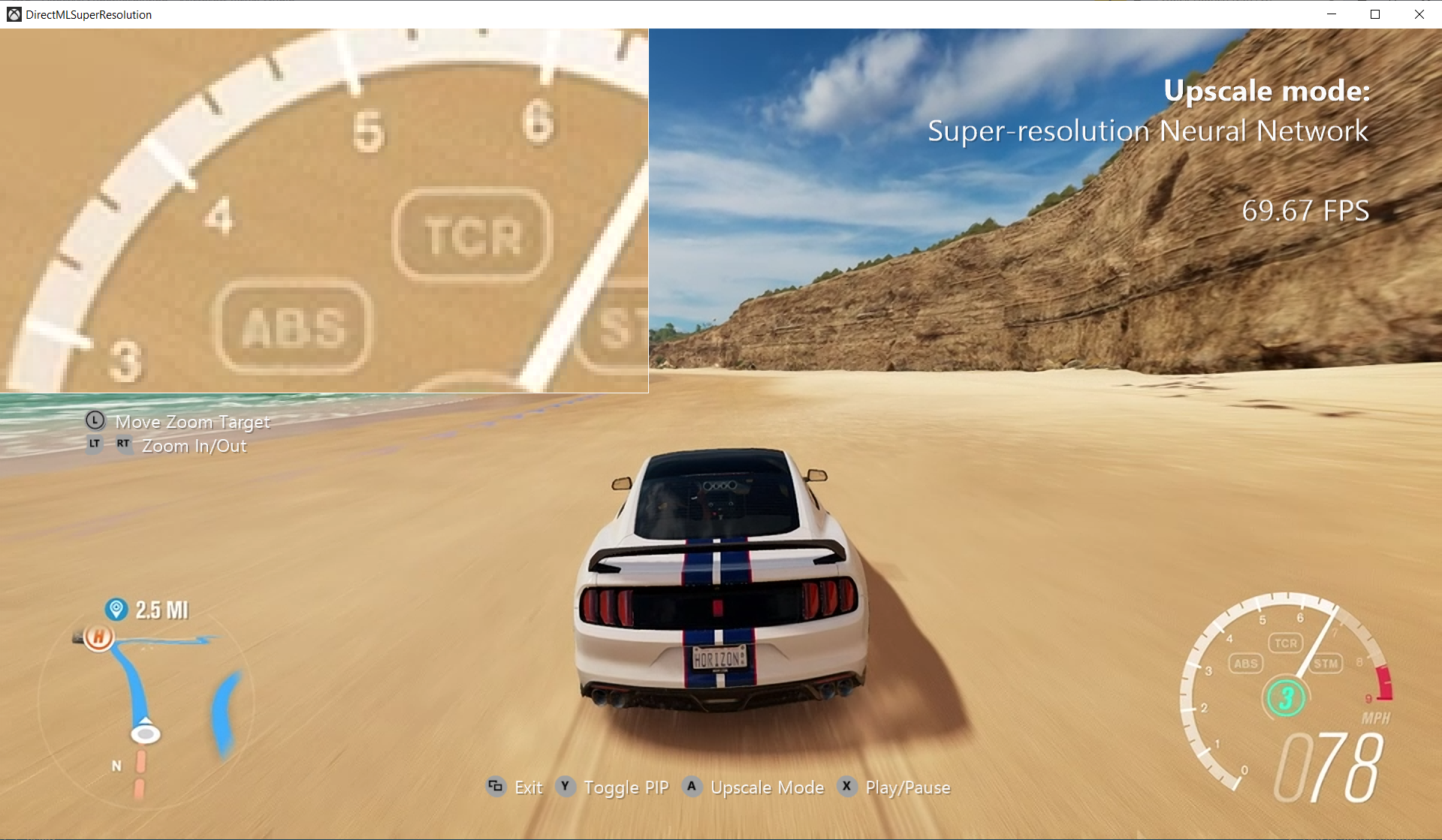


DirectML Super-Resolution Sample

*\* This sample is compatible with the Windows 10 2019 H1 Update. The earliest compatible version is Windows 10 Insider Preview, version 1903 (10.0; Build 18309).*

# Description



This sample demonstrates the DirectML API by implementing a super-resolution machine learning (ML) model on the GPU. This includes converting between image and tensor formats, initializing and executing ML operators, and interleaving graphics and ML work. The ML model performs a smart upscale of an image to double its original resolution. For example, it can scale a 540p image to 1080p.

**IMPORTANT:** While the code in this sample is complete, it currently includes weights trained on a limited data set for the given model. This means that the visual results of this super-resolution demo do not generalize as well as they could. The weights will be improved in a future update to the sample.

# Recommended hardware

This sample requires a GPU with support for FP16 data types. For best performance, a GPU with dedicated ML support is recommended.

# Using the sample

|  |  |  |
| --- | --- | --- |
| Action | Gamepad | Keyboard |
| Toggle between ML super-resolution and bilinear upscale | A button | Space |
| Play/pause video | X button | Enter |
| Show/hide zoomed picture-in-picture view | Y button | Z |
| Move zoom target | Left thumbstick | Arrow keys |
| Change zoom level | LT and RT buttons | W and S |
| Exit | View Button | Esc |

# Implementation notes

The primary purpose of this sample is to demonstrate usage of the DirectML API for real-time graphics processing. The particular super-resolution model is rather heavyweight, so a GPU and driver with dedicated ML support is recommended to run at higher framerates.

* **CreateDeviceDependentResources**: This calls into several methods to create and initialize the resources required to run the ML model and render the results. In particular, **CreateDirectMLResources** creates all the DML operators that comprise the model, as well as the D3D resources used to store intermediate results, and shaders to convert between texture and tensor formats. **InitializeDirectMLResources** binds the D3D resources to the operators and runs operator initialization.
* **Render**: If DirectML processing is enabled, this runs the operators using their bound resources to perform the super-resolution method on the current video frame. When DML is disabled, it uses a simple bilinear upscale. The result is rendered to the screen.

Several properties of the sample can be configured at compile time by changing constants at the beginning of DirectMLSuperResolution.cpp:

* **USE\_VIDEO**: Normally, the sample runs on a video. When set to 0, it will load and display a static image instead.
* **FORCE\_NCHW**: Setting this to 1 forces the sample to use the standard NCHW (batch size, channels, height, width) tensor format in all cases. This will run more slowly on recent NVIDIA hardware (Volta and later), but faster on older NVIDIA hardware. See Known Issues below for more details.
* **DML\_MANAGED\_WEIGHTS**: For tensor data that never changes, like weights, DirectML can load it during operator initialization and store it in a persistent resource. This allows optimizations on some hardware, since DML can reformat the data based on the current device. Note that not all hardware benefits from this, but there should never be a penalty to enabling it. Setting this to 0 disables this path.
* **c\_videoPath** and **c\_imagePath**: Paths to the video or image (depending on USE\_VIDEO) used as input for the super-resolution model. If you want to use your own content, add it to the project and change the appropriate path.

# Known issues

Different GPUs may require different tensor layouts for best performance. DirectML does not currently provide an API for determining the preferred layout., so the sample currently makes a coarse-grained decision based on the GPU vendor:

* NVIDIA: Uses NHWC (batch size, height, width, channels)
* All others: Uses NCHW (batch size, channels, height, width)

Notably, this is incorrect for older NVIDIA hardware (pre-Volta). For best performance in this case, set FORCE\_NCHW to 1 as described above.

# Legal notice

This sample includes content from Forza Horizon 3 for demonstration purposes. Video and images from Forza Horizon 3 are not for commercial use.

# Update history

Initial release April 2019.