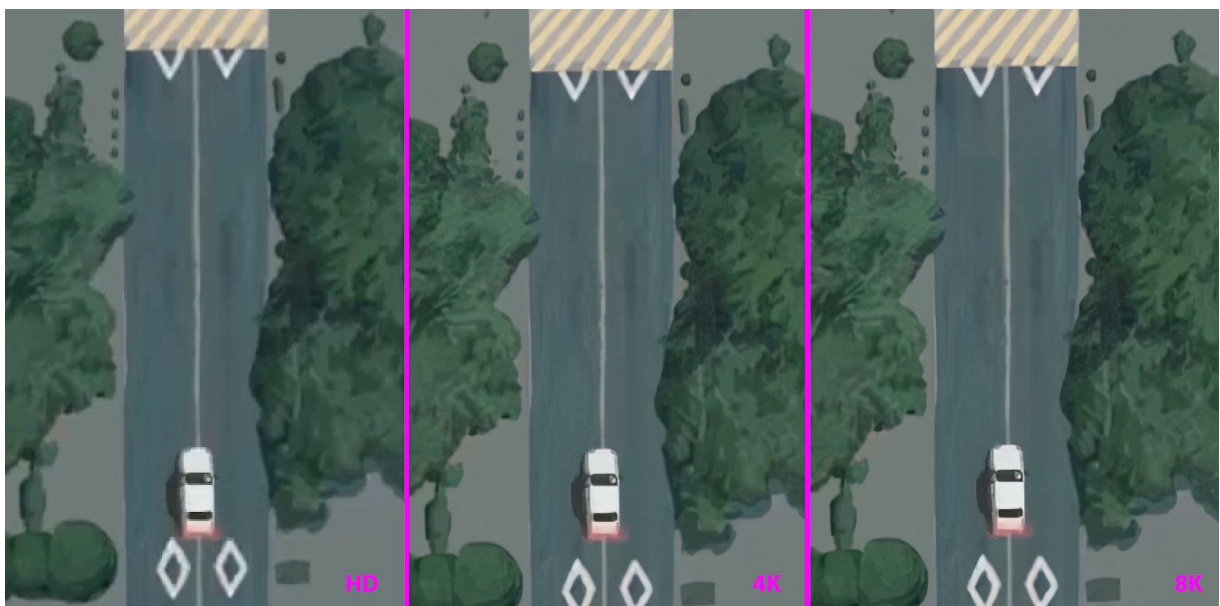


Figure 8. Comparing HD scaled to 400% (nearest neighbor) with 200% 4K upscale & 100% 8K upscale via Topaz Video Enhance A



7 Conclusion

Upscaling of video files with AI is a novel workflow coming to fruition recently. The different approaches discussed in this document show that results may vary.

First, we introduced [Waifu2x-Caffe](#), a standalone open-source application that comes with various pre-trained models and can be used primarily for animation-like content. The installation of Waifu2x-Caffe is easy as no additional software is required. It is equipped with a self-explanatory and easy-to-use interface. Furthermore, it is possible to train and to share user specific, pre-trained models by using the original implementation of Waifu2x. All in all it is a nice experimental software to try out.

Working with Adobe Creative Cloud Applications (see [Image Upscale](#) & [Video Upscale](#)) gives a familiar environment for most users. As an industry standard in media creation it is adopted and used in many workflows. Nevertheless their Preserve Details 2.0 Upscaling method lacks some of the upscaling fidelity in both picture and video formats, therefore feels a bit behind on recent AI upscaling developments. In spite of that Adobe CC's straight forward integration into video production pipelines makes it still a viable and stable toolset.

Using Topaz Video Enhance AI shows very promising upscaling results. Nonetheless it is a newly released stand-alone application with some limitations, therefore it is critical to use a certain workflow to maximise it's final output quality (see [Video Enhance AI workflow](#)). Upscaling with Topaz Video Enhance AI leads to the best results of all applications tested, despite working with a tool outside of professional workflow pipelines.