ISEE算法模块接口定义

version 0.2

1. 基本定义

1. /\*\* @file basic\_define.h
2. \*  @brief   Basic definitions used in the code for LPRC.
3. \*  @author  CRIPAC
4. \*  @version 0.1
5. \*  @date    2018/04/08
6. \*/
8. #ifndef \_LCPR\_BASIC\_DEFINE\_H\_
9. #define \_LCPR\_BASIC\_DEFINE\_H\_
11. /\*Error list.\*/
12. **enum** LPRC\_ErrorsList {
13. LCPR\_SUCCESS = 0,
14. LCPR\_LOAD\_IMG\_FAILED = -1,
15. LCPR\_LOAD\_MODEL\_FAILED = -2,
16. LCPR\_LOAD\_WEIGHTS\_FAILED = -3,
17. LCPR\_LOAD\_PARAMS\_FILE\_FAILED = -4,
18. LCPR\_BAD\_VAL = -5,
19. };
21. /\* Information of input image.\*/
22. **typedef** **struct** lcpr\_input\_image\_t {
23. **int** img\_w;
24. **int** img\_h;
25. **int** num\_channels;
26. unsigned **char**\* img\_data;  // bgr, bgr, bgr, ...
27. } LCPRInputIMG;
29. /\* Bounding box \*/
30. **typedef** **struct** lcpr\_bounding\_box\_t {
31. **int** top;
32. **int** left;
33. **int** bottom;
34. **int** right;
35. } LCPRBoundingBox;
37. #endif  //\_LCPR\_BASIC\_DEFINE\_H\_

2. 各模块接口定义

(1). 目标检测

1. /\*\* @file pedestrian\_detection.h
2. \*  @brief   Interfaces for pedestrian detection for a single image.
3. \*  @author  CRIPAC
4. \*  @version 0.1
5. \*  @date    2014/04/09
6. \*/
7. #ifndef \_LCPR\_PEDESTRIAN\_DETECTION\_H\_
8. #define \_LCPR\_PEDESTRIAN\_DETECTION\_H\_
10. #include "basic\_define.h"
12. **namespace** lcpr {
13. #define MAX\_PEDESTRIAN\_NUM 128
15. /\* Detected pedestrian information.\*/
16. **typedef** **struct** pedstrian\_t {
17. **float** score;
18. LCPRBoundingBox bb;
19. } Pedestrian;
21. /\* Ouput information.\*/
22. **typedef** **struct** detection\_output\_t {
23. **int** num;
24. Pedestrian detected\_pedestrians[MAX\_PEDESTRIAN\_NUM];
25. } DetectionOutput;
27. /\*\* @class LCPRPedestrianDetection
28. \*/
29. **class** PedestrianDetection {
31. **public**:
32. /\* Constructor and destructor.\*/
33. PedestrianDetection(**void**){}
34. ~PedestrianDetection(**void**){}
36. /\*\*
37. \* \brief Initialization.
38. \* \param[IN] params - the necessary parameters used for pedestrian detection.
39. \*            (It can be a pointer to a struct of paramters or a path of a file
40. \*             including all the parameters.)
41. \* \return a handle (a pointor to the inside object to achieve detection).
42. \*/
43. **long** initialize(**void**\* params);
45. /\*\*
46. \* \brief detect the pedestrian(s) in a single image.
47. \* \param[IN] handle
48. \* \param[IN] img - the rgb data of input image and its basic information.
49. \* \param[OUT] detected - detection results.
50. \* \return error code (as shown in basic\_define.h).
51. \*/
52. **int** detect(**long** handle, **const** LCPRInputIMG& img, DetectionOutput& detected);
54. /\*\*
55. \* \brief realse resources.
56. \* \param[IN] handle
57. \*/
58. **void** release(**long** handle);
59. }; // PedestrianDetection
60. }  // lcpr
61. #endif //\_LCPR\_PEDESTRIAN\_DETECTION\_H\_

说明：

1). 函数initialize用于初始化网络，指定计算资源，以及为一些必要参数赋值等（输入可以是指向“参数结构体的指针”或包含所需参数的文件路径），初始化成功返回“句柄”，即指向用于检测的实例化对象的指针，该句柄用于之后的检测和资源释放函数。

2). 函数detect即对输入的一帧RGB图像进行行人检测，这里数据格式与OpenCV (c/c++)相同，为BGR,BGR,...排列方式。

3). 函数release用于释放资源。

(2). 属性识别

1. /\*\*
2. \* @file    attribute\_recognition.h
3. \* @brief   The interface to do attribute recognition through calling the trained
4. \*          model which is provided by users.
5. \* @author  CRIPAC
6. \* @version 0.1
7. \* @date    2018/04/09
8. \*/
9. #ifndef \_LCPR\_ATTR\_RECOGNTION\_H\_
10. #define \_LCPR\_ATTR\_RECOGNTION\_H\_
12. #include "basic\_define.h"
14. **namespace** lcpr{
15. **class** AttributeRecogntion {
16. **public**:
17. /\*Constructor and destructor.\*/
18. AttributeRecogntion(**void**){}
19. ~AttributeRecogntion(**void**){}
21. /\*\* \brief Initialization.
22. \*  \param[IN] params - parameters used in initialization (a pointer to a struct
23. \*             or a path about the file including all parameters).
24. \*  \return a handle (a pointor to the instantiated object to attribute recognition)
25. \*/
26. **long** initialize(**void**\* params);
28. /\*\* \brief To recognize the attributes of an image.
29. \*  \param[IN] handle
30. \*  \param[IN] img - the rgb data of input image and its basic information.
31. \*  \return recognition results.
32. \*/
33. **const** **float**\* recognize(**long** handle, **const** LCPRInputIMG& img);
35. /\*\* \brief To recognize the attributes of multiple images.
36. \*  \param[IN] handle
37. \*  \param[IN] batch\_size - the number of input images.
38. \*  \param[IN] imgs - an array of images.
39. \*  \return recognition results.
40. \*/
41. **const** **float**\* recognize(**long** handle, **int** batch\_size, LCPRInputIMG\* imgs);
43. /\*\* \brief release the resources.
44. \*  \param[IN] handle
45. \*/
46. **void** release(**long** handle);
47. };  // AttributeRecognition
48. } // lcpr
50. #endif //\_LCPR\_ATTR\_RECOGNTION\_H\_

说明：

1). 初始化函数的输入参数同“行人检测”要求。

2). 输入图像并不是完整图像，而是检测到的矩形框内对应的数据。

3). 需要显式给出输出层维数，若为一次处理多帧，请确保输出结果顺序与输入一致。

4). 需要在提供的说明文档中明确指明输出层结果与属性名称之间的对应关系。

(3). 行人再识别

1. /\*\*
2. \* @file    reid\_feature\_extraction.h
3. \* @brief   The interface to extract reid features through calling the trained
4. \*          model which is provided by users.
5. \* @author  CRIPAC
6. \* @version 0.1
7. \* @date    2018/04/10
8. \*/
10. #ifndef \_LCPR\_REID\_FEATURE\_EXTRACTION\_H\_
11. #define \_LCPR\_REID\_FEATURE\_EXTRACTION\_H\_
13. #include "basic\_define.h"
15. **namespace** lcpr{
16. **class** ReIDFeatureExtraction {
17. **public**:
18. /\*Constructor and destructor.\*/
19. ReIDFeatureExtraction(**void**){}
20. ~ReIDFeatureExtraction(**void**){}
22. /\*\* \brief Initialization.
23. \*  \param[IN] params - parameters used in ReID feature extraction (a pointer to a struct
24. \*             or a file path to the parameters file).
25. \*  \return a handle (a pointor to the instantiated object to attribute recognition)
26. \*/
27. **long** initialize(**void**\* params);
29. /\*\* \brief To extract the reid features of a single image.
30. \*  \param[IN] handle
31. \*  \param[IN] img - the rgb data of input image and its basic information.
32. \*  \return reid features.
33. \*/
34. **const** **float**\* extract(**long** handle, **const** LCPRInputIMG& img);
36. /\*\* \brief To extract reid features of multiple images.
37. \*  \param[IN] handle
38. \*  \param[IN] batch\_size - the number of input images.
39. \*  \param[IN] imgs - an array of images.
40. \*  \return reid features.
41. \*/
42. **const** **float**\* extract(**long** handle, **int** batch\_size, LCPRInputIMG\* imgs);
44. /\*\* \brief release the resources.
45. \*  \param[IN] handle
46. \*/
47. **void** release(**long** handle);
48. };  // ReIDFeatureExtraction
49. }   // lcpr
51. #endif // \_LCPR\_REID\_FEATURE\_EXTRACTION\_H\_

说明：

1). 该接口目的为提取输入图像的ReID特征。

2). 其他注意事项同“属性识别”：1). 2). 3). 4).

**注：**

**1). 请依据上述接口给出可正常运行的单机版测试代码。**

**2). 目前系统平台只支持Caffe(2)，Pytorch以及Tensorflow框架训练的模型，且网络结构中不能包含自定义的layer，否则请提供源代码，或封装好的动态链接库。**

**3). 这里的源代码最好为c/c++，或者底层为python，最外层再裹一层c/c++，且应给出详细说明文档，包括依赖环境和版本等。**

**4). 若提供源代码请确保代码内部无“内存泄露”情况，建议使用valgrind自行进行检测。**

**5). 目前只能调用在linux环境下封装的代码。**

**6). 集群环境中cuda版本为8.0，cudnn为6.0.**