#ifndef LOCAL\_LAYER\_H

#define LOCAL\_LAYER\_H

#include "dark\_cuda.h"

#include "image.h"

#include "activations.h"

#include "layer.h"

#include "network.h"

typedef layer local\_layer;

#ifdef \_\_cplusplus

extern "C" {

#endif

#ifdef GPU

void forward\_local\_layer\_gpu(local\_layer layer, network\_state state);

void backward\_local\_layer\_gpu(local\_layer layer, network\_state state);

void update\_local\_layer\_gpu(local\_layer layer, int batch, float learning\_rate, float momentum, float decay, float loss\_scale);

void push\_local\_layer(local\_layer layer);

void pull\_local\_layer(local\_layer layer);

#endif

local\_layer make\_local\_layer(int batch, int h, int w, int c, int n, int size, int stride, int pad, ACTIVATION activation);

void forward\_local\_layer(const local\_layer layer, network\_state state);

void backward\_local\_layer(local\_layer layer, network\_state state);

void update\_local\_layer(local\_layer layer, int batch, float learning\_rate, float momentum, float decay);

void bias\_output(float \*output, float \*biases, int batch, int n, int size);

void backward\_bias(float \*bias\_updates, float \*delta, int batch, int n, int size);

#ifdef \_\_cplusplus

}

#endif

#endif