计算机视觉总结

SUMMARY OF COMPUTER VISION

(第1版)

LVSHUAILIN

OPEN SOURCE
BEIJING

VERSION 1

一. 数据结构与算法-LeetCode Hot 100

二. PYTHON: 1) NUMPY; 2) PANDAS; 3) PYTHON多进程; 4) PYTHON分布式; 5) PYTHON界面;

三. 深度学习: TensorFlow 2.0; PYTORCH;

四. 图像配准

五. 强化学习

六. OTHERS: 1) Model INFERENCE by EXE; 2) GIT; 3) DOCKER

LVSHUAILIN

2020年2月

目　　录

第1章　绪论

Goals to Achieve

1. unordered_map.

§ 1.1 两数之和

HOT100 1.1　问题描述

给定一个整数数组**nums** 和一个目标值**target**, 请你在该数组中找出和为目标值的那两个整数, 并返回他们的数组下标. 你可以假设每种输入只会对应一个答案. 但是, 你不能重复利用这个数组中同样的元素.

示例: 给定nums = [2, 7, 11, 15], target = 9; 因为nums[0] + nums[1] = 2 + 7 = 9;

所以返回[0, 1]

https://leetcode-cn.com/problems/two-sum

1.1.1 解题思路

这里用c++的unordered_map来解决, unordered_map内部是一个关联容器, 采用hash 表结构, 有快速检索的功能.

哈希表是通过key关键字直接访问对应value值的数据结构. 特点是键和值一一对应, 查找时间复杂度**O(1)**.

unordered_map example code

```
#include <iostream>
#include <unordered_map>
#include <string>
using namespace std;
int main()
{
        unordered_map<int, string> myMap = { { 5, "王五" },{ 6, "赵
            六" } };//使用赋值{}
        myMap[2] = "李二";  //使用[ 进行单个插入，若已存在键值，则赋值修改，
            若无则插入。]2
        myMap.insert(pair<int, string>(3, "张三"));//使用和插
            入insertpair
        myMap[1] = "陈一";
        //遍历输出
        auto iter = myMap.begin();
        while (iter != myMap.end()) {
                cout << iter->first << "," << iter->second << endl;
                ++iter;
        }

        //查找元素并输出
        auto iterator = myMap.find(2);//find()返回一个指向的迭代器2
        if (iterator != myMap.end())
                cout << endl << iterator->first << "," << iterator->
                    second << endl;
        system("pause");
        return 0;
}
```

1.1.2 解题代码

```
#include <iostream>
#include <unordered_map>
#include <vector>
using namespace std;
```

```
vector<int> twoSum(vector<int>& nums, int target)
{
    unordered_map<int, int> map;
    vector<int> result={};
    int n = (int)nums.size();
    for(int i = 0; i < n; ++i) {
        auto p = map.find(target-nums[i]);
        if(p != map.end()) {
        result.push_back(p->second);
        result.push_back(i);
        }
    map[nums[i]] = i;
    }
    return result;
}

int main()
{
    vector< int > nums = {2,7,11,15};
    vector<int> result;
    result = twoSum(nums,9);
    cout<<"["<<result[0] << "," <<result[1]<<"]"<<endl;
    return 0;
}
```