ONNX Pre-processing WG

Monthly meeting - June 14, 2023

Agenda

- Status
- Image decoder operator proposal
- Open floor

Status

PRs:

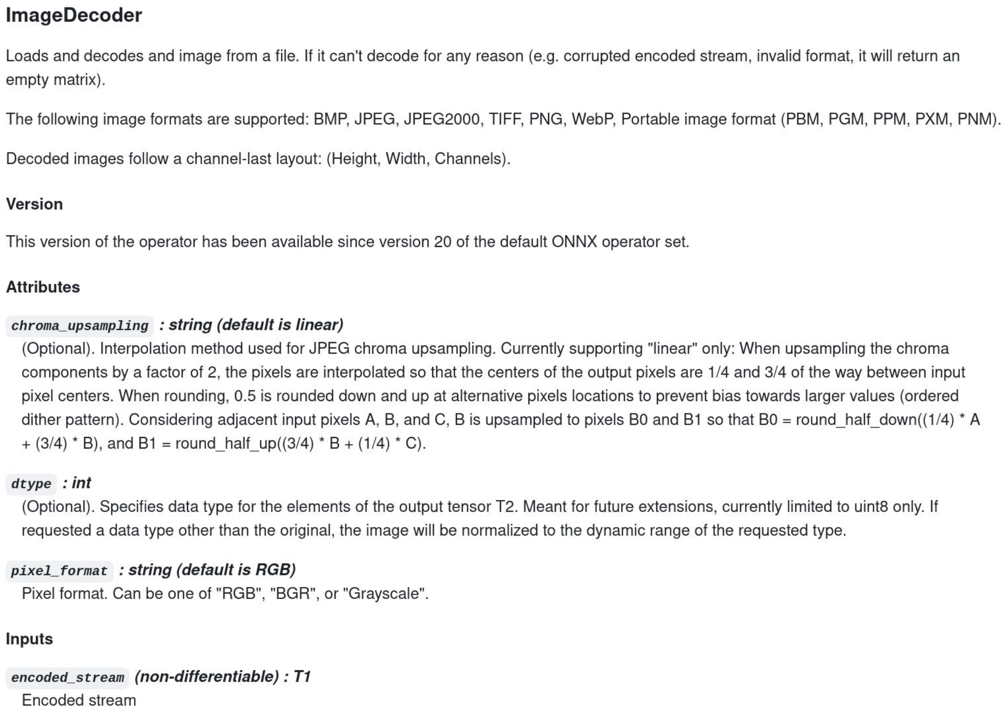
[IN PROGRESS] Affine grid generator operator - https://github.com/onnx/onnx/pull/5225

[IN PROGRESS] Image decoder operator - https://github.com/onnx/onnx/pull/5294

[MERGED] Using ONNX parser in SequenceMap tutorial (waiting for next ONNX runtime release)
https://github.com/onnx/tutorials/pull/277

Image decoder operator proposal

https://github.com/onnx/onnx/pull/5294



ImageDecoder

Loads and decodes and image from a file. If it can't decode for any reason (e.g. corrupted encoded stream, invalid format, it will return an empty matrix).

The following image formats are supported: BMP, JPEG, JPEG2000, TIFF, PNG, WebP, Portable image format (PBM, PGM, PPM, PXM, PNM).

Decoded images follow a channel-last layout: (Height, Width, Channels).

Version

This version of the operator has been available since version 20 of the default ONNX operator set.

Attributes

`chroma_upsampling` *: string (default is linear)*
(Optional). Interpolation method used for JPEG chroma upsampling. Currently supporting "linear" only: When upsampling the chroma components by a factor of 2, the pixels are interpolated so that the centers of the output pixels are 1/4 and 3/4 of the way between input pixel centers. When rounding, 0.5 is rounded down and up at alternative pixels locations to prevent bias towards larger values (ordered dither pattern). Considering adjacent input pixels A, B, and C, B is upsampled to pixels B0 and B1 so that B0 = round_half_down((1/4) * A + (3/4) * B), and B1 = round_half_up((3/4) * B + (1/4) * C).

`dtype` *: int*
(Optional). Specifies data type for the elements of the output tensor T2. Meant for future extensions, currently limited to uint8 only. If requested a data type other than the original, the image will be normalized to the dynamic range of the requested type.

`pixel_format` *: string (default is RGB)*
Pixel format. Can be one of "RGB", "BGR", or "Grayscale".

Inputs

`encoded_stream` *(non-differentiable) : T1*
Encoded stream

Open questions:

- Test images?
- Dependency with opencv?
  - reference implementation
  - tests
- Currently using generated images, encoded with opencv2

Open floor