```cpp
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <string>
#include <fstream>
#include <vector>
#include <opencv2/opencv.hpp>


using namespace std;
using namespace cv;


int findContrast(IplImage *, int *, int);


int main(int argc, char ** argv) {
    int dep = 256;

    vector <int> C;

    string line;
    ifstream out(argv[1]);

    int tot = 0, cnt = 0;

    while(getline(out, line)) {
        const char * imgName = line.c_str();
        IplImage * img = cvLoadImage(imgName, CV_LOAD_IMAGE_GRAYSCALE);
        int hist[dep];
        int c = findContrast(img, hist, cnt);
        C.push_back(c);
        tot += c;
        cnt++;
        cvReleaseImage(&img);
        cout << cnt << endl;
    }

    cout << cnt << " images..." << endl;

    int M = tot / cnt;
    cout << "average contrast: " << M << endl;

    double SD = 0;

    for (int i = 0; i < C.size(); i++)
        SD += pow(C[i] - M, 2);

    SD = sqrt(SD / C.size());

    cout << "Standard Deviation: " << SD << endl;

    out.close();

    return 0;
}
```

```
59
60  /// ////////////////////////////////////////////////////////////////////
61  /// ////////////////////////////////////////////////////////////////////
62
63
64      /*********************************************************************
65      * Just a support function to 'drawHistogram', 'drawHistChannels', and        *
66      * 'getHistogram'.                                                            *
67      *********************************************************************/
68  void func(uchar c, int * h) {
69      unsigned x = c;
70      (*(h + x))++;
71  }
72
73
74  /// ////////////////////////////////////////////////////////////////////
75  /// ////////////////////////////////////////////////////////////////////
76
77
78  int findContrast(IplImage * gray, int hist [], int l) {
79      int st = 4;       // To separate lines in the histograms, for better look ;)
80      int dep = 256;
81          /// Initialize histograms
82      for (int i = 0; i < dep; i++)
83          hist[i] = 0;
84
85          /// Calculate histogram
86      for(int i = 0; i < gray->height; i++) {
87          char * ptr = gray->imageData + i*gray->widthStep;
88          for(int j = 0; j < gray->width; j++) {
89                  func((uchar)(*ptr), hist);
90                  ptr++;
91          }
92      }
93
94          /// Create image for the histogram
95      IplImage * his = cvCreateImage(cvSize(st*dep,600), IPL_DEPTH_8U, 1);
96      cvSet(his, 0);                    // Initialize image (all black)
97      his->origin = IPL_ORIGIN_BL;      // Set the origin in the bottom left corner
98
99
100     for (int i = 0; i < dep; i++)
101         if (hist[i] != 0)
102             cvLine(his, cvPoint(i*st, 0), cvPoint(i*st, hist[i]/10), 150, 1, 4);
103
104     //cvSaveImage("Histogram.png", his);
105
106     int cnt = 0, TOT = (gray->height * gray->width);
107     int p5, p95;
108     bool f1 = true, f2 = true;
109     for(int i = 0; i < 256; i++) {
110         cnt += hist[i];
111         if(f1 && ((cnt * 100) / TOT) >= 5) {
112             p5 = i;
113             f1 = false;
114             cout << endl;
115             cout << ((cnt * 100) / TOT) << " percentile: " << i << endl;
116         }
```

```
117
118         if(f2 && ((cnt * 100) / TOT) >= 95) {
119             p95 = i;
120             f2 = false;
121             cout << ((cnt * 100) / TOT) << " percentile: " << i << endl;
122         }
123
124     }
125     /*
126     IplImage * Chis = cvCreateImage(cvGetSize(his), IPL_DEPTH_8U, 3);
127     cvCvtColor(his, Chis, COLOR_GRAY2RGB);
128     cvFlip(Chis, Chis, 0);
129
130     cvLine(Chis, cvPoint(p5*st, 0), cvPoint(p5*st, 600), cvScalar(0,0,255), 2, 4);
131     cvLine(Chis, cvPoint(p95*st, 0), cvPoint(p95*st, 600), cvScalar(0,255,0), 2, 4);
132
133     string name = "Histogram";
134     string tr = "track";
135     string sufix = ".png";
136     char numstr[21];
137     string result = "test";
138
139     sprintf(numstr, "%d", 1);
140     result = name + numstr + sufix;
141     const char * pd = result.c_str();
142     cvSaveImage(pd, his);
143     result = tr + numstr + sufix;
144     pd = result.c_str();
145     cvSaveImage(pd, Chis);
146     cvReleaseImage(&Chis);
147
148     cvReleaseImage(&his);*/
149
150     cout << "contrast: " << (p95 - p5) << endl;
151     cout << endl;
152
153     return (p95 - p5);
154 }
155
156
```