ONNX Pre-processing WG

Monthly meeting - May 10, 2023

Agenda

- Status
- Image decoder operator proposal
- Open floor

Status

PRs:

[MERGED] Extend python API documentation
https://github.com/onnx/onnx/pull/5156

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

Image reader/decoder operator proposal

ImageRead

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)

Inputs filepath (heterogeneous) - uint8: 1-D tensor (null terminated string representing the file path)

Outputs image (heterogeneous) - T2: 3-D tensor (T2: uint8, uint16, float)

Attributes:

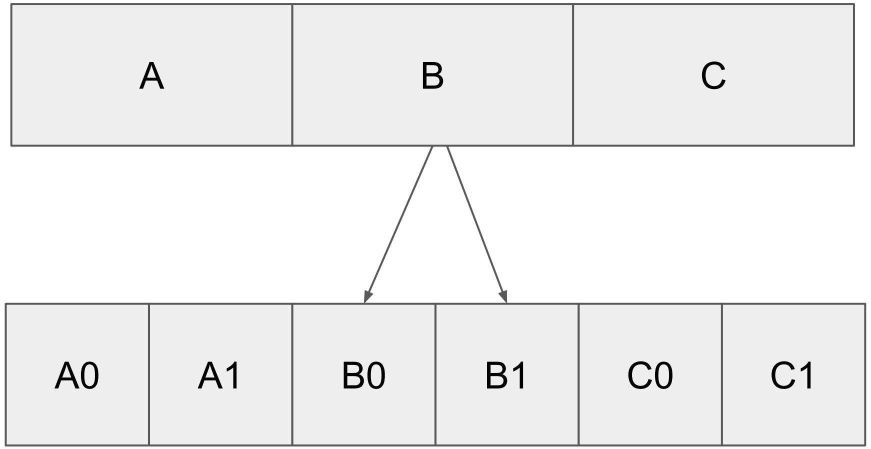
- format: INT (default PixelFormat.COLOR_RGB)
  Pixel format. Strictly must be one of the types from PixelFormat.
  Supported values are: COLOR_RGB, COLOR_BGR, GRAYSCALE, COLOR_UNCHANGED
- dtype - INT (default DataType.UINT8)
  The data type of the pixel values. Strictly must be one of the types from DataType enum in TensorProto.
  Supported values are: UINT8, UINT16, FLOAT
- chroma_upsampling - STRING (default "linear")
  Interpolation method used for JPEG chroma upsampling.
  "linear" "nearest"
- ignore_exif_orientation: INT (0, 1)
  If set to 1, EXIF orientation information from the encoded stream is not applied

ImageDecode

(Same as ImageRead, only changing the meaning of the input)

Inputs encoded_stream (heterogeneous) - uint8: 1-D tensor

JPEG chroma upsampling - linear interpolation



The centers of the output pixels are 1/4 and 3/4 of the way between input pixel centers.

Rounding fractional values to integer, we do not want to always round 0.5 up to the next integer, to avoid bias towards larger values.
Instead, 0.5 will be rounded up or down at alternate pixel locations (ordered dither pattern)

Note: Used in libjpeg by default (therefore, also in OpenCV, Pillow)

B0 = round_half_down(¾ * B + ¼ * A)
B1 = round_half_up(¾ * B + ¼ * C)

Open floor