#ifndef UTILS\_H

#define UTILS\_H

#include "darknet.h"

#include "list.h"

#include <stdio.h>

#include <time.h>

#ifndef M\_PI

#define M\_PI 3.14159265358979323846 // pi

#endif

#ifdef \_\_cplusplus

extern "C" {

#endif

LIB\_API void free\_ptrs(void \*\*ptrs, int n);

LIB\_API void top\_k(float \*a, int n, int k, int \*index);

void \*xmalloc(size\_t size);

void \*xcalloc(size\_t nmemb, size\_t size);

void \*xrealloc(void \*ptr, size\_t size);

double what\_time\_is\_it\_now();

int \*read\_map(char \*filename);

void shuffle(void \*arr, size\_t n, size\_t size);

void sorta\_shuffle(void \*arr, size\_t n, size\_t size, size\_t sections);

char \*basecfg(char \*cfgfile);

int alphanum\_to\_int(char c);

char int\_to\_alphanum(int i);

int read\_int(int fd);

void write\_int(int fd, int n);

void read\_all(int fd, char \*buffer, size\_t bytes);

void write\_all(int fd, char \*buffer, size\_t bytes);

int read\_all\_fail(int fd, char \*buffer, size\_t bytes);

int write\_all\_fail(int fd, char \*buffer, size\_t bytes);

LIB\_API void find\_replace(const char\* str, char\* orig, char\* rep, char\* output);

void replace\_image\_to\_label(const char\* input\_path, char\* output\_path);

void error(const char \*s);

void malloc\_error();

void calloc\_error();

void realloc\_error();

void file\_error(char \*s);

void strip(char \*s);

void strip\_args(char \*s);

void strip\_char(char \*s, char bad);

list \*split\_str(char \*s, char delim);

char \*fgetl(FILE \*fp);

list \*parse\_csv\_line(char \*line);

char \*copy\_string(char \*s);

int count\_fields(char \*line);

float \*parse\_fields(char \*line, int n);

void normalize\_array(float \*a, int n);

void scale\_array(float \*a, int n, float s);

void translate\_array(float \*a, int n, float s);

int max\_index(float \*a, int n);

int top\_max\_index(float \*a, int n, int k);

float constrain(float min, float max, float a);

int constrain\_int(int a, int min, int max);

float mse\_array(float \*a, int n);

float rand\_normal();

size\_t rand\_size\_t();

float rand\_uniform(float min, float max);

float rand\_scale(float s);

int rand\_int(int min, int max);

float sum\_array(float \*a, int n);

float mean\_array(float \*a, int n);

void mean\_arrays(float \*\*a, int n, int els, float \*avg);

float variance\_array(float \*a, int n);

float mag\_array(float \*a, int n);

float mag\_array\_skip(float \*a, int n, int \* indices\_to\_skip);

float dist\_array(float \*a, float \*b, int n, int sub);

float \*\*one\_hot\_encode(float \*a, int n, int k);

float sec(clock\_t clocks);

int find\_int\_arg(int argc, char \*\*argv, char \*arg, int def);

float find\_float\_arg(int argc, char \*\*argv, char \*arg, float def);

int find\_arg(int argc, char\* argv[], char \*arg);

char \*find\_char\_arg(int argc, char \*\*argv, char \*arg, char \*def);

int sample\_array(float \*a, int n);

int sample\_array\_custom(float \*a, int n);

void print\_statistics(float \*a, int n);

unsigned int random\_gen\_fast(void);

float random\_float\_fast();

int rand\_int\_fast(int min, int max);

unsigned int random\_gen();

float random\_float();

float rand\_uniform\_strong(float min, float max);

float rand\_precalc\_random(float min, float max, float random\_part);

double double\_rand(void);

unsigned int uint\_rand(unsigned int less\_than);

int check\_array\_is\_nan(float \*arr, int size);

int check\_array\_is\_inf(float \*arr, int size);

int int\_index(int \*a, int val, int n);

int \*random\_index\_order(int min, int max);

int max\_int\_index(int \*a, int n);

boxabs box\_to\_boxabs(const box\* b, const int img\_w, const int img\_h, const int bounds\_check);

int make\_directory(char \*path, int mode);

#define max\_val\_cmp(a,b) (((a) > (b)) ? (a) : (b))

#define min\_val\_cmp(a,b) (((a) < (b)) ? (a) : (b))

#ifdef \_\_cplusplus

}

#endif

#endif