

cellUview

1.0.0

Generated by Doxygen 1.9.5

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 Camera Class Reference	7
4.1.1 Member Function Documentation	7
4.1.1.1 getParamLabel()	7
4.1.1.2 receiveFrame()	8
4.1.1.3 updateSettings()	8
4.2 cellUviewWelcome Class Reference	8
4.2.1 Detailed Description	8
4.3 contrastEnhancement Class Reference	8
4.3.1 Member Function Documentation	9
4.3.1.1 getParamLabel()	9
4.3.1.2 receiveFrame()	9
4.3.1.3 updateSettings()	9
4.4 dilation Class Reference	10
4.4.1 Member Function Documentation	10
4.4.1.1 getParamLabel()	10
4.4.1.2 receiveFrame()	10
4.4.1.3 updateSettings()	10
4.5 edgeDetection Class Reference	11
4.5.1 Member Function Documentation	11
4.5.1.1 getParamLabel()	11
4.5.1.2 receiveFrame()	11
4.5.1.3 updateSettings()	12
4.6 erosion Class Reference	12
4.6.1 Member Function Documentation	12
4.6.1.1 getParamLabel()	12
4.6.1.2 receiveFrame()	12
4.6.1.3 updateSettings()	13
4.7 flatFieldCorrect Class Reference	13
4.7.1 Member Function Documentation	13
4.7.1.1 getParamLabel()	13
4.7.1.2 receiveFrame()	14
4.7.1.3 updateSettings()	14
4.8 frame Class Reference	14

4.9 Gallery Class Reference	15
4.10 grayScale Class Reference	15
4.10.1 Member Function Documentation	15
4.10.1.1 getParamLabel()	15
4.10.1.2 receiveFrame()	16
4.10.1.3 updateSettings()	16
4.11 Gui Class Reference	16
4.11.1 Detailed Description	16
4.11.2 Constructor & Destructor Documentation	17
4.11.2.1 Gui()	17
4.11.3 Member Function Documentation	18
4.11.3.1 receiveFrame()	18
4.11.3.2 SetVisible()	18
4.12 imageProcessor Class Reference	18
4.12.1 Member Function Documentation	19
4.12.1.1 receiveFrame()	19
5 File Documentation	21
5.1 camera.h	21
5.2 cellUviewWelcome.h	21
5.3 contrastEnhancement.h	22
5.4 dilation.h	22
5.5 edgeDetection.h	23
5.6 erosion.h	23
5.7 flatFieldCorrect.h	24
5.8 frame.h	24
5.9 gallery.h	25
5.10 grayScale.h	26
5.11 gui.h	26
5.12 imageProcessor.h	27
Index	29

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

cellUviewWelcome	8
frame	14
Gallery	15
imageProcessor	18
Camera	7
Gui	16
contrastEnhancement	8
dilation	10
edgeDetection	11
erosion	12
flatFieldCorrect	13
grayScale	15
QWidget	
Gui	16

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Camera	7
cellUviewWelcome	8
contrastEnhancement	8
dilation	10
edgeDetection	11
erosion	12
flatFieldCorrect	13
frame	14
Gallery	15
grayScale	15
Gui	16
imageProcessor	18

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

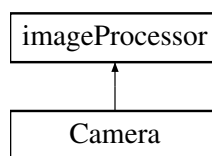
src/ camera.h	21
src/ cellUviewWelcome.h	21
src/ contrastEnhancement.h	22
src/ dilation.h	22
src/ edgeDetection.h	23
src/ erosion.h	23
src/ flatFieldCorrect.h	24
src/ frame.h	24
src/ gallery.h	25
src/ grayScale.h	26
src/ gui.h	26
src/ imageProcessor.h	27

Chapter 4

Class Documentation

4.1 Camera Class Reference

Inheritance diagram for Camera:



Public Member Functions

- void **start** (int deviceID=0, int apiID=0)
- void **stop** ()
- bool **getIsOn** ()
- void **captureMetadata** ()
- void **setExposure** (int)
- std::string **getParamLabel** ()
- void **updateSettings** (std::map< std::string, std::string >)
- void **receiveFrame** (frame newFrame)

Additional Inherited Members

4.1.1 Member Function Documentation

4.1.1.1 getParamLabel()

```
std::string Camera::getParamLabel ( ) [inline], [virtual]
```

Implements [imageProcessor](#).

4.1.1.2 receiveFrame()

```
void Camera::receiveFrame (
    frame newFrame ) [inline], [virtual]
```

Implements [imageProcessor](#).

4.1.1.3 updateSettings()

```
void Camera::updateSettings (
    std::map< std::string, std::string > metadata ) [virtual]
```

Implements [imageProcessor](#).

The documentation for this class was generated from the following files:

- src/camera.h
- src/camera.cpp

4.2 cellUviewWelcome Class Reference

```
#include <cellUviewWelcome.h>
```

Static Public Member Functions

- static void **welcomeMessage** ()

4.2.1 Detailed Description

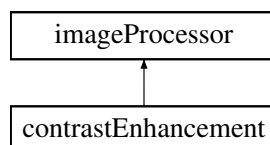
Prints welcome message ASCII art on application start.

The documentation for this class was generated from the following file:

- src/cellUviewWelcome.h

4.3 contrastEnhancement Class Reference

Inheritance diagram for contrastEnhancement:



Public Member Functions

- void [receiveFrame](#) ([frame](#) newFrame)
- void [updateThreshold](#) (int value)
- void [updateSettings](#) (std::map< std::string, std::string >)
- std::string [getParamLabel](#) ()

Public Attributes

- float [threshold](#) = 0
- float [sliderThreshold](#) = 10

Additional Inherited Members

4.3.1 Member Function Documentation

4.3.1.1 [getParamLabel\(\)](#)

```
std::string contrastEnhancement::getParamLabel ( ) [inline], [virtual]
```

Implements [imageProcessor](#).

4.3.1.2 [receiveFrame\(\)](#)

```
void contrastEnhancement::receiveFrame (
    frame newFrame ) [virtual]
```

Implements [imageProcessor](#).

4.3.1.3 [updateSettings\(\)](#)

```
void contrastEnhancement::updateSettings (
    std::map< std::string, std::string > metadata ) [virtual]
```

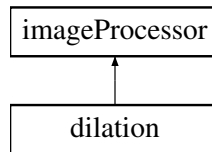
Implements [imageProcessor](#).

The documentation for this class was generated from the following files:

- src/contrastEnhancement.h
- src/contrastEnhancement.cpp

4.4 dilation Class Reference

Inheritance diagram for dilation:



Public Member Functions

- void [receiveFrame](#) ([frame](#) newFrame)
- std::string [getParamLabel](#) ()
- void [updateSettings](#) (std::map< std::string, std::string >)

Additional Inherited Members

4.4.1 Member Function Documentation

4.4.1.1 getParamLabel()

```
std::string dilation::getParamLabel ( ) [inline], [virtual]
```

Implements [imageProcessor](#).

4.4.1.2 receiveFrame()

```
void dilation::receiveFrame (
    frame newFrame ) [virtual]
```

Implements [imageProcessor](#).

4.4.1.3 updateSettings()

```
void dilation::updateSettings (
    std::map< std::string, std::string > metadata ) [virtual]
```

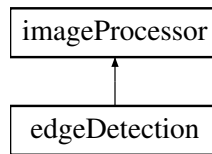
Implements [imageProcessor](#).

The documentation for this class was generated from the following files:

- src/dilation.h
- src/dilation.cpp

4.5 edgeDetection Class Reference

Inheritance diagram for edgeDetection:



Public Member Functions

- void `receiveFrame` (`frame` newFrame)
- void `updateThreshold` (int value)
- void `updateSettings` (std::map< std::string, std::string >)
- std::string `getParamLabel` ()

Public Attributes

- int `threshold` = 0
- int `sliderThreshold` = 100

Additional Inherited Members

4.5.1 Member Function Documentation

4.5.1.1 getParamLabel()

```
std::string edgeDetection::getParamLabel ( ) [inline], [virtual]
```

Implements `imageProcessor`.

4.5.1.2 receiveFrame()

```
void edgeDetection::receiveFrame (
    frame newFrame ) [virtual]
```

Implements `imageProcessor`.

4.5.1.3 updateSettings()

```
void edgeDetection::updateSettings (
    std::map< std::string, std::string > metadata ) [virtual]
```

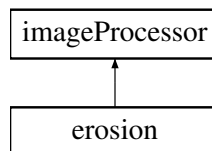
Implements [imageProcessor](#).

The documentation for this class was generated from the following files:

- src/edgeDetection.h
- src/edgeDetection.cpp

4.6 erosion Class Reference

Inheritance diagram for erosion:



Public Member Functions

- void [receiveFrame](#) ([frame](#) newFrame)
- std::string [getParamLabel](#) ()
- void [updateSettings](#) (std::map< std::string, std::string >)

Additional Inherited Members

4.6.1 Member Function Documentation

4.6.1.1 getParamLabel()

```
std::string erosion::getParamLabel ( ) [inline], [virtual]
```

Implements [imageProcessor](#).

4.6.1.2 receiveFrame()

```
void erosion::receiveFrame (
    frame newFrame ) [virtual]
```

Implements [imageProcessor](#).

4.6.1.3 updateSettings()

```
void erosion::updateSettings (
    std::map< std::string, std::string > metadata ) [virtual]
```

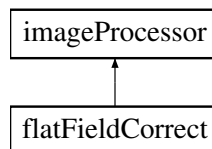
Implements [imageProcessor](#).

The documentation for this class was generated from the following files:

- src/erosion.h
- src/erosion.cpp

4.7 flatFieldCorrect Class Reference

Inheritance diagram for flatFieldCorrect:



Public Member Functions

- void [receiveFrame](#) ([frame](#) newFrame)
- std::string [getParamLabel](#) ()
- void [setUpdateFlag](#) ()
- void [updateSettings](#) (std::map< std::string, std::string >)

Additional Inherited Members

4.7.1 Member Function Documentation

4.7.1.1 getParamLabel()

```
std::string flatFieldCorrect::getParamLabel ( ) [inline], [virtual]
```

Implements [imageProcessor](#).

4.7.1.2 receiveFrame()

```
void flatFieldCorrect::receiveFrame (
    frame newFrame ) [virtual]
```

Implements [imageProcessor](#).

4.7.1.3 updateSettings()

```
void flatFieldCorrect::updateSettings (
    std::map< std::string, std::string > metadata ) [virtual]
```

Implements [imageProcessor](#).

The documentation for this class was generated from the following files:

- src/flatFieldCorrect.h
- src/flatFieldCorrect.cpp

4.8 frame Class Reference

Public Member Functions

- **frame** (cv::Mat matIn)
- void **copyFrom** ([frame](#) *copyFrom)
- void **setParameter** (std::string, std::string)
- std::string **encodeMetadata** ()
- std::string **getParam** (std::string)
- **frame** ([frame](#) const &)=default
- int **getParamSize** ()

Public Attributes

- cv::Mat **image**
- bool **doMeta** = false

The documentation for this class was generated from the following files:

- src/frame.h
- src/frame.cpp

4.9 Gallery Class Reference

Public Member Functions

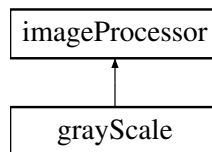
- void **captureFrame** ([frame](#), bool flatfield=false, int flatFieldCounter=0)
- void **updateImgName** (std::string)
- void **updateIndex** ()
- std::map< std::string, std::string > **getMetadata** (std::string="")
- std::string **getPathname** ()
- std::string **getCaptureFname** ()

The documentation for this class was generated from the following files:

- src/gallery.h
- src/gallery.cpp

4.10 grayScale Class Reference

Inheritance diagram for grayScale:



Public Member Functions

- void [receiveFrame](#) ([frame](#))
- std::string [getParamLabel](#) ()
- void [updateSettings](#) (std::map< std::string, std::string >)

Additional Inherited Members

4.10.1 Member Function Documentation

4.10.1.1 getParamLabel()

```
std::string grayScale::getParamLabel ( ) [inline], [virtual]
```

Implements [imageProcessor](#).

4.10.1.2 receiveFrame()

```
void grayScale::receiveFrame (
    frame newFrame ) [virtual]
```

Implements [imageProcessor](#).

4.10.1.3 updateSettings()

```
void grayScale::updateSettings (
    std::map< std::string, std::string > metadata ) [virtual]
```

Implements [imageProcessor](#).

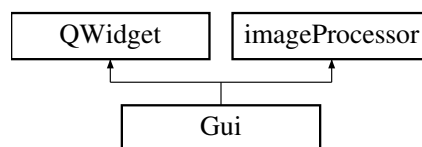
The documentation for this class was generated from the following files:

- src/grayScale.h
- src/grayScale.cpp

4.11 Gui Class Reference

```
#include <gui.h>
```

Inheritance diagram for Gui:



Public Member Functions

- void [receiveFrame](#) ([frame](#) newFrame)
- [Gui](#) (QMainWindow *, Ui_GUI *, [Gallery](#) *, std::vector< [imageProcessor](#) * > &)
- void [SetVisible](#) (bool visible)

Additional Inherited Members

4.11.1 Detailed Description

A class which handles GUI connections and functionality.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 Gui()

```
Gui::Gui (
    QMainWindow * win,
    Ui_GUI * ui_win,
    Gallery * galleryIn,
    std::vector< imageProcessor * > & blocksIn )
```

Constructor to initialise the GUI and set connections

Parameters

<i>win</i>	points to QMainWindow
<i>ui_win</i>	points to Ui_GUI
<i>gallery↔ In</i>	points to Gallery instance
<i>blocksIn</i>	is a std::vector of the image processing blocks

4.11.3 Member Function Documentation

4.11.3.1 receiveFrame()

```
void Gui::receiveFrame (
    frame newFrame ) [virtual]
```

Function to recieve callbacks frames from image processor blocks

Parameters

<i>newFrame</i>	frame structure from processing block via callback interface
-----------------	--

Implements [imageProcessor](#).

4.11.3.2 SetVisible()

```
void Gui::SetVisible (
    bool visible )
```

Sets UI visibility

Parameters

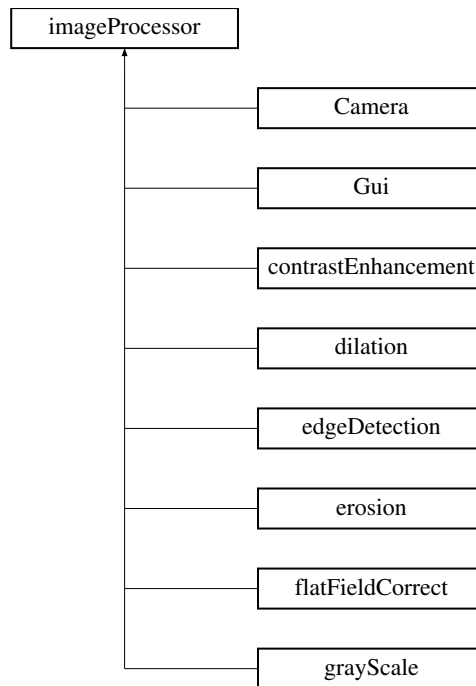
<i>visible</i>	true to make visible
----------------	----------------------

The documentation for this class was generated from the following files:

- src/gui.h
- src/gui.cpp

4.12 imageProcessor Class Reference

Inheritance diagram for imageProcessor:



Public Member Functions

- virtual void [receiveFrame](#) ([frame](#) newFrame)=0
- virtual void **updateSettings** (std::map< std::string, std::string >)=0
- virtual std::string **getParamLabel** ()=0
- void **registerCallback** ([imageProcessor](#) *cb)
- bool **toggleEnable** ()
- bool **getEnabled** ()

Protected Attributes

- [imageProcessor](#) * **frameCb** = nullptr
- bool **enabled** = false

4.12.1 Member Function Documentation

4.12.1.1 receiveFrame()

```
virtual void imageProcessor::receiveFrame (
    frame newFrame ) [pure virtual]
```

Implemented in [Gui](#).

The documentation for this class was generated from the following file:

- src/imageProcessor.h

Chapter 5

File Documentation

5.1 camera.h

```
1 #ifndef CELLUVIEW_CAMERA_H
2 #define CELLUVIEW_CAMERA_H
3
4
5 #include <opencv2/core.hpp>
6 #include <opencv2/videoio.hpp>
7 #include <iostream>
8 #include <stdlib.h>
9 #include <thread>
10 #include "imageProcessor.h"
11
12 #include "frame.h"
13
14
15 class Camera: public imageProcessor{
16 public:
17     Camera() = default;
18     void start(int deviceID = 0, int apiID=0);
19     void stop();
20     //void receiveFrame(frame newFrame);
21     bool getIsOn();
22     //void registerCallback(imageProcessor*);
23     void captureMetadata();
24     void setExposure(int);
25     std::string getParamLabel(){return paramLabel;};
26     void updateSettings(std::map<std::string, std::string>);
27     void receiveFrame(frame newFrame){return;};
28
29 private:
30     frame f;
31     void postFrame();
32     void threadLoop();
33     cv::VideoCapture videoCapture;
34     std::thread cameraThread;
35     std::string exposureState = "OFF";
36     bool isOn = false;
37     std::string paramLabel = "exposure";
38     //imageProcessor* frameCb = nullptr;
39     bool doMeta = false;
40
41
42 };
43
44 #endif // CELLUVIEW_CAMERA_H
```

5.2 cellUviewWelcome.h

```
1 #ifndef CELLUVIEW_WELCOME_H
2 #define CELLUVIEW_WELCOME_H
3 #include <stdlib.h>
4 #include <stdio.h>
5 #include "iostream"
6
11 class cellUviewWelcome {
```



```

17     dilation() = default;
18     void receiveFrame(frame newFrame);
19     std::string getParamLabel(){return paramLabel;};
20     void updateSettings(std::map<std::string, std::string>);
21
22 private:
23     //add any other methods here
24     void dilate(frame); //image dilation
25     std::string paramLabel = "dilation";
26 };
27 #endif // CELLUVIEW_DILATION_H

```

5.5 edgeDetection.h

```

1
2 #ifndef CELLUVIEW_EDGE_DETECTION_H
3 #define CELLUVIEW_EDGE_DETECTION_H
4
5
6 #include <opencv2/core.hpp>
7 #include <opencv2/videoio.hpp>
8 #include <iostream>
9 #include <stdlib.h>
10 #include <thread>
11 #include "imageProcessor.h"
12
13
14 class edgeDetection: public imageProcessor{
15
16 public:
17     edgeDetection() = default;
18     int threshold= 0;
19     int sliderThreshold = 100; //slider threshold used for metadata purposes
20     void receiveFrame(frame newFrame);
21     void updateThreshold(int value);
22     void updateSettings(std::map<std::string, std::string>);
23     std::string getParamLabel(){return paramLabel;};
24
25 private:
26     //add any other methods here
27     void enhanceEdge(frame); //edge detection
28     std::string paramLabel = "edgeThreshold";
29 };
30 #endif // CELLUVIEW_EDGE_DETECTION_H

```

5.6 erosion.h

```

1
2 #ifndef CELLUVIEW_EROSION_H
3 #define CELLUVIEW_EROSION_H
4
5
6 #include <opencv2/core.hpp>
7 #include <opencv2/videoio.hpp>
8 #include <iostream>
9 #include <stdlib.h>
10 #include <thread>
11 #include "imageProcessor.h"
12
13
14 class erosion: public imageProcessor{
15 public:
16     erosion() = default;
17     void receiveFrame(frame newFrame);
18     std::string getParamLabel(){return paramLabel;};
19
20     void updateSettings(std::map<std::string, std::string>);
21
22 private:
23     //add any other methods here
24     void erode(frame); //edge detection
25     std::string paramLabel = "erosion";
26 };
27 #endif // CELLUVIEW_EROSION_H

```

5.7 flatFieldCorrect.h

```

1
2 #ifndef OPENFLEXURE_FLAT_FIELD_CORRECT_H
3 #define OPENFLEXURE_FLAT_FIELD_CORRECT_H
4
5
6 #include <opencv2/core.hpp>
7 #include <opencv2/videoio.hpp>
8 #include <iostream>
9 #include <stdlib.h>
10 #include <thread>
11 #include "imageProcessor.h"
12 #include "gallery.h"
13
14
15 class flatFieldCorrect: public imageProcessor{
16 public:
17     flatFieldCorrect() = default;
18
19     void receiveFrame(frame newFrame);
20     std::string getParamLabel() {return paramLabel;};
21     void setUpdateFlag();
22
23
24     void updateSettings(std::map<std::string, std::string>);
25 private:
26     //add any other methods here
27     void updateAverage(frame);
28     void flatField(frame); //Flat Field Correction
29     std::string paramLabel = "flatField";
30     cv::Mat current_correction_factor;
31     bool calculateAverageEnabled;
32 };
33 #endif // OPENFLEXURE_FLAT_FIELD_CORRECT_H

```

5.8 frame.h

```

1 #ifndef CELLUVIEW_FRAME_H
2 #define CELLUVIEW_FRAME_H
3 #include <opencv2/core.hpp>
4 #include <opencv2/videoio.hpp>
5
6 #include <iostream>
7 #include <stdlib.h>
8 #include <thread>
9 #include <typeinfo>
10 #include <map>
11
12 #define metaDataPairDelim "|"
13 #define metaDataItemDelim "*"
14
15
16 class frame{
17 public:
18     frame() = default;
19     frame(cv::Mat matIn): image(matIn){}
20
21     //custom copy constructor replacement to allow test to pass,
22     // provide a pointer to existing frame
23     //and copy from there into a constructed second frame
24     void copyFrom(frame* copyFrom){
25         this->image = copyFrom->image;
26         this->doMeta = copyFrom->doMeta;
27         this->params = copyFrom->params;
28     }
29
30
31
32     cv::Mat image;
33     void setParameter(std::string, std::string);
34     std::string encodeMetadata();
35     bool doMeta = false;
36
37     std::string getParam(std::string);
38
39     frame(frame const & ) = default; //default copy constructor
40
41     int getParamSize();
42
43
44 private:
45     std::map<std::string, std::string> params;

```

```

46
47     const std::vector<std::string> validParams= {
48         "edgeThreshold",
49         "erosion",
50         "dilation",
51         "grayScale",
52         "contrastThreshold",
53         "flatField",
54         "exposure"
55         //ADD MORE PARAMETERS HERE
56     };
57
58     std::string encodedString;
59
60
61 };
62
63
64
65 #endif //CELLUVIEW_FRAME_H

```

5.9 gallery.h

```

1  #ifndef CELLUVIEW_GALLERY_H
2  #define CELLUVIEW_GALLERY_H
3
4  #include "stdlib.h"
5  #include <opencv2/core.hpp>
6  #include <opencv2/imgproc.hpp>
7  #include <opencv2/imgcodecs.hpp>
8  #include "imageProcessor.h"
9  #include <sys/types.h>
10 #include <sys/stat.h>
11 #include <dirent.h>
12 #include "cpp_exiftool/src/ExifTool.h"
13 #include <sstream>
14 #include <iterator>
15 #include <string>
16 #include "frame.h"
17 #include <cstdio>
18
19 class Gallery{
20
21     public:
22         Gallery();
23         void captureFrame(frame, bool flatfield=false, int flatFieldCounter = 0);
24         void updateImgName(std::string);
25         void updateIndex();
26         std::map<std::string, std::string> getMetadata(std::string = "");
27
28         std::string getPathname(){return pathname;}; //testing only
29         // std::string getFlatFieldPath(){return flatFieldPath;};
30         std::string getCaptureFname(){return captureFname;}; //testing only
31
32
33
34     private:
35         int initialiseDirectory(std::string, std::string);
36         void writeMetadata(frame, std::string);
37
38         DIR *dir;
39         struct dirent *ent;
40         std::string pathname = "";
41         std::string flatFieldPath = "";
42         std::string imgName = "capture";
43         int captureImgCounter = 0;
44         std::string captureFname = "";
45         int index;
46         int lastHighestIndex = -1;
47         int indexLen = 1;
48         std::string existingCaptureFname;
49
50
51         cv::Mat img;
52
53         std::string tagName;
54         std::string receivedMetadata;
55         std::string MetadataToWrite;
56
57         ExifTool *et = new ExifTool();
58
59         std::map<std::string, std::string> restoredParams;
60

```

```

61
62 };
63 #endif //CELLUVIEW_GALLERY_H

```

5.10 grayScale.h

```

1 #ifndef CELLUVIEW_GREY_SCALE_H
2 #define CELLUVIEW_GREY_SCALE_H
3
4 #include <opencv2/core.hpp>
5 #include <opencv2/videoio.hpp>
6 #include <iostream>
7 #include <stdlib.h>
8 #include <thread>
9 #include "imageProcessor.h"
10
11 class grayScale: public imageProcessor{
12
13 public:
14     grayScale() = default;
15     void receiveFrame(frame);
16     std::string getParamLabel(){return paramLabel;};
17     void updateSettings(std::map<std::string, std::string>);
18
19 private:
20
21     void grayEnhance(frame); // grayscale conversion
22     std::string paramLabel = "grayScale";
23 };
24 #endif // CELLUVIEW_GREY_SCALE_H

```

5.11 gui.h

```

1 #ifndef CELLUVIEW_GUI_H
2 #define CELLUVIEW_GUI_H
3 #include <QMainWindow>
4
5 #include "QT/qtWindow.h" //compiled header file from qtcreator
6 #include "stdlib.h"
7 #include <opencv2/core.hpp>
8 #include <opencv2/imgproc.hpp>
9 #include <opencv2/imgcodecs.hpp>
10 #include "imageProcessor.h"
11 #include "edgeDetection.h"
12 #include "gallery.h"
13 #include "grayScale.h"
14 #include "frame.h"
15 #include "dilation.h"
16 #include "erosion.h"
17 #include "camera.h"
18 #include "contrastEnhancement.h"
19 #include "flatFieldCorrect.h"
20
21
22
23
24
25 class Gui : public QWidget, public imageProcessor{
26     Q_OBJECT
27
28 public:
29     void receiveFrame(frame newFrame);
30     Gui(QMainWindow*, Ui_GUI*, Gallery*, std::vector <imageProcessor *>&);
31     void SetVisible(bool visible);
32
33 private:
34     QMainWindow *widget;
35     Ui_GUI *ui;
36     bool doCapture = false;
37     void captureNextFrame();
38     void restoreSettings(std::string = "");
39     void updateSettings(std::map<std::string, std::string>);
40     std::string getParamLabel(){return ""};
41     bool updateFlatField = false;
42     bool flatFieldEnable = false;
43     void setUpdateFlatField();
44     int flatFieldCounter = 0;
45
46
47     cv::Mat img;
48     Gallery* gallery;
49     Camera* cam;

```

```

50     std::vector<imageProcessor*> blocks;
51
52     std::map<std::string, std::string> metadata;
53
54
55
56     edgeDetection *edgeDetector;
57     grayScale *grayDetector;
58 };
59 #endif // CELLUVIEW_GUI_H

```

5.12 imageProcessor.h

```

1  #ifndef CELLUVIEW_IMAGEPROCESSOR_H
2  #define CELLUVIEW_IMAGEPROCESSOR_H
3  #include <opencv2/core.hpp>
4  #include <opencv2/videoio.hpp>
5  #include "frame.h"
6
7  #include <iostream>
8  #include <stdlib.h>
9  #include <thread>
10
11 //abstract class
12 class imageProcessor{
13 public:
14     imageProcessor() = default;
15
16     virtual void receiveFrame(frame newFrame) = 0;
17
18     virtual void updateSettings(std::map<std::string, std::string>) = 0;
19
20     virtual std::string getParamLabel() = 0;
21
22     void registerCallback(imageProcessor* cb){
23         frameCb = cb;
24     }
25
26     bool toggleEnable(){
27         enabled = !enabled;
28         return enabled;
29     }
30
31     bool getEnabled(){
32         return enabled;
33     }
34
35
36 protected:
37     imageProcessor* frameCb = nullptr;
38     bool enabled = false;
39
40
41 };
42
43 #endif

```


Index

- Camera, [7](#)
 - [getParamLabel](#), [7](#)
 - [receiveFrame](#), [7](#)
 - [updateSettings](#), [8](#)
- [cellUviewWelcome](#), [8](#)
- [contrastEnhancement](#), [8](#)
 - [getParamLabel](#), [9](#)
 - [receiveFrame](#), [9](#)
 - [updateSettings](#), [9](#)
- [dilation](#), [10](#)
 - [getParamLabel](#), [10](#)
 - [receiveFrame](#), [10](#)
 - [updateSettings](#), [10](#)
- [edgeDetection](#), [11](#)
 - [getParamLabel](#), [11](#)
 - [receiveFrame](#), [11](#)
 - [updateSettings](#), [11](#)
- [erosion](#), [12](#)
 - [getParamLabel](#), [12](#)
 - [receiveFrame](#), [12](#)
 - [updateSettings](#), [12](#)
- [flatFieldCorrect](#), [13](#)
 - [getParamLabel](#), [13](#)
 - [receiveFrame](#), [13](#)
 - [updateSettings](#), [14](#)
- [frame](#), [14](#)
- [Gallery](#), [15](#)
 - [getParamLabel](#)
 - Camera, [7](#)
 - [contrastEnhancement](#), [9](#)
 - [dilation](#), [10](#)
 - [edgeDetection](#), [11](#)
 - [erosion](#), [12](#)
 - [flatFieldCorrect](#), [13](#)
 - [grayScale](#), [15](#)
- [grayScale](#), [15](#)
 - [getParamLabel](#), [15](#)
 - [receiveFrame](#), [15](#)
 - [updateSettings](#), [16](#)
- [Gui](#), [16](#)
 - [Gui](#), [17](#)
 - [receiveFrame](#), [18](#)
 - [SetVisible](#), [18](#)
- [imageProcessor](#), [18](#)
 - [receiveFrame](#), [19](#)
- [receiveFrame](#)
 - Camera, [7](#)
 - [contrastEnhancement](#), [9](#)
 - [dilation](#), [10](#)
 - [edgeDetection](#), [11](#)
 - [erosion](#), [12](#)
 - [flatFieldCorrect](#), [13](#)
 - [grayScale](#), [15](#)
 - [Gui](#), [18](#)
 - [imageProcessor](#), [19](#)
- [SetVisible](#)
 - [Gui](#), [18](#)
- [src/camera.h](#), [21](#)
- [src/cellUviewWelcome.h](#), [21](#)
- [src/contrastEnhancement.h](#), [22](#)
- [src/dilation.h](#), [22](#)
- [src/edgeDetection.h](#), [23](#)
- [src/erosion.h](#), [23](#)
- [src/flatFieldCorrect.h](#), [24](#)
- [src/frame.h](#), [24](#)
- [src/gallery.h](#), [25](#)
- [src/grayScale.h](#), [26](#)
- [src/gui.h](#), [26](#)
- [src/imageProcessor.h](#), [27](#)
- [updateSettings](#)
 - Camera, [8](#)
 - [contrastEnhancement](#), [9](#)
 - [dilation](#), [10](#)
 - [edgeDetection](#), [11](#)
 - [erosion](#), [12](#)
 - [flatFieldCorrect](#), [14](#)
 - [grayScale](#), [16](#)