Tutorial for basic C/C++ for OpenCV

Image Processing with Deep Learning

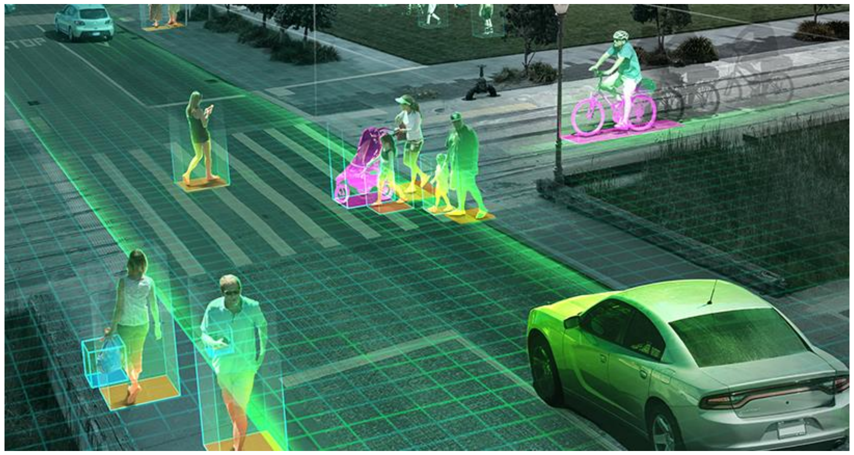
한동대학교
HANDONG GLOBAL UNIVERSITY

OpenCV Introduction

What is Open-source Computer Vision Library?

- The OpenCV Library has >2500 algorithms, extensive documentation and sample code for real-time computer vision.

- You can see basic information about OpenCV at the following sites.
  - Homepage: https://opencv.org
  - Documentation: https://docs.opencv.org
  - Source code: https://github.com/opencv
  - Tutorial: https://docs.opencv.org/master
  - Books: https://opencv.org.books.html



OpenCV

OpenCV Example Code

- **Image File Read / Write / Display**

```
#include <iostream>
#include <opencv2/opencv.hpp>

using namespace std;
using namespace cv;

int main()
{
	/*  read image  */
	String filename1 = "image.jpg";
	Mat img = imread(filename1);
	Mat img_gray = imread("image.jpg", 0);  // read in grayscale

	/*  write image  */
	String filename2 = "writeTest.jpg";
	imwrite(filename2, img);

	/*  display image  */
	namedWindow("image", CV_WINDOW_AUTOSIZE);
	imshow("image", img);

	namedWindow("image_gray", CV_WINDOW_AUTOSIZE);
	imshow("image_gray", img_gray);

	waitKey(0);
}
```

You need to know

namesapce

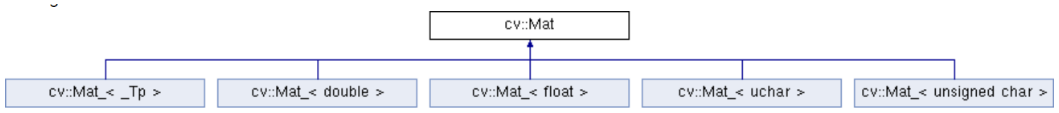
class

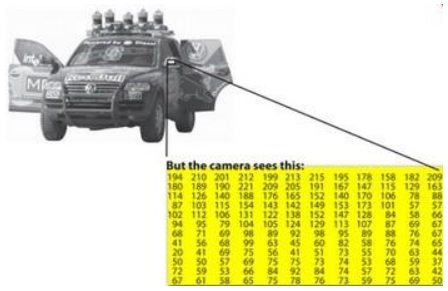
C++ class/syntax
(String, cout, cin..)

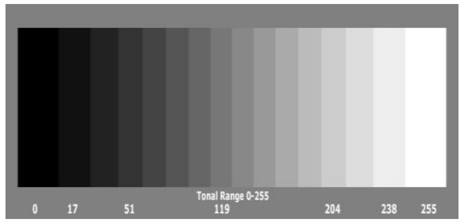
OpenCV Example Code

- **Mat Class**
  - The image data are in forms of 1D, 2D, 3D arrays with values 0~255 or 0~1
  - OpenCV provides the Mat class for operating images

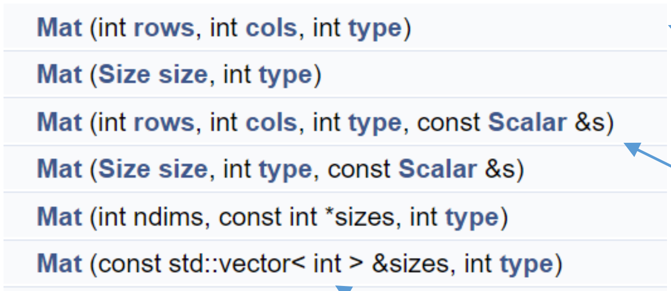
Mat class for various number type(char, integer, double, float)







You need to know



Mat (int rows, int cols, int type)
Mat (Size size, int type)
Mat (int rows, int cols, int type, const Scalar &s)
Mat (Size size, int type, const Scalar &s)
Mat (int ndims, const int *sizes, int type)
Mat (const std::vector< int > &sizes, int type)

overloading

reference

template

C++ for OpenCV

Header file include (C / C++)

description	C programming	C++ programming
- stdio: standard Input and Output Library - stdlib: standard Utility Functions	#include <stdio.h> #include <stdlib.h>	#include <iostream>

A header file containing all C++ streams for performing input/output.

In C++, many functions can be used through iostream's include.

C++ for OpenCV

OpenCV is provided in C++, Python, Java

We will learn how to use OpenCV in

1) C++ (general image processing)
2) Python (for Deep learning processing)

For C++, we need to learn

- Basic C++ syntax
- Class
- Overloading, namespace, template
- Reference

C++

C++ Introduction

- C++ is a general-purpose programming language created by Bjarne Stroustrup as an **extension of the C programming language**.
- C++ is portable and can be used to develop applications that can be adapted to multiple platforms. (cross-platform)

- This course is assumed that you have knowledge of C programming.

- You can see basic C++ tutorials in following site.
- https://www.w3schools.com/cpp/
- https://www.cplusplus.com/doc/tutorial/variables/

Workspace

Project Workspace

Workspace Folder

1) Create the lecture workspace as **C:\Users\yourID\source\repos**

e.g. **C:\Users\ykkim\source\repos**

2) Create sub-directories such as :

- **C:\Users\yourID\source\repos\DLIP**
- **C:\Users\yourID\source\repos\DLIP\Tutorial**
- **C:\Users\yourID\source\repos\DLIP\Include**
- **C:\Users\yourID\source\repos\DLIP\Assignment**
- **C:\Users\yourID\source\repos\DLIP\LAB**
- **C:\Users\yourID\source\repos\DLIP\Image**

Project Workspace

C++ Tutorial Workspace Folder

1) Create this tutorial workspace as

C:\Users\yourID\source\repos\DLIP\Tutorial\Tutorial_Cpp\

Define Function

Define Function

Original code

```
#include <iostream>


int main() {

    int val1 = 11;
    int val2 = 22;

    int sum = val1 + val2;

    std::cout << sum << std::endl;

    system("pause");
    return 0;
}
```

Download Code

Same code with function

Declare function (TU_DLIP.h)

```
#include <iostream>

int sum(int val1, int val2);
```

```
int main() {

    int val1 = 11;
    int val2 = 22;
```

Call function

```
    int out = sum(val1, val2);

    std::cout << out << std::endl;

    system("pause");
    return 0;
}
```

Define function (TU_DLIP.cpp)

```
int sum(int val1, int val2) {
    return val1 + val2;
}
```

Exercise 1

Declare and Define Functions in Header File

1) **Create header files: "TU_DLIP.h", "TU_DLIP.cpp".**
   under C:\Users\yourID\source\repos\DLIP\Tutorial\Tutorial_Cpp\

   Download individual code files from here

   or download zip file

2) **Declare the function in the header file ("TU_DLIP.h").**

```
int sum(int val1, int val2);
```

3) **Define the function in the header file ("TU_DLIP.cpp").**

```
int sum(int val1, int val2){…}
```

4) **Run the main() in** "DLIP_Tutorial_C++_student.cpp" **and print the sum value.**

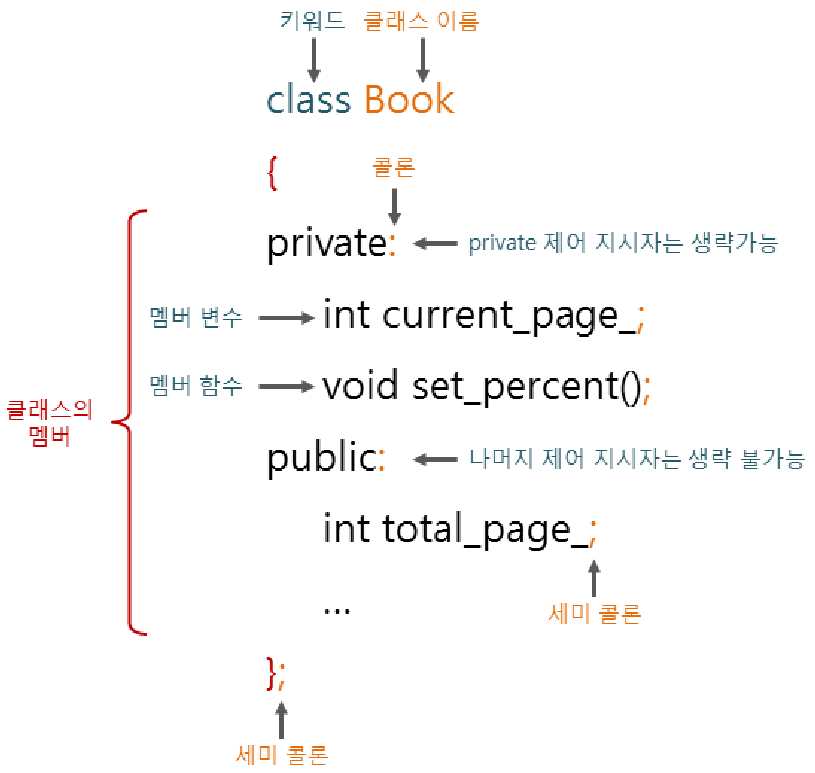
C++ Tutorial:

Class

Class(C++)

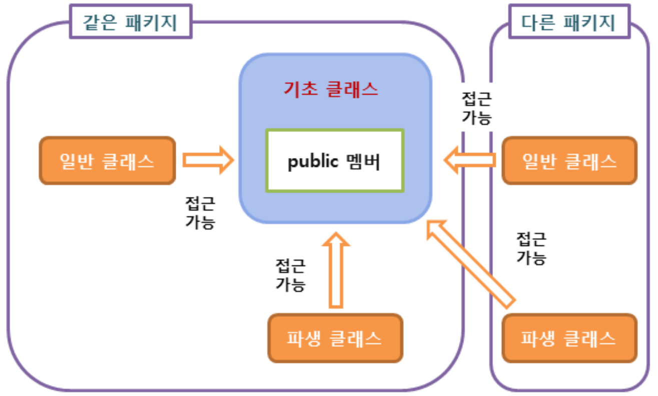
Similar to C structure.

Group variables, functions definition/declaration, other classes

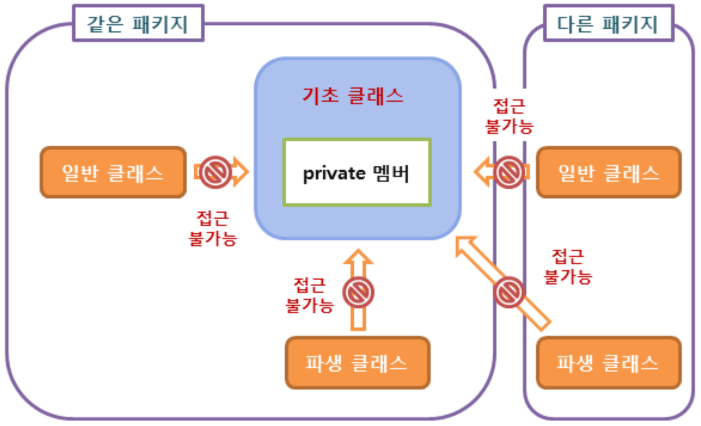


```
class Book
{
private:
    int current_page_;
    void set_percent();
public:
    int total_page_;
    ...
};
```

Public:



Private:



Structure(C) vs. Class(C++)

- Structure: Cannot include functions. Only variables

  Class: Can include variables, functions definition/declaration, other class

Structure (C language)

```
#include <stdio.h>
#include <stdlib.h>

typedef struct {
    char number[20];
    char password[20];
    char name[20];
    int balance;
}Account;
```

Class (C++)

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

class Account{
    public:
    char number[20];
    char password[20];
    char name[20];
    int balance;
    void deposit(int money);
    void withdraw(int money);
};

  void Account::deposit(int money){
        balance+=money;
    }

    void Account::withdraw(int money){
        balance-=money;
    }
```

Class definition

Can include functions

Class function definition

Class

Constructor

Special method automatically called when an object of a class is created

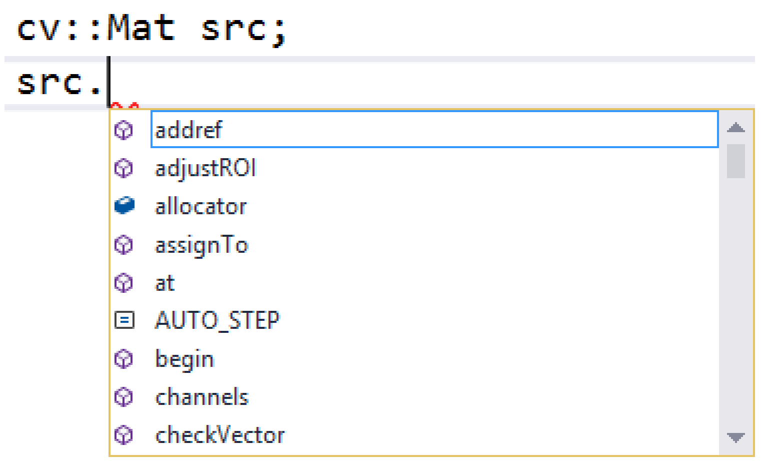
1) Use the **same** name as the class, followed by parentheses `()`:
2) it is always `public`
3) It does not have any return value

```
class MyNum {
  public:
  MyNum();  //  Constructor 1
  MyNum(int x); // Constructor 2

  int num;
};


// Class Constructor1
MyNum::MyNum(){}

// Class Constructor2
MyNum::MyNum(int x)
{ num = x; }
```

```
int main()
{
   // Creating object by constructor1
   MyNum mynum;
   mynum.num = 10;

   // Creating object by constructor2
   MyNum mynum2(10);
}
```

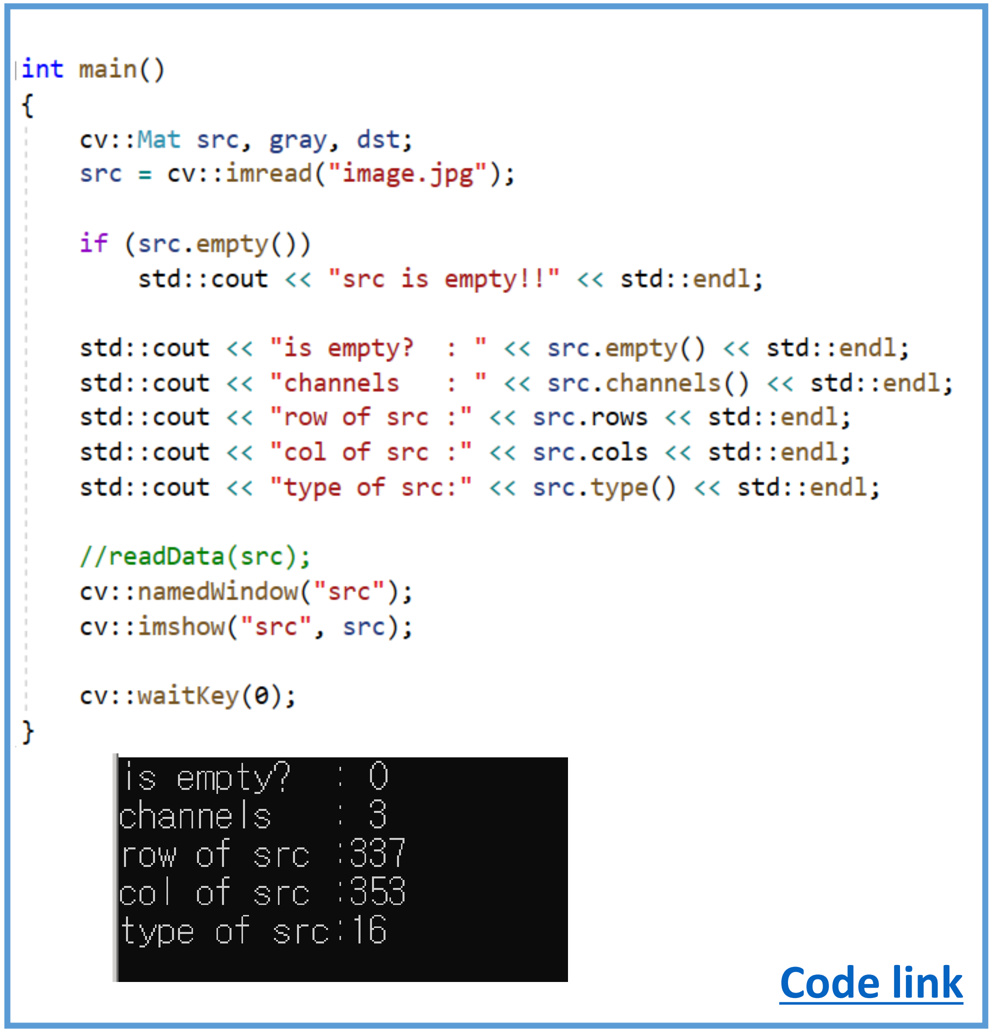
Class

OpenCV Mat is a class object



```
cv::Mat src;
src.
```

Using member variables/methods



```
int main()
{
    cv::Mat src, gray, dst;
    src = cv::imread("image.jpg");

    if (src.empty())
        std::cout << "src is empty!!" << std::endl;

    std::cout << "is empty?  : " << src.empty() << std::endl;
    std::cout << "channels   : " << src.channels() << std::endl;
    std::cout << "row of src :" << src.rows << std::endl;
    std::cout << "col of src :" << src.cols << std::endl;
    std::cout << "type of src:" << src.type() << std::endl;

    //readData(src);
    cv::namedWindow("src");
    cv::imshow("src", src);

    cv::waitKey(0);
}
```

```
is empty?  : 0
channels   : 3
row of src :337
col of src :353
type of src:16
```

Code link

Exercise 2

Create a Class 'myNum'

1) Declare a class member named as 'myNum' in "DLIP_Tutorial_C++_student.cpp"

- Constructor:	MyNum()	
- Member variables:	val1, val2	// integer type
- Member functions:	int sum()	// returns the sum of val1 and val2
- Member functions:	void print()	// prints values of "val1, val2, and sum"

2) Split the declaration and definitions of this class:

"TU_DLIP.h" and **"TU_DLIP.cpp"**

Solution Code

C++ Tutorial:

Namespace

Namespace

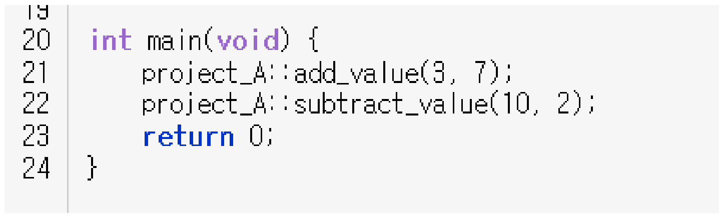
A namespace provides a scope to the identifiers (the names of types, functions, variables, etc) inside it.

Uses ‘::’ as scope resolution operator

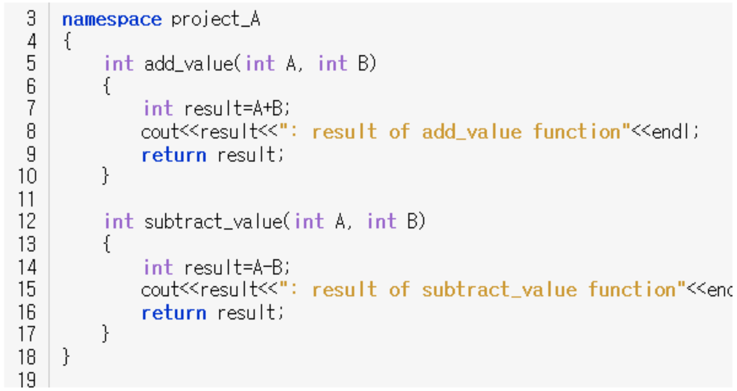
Use “namespace” in order to avoid collision using functions with the same name

e.g 김한동 → 15학번::김한동, 16학번::김한동

Method 1) calling specific function(recommended)

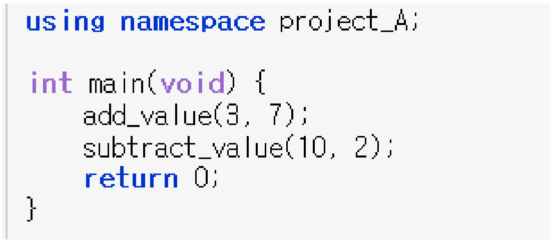


```cpp
int main(void) {
    project_A::add_value(3, 7);
    project_A::subtract_value(10, 2);
    return 0;
}
```



```cpp
namespace project_A
{
    int add_value(int A, int B)
    {
        int result=A+B;
        cout<<result<<": result of add_value function"<<endl;
        return result;
    }

    int subtract_value(int A, int B)
    {
        int result=A-B;
        cout<<result<<": result of subtract_value function"<<en
        return result;
    }
}
```

Method 2) calling all function in the namespace



```cpp
using namespace project_A;

int main(void) {
    add_value(3, 7);
    subtract_value(10, 2);
    return 0;
}
```

※ std::cout, std::cin, std::endl are also defined in “iostream”

// Method 1
std::cout << “print this” << std::endl;

// Method 2
using namespace std
cout << “print this” << endl;

Namespace

Namespace for OpenCV **cv::___**

`cv::Mat img;` ➔ create variable "img" to contain image

`img = cv::imread("file.jpg");` ➔ Read image file and save it in 'img'

Method 1) recommended

```
#include <opencv.hpp>
#include <iostream>



void main(){
  cv::Mat src, gray, dst;
  src = cv::imread("testImage.jpg");

  if (src.empty())
    std::cout << "src is empty!!" << std::endl;

  cv::namedWindow("src");
  cv::imshow("src", src);

  cv::waitKey(0);
}
```

Code link

Method 2)

```
#include <opencv.hpp>
#include <iostream>

using namespace std;
using namespace cv;

void main(){
  Mat src, gray, dst;
  src = imread("testImage.jpg");

  if (src.empty())
    cout << "src is empty!!" << endl;

  namedWindow("src");
  imshow("src", src);

  waitKey(0);
}
```

Code link

Exercise

Create another Class ‘myNum’

1) Declare class member variables like this in “DLIP_Tutorial_C++_namespace_student.cpp”:

Constructor / val1 / val2 / val3 / sum / print

- val1, val2, val3: member variable of integer type
- sum(): member function that returns the sum of val1, val2, and val3
- print(): member function that prints val1, val2, val3, and sum

2) Build

3) Use namespace to clearly identify two classes

- First ‘myNum’ class: namespace name ‘proj_A’
- Second ‘myNum’ class : namespace name ‘proj_B’

4) Build and compare