```
1  #pragma once
2  #define SEGMENT_VERSION 1
3
4  #include <vector>
5  #include <queue>
6  #include <string>
7  #include <stdint.h>
8  #include <iostream>
9
10 #include "opencv2/imgproc/imgproc.hpp"
11
12 #include "ImageProcessing.h"
13 #include "MorphologicalFilter.h"
14 #include "../SoilMath/SoilMath.h"
15
16 namespace Vision
17 {
18     class Segment :
19         public ImageProcessing
20     {
21     public:
22         /*! Enumerator to indicate what kind of object to extract  */
23         enum TypeOfObjects
24         {
25             Bright,     /*!< Enum value Bright object */
26             Dark        /*!< Enum value Dark object. */
27         };
28
29         /*! Enumerator to indicate how the pixel correlate between each other in a blob*/
30         enum Connected
31         {
32             Four,   /*!< Enum Four connected, relation between Center, North, East, South and West*/
33             Eight   /*!< Enum Eight connected, relation between Center, North, NorthEast, East, SouthEast, South, SouthWest, West and
                NorthWest */
34         };
35
36         cv::Mat LabelledImg;    /*!< Image with each individual blob labeled with a individual number */
37         uint16_t MaxLabel = 0;  /*!< Maximum labels found in the labelled image*/
38         uint16_t noOfFilteredBlobs = 0; /*!< Total numbers of blobs that where filtered beacuse the where smaller than the
              minBlobArea*/
39
```

```
40      /*! Coordinates for the region of interest*/
41      typedef struct Rect
42      {
43          uint16_t leftX;     /*!< Left X coordinate*/
44          uint16_t leftY;     /*!< Left Y coordinate*/
45          uint16_t rightX;    /*!< Right X coordinate*/
46          uint16_t rightY;    /*!< Right Y coordinate*/
47      } Rect;
48
49      /*! Individual blob*/
50      typedef struct Blob
51      {
52          ushort Label;   /*!< ID of the blob*/
53          cv::Mat Img;    /*!< BW image of the blob all the pixel belonging to the blob are set to 1 others are 0*/
54          Rect ROI;       /*!< Coordinates for the blob in the original picture*/
55          cv::Rect cvROI; /*!< Coordinates for the blob in the original picture as a cv::Rect*/
56      } Blob;
57
58      vector<Blob> BlobList; /*!< vector with all the individual blobs*/
59
60      ucharStat_t OriginalImgStats; /*!< Statistical data from the original image*/
61      uint8_t ThresholdLevel = 0;             /*!< Current calculated threshold level*/
62
63      Segment();
64      Segment(const Mat &src);
65      ~Segment();
66
67      void ConvertToBW(TypeOfObjects Typeobjects);
68      void ConvertToBW(const Mat &src, Mat &dst, TypeOfObjects Typeobjects);
69
70      void GetEdges(bool chain = false, Connected conn = Eight);
71      void GetEdges(const Mat &src, Mat &dst, bool chain = false, Connected conn = Eight);
72
73      void GetEdgesEroding(bool chain = false);
74
75      void GetBlobList(bool chain = false, Connected conn = Eight);
76      void GetBlobList(const Mat &src, Mat &dst, bool chain = false, Connected conn = Eight);
77
78      void Threshold(uchar t, TypeOfObjects Typeobjects);
79
80      void LabelBlobs(bool chain = false, uint16 t minBlobArea = 25, Connected conn = Eight);
```

```
81
82          void RemoveBorderBlobs(bool chain = false, Connected conn = Eight);
83
84          void FillHoles(bool chain = false);
85
86      private:
87
88          uint8_t GetThresholdLevel(TypeOfObjects TypeObject);
89          void SetBorder(uchar *P, uchar setValue);
90          void FloodFill(uchar *O, uchar *P, uint16_t x, uint16_t y, uchar fillValue, uchar OldValue);
91          void makeConsecutive(uint16_t LastLabelUsed, uint16_t * tempLUT, uint16_t * &LUT_newVal);
92      };
93  }
```