Color normalization using the *White Patch* method.

Goals:

- Learning about the color unbalancing problem due to illumination changes or hardware differences.
- Applying a pixel-wise image processing method.
- Get to know methods to obtain image statistics.
- Obtaining a deeper understanding of OpenCV: Mat, imread, imwrite, traversing an image and selecting color channels. Defining a GUI.

Description:

The white patch method aims at correcting color deficiencies in illumination that makes white light to appear as a different color.
To prevent that problem, find the brightest image pixel and calculate the transform that makes it white, i.e., RGB= (255,255,255). Then, scale all image points accordingly. Please notice that the computed correction might make some pixels go out of range. In that case, saturate to the maximum value 255.

In the simplest implementation, your program work as a command line as:

```
wpCorrection  img-in img-out [-p <float>=0.0] [-i]
```

where:

- ESC key should finish the program.

Mandatory requirements (Up to 6 points):

- Correcting color using a single pixel.

Optional requirements (+2 each):

- Correcting color using the average of the % brightest colors:
    - Add command line option -p to indicate the percentage of (0,1) of the brightest points used to compute the mean value employed to do the correction. The value 0 means using a single point.
- Interactive mode: use sliders to adjust the values p

Add -i command line option means interactive mode. In that mode, the program shows the original and corrected images, and with a slider, the user must be able to change the value of the parameter p.

How to easily use command line arguments

```
//Class to manage command line arguments.
// Copy in your program!
class CmdLineParser{
   int argc;
   char **argv;
public:
   CmdLineParser(int _argc,char **_argv):argc(_argc),argv(_argv){}
   bool operator[] ( string param ) {int idx=-1;  for ( int i=0; i<argc &&
idx==-1; i++ ) if ( string ( argv[i] ) ==param ) idx=i;    return ( idx!=-1 ) ;
}
   string operator()(string param,string defvalue="-1"){int idx=-1;    for ( int
i=0; i<argc && idx==-1; i++ ) if ( string ( argv[i] ) ==param ) idx=i; if (
idx==-1 ) return defvalue;   else  return ( argv[  idx+1] ); }
};

//Example of use
int main(int argc,char **argv){

   CmdLineParser cml(argc,argv);
   //check if a command is present
   if (cml["-p"]){
       cout<<"-p option is in the command line";
       float f= std::stof( cml("-p"));//convert to float
       int i= std::stoi( cml("-p"));//convert to int
   }
   //used with default values
   float f=std::stof(cml("-p","1"));//if -p is not, then, return 1
}
```