#ifndef BOX\_H

#define BOX\_H

#include "darknet.h"

//typedef struct{

// float x, y, w, h;

//} box;

typedef struct{

float dx, dy, dw, dh;

} dbox;

//typedef struct detection {

// box bbox;

// int classes;

// float \*prob;

// float \*mask;

// float objectness;

// int sort\_class;

//} detection;

typedef struct detection\_with\_class {

detection det;

// The most probable class id: the best class index in this->prob.

// Is filled temporary when processing results, otherwise not initialized

int best\_class;

} detection\_with\_class;

#ifdef \_\_cplusplus

extern "C" {

#endif

box float\_to\_box(float \*f);

box float\_to\_box\_stride(float \*f, int stride);

float box\_iou(box a, box b);

float box\_iou\_kind(box a, box b, IOU\_LOSS iou\_kind);

float box\_rmse(box a, box b);

dxrep dx\_box\_iou(box a, box b, IOU\_LOSS iou\_loss);

float box\_giou(box a, box b);

float box\_diou(box a, box b);

float box\_ciou(box a, box b);

dbox diou(box a, box b);

boxabs to\_tblr(box a);

void do\_nms(box \*boxes, float \*\*probs, int total, int classes, float thresh);

void do\_nms\_sort\_v2(box \*boxes, float \*\*probs, int total, int classes, float thresh);

//LIB\_API void do\_nms\_sort(detection \*dets, int total, int classes, float thresh);

//LIB\_API void do\_nms\_obj(detection \*dets, int total, int classes, float thresh);

//LIB\_API void diounms\_sort(detection \*dets, int total, int classes, float thresh, NMS\_KIND nms\_kind, float beta1);

box decode\_box(box b, box anchor);

box encode\_box(box b, box anchor);

// Creates array of detections with prob > thresh and fills best\_class for them

// Return number of selected detections in \*selected\_detections\_num

detection\_with\_class\* get\_actual\_detections(detection \*dets, int dets\_num, float thresh, int\* selected\_detections\_num, char \*\*names);

#ifdef \_\_cplusplus

}

#endif

#endif