#ifndef MAXPOOL\_LAYER\_H

#define MAXPOOL\_LAYER\_H

#include "image.h"

#include "dark\_cuda.h"

#include "layer.h"

#include "network.h"

typedef layer maxpool\_layer;

#ifdef \_\_cplusplus

extern "C" {

#endif

image get\_maxpool\_image(maxpool\_layer l);

maxpool\_layer make\_maxpool\_layer(int batch, int h, int w, int c, int size, int stride\_x, int stride\_y, int padding, int maxpool\_depth, int out\_channels, int antialiasing, int avgpool, int train);

void resize\_maxpool\_layer(maxpool\_layer \*l, int w, int h);

void forward\_maxpool\_layer(const maxpool\_layer l, network\_state state);

void backward\_maxpool\_layer(const maxpool\_layer l, network\_state state);

void forward\_local\_avgpool\_layer(const maxpool\_layer l, network\_state state);

void backward\_local\_avgpool\_layer(const maxpool\_layer l, network\_state state);

#ifdef GPU

void forward\_maxpool\_layer\_gpu(maxpool\_layer l, network\_state state);

void backward\_maxpool\_layer\_gpu(maxpool\_layer l, network\_state state);

void cudnn\_maxpool\_setup(maxpool\_layer \*l);

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

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

#endif // GPU

#ifdef \_\_cplusplus

}

#endif

#endif