#ifndef IM2COL\_H

#define IM2COL\_H

#include <stddef.h>

#include <stdint.h>

#include "darknet.h"

#ifdef \_\_cplusplus

extern "C" {

#endif

void im2col\_cpu(float\* data\_im,

int channels, int height, int width,

int ksize, int stride, int pad, float\* data\_col);

float im2col\_get\_pixel(float\* im, int height, int width, int channels,

int row, int col, int channel, int pad);

void im2col\_cpu\_ext(const float\* data\_im, const int channels,

const int height, const int width, const int kernel\_h, const int kernel\_w,

const int pad\_h, const int pad\_w,

const int stride\_h, const int stride\_w,

const int dilation\_h, const int dilation\_w,

float\* data\_col);

#ifdef GPU

void im2col\_ongpu(float \*im,

int channels, int height, int width,

int ksize, int stride, int pad,float \*data\_col);

void im2col\_gpu\_ext(const float\* data\_im, const int channels,

const int height, const int width, const int kernel\_h, const int kernel\_w,

const int pad\_h, const int pad\_w,

const int stride\_h, const int stride\_w,

const int dilation\_h, const int dilation\_w,

float\* data\_col);

void im2col\_align\_ongpu(float \*im,

int channels, int height, int width,

int ksize, int stride, int pad, float \*data\_col, int bit\_align);

void im2col\_align\_bin\_ongpu(float \*im,

int channels, int height, int width,

int ksize, int stride, int pad, float \*data\_col, int bit\_align);

void float\_to\_bit\_gpu(float \*src, unsigned char \*dst, size\_t size);

void transpose\_bin\_gpu(unsigned char \*A, unsigned char \*B, const int n, const int m,

const int lda, const int ldb, const int block\_size);

void transpose\_uint32\_gpu(uint32\_t \*src, uint32\_t \*dst, int src\_h, int src\_w, int src\_align, int dst\_align);

void transpose\_uint32\_gpu\_2(uint32\_t \*src, uint32\_t \*dst, int src\_h, int src\_w, int src\_align, int dst\_align);

void repack\_input\_gpu(float \*input, float \*re\_packed\_input, int w, int h, int c);

void repack\_input\_gpu\_2(float \*input, float \*re\_packed\_input, int w, int h, int c);

void repack\_input\_gpu\_bin(float \*input, uint32\_t \*re\_packed\_input\_bin, int w, int h, int c);

void fill\_int8\_gpu(unsigned char \*src, unsigned char val, size\_t size);

// shared\_memory + partial coalescing = GOOD

void gemm\_nn\_custom\_bin\_mean\_transposed\_gpu(int M, int N, int K,

unsigned char \*A, int lda,

unsigned char \*B, int ldb,

float \*C, int ldc, float \*mean\_arr, float \*bias, int leaky\_activation,

float \*shortcut\_in\_gpu, float \*shortcut\_out\_gpu);

// sequentially - BAD

void gemm\_nn\_custom\_bin\_mean\_transposed\_sequentially\_gpu(int M, int N, int K,

unsigned char \*A, int lda,

unsigned char \*B, int ldb,

float \*C, int ldc, float \*mean\_arr);

void convolve\_gpu(float \*input, float \*weights, float \*output, int in\_w, int in\_h, int in\_c, int n, int size, int pad);

void convolve\_bin\_gpu(float \*input, float \*weights, float \*output, int in\_w, int in\_h, int in\_c, int n, int size, int pad,

int new\_lda, float \*mean\_arr\_gpu);

//void convolve\_bin\_cpu(float \*input, float \*weights, float \*output, int in\_w, int in\_h, int in\_c, int n, int size, int pad, int new\_lda, float \*mean\_arr\_gpu);

//void convolve\_cpu(float \*input, float \*weights, float \*output, int in\_w, int in\_h, int in\_c, int n, int size, int pad);

#endif

#ifdef \_\_cplusplus

}

#endif

#endif