```
#pragma once
#define ENHANCE_VERSION 1

#include "ImageProcessing.h"
#include "../SoilMath/SoilMath.h"

using namespace std;
using namespace SoilMath;

namespace Vision
{
    class Enhance:
        public ImageProcessing
    {
    private:
        void CalculateSumOfNeighboringPixels(uchar *O, int i, int hKsize, int nCols, uint32_t &sum);
        float CalculateStdOfNeighboringPixels(uchar *O,  int i, int hKsize, int nCols, int noNeighboursPix, float mean);
    public:
        /*! Enumerator indicating the requested enhancement operation*/
        enum EnhanceOperation
        {
            _AdaptiveContrastStretch,   /*!< custom adaptive contrast stretch operation*/
            _Blur,                      /*!< Blur operation*/
            _HistogramEqualization      /*!< Histogram equalization*/
        };

        Enhance();
        Enhance(const Mat& src);
        Enhance(const Mat& src, Mat& dst, uint8_t kernelsize = 9, float factor = 1.0, EnhanceOperation operation = _Blur);

        ~Enhance();

        void AdaptiveContrastStretch(uint8_t kernelsize, float factor, bool chain = false);
        void AdaptiveContrastStretch(const Mat& src, Mat& dst, uint8_t kernelsize, float factor);

        void Blur(uint8_t kernelsize, bool chain = false);
        void Blur(const Mat& src, Mat& dst, uint8_t kernelsize);

        void HistogramEqualization(bool chain = false);
        void HistogramEqualization(const Mat& src, Mat& dst);
    };
}
```