ImageProcess API

Contents

Chapter 1

File Documentation

1.1 image_proc.h File Reference

Full APIs which support in this demo.

Classes

- struct RGBAffineInfo

  Affine transformation info structure.
- struct batch_resize_param

  multiple batch resize info structure.
- struct multi_roi_resize_param

  multiple roi resize info structure.

Macros

- #define ROTATE_90_CLOCKWISE 0
- #define ROTATE_180_CLOCKWISE 1
- #define ROTATE_270_CLOCKWISE 2

Typedefs

- typedef struct RGBAffineInfo RGBAffineInfo_t

  Affine transformation info structure.

Functions

- void RGBResizeBilinear (uint8_t *in_buf, uint8_t *out_buf, int w_in, int h_in, int w_out, int h_out, cuda↩ Stream_t stream)

  Resize RGB image with bilinear interpolation.
- void RGBResizeNearest (uint8_t *in_buf, uint8_t *out_buf, int w_in, int h_in, int w_out, int h_out, cuda↩ Stream_t stream)

  Resize RGB image with nearest neighbor interpolation.
- void GrayResizeBilinear (uint8_t *in_buf, uint8_t *out_buf, int w_in, int h_in, int w_out, int h_out, cuda↩ Stream_t stream)

  Resize gray image with bilinear interpolation.
- void GrayResizeNearest (uint8_t *in_buf, uint8_t *out_buf, int in_w, int in_h, int out_w, int out_h, cuda↩ Stream_t stream)

  Resize gray image with nearest neighbor interpolation.
- void YUVNv12ResizeNearest (uint8_t *in_buf, uint8_t *out_buf, int in_w, int in_h, int out_w, int out_h, cuda↩ Stream_t stream)

  Resize YUV(nv12) image with nearest neighbor interpolation.
- void YUVNv21ResizeNearest (uint8_t *in_buf, uint8_t *out_buf, int in_w, int in_h, int out_w, int out_h, cuda↩ Stream_t stream)

  Resize YUV(nv21) image with nearest neighbor interpolation.
- void YUVI420ResizeNearest (uint8_t *in_buf, uint8_t *out_buf, int in_w, int in_h, int out_w, int out_h, cuda↩ Stream_t stream)

  Resize YUV(i420) image with nearest neighbor interpolation.
- void YUVNv12ResizeBilinear (uint8_t *in_buf, uint8_t *out_buf, int in_w, int in_h, int out_w, int out_h, cuda↩ Stream_t stream)

  Resize YUV(nv12) image with bilinear interpolation.
- void YUVNv21ResizeBilinear (uint8_t *in_buf, uint8_t *out_buf, int in_w, int in_h, int out_w, int out_h, cuda↩ Stream_t stream)

  Resize YUV(nv21) image with bilinear interpolation.
- void YUVI420ResizeBilinear (uint8_t *in_buf, uint8_t *out_buf, int in_w, int in_h, int out_w, int out_h, cuda↩ Stream_t stream)

  Resize YUV(i420) image with bilinear interpolation.
- void YUVYu12ToRGB (uint8_t *in_buf, uint8_t *out_buf, int in_w, int in_h, cudaStream_t stream)

  Convert YUV(yu12, also called i420) image to RGB image.
- void YUVNv12ToRGB (uint8_t *in_buf, uint8_t *out_buf, int in_w, int in_h, cudaStream_t stream)

  Convert YUV(nv12) image to RGB image.
- void YUVYu12ToRGBFloat (uint8_t *in_buf, float *out_buf, int in_w, int in_h, cudaStream_t stream)

  Convert YUV(yu12, also called i420) image to RGB image.
- void YUVNv12ToRGBFloat (uint8_t *in_buf, float *out_buf, int in_w, int in_h, cudaStream_t stream)

  Convert YUV(nv12) image to RGB image.
- void YUVYu12ToRGBPlane (uint8_t *in_buf, uint8_t *out_buf, int in_w, int in_h, cudaStream_t stream)

  Convert YUV(yu12, also called i420) image to RGB image.
- void YUVNv12ToRGBPlane (uint8_t *in_buf, uint8_t *out_buf, int in_w, int in_h, cudaStream_t stream)

  Convert YUV(nv12) image to RGB image.
- void RGBResizeWithROINearest (uint8_t *in_buf, uint8_t *out_buf, int w_in, int h_in, int w_out, int h_out, int roi_w_start, int roi_h_start, int roi_w, int roi_h, cudaStream_t stream)

  Resize RGB image ROI area with nearset interpolation.
- void RGBResizeWithROIBilinear (uint8_t *in_buf, uint8_t *out_buf, int w_in, int h_in, int w_out, int h_out, int roi_w_start, int roi_h_start, int roi_w, int roi_h, cudaStream_t stream)

  Resize RGB image ROI area with bilinear interpolation.
- void RGBResizePlaneBilinear (uint8_t *in_buf, uint8_t *out_buf, int w_in, int h_in, int w_out, int h_out, cuda↩ Stream_t stream)

  Resize RGB image with bilinear interpolation.

- void RGBResizePlaneNearest (uint8_t *in_buf, uint8_t *out_buf, int w_in, int h_in, int w_out, int h_out, cuda↩ Stream_t stream)

  Resize RGB image with nearest interpolation.
- void RGBResizePlanePadBilinear (uint8_t *in_buf, uint8_t *out_buf, int w_in, int h_in, int w_out, int h_out, int w_box, int h_box, int w_b, int h_b, cudaStream_t stream)

  Resize RGB image with bilinear interpolation.
- void RGBResizePlanePadNearest (uint8_t *in_buf, uint8_t *out_buf, int w_in, int h_in, int w_out, int h_out, int w_box, int h_box, int w_b, int h_b, cudaStream_t stream)

  Resize RGB image with nearest interpolation.
- void RGBCrop (uint8_t *in_buf, uint8_t *out_buf, int start_w, int start_h, int w_in, int h_in, int w_out, int h_out, cudaStream_t stream)

  Crop RGB image.
- void YU12Crop (uint8_t *in_buf, uint8_t *out_buf, int start_w, int start_h, int w_in, int h_in, int w_out, int h_out, cudaStream_t stream)

  Crop YU12 image.
- void NV12Crop (uint8_t *in_buf, uint8_t *out_buf, int start_w, int start_h, int w_in, int h_in, int w_out, int h_out, cudaStream_t stream)

  Crop NV12 image.
- void RGBNormalization (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int in_c, float mean, float standard, float scale, cudaStream_t stream)

  RGB image normalization.
- void RGBNormalization_3Channels (uint8_t *in_buf, float *out_buf, int in_w, int in_h, float mean1, float mean2, float mean3, float standard1, float standard2, float standard3, float scale, bool input_plane, bool output_plane, bool channel_rev, cudaStream_t stream)

  RGB image normalization.
- void NV12ToRGBBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  NV12 image convert to RGB and resize with padding and normalization.
- void YU12ToRGBBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  YU12(i420) image convert to RGB and resize with padding and normalization.
- void NV12ToRGBNearestResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  NV12 image convert to RGB and resize with padding and normalization.
- void YU12ToRGBNearestResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  YU12(i420) image convert to RGB and resize with padding and normalization.
- void NV12ToBGRBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  NV12 image convert to BGR and resize with padding and normalization.
- void YU12ToBGRBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  YU12(i420) image convert to BGR and resize with padding and normalization.
- void NV12ToBGRNearestResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  NV12 image convert to BGR and resize with padding and normalization.

- void YU12ToBGRNearestResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  YU12(i420) image convert to BGR and resize with padding and normalization.
- void RGBROINearestResizeNormPlane (uint8_t *in_buf, float *out_buf, int w_in, int h_in, int w_out, int h_out, int roi_w_start, int roi_h_start, int roi_w, int roi_h, float scale, float mean1, float mean2, float mean3, float std1, float std2, float std3, bool channel_rev, cudaStream_t stream)

  Resize RGB/BGR image ROI area with nearset interpolation and normalization.
- void RGBROIBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int w_in, int h_in, int w_out, int h_out, int roi_w_start, int roi_h_start, int roi_w, int roi_h, float scale, float mean1, float mean2, float mean3, float std1, float std2, float std3, bool channel_rev, cudaStream_t stream)

  Resize RGB/BGR image ROI area with bilinear interpolation and normalization.
- void RoiNV12ToRGBBilinearResizePlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float pad1, float pad2, float pad3, cudaStream_t stream)

  NV12 image roi area convert to RGB and resize with padding.
- void RoiYU12ToRGBBilinearResizePlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float pad1, float pad2, float pad3, cudaStream_t stream)

  YU12(i420) image roi area convert to RGB and resize with padding and normalization.
- void RoiNV12ToBGRBilinearResizePlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float pad1, float pad2, float pad3, cudaStream_t stream)

  NV12 image roi area convert to BGR and resize with padding.
- void RoiYU12ToBGRBilinearResizePlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float pad1, float pad2, float pad3, cudaStream_t stream)

  YU12(i420) image roi area convert to BGR and resize with padding and normalization.
- void RoiNV12ToRGBBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  NV12 image roi area convert to RGB and resize with padding and normalization.
- void RoiYU12ToRGBBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  YU12(i420) image roi area convert to RGB and resize with padding and normalization.
- void FullRangeNV12ToRGBBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  NV12 full range(0∼255) image convert to RGB and resize with padding and normalization.
- void FullRangeYU12ToRGBBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  YU12(i420) full range(0∼255) image convert to RGB and resize with padding and normalization.
- void GrayAffine (uint8_t *input, uint8_t *output, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int out_w, int out_h, float m[6], cudaStream_t stream)

  Affine transformation gray image with bilinear interpolation.
- void RGBAffine (uint8_t *input, uint8_t *output, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int out_w, int out_h, float m[6], cudaStream_t stream)

  Affine transformation rgb image with bilinear interpolation.
- void YU122RGBAffine (uint8_t *input, uint8_t *output, int in_w, int in_h, int out_w, int out_h, float m[6], cudaStream_t stream)

YU12(i420) image convert to RGB and affine transformation.

- void RoiYUV444PToRGBBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int roi↩ _w_start, int roi_h_start, int roi_w, int roi_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  YUV444P image roi area convert to RGB and bilinear resize with padding and normalization.
- void RoiYUV400PToRGBBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int roi↩ _w_start, int roi_h_start, int roi_w, int roi_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean, float std, float scale, float pad, cudaStream_t stream)

  YUV400P image roi area convert to RGB and bilinear resize with padding and normalization.
- void RoiYUV422PToRGBBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int roi↩ _w_start, int roi_h_start, int roi_w, int roi_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  YUV422P image roi area convert to RGB and bilinear resize with padding and normalization.
- void RoiYUV422ToRGBBilinearResizeNormPlane (uint8_t *in_buf, float *out_buf, int in_w, int in_h, int roi↩ _w_start, int roi_h_start, int roi_w, int roi_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  YUV422 image roi area convert to RGB and bilinear resize with padding and normalization.
- void RoiNv122RGBAffineNorm (uint8_t *input, float *output, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int out_w, int out_h, float m[6], float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  NV12 image roi area convert to RGB and affine transformation with padding and normalization.
- void RoiYU122RGBAffineNorm (uint8_t *input, float *output, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int out_w, int out_h, float m[6], float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  YU12 image roi area convert to RGB and affine transformation with padding and normalization.
- void RoiYU122BGRAffineNorm (uint8_t *input, float *output, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int out_w, int out_h, float m[6], float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, float pad1, float pad2, float pad3, cudaStream_t stream)

  YU12 image roi area convert to BGR and affine transformation with padding and normalization.
- void RoiNV12ToRGBBilinearResizeQuantizePlane (uint8_t *in_buf, uint8_t *out_buf, uchar4 *ws, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float pad1, float pad2, float pad3, float zero_point, float scales_input, cudaStream_t stream)

  NV12 image roi area convert to RGB, bilinear resize, padding and quantize output to uint8.
- void RoiYU12ToRGBBilinearResizeQuantizePlane (uint8_t *in_buf, uint8_t *out_buf, uchar4 *ws, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int img_w, int img_h, int out_w, int out_h, int pad_w, int pad_h, float pad1, float pad2, float pad3, float zero_point, float scales_input, cudaStream_t stream)

  YU12 image roi area convert to RGB, bilinear resize, padding and quantize output to uint8.
- void RoiYU12ToRGBBilinearResizeNormPlaneV2 (uint8_t *in_buf, float *out_buf, uchar4 *ws, int in_w, int in_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, int out_w, int out_h, int input_pad_w, int input_pad_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, uint8_t pad1, uint8_t pad2, uint8_t pad3, cudaStream_t stream)

  YU12(i420) image roi area convert to RGB and resize(compatible with transforms.Resize of torchvison) with padding and normalization.
- void RGB2YU12 (uint8_t *in_buf, uint8_t *out_buf, int w, int h, cudaStream_t stream)

  RGB image convert to YUV420p(I420).
- template<typename DataType , typename IndxType , bool IsAscending = false>
  int32_t RadixSortFunc (DataType *data_dev, IndxType *indxs_dev, int32_t total_num, DataType *data_out↩ _dev=nullptr, IndxType *indxs_out_dev=nullptr, void *workspace_dev=nullptr, cudaStream_t *stream=nullptr)

  Sort data with radix sort algorithm.
- void non_max_suppression (float *predict, int num_batch, int num_bboxes, int num_classes, float confidence_threshold, float nms_threshold, float *pout, int max_objects, cudaStream_t stream)

  Non max suppression inner class.

- void YUV444pToYUV420p (uint8_t *sy, uint8_t *su, uint8_t *sv, uint8_t *dy, uint8_t *du, uint8_t *dv, int w, int h, int align_w, cudaStream_t stream)

  yuv444p transform to yuv420p.
- void YUV440pToYUV420p (uint8_t *sy, uint8_t *su, uint8_t *sv, uint8_t *dy, uint8_t *du, uint8_t *dv, int w, int h, int align_w, cudaStream_t stream)

  yuv440p transform to yuv420p.
- void RGBROINearestResizeNormPadPlane (uint8_t *in_buf, float *out_buf, int w_in, int h_in, int w_out, int h_out, int img_w, int img_h, int pad_w, int pad_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, float scale, float mean1, float mean2, float mean3, float std1, float std2, float std3, float pad1, float pad2, float pad3, bool channel_rev, cudaStream_t stream)

  Resize RGB/BGR image ROI area with nearset interpolation and normalization.
- void RGBROIBilinearResizeNormPadPlane (uint8_t *in_buf, float *out_buf, int w_in, int h_in, int w_out, int h_out, int img_w, int img_h, int pad_w, int pad_h, int roi_w_start, int roi_h_start, int roi_w, int roi_h, float scale, float mean1, float mean2, float mean3, float std1, float std2, float std3, float pad1, float pad2, float pad3, bool channel_rev, cudaStream_t stream)

  Resize RGB/BGR image ROI area with bilinear interpolation and normalization.
- void RGBBilinearResizeCropNormPlaneV2 (uint8_t *in_buf, float *out_buf, uchar4 *ws, int in_w, int in_h, int resized_w, int resized_h, int crop_w_start, int crop_h_start, int crop_w, int crop_h, float mean1, float mean2, float mean3, float std1, float std2, float std3, float scale, bool fmt_cvt, cudaStream_t stream)

  RGB resize(compatible with transforms.Resize of torchvison) with crop and normalization.
- void RGBRotate (uint8_t *in, uint8_t *out, int in_w, int in_h, int rotate_mode, cudaStream_t stream)

  Rotate RGB image with specific rotate mode.
- void BatchRGBAffine (uint8_t *input, uint8_t *output, int batch, int out_batch_offset, int in_w, int in_h, RGBAffineInfo_t *param, cudaStream_t stream)

  Affine transformation batch of rgb image with bilinear interpolation.
- void BatchRGBBilinearResizeNormPadPlane (uint8_t *in_buf, float *out_buf, int batch, int input_batch_offset, int output_batch_offset, batch_resize_param *param, float scale, float mean1, float mean2, float mean3, float std1, float std2, float std3, float pad1, float pad2, float pad3, bool channel_rev, cudaStream_t stream)

  Multiple batch resize RGB/BGR image with bilinear interpolation and normalization.
- void MultiRoiRGBBilinearResizeNormPadPlane (uint8_t *in_buf, float *out_buf, int batch, int in_h, int in_w, int output_batch_offset, multi_roi_resize_param *param, float scale, float mean1, float mean2, float mean3, float std1, float std2, float std3, float pad1, float pad2, float pad3, bool channel_rev, cudaStream_t stream)

  Multiple roi resize RGB/BGR image ROI area with bilinear interpolation and normalization.

1.1.1 Detailed Description

Full APIs which support in this demo.

1.1.2 Macro Definition Documentation

1.1.2.1 #define ROTATE_180_CLOCKWISE 1

1.1.2.2 #define ROTATE_270_CLOCKWISE 2

1.1.2.3 #define ROTATE_90_CLOCKWISE 0

1.1.3 Typedef Documentation

1.1.3.1 typedef struct RGBAffineInfo RGBAffineInfo_t

Affine transformation info structure.

Parameters

roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
out_w	The width of output image.
out_h	The height of output image.
m[6]	Affine transformation matrix.

1.1.4 Function Documentation

1.1.4.1 void BatchRGBAffine (uint8_t * *input*, uint8_t * *output*, int *batch*, int *out_batch_offset*, int *in_w*, int *in_h*, RGBAffineInfo_t * *param*, cudaStream_t *stream*)

Affine transformation batch of rgb image with bilinear interpolation.

This API is suitable for a lot of small rgb images compare with RGBAffine API.

Parameters

input	The input buffer allocate in device memory. The size is (in_w * in_h * 3) * sizeof(uint8_t).
output	The output buffer allocate in device memory. The size is (out_w * out_h * 3 * batch) * sizeof(uint8_t).
batch	Number of input images.
out_batch_offset	The offset of per out image.
in_w	The width of input image.
in_h	The height of input image.
param	The information of per affine transformation.
stream	CU kernel run in the stream.

1.1.4.2 void BatchRGBBilinearResizeNormPadPlane (uint8_t * *in_buf*, float * *out_buf*, int *batch*, int *input_batch_offset*, int *output_batch_offset*, batch_resize_param * *param*, float *scale*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *pad1*, float *pad2*, float *pad3*, bool *channel_rev*, cudaStream_t *stream*)

Multiple batch resize RGB/BGR image with bilinear interpolation and normalization.

Output format is RRRGGGBBB / BBBGGGRRR. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in * h_in * 3) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out * h_out * 3) * sizeof(float).
batch	Number of input images.
input_batch_offset	The stride of input per batch.

Parameters

output_batch_offset	The stride of output per batch.
param	The information of per batch.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
channel_rev	True mean rgb convert to bgr or bgr convert to rgb. False mean input and output image is same format.
stream	CU kernel run in the stream.

1.1.4.3 void FullRangeNV12ToRGBBilinearResizeNormPlane (uint8_t ∗ *in_buf*, float ∗ *out_buf*, int *in_w*, int *in_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

NV12 full range(0∼255) image convert to RGB and resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3 / 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_w ∗ 3) ∗ sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.4 void FullRangeYU12ToRGBBilinearResizeNormPlane (uint8_t ∗ *in_buf*, float ∗ *out_buf*, int *in_w*, int *in_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

YU12(i420) full range(0∼255) image convert to RGB and resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3 / 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_w ∗ 3) ∗ sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.5 void GrayAffine (uint8_t ∗ *input,* uint8_t ∗ *output,* int *in_w,* int *in_h,* int *roi_w_start,* int *roi_h_start,* int *roi_w,* int *roi_h,* int *out_w,* int *out_h,* float *m[6],* cudaStream_t *stream*)

Affine transformation gray image with bilinear interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_h) ∗ sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
out_w	The width of output image.
out_h	The height of output image.
m[6]	Affine transformation matrix.
stream	CU kernel run in the stream.

1.1.4.6 void GrayResizeBilinear (uint8_t ∗ *in_buf,* uint8_t ∗ *out_buf,* int *w_in,* int *h_in,* int *w_out,* int *h_out,* cudaStream_t *stream*)

Resize gray image with bilinear interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in ∗ h_in ∗ 3) ∗ sizeof(uint8_t).

Parameters

out_buf	The output buffer allocate in device memory. The size is (w_out ∗ h_out ∗ 3) ∗ sizeof(uint8_t).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
stream	CU kernel run in the stream.

1.1.4.7 void GrayResizeNearest (uint8_t ∗ *in_buf,* uint8_t ∗ *out_buf,* int *in_w,* int *in_h,* int *out_w,* int *out_h,* cudaStream_t *stream*)

Resize gray image with nearest neighbor interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3 / 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_h ∗ 3 / 2) ∗ sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
out_w	The width of output image.
out_h	The height of output image.
stream	CU kernel run in the stream.

1.1.4.8 void MultiRoiRGBBilinearResizeNormPadPlane (uint8_t ∗ *in_buf,* float ∗ *out_buf,* int *batch,* int *in_h,* int *in_w,* int *output_batch_offset,* multi_roi_resize_param ∗ *param,* float *scale,* float *mean1,* float *mean2,* float *mean3,* float *std1,* float *std2,* float *std3,* float *pad1,* float *pad2,* float *pad3,* bool *channel_rev,* cudaStream_t *stream*)

Multiple roi resize RGB/BGR image ROI area with bilinear interpolation and normalization.

Output format is RRRGGGBBB / BBBGGGRRR. Normalization function is: output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in ∗ h_in ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out ∗ h_out ∗ 3) ∗ sizeof(float).
batch	Number of input images.
in_w	The width of input image.
in_h	The height of input image.
output_batch_offset	The stride of output per batch.
param	The information of per batch.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
channel_rev	True mean rgb convert to bgr or bgr convert to rgb. False mean input and output image is same format.
stream	CU kernel run in the stream.

1.1.4.9 void non_max_suppression (float * *predict*, int *num_batch*, int *num_bboxes*, int *num_classes*, float *confidence_threshold*, float *nms_threshold*, float * *pout*, int *max_objects*, cudaStream_t *stream*)

Non max suppression inner class.

The num_bboxes need has same value between batches. At each one batch, the predict data and out data: predict size: num_bboxes * (5 + num_classes); predict item: [[cx, cy, w, h, obj_conf, cls0_conf, ..., clsN_conf] ... [cx, cy, w, h, obj_conf, cls0_conf, ..., clsN_conf]]; pout size: 1 + max_objects * 7 pout item: [box_cnt, [x1, y1, x2, y2, confidence, class_id, keep_flag], ... [x1, y1, x2, y2, confidence, keep_flag]]

Parameters

predict	The multi-batches input buffer allocate in device memory.
num_batch	The number of batch.
num_classes	The number of class.
confidence_threshold	The confidence threshold used for filter out boxes.
nms_threshold	The nms threshold used for merge boxes.
pout	The multi-batches output buffer allocate in device memory.
max_objects	The max boxes to keep
stream	The CU kernel run in the stream.

1.1.4.10 void NV12Crop (uint8_t * *in_buf*, uint8_t * *out_buf*, int *start_w*, int *start_h*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, cudaStream_t *stream*)

Crop NV12 image.

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in * h_in * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out * h_out * 3 / 2) * sizeof(uint8_t).
start_w	The start index of crop area in the width dimension of input image.
start_h	The start index of crop area in the height dimension of input image.
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
stream	CU kernel run in the stream.

1.1.4.11 void NV12ToBGRBilinearResizeNormPlane (uint8_t * *in_buf*, float * *out_buf*, int *in_w*, int *in_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

NV12 image convert to BGR and resize with padding and normalization.

The order of BGR image is BBBGGGRRR. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3 / 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_w ∗ 3) ∗ sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.12 void NV12ToBGRNearestResizeNormPlane (uint8_t ∗ *in_buf,* float ∗ *out_buf,* int *in_w,* int *in_h,* int *img_w,* int *img_h,* int *out_w,* int *out_h,* int *pad_w,* int *pad_h,* float *mean1,* float *mean2,* float *mean3,* float *std1,* float *std2,* float *std3,* float *scale,* float *pad1,* float *pad2,* float *pad3,* cudaStream_t *stream*)

NV12 image convert to BGR and resize with padding and normalization.

The order of BGR image is BBBGGGRRR. Normalization function is: output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3 / 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_w ∗ 3) ∗ sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.13 void NV12ToRGBBilinearResizeNormPlane (uint8_t * *in_buf*, float * *out_buf*, int *in_w*, int *in_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

NV12 image convert to RGB and resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] * scale - mean[channel]) * std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.14 void NV12ToRGBNearestResizeNormPlane (uint8_t * *in_buf*, float * *out_buf*, int *in_w*, int *in_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

NV12 image convert to RGB and resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.

Parameters

pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.15 template<typename DataType , typename IndxType , bool IsAscending = false> int32_t RadixSortFunc (DataType ∗ *data_dev,* IndxType ∗ *indxs_dev,* int32_t *total_num,* DataType ∗ *data_out_dev* = `nullptr`, IndxType ∗ *indxs_out_dev* = `nullptr`, void ∗ *workspace_dev* = `nullptr`, cudaStream_t ∗ *stream* = `nullptr`)

Sort data with radix sort algorithm.

Parameters

data_dev	The keys for sort.
indxs_dev	The indexes which pair with key. Calculate by call this function with nullptr of data_dev, indxs_dev, data_out_dev and indxs_out_dev.
total_num	The total number of keys need to be sort.
data_out_dev	The sorted keys.
data_out_dev	The sorted indexes.
workspace_dev	The workspace buffer allocate in device memory.
stream	CU kernel run in the stream.

1.1.4.16 void RGB2YU12 (uint8_t ∗ *in_buf,* uint8_t ∗ *out_buf,* int *w,* int *h,* cudaStream_t *stream*)

RGB image convert to YUV420p(I420).

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_w ∗ 3 / 2) ∗ sizeof(uint8_t).
w	The width of input image.
h	The height of input image.
stream	CU kernel run in the stream.

1.1.4.17 void RGBAffine (uint8_t ∗ *input,* uint8_t ∗ *output,* int *in_w,* int *in_h,* int *roi_w_start,* int *roi_h_start,* int *roi_w,* int *roi_h,* int *out_w,* int *out_h,* float *m[6],* cudaStream_t *stream*)

Affine transformation rgb image with bilinear interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_h ∗ 3) ∗ sizeof(uint8_t).

Parameters

in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
out_w	The width of output image.
out_h	The height of output image.
m[6]	Affine transformation matrix.
stream	CU kernel run in the stream.

1.1.4.18 void RGBBilinearResizeCropNormPlaneV2 (uint8_t * *in_buf*, float * *out_buf*, uchar4 * *ws*, int *in_w*, int *in_h*, int *resized_w*, int *resized_h*, int *crop_w_start*, int *crop_h_start*, int *crop_w*, int *crop_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, bool *fmt_cvt*, cudaStream_t *stream*)

RGB resize(compatible with transforms.Resize of torchvison) with crop and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
ws	The workspace buffer allocate in device memory.
in_w	The width of input image.
in_h	The height of input image.
resized_w	The width of resized image.
resized_h	The height of resized image.
crop_w_start	The start index of crop area in the width dimension of resized image.
crop_h_start	The start index of crop area in the height dimension of resized image.
crop_w	The width of output image.
crop_h	The height of output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] * scale - mean[channel]) * std[channel]".
fmt_cvt	Convert rgb to bgr or not.
stream	CU kernel run in the stream.

1.1.4.19 void RGBCrop (uint8_t * *in_buf*, uint8_t * *out_buf*, int *start_w*, int *start_h*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, cudaStream_t *stream*)

Crop RGB image.

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in ∗ h_in ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out ∗ h_out ∗ 3) ∗ sizeof(uint8_t).
start_w	The start index of crop area in the width dimension of intput image.
start_h	The start index of crop area in the height dimension of intput image.
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
stream	CU kernel run in the stream.

1.1.4.20 void RGBNormalization (uint8_t ∗ *in_buf*, float ∗ *out_buf*, int *in_w*, int *in_h*, int *in_c*, float *mean*, float *standard*, float *scale*, cudaStream_t *stream*)

RGB image normalization.

output = (input ∗ scale - mean) ∗ standard

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ in_c) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (in_w ∗ in_h ∗ in_c) ∗ sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
in_c	The channels of input image.
mean,standard,scale	The parameter in "output = (input ∗ scale - mean) ∗ standard"
stream	CU kernel run in the stream.

1.1.4.21 void RGBNormalization_3Channels (uint8_t ∗ *in_buf*, float ∗ *out_buf*, int *in_w*, int *in_h*, float *mean1*, float *mean2*, float *mean3*, float *standard1*, float *standard2*, float *standard3*, float *scale*, bool *input_plane*, bool *output_plane*, bool *channel_rev*, cudaStream_t *stream*)

RGB image normalization.

output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3) ∗ sizeof(float).
in_w	The width of input image.
in_h	The height of input image.

Parameters

mean1,mean2,mean3,standard1,standard2,standard3,scale	The parameter in "output[channel_out] = (input[channel] ∗ scale - mean[channel]) ∗ standard[channel]".
input_plane	True mean input image is RRRGGGBBB. False mean input image is RGBRGBRGB.
output_plane	True mean output image is RRRGGGBBB. False mean output image is RGBRGBRGB.
channel_rev	True mean rgb convert to bgr or bgr convert to rgb. False mean input and output image is same format.
stream	CU kernel run in the stream.

1.1.4.22 void RGBResizeBilinear (uint8_t ∗ *in_buf*, uint8_t ∗ *out_buf*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, cudaStream_t *stream*)

Resize RGB image with bilinear interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in ∗ h_in ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out ∗ h_out ∗ 3) ∗ sizeof(uint8_t).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
stream	CU kernel run in the stream.

1.1.4.23 void RGBResizeNearest (uint8_t ∗ *in_buf*, uint8_t ∗ *out_buf*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, cudaStream_t *stream*)

Resize RGB image with nearest neighbor interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in ∗ h_in ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out ∗ h_out ∗ 3) ∗ sizeof(uint8_t).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
stream	CU kernel run in the stream.

1.1.4.24 void RGBResizePlaneBilinear (uint8_t ∗ *in_buf*, uint8_t ∗ *out_buf*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, cudaStream_t *stream*)

Resize RGB image with bilinear interpolation.

The order of RGB image is RRRGGGBBB.

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in ∗ h_in ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out ∗ h_out ∗ 3) ∗ sizeof(uint8_t).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
stream	CU kernel run in the stream.

1.1.4.25 void RGBResizePlaneNearest (uint8_t ∗ *in_buf*, uint8_t ∗ *out_buf*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, cudaStream_t *stream*)

Resize RGB image with nearest interpolation.

The order of RGB image is RRRGGGBBB.

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in ∗ h_in ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out ∗ h_out ∗ 3) ∗ sizeof(uint8_t).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
stream	CU kernel run in the stream.

1.1.4.26 void RGBResizePlanePadBilinear (uint8_t ∗ *in_buf*, uint8_t ∗ *out_buf*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, int *w_box*, int *h_box*, int *w_b*, int *h_b*, cudaStream_t *stream*)

Resize RGB image with bilinear interpolation.

The order of input and output RGB image is RRRGGGBBB. Output image has padding.

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in ∗ h_in ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_box ∗ h_box ∗ 3) ∗ sizeof(uint8_t).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of resized image.
h_out	The height of resized image.
w_box	The width of output image.
h_box	The height of output image.
w_b	The offset of width dimension of resized image in output image.
h_b	The offset of height dimension of resized image in output image.
stream	CU kernel run in the stream.

1.1.4.27 void RGBResizePlanePadNearest (uint8_t * *in_buf*, uint8_t * *out_buf*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, int *w_box*, int *h_box*, int *w_b*, int *h_b*, cudaStream_t *stream*)

Resize RGB image with nearest interpolation.

The order of input and output RGB image is RRRGGGBBB. Output image has padding.

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in * h_in * 3) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_box * h_box * 3) * sizeof(uint8_t).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of resized image.
h_out	The height of resized image.
w_box	The width of output image.
h_box	The height of output image.
w_b	The offset of width dimension of resized image in output image.
h_b	The offset of height dimension of resized image in output image.
stream	CU kernel run in the stream.

1.1.4.28 void RGBResizeWithROIBilinear (uint8_t * *in_buf*, uint8_t * *out_buf*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, cudaStream_t *stream*)

Resize RGB image ROI area with bilinear interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in * h_in * 3) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out * h_out * 3) * sizeof(uint8_t).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
stream	CU kernel run in the stream.

1.1.4.29 void RGBResizeWithROINearest (uint8_t * *in_buf*, uint8_t * *out_buf*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, cudaStream_t *stream*)

Resize RGB image ROI area with nearset interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in ∗ h_in ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out ∗ h_out ∗ 3) ∗ sizeof(uint8_t).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
stream	CU kernel run in the stream.

1.1.4.30 void RGBROIBilinearResizeNormPadPlane (uint8_t ∗ *in_buf*, float ∗ *out_buf*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, int *img_w*, int *img_h*, int *pad_w*, int *pad_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, float *scale*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *pad1*, float *pad2*, float *pad3*, bool *channel_rev*, cudaStream_t *stream*)

Resize RGB/BGR image ROI area with bilinear interpolation and normalization.

Output format is RRRGGGBBB / BBBGGGRRR. Normalization function is: output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in ∗ h_in ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out ∗ h_out ∗ 3) ∗ sizeof(float).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
img_w	The width of resized image.
img_h	The height of resized image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
channel_rev	True mean rgb convert to bgr or bgr convert to rgb. False mean input and output image is same format.
stream	CU kernel run in the stream.

1.1.4.31 void RGBROIBilinearResizeNormPlane (uint8_t * *in_buf*, float * *out_buf*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, float *scale*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, bool *channel_rev*, cudaStream_t *stream*)

Resize RGB/BGR image ROI area with bilinear interpolation and normalization.

Output format is RRRGGGBBB / BBBGGGRRR. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in * h_in * 3) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out * h_out * 3) * sizeof(float).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] * scale - mean[channel]) * std[channel]".
channel_rev	True mean rgb convert to bgr or bgr convert to rgb. False mean input and output image is same format.
stream	CU kernel run in the stream.

1.1.4.32 void RGBROINearestResizeNormPadPlane (uint8_t * *in_buf*, float * *out_buf*, int *w_in*, int *h_in*, int *w_out*, int *h_out*, int *img_w*, int *img_h*, int *pad_w*, int *pad_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, float *scale*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *pad1*, float *pad2*, float *pad3*, bool *channel_rev*, cudaStream_t *stream*)

Resize RGB/BGR image ROI area with nearset interpolation and normalization.

Output format is RRRGGGBBB / BBBGGGRRR. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in * h_in * 3) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out * h_out * 3) * sizeof(float).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.

Parameters

img_w	The width of resized image.
img_h	The height of resized image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
channel_rev	True mean rgb convert to bgr or bgr convert to rgb. False mean input and output image is same format.
stream	CU kernel run in the stream.

1.1.4.33 void RGBROINearestResizeNormPlane (uint8_t ∗ *in_buf,* float ∗ *out_buf,* int *w_in,* int *h_in,* int *w_out,* int *h_out,* int *roi_w_start,* int *roi_h_start,* int *roi_w,* int *roi_h,* float *scale,* float *mean1,* float *mean2,* float *mean3,* float *std1,* float *std2,* float *std3,* bool *channel_rev,* cudaStream_t *stream*)

Resize RGB/BGR image ROI area with nearset interpolation and normalization.

Output format is RRRGGGBBB / BBBGGGRRR. Normalization function is: output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in ∗ h_in ∗ 3) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out ∗ h_out ∗ 3) ∗ sizeof(float).
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
channel_rev	True mean rgb convert to bgr or bgr convert to rgb. False mean input and output image is same format.
stream	CU kernel run in the stream.

1.1.4.34 void RGBRotate (uint8_t * *in*, uint8_t * *out*, int *in_w*, int *in_h*, int *rotate_mode*, cudaStream_t *stream*)

Rotate RGB image with specific rotate mode.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
rotate_mode	The rotate angle of clockwise which include 90 degree, 180 degree, 270 degree.
stream	CU kernel run in the stream.

1.1.4.35 void RoiNv122RGBAffineNorm (uint8_t * *input*, float * *output*, int *in_w*, int *in_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, int *out_w*, int *out_h*, float *m[6]*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

NV12 image roi area convert to RGB and affine transformation with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
out_w	The width of output image.
out_h	The height of output image.
m[6]	Affine transformation matrix.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] * scale - mean[channel]) * std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.36 void RoiNV12ToBGRBilinearResizePlane (uint8_t * *in_buf*, float * *out_buf*, int *in_w*, int *in_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

NV12 image roi area convert to BGR and resize with padding.

The order of BGR image is BBBGGGRRR.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3 / 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_w ∗ 3) ∗ sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.37 void RoiNV12ToRGBBilinearResizeNormPlane (uint8_t ∗ *in_buf*, float ∗ *out_buf*, int *in_w*, int *in_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

NV12 image roi area convert to RGB and resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3 / 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_w ∗ 3) ∗ sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.

Parameters

pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.38 void RoiNV12ToRGBBilinearResizePlane (uint8_t ∗ *in_buf,* float ∗ *out_buf,* int *in_w,* int *in_h,* int *roi_w_start,* int *roi_h_start,* int *roi_w,* int *roi_h,* int *img_w,* int *img_h,* int *out_w,* int *out_h,* int *pad_w,* int *pad_h,* float *pad1,* float *pad2,* float *pad3,* cudaStream_t *stream*)

NV12 image roi area convert to RGB and resize with padding.

The order of RGB image is RRRGGGBBB.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3 / 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_w ∗ 3) ∗ sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.39 void RoiNV12ToRGBBilinearResizeQuantizePlane (uint8_t ∗ *in_buf,* uint8_t ∗ *out_buf,* uchar4 ∗ *ws,* int *in_w,* int *in_h,* int *roi_w_start,* int *roi_h_start,* int *roi_w,* int *roi_h,* int *img_w,* int *img_h,* int *out_w,* int *out_h,* int *pad_w,* int *pad_h,* float *pad1,* float *pad2,* float *pad3,* float *zero_point,* float *scales_input,* cudaStream_t *stream*)

NV12 image roi area convert to RGB, bilinear resize, padding and quantize output to uint8.

The order of RGB image is RRRGGGBBB. Quantize function is: output[channel] = (input ∗ scales_input + zero_↩ point

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3 / 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_h ∗ 3) ∗ sizeof(uint8_t).

Parameters

ws	The workspace buffer allocate in device memory. The size is (roi_w ∗ roi_h) ∗ sizeof(uchar4).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
pad1,pad2,pad3	The padding value of output.
zero_point	The output quantize parameter.
scales_input	The output quantize parameter.
stream	CU kernel run in the stream.

1.1.4.40 void RoiYU122BGRAffineNorm (uint8_t ∗ *input,* float ∗ *output,* int *in_w,* int *in_h,* int *roi_w_start,* int *roi_h_start,* int *roi_w,* int *roi_h,* int *out_w,* int *out_h,* float *m[6],* float *mean1,* float *mean2,* float *mean3,* float *std1,* float *std2,* float *std3,* float *scale,* float *pad1,* float *pad2,* float *pad3,* cudaStream_t *stream*)

YU12 image roi area convert to BGR and affine transformation with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3 / 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_w ∗ 3) ∗ sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
out_w	The width of output image.
out_h	The height of output image.
m[6]	Affine transformation matrix.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.41 void RoiYU122RGBAffineNorm (uint8_t * *input*, float * *output*, int *in_w*, int *in_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, int *out_w*, int *out_h*, float *m[6]*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

YU12 image roi area convert to RGB and affine transformation with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
out_w	The width of output image.
out_h	The height of output image.
m[6]	Affine transformation matrix.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] * scale - mean[channel]) * std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.42 void RoiYU12ToBGRBilinearResizePlane (uint8_t * *in_buf*, float * *out_buf*, int *in_w*, int *in_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

YU12(i420) image roi area convert to BGR and resize with padding and normalization.

The order of BGR image is BBBGGGRRR. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.

Parameters

img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.43 void RoiYU12ToRGBBilinearResizeNormPlane (uint8_t * *in_buf*, float * *out_buf*, int *in_w*, int *in_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

YU12(i420) image roi area convert to RGB and resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] * scale - mean[channel]) * std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.44 void RoiYU12ToRGBBilinearResizeNormPlaneV2 (uint8_t * *in_buf*, float * *out_buf*, uchar4 * *ws*, int *in_w*, int *in_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, int *out_w*, int *out_h*, int *input_pad_w*, int *input_pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, uint8_t *pad1*, uint8_t *pad2*, uint8_t *pad3*, cudaStream_t *stream*)

YU12(i420) image roi area convert to RGB and resize(compatible with transforms.Resize of torchvison) with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
ws	The workspace buffer allocate in device memory.
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
input_pad_w	The pad of width dimension of input image.
input_pad_h	The pad of height dimension of input image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] * scale - mean[channel]) * std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.45 void RoiYU12ToRGBBilinearResizePlane (uint8_t * *in_buf*, float * *out_buf*, int *in_w*, int *in_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

YU12(i420) image roi area convert to RGB and resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.

Parameters

in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.46 void RoiYU12ToRGBBilinearResizeQuantizePlane (uint8_t * *in_buf,* uint8_t * *out_buf,* uchar4 * *ws,* int *in_w,* int *in_h,* int *roi_w_start,* int *roi_h_start,* int *roi_w,* int *roi_h,* int *img_w,* int *img_h,* int *out_w,* int *out_h,* int *pad_w,* int *pad_h,* float *pad1,* float *pad2,* float *pad3,* float *zero_point,* float *scales_input,* cudaStream_t *stream*)

YU12 image roi area convert to RGB, bilinear resize, padding and quantize output to uint8.

The order of RGB image is RRRGGGBBB. Quantize function is: output[channel] = (input * scales_input + zero_↩ point

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_h * 3) * sizeof(uint8_t).
ws	The workspace buffer allocate in device memory. The size is (roi_w * roi_h) * sizeof(uchar4).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
pad1,pad2,pad3	The padding value of output.
zero_point	The output quantize parameter.
scales_input	The output quantize parameter.
stream	CU kernel run in the stream.

1.1.4.47 void RoiYUV400PToRGBBilinearResizeNormPlane (uint8_t * *in_buf*, float * *out_buf*, int *in_w*, int *in_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean*, float *std*, float *scale*, float *pad*, cudaStream_t *stream*)

YUV400P image roi area convert to RGB and bilinear resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean,std,scale	The parameter in "output[channel] = (input[channel] * scale - mean[channel]) * std[channel]".
pad	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.48 void RoiYUV422PToRGBBilinearResizeNormPlane (uint8_t * *in_buf*, float * *out_buf*, int *in_w*, int *in_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

YUV422P image roi area convert to RGB and bilinear resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.

Parameters

roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.49 void RoiYUV422ToRGBBilinearResizeNormPlane (uint8_t ∗ *in_buf*, float ∗ *out_buf*, int *in_w*, int *in_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

YUV422 image roi area convert to RGB and bilinear resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_w ∗ 3) ∗ sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.50 void RoiYUV444PToRGBBilinearResizeNormPlane (uint8_t * *in_buf*, float * *out_buf*, int *in_w*, int *in_h*, int *roi_w_start*, int *roi_h_start*, int *roi_w*, int *roi_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

YUV444P image roi area convert to RGB and bilinear resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
roi_w_start	The start index of ROI area in the width dimension of intput image.
roi_h_start	The start index of ROI area in the height dimension of intput image.
roi_w	The width of ROI area in the width dimension of intput image.
roi_h	The height of ROI area in the height dimension of intput image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] * scale - mean[channel]) * std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.51 void YU122RGBAffine (uint8_t * *input*, uint8_t * *output*, int *in_w*, int *in_h*, int *out_w*, int *out_h*, float *m[6]*, cudaStream_t *stream*)

YU12(i420) image convert to RGB and affine transformation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
out_w	The width of output image.
out_h	The height of output image.
m[6]	Affine transformation matrix.
stream	CU kernel run in the stream.

1.1.4.52 void YU12Crop (uint8_t * *in_buf,* uint8_t * *out_buf,* int *start_w,* int *start_h,* int *w_in,* int *h_in,* int *w_out,* int *h_out,* cudaStream_t *stream*)

Crop YU12 image.

Parameters

in_buf	The input buffer allocate in device memory. The size is (w_in * h_in * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (w_out * h_out * 3 / 2) * sizeof(uint8_t).
start_w	The start index of crop area in the width dimension of input image.
start_h	The start index of crop area in the height dimension of input image.
w_in	The width of input image.
h_in	The height of input image.
w_out	The width of output image.
h_out	The height of output image.
stream	CU kernel run in the stream.

1.1.4.53 void YU12ToBGRBilinearResizeNormPlane (uint8_t * *in_buf,* float * *out_buf,* int *in_w,* int *in_h,* int *img_w,* int *img_h,* int *out_w,* int *out_h,* int *pad_w,* int *pad_h,* float *mean1,* float *mean2,* float *mean3,* float *std1,* float *std2,* float *std3,* float *scale,* float *pad1,* float *pad2,* float *pad3,* cudaStream_t *stream*)

YU12(i420) image convert to BGR and resize with padding and normalization.

The order of BGR image is BBBGGGRRR. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] * scale - mean[channel]) * std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.54 void YU12ToBGRNearestResizeNormPlane (uint8_t * *in_buf*, float * *out_buf*, int *in_w*, int *in_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

YU12(i420) image convert to BGR and resize with padding and normalization.

The order of BGR image is BBBGGGRRR. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] * scale - mean[channel]) * std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.55 void YU12ToRGBBilinearResizeNormPlane (uint8_t * *in_buf*, float * *out_buf*, int *in_w*, int *in_h*, int *img_w*, int *img_h*, int *out_w*, int *out_h*, int *pad_w*, int *pad_h*, float *mean1*, float *mean2*, float *mean3*, float *std1*, float *std2*, float *std3*, float *scale*, float *pad1*, float *pad2*, float *pad3*, cudaStream_t *stream*)

YU12(i420) image convert to RGB and resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] * scale - mean[channel]) * standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_w * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.

Parameters

pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.56 void YU12ToRGBNearestResizeNormPlane (uint8_t ∗ *in_buf,* float ∗ *out_buf,* int *in_w,* int *in_h,* int *img_w,* int *img_h,* int *out_w,* int *out_h,* int *pad_w,* int *pad_h,* float *mean1,* float *mean2,* float *mean3,* float *std1,* float *std2,* float *std3,* float *scale,* float *pad1,* float *pad2,* float *pad3,* cudaStream_t *stream*)

YU12(i420) image convert to RGB and resize with padding and normalization.

The order of RGB image is RRRGGGBBB. Normalization function is: output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ standard[channel]

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3 / 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w ∗ out_w ∗ 3) ∗ sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
img_w	The width of resized image.
img_h	The height of resized image.
out_w	The width of output image.
out_h	The height of output image.
pad_w	The offset of width dimension of resized image in output image.
pad_h	The offset of height dimension of resized image in output image.
mean1,mean2,mean3,std1,std2,std3,scale	The parameter in "output[channel] = (input[channel] ∗ scale - mean[channel]) ∗ std[channel]".
pad1,pad2,pad3	The padding value of output.
stream	CU kernel run in the stream.

1.1.4.57 void YUV440pToYUV420p (uint8_t ∗ *sy,* uint8_t ∗ *su,* uint8_t ∗ *sv,* uint8_t ∗ *dy,* uint8_t ∗ *du,* uint8_t ∗ *dv,* int *w,* int *h,* int *align_w,* cudaStream_t *stream*)

yuv440p transform to yuv420p.

Parameters

sy,su,sv	The input buffer of [Y, U, V] 3 channels allocate in device memory.
dy,du,dv	The output buffer of [Y, U, V] 3 channels allocate in device memory.
w	The width of image.
h	The height of image.
align_w	The stride of height of input image.
stream	CU kernel run in the stream.

1.1.4.58 void YUV444pToYUV420p (uint8_t * *sy,* uint8_t * *su,* uint8_t * *sv,* uint8_t * *dy,* uint8_t * *du,* uint8_t * *dv,* int *w,* int *h,* int *align_w,* cudaStream_t *stream*)

yuv444p transform to yuv420p.

Parameters

sy,su,sv	The input buffer of [Y, U, V] 3 channels allocate in device memory.
dy,du,dv	The output buffer of [Y, U, V] 3 channels allocate in device memory.
w	The width of image.
h	The height of image.
align_w	The stride of height of input image.
stream	CU kernel run in the stream.

1.1.4.59 void YUVI420ResizeBilinear (uint8_t * *in_buf,* uint8_t * *out_buf,* int *in_w,* int *in_h,* int *out_w,* int *out_h,* cudaStream_t *stream*)

Resize YUV(i420) image with bilinear interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_h * 3 / 2) * sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
out_w	The width of output image.
out_h	The height of output image.
stream	CU kernel run in the stream.

1.1.4.60 void YUVI420ResizeNearest (uint8_t * *in_buf,* uint8_t * *out_buf,* int *in_w,* int *in_h,* int *out_w,* int *out_h,* cudaStream_t *stream*)

Resize YUV(i420) image with nearest neighbor interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_h * 3 / 2) * sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
out_w	The width of output image.
out_h	The height of output image.
stream	CU kernel run in the stream.

1.1.4.61 void YUVNv12ResizeBilinear (uint8_t * *in_buf*, uint8_t * *out_buf*, int *in_w*, int *in_h*, int *out_w*, int *out_h*, cudaStream_t *stream*)

Resize YUV(nv12) image with bilinear interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_h * 3 / 2) * sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
out_w	The width of output image.
out_h	The height of output image.
stream	CU kernel run in the stream.

1.1.4.62 void YUVNv12ResizeNearest (uint8_t * *in_buf*, uint8_t * *out_buf*, int *in_w*, int *in_h*, int *out_w*, int *out_h*, cudaStream_t *stream*)

Resize YUV(nv12) image with nearest neighbor interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_h * 3 / 2) * sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
out_w	The width of output image.
out_h	The height of output image.
stream	CU kernel run in the stream.

1.1.4.63 void YUVNv12ToRGB (uint8_t * *in_buf*, uint8_t * *out_buf*, int *in_w*, int *in_h*, cudaStream_t *stream*)

Convert YUV(nv12) image to RGB image.

The order of RGB image is RGBRGBRGB.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (in_w * in_h * 3) * sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
stream	CU kernel run in the stream.

1.1.4.64 void YUVNv12ToRGBFloat (uint8_t * *in_buf,* float * *out_buf,* int *in_w,* int *in_h,* cudaStream_t *stream*)

Convert YUV(nv12) image to RGB image.

The order of RGB image is RGBRGBRGB.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (in_w * in_h * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
stream	CU kernel run in the stream.

1.1.4.65 void YUVNv12ToRGBPlane (uint8_t * *in_buf,* uint8_t * *out_buf,* int *in_w,* int *in_h,* cudaStream_t *stream*)

Convert YUV(nv12) image to RGB image.

The order of RGB image is RRRGGGBBB.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (in_w * in_h * 3) * sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
stream	CU kernel run in the stream.

1.1.4.66 void YUVNv21ResizeBilinear (uint8_t * *in_buf,* uint8_t * *out_buf,* int *in_w,* int *in_h,* int *out_w,* int *out_h,* cudaStream_t *stream*)

Resize YUV(nv21) image with bilinear interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_h * 3 / 2) * sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
out_w	The width of output image.
out_h	The height of output image.
stream	CU kernel run in the stream.

1.1.4.67 void YUVNv21ResizeNearest (uint8_t * *in_buf,* uint8_t * *out_buf,* int *in_w,* int *in_h,* int *out_w,* int *out_h,* cudaStream_t *stream*)

Resize YUV(nv21) image with nearest neighbor interpolation.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (out_w * out_h * 3 / 2) * sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
out_w	The width of output image.
out_h	The height of output image.
stream	CU kernel run in the stream.

1.1.4.68 void YUVYu12ToRGB (uint8_t * *in_buf,* uint8_t * *out_buf,* int *in_w,* int *in_h,* cudaStream_t *stream*)

Convert YUV(yu12, also called i420) image to RGB image.

The order of RGB image is RGBRGBRGB.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (in_w * in_h * 3) * sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
stream	CU kernel run in the stream.

1.1.4.69 void YUVYu12ToRGBFloat (uint8_t * *in_buf,* float * *out_buf,* int *in_w,* int *in_h,* cudaStream_t *stream*)

Convert YUV(yu12, also called i420) image to RGB image.

The order of RGB image is RGBRGBRGB.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w * in_h * 3 / 2) * sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (in_w * in_h * 3) * sizeof(float).
in_w	The width of input image.
in_h	The height of input image.
stream	CU kernel run in the stream.

1.1.4.70 void YUVYu12ToRGBPlane (uint8_t * *in_buf,* uint8_t * *out_buf,* int *in_w,* int *in_h,* cudaStream_t *stream*)

Convert YUV(yu12, also called i420) image to RGB image.

The order of RGB image is RRRGGGBBB.

Parameters

in_buf	The input buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3 / 2) ∗ sizeof(uint8_t).
out_buf	The output buffer allocate in device memory. The size is (in_w ∗ in_h ∗ 3) ∗ sizeof(uint8_t).
in_w	The width of input image.
in_h	The height of input image.
stream	CU kernel run in the stream.

Index