BIOPHOTONICS & BIOIMAGING LABORATORY

Week 1 OpenCV

陳立達 @ 2017 Summer

NTUBIME Lab405

Input Image

- Mat cv::imread(const String & filename,

  int flags = IMREAD_COLOR)

- Parameters

  - filenameName of file to be loaded.

  - flags Flag that can take values of cv::ImreadModes

Input Image

* Flags

IMREAD_UNCHANGED	**If set, return the loaded image as is (with alpha channel, otherwise it gets cropped).**
IMREAD_GRAYSCALE	**If set, always convert image to the single channel grayscale image.**
IMREAD_COLOR	**If set, always convert image to the 3 channel BGR color image.**
IMREAD_ANYDEPTH	**If set, return 16-bit/32-bit image when the input has the corresponding depth, otherwise convert it to 8-bit.**
IMREAD_ANYCOLOR	**If set, the image is read in any possible color format.**
IMREAD_LOAD_GDAL	**If set, use the gdal driver for loading the image.**
IMREAD_REDUCED_GRAYSCALE_2	**If set, always convert image to the single channel grayscale image and the image size reduced 1/2.**
IMREAD_REDUCED_COLOR_2	**If set, always convert image to the 3 channel BGR color image and the image size reduced 1/2.**
IMREAD_REDUCED_GRAYSCALE_4	**If set, always convert image to the single channel grayscale image and the image size reduced 1/4.**
IMREAD_REDUCED_COLOR_4	**If set, always convert image to the 3 channel BGR color image and the image size reduced 1/4.**
IMREAD_REDUCED_GRAYSCALE_8	**If set, always convert image to the single channel grayscale image and the image size reduced 1/8.**
IMREAD_REDUCED_COLOR_8	**If set, always convert image to the 3 channel BGR color image and the image size reduced 1/8.**
IMREAD_IGNORE_ORIENTATION	**If set, do not rotate the image according to EXIF's orientation flag.**

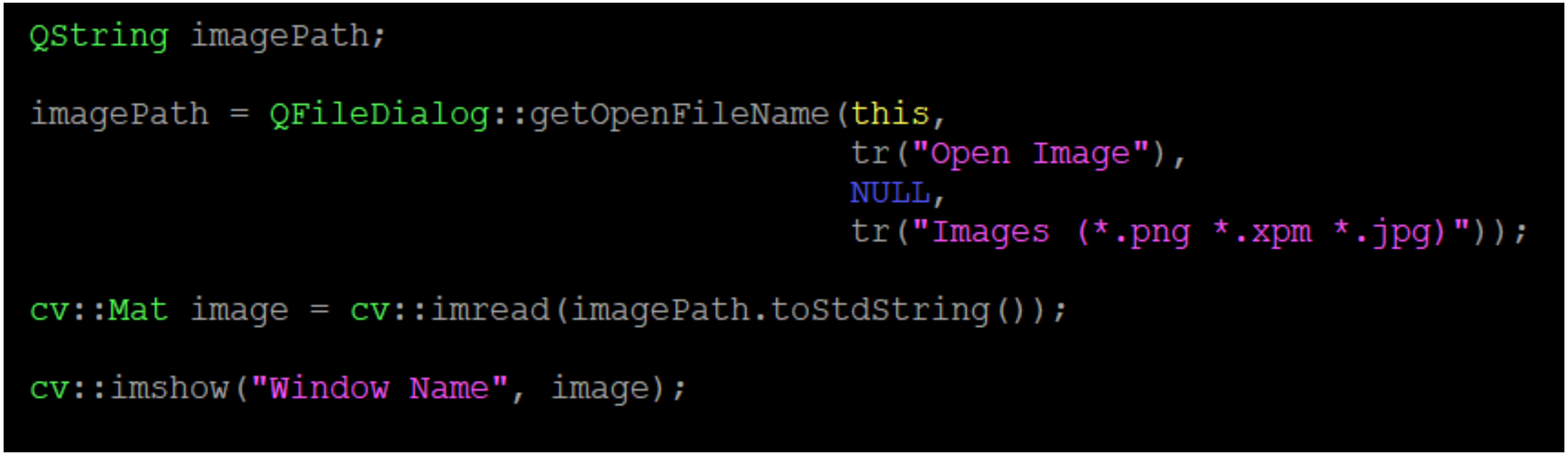
Display Image

- void cv::imshow(const String & winname, InputArray mat)

- Parameters

- winname	Name of the window.
- mat	Image to be shown.

Example



```
QString imagePath;

imagePath = QFileDialog::getOpenFileName(this,
                                         tr("Open Image"),
                                         NULL,
                                         tr("Images (*.png *.xpm *.jpg)"));

cv::Mat image = cv::imread(imagePath.toStdString());

cv::imshow("Window Name", image);
```

Save Image

* bool cv::imwrite(const String & filename,

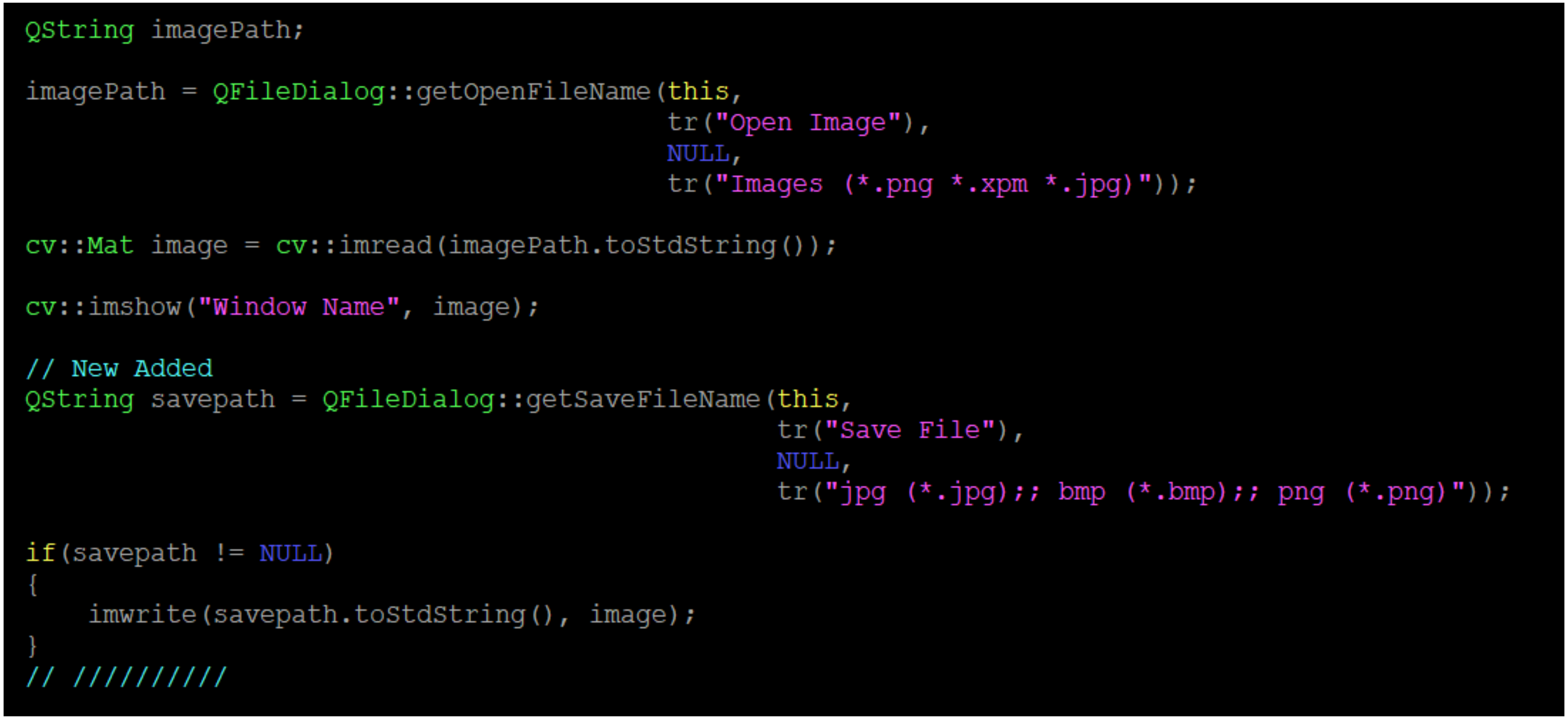
  InputArray img,

  const std::vector< int > & params = std::vector< int >())

* Parameters

- filename Path of the file to be saved.
- img Image to be saved.

Example



```
QString imagePath;

imagePath = QFileDialog::getOpenFileName(this,
                                         tr("Open Image"),
                                         NULL,
                                         tr("Images (*.png *.xpm *.jpg)"));

cv::Mat image = cv::imread(imagePath.toStdString());

cv::imshow("Window Name", image);

// New Added
QString savepath = QFileDialog::getSaveFileName(this,
                                                tr("Save File"),
                                                NULL,
                                                tr("jpg (*.jpg);; bmp (*.bmp);; png (*.png)"));

if(savepath != NULL)
{
    imwrite(savepath.toStdString(), image);
}
// //////////
```

Color Space Transform

- void cv::cvtColor (InputArray src, OutputArray dst,

  int code, int dstCn = 0)

Parameters

- src input image

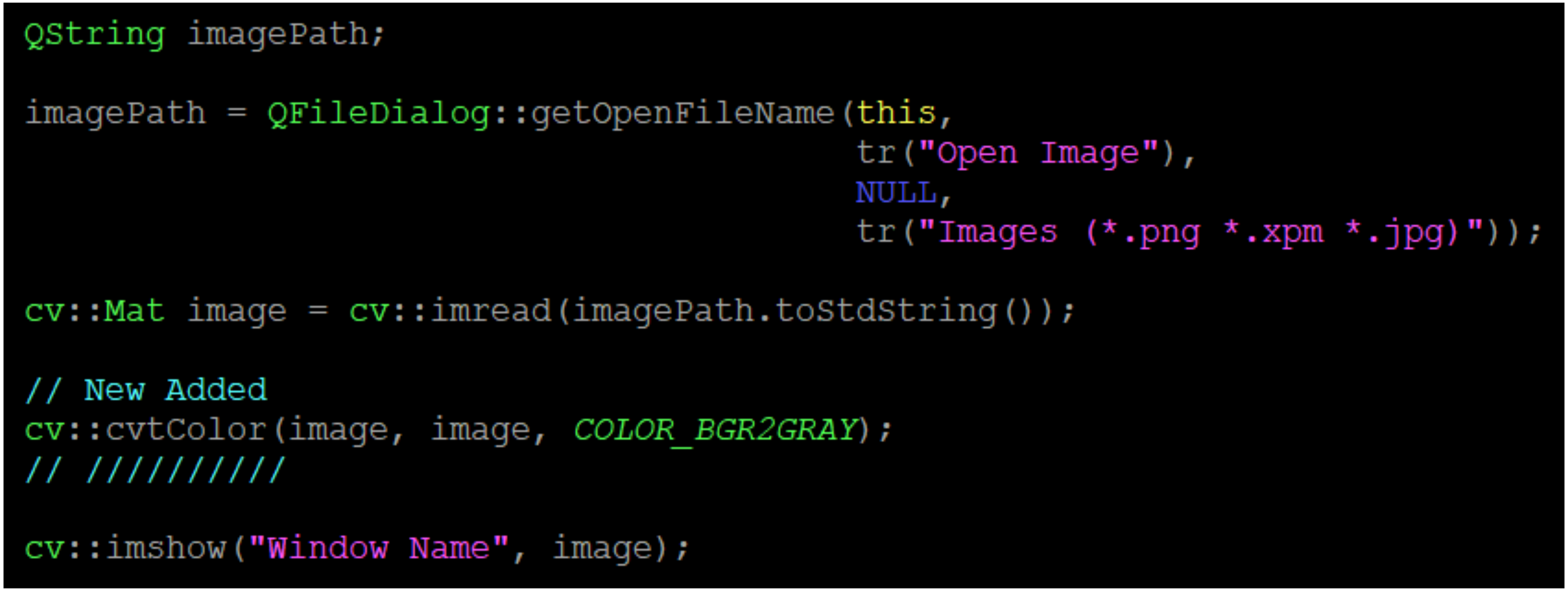
- dst output image of the same size and depth as src.

- code color space conversion code (see cv::ColorConversionCodes).

- dstCn number of channels in the destination image; if the parameter is 0,

  the number of the channels is derived automatically from src and code.

Example: Grayscale



```
QString imagePath;

imagePath = QFileDialog::getOpenFileName(this,
                                         tr("Open Image"),
                                         NULL,
                                         tr("Images (*.png *.xpm *.jpg)"));

cv::Mat image = cv::imread(imagePath.toStdString());

// New Added
cv::cvtColor(image, image, COLOR_BGR2GRAY);
// //////////

cv::imshow("Window Name", image);
```

Draw a circle on image

* void cv::circle (

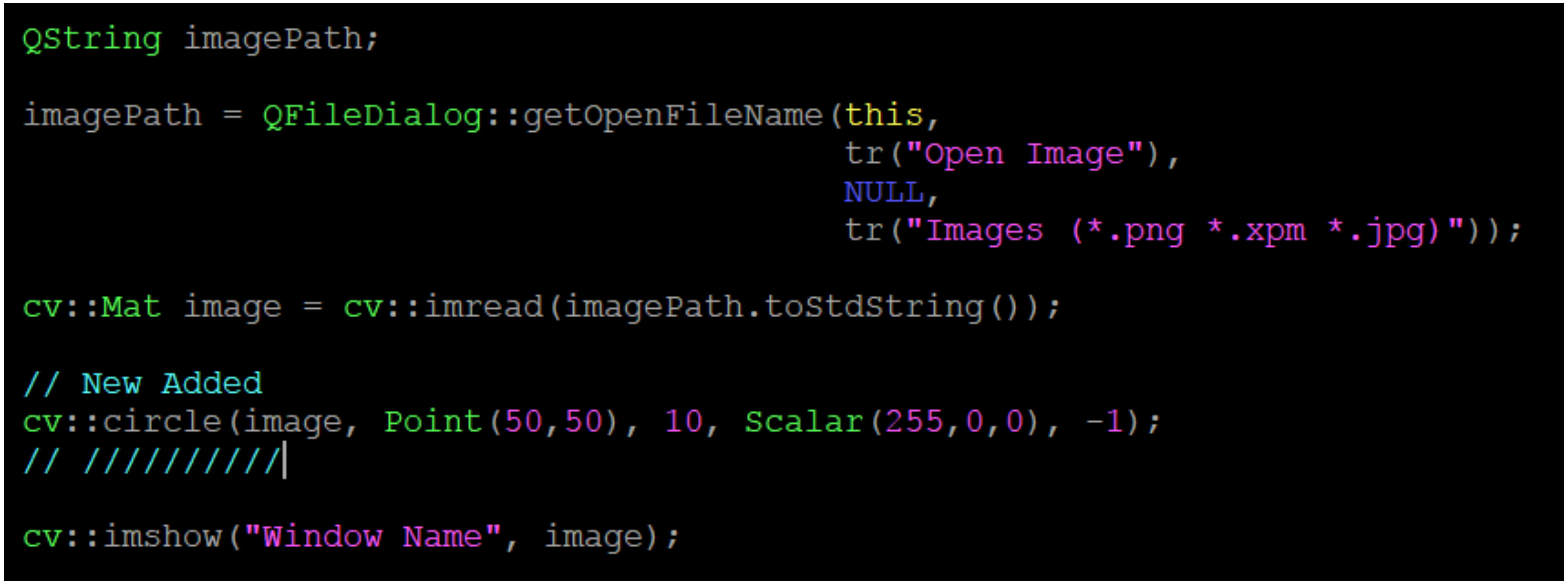
```
InputOutputArray    img,
Point center,
int radius,
const Scalar & color,
int    thickness = 1,
int    lineType = LINE_8,
int    shift = 0
)
```

Draw a circle on image

* Parameters

-	img	Image where the circle is drawn.
-	center	Center of the circle.
-	radius	Radius of the circle.
-	color	Circle color.
-	thickness	Thickness of the circle outline, if positive. Negative thickness means that a filled circle is to be drawn.
-	lineType	Type of the circle boundary. See the line description.
-	Shift	Number of fractional bits in the coordinates of the center and in the radius value.

Example



```
QString imagePath;

imagePath = QFileDialog::getOpenFileName(this,
                                         tr("Open Image"),
                                         NULL,
                                         tr("Images (*.png *.xpm *.jpg)"));

cv::Mat image = cv::imread(imagePath.toStdString());

// New Added
cv::circle(image, Point(50,50), 10, Scalar(255,0,0), -1);
// //////////

cv::imshow("Window Name", image);
```

Draw a rectangle on image

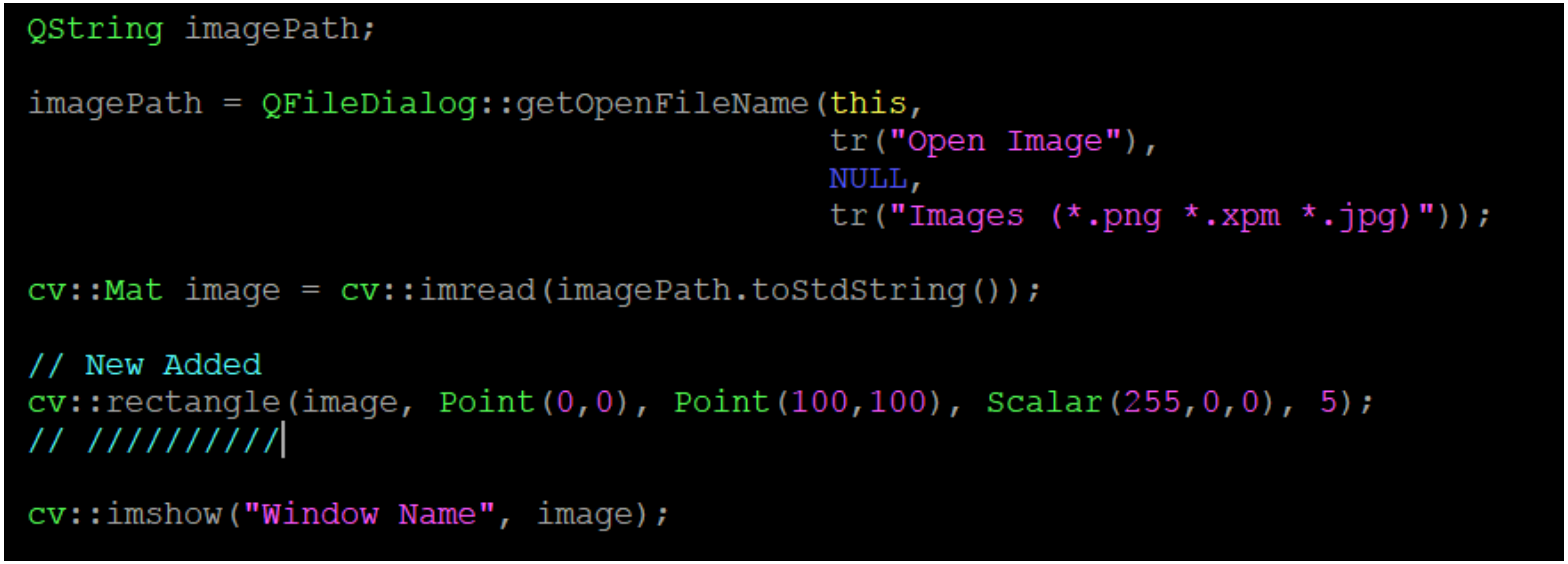
*
  ```
  void cv::rectangle(  InputOutputArray    img,
                       Point pt1,
                       Point pt2,
                       const Scalar & color,
                       int thickness = 1,
                       int lineType = LINE_8,
                       int shift = 0
                       )
  ```

Draw a rectangle on image

* Parameters

- img	Image.
- pt1	Vertex of the rectangle.
- pt2	Vertex of the rectangle opposite to pt1 .
- color	Rectangle color or brightness (grayscale image).
- thickness	Thickness of lines that make up the rectangle. Negative values, like CV_FILLED , mean that the function has to draw a filled rectangle.
- lineType	Type of the line. See the line description.
- shift	Number of fractional bits in the point coordinates.

Example



```
QString imagePath;

imagePath = QFileDialog::getOpenFileName(this,
                                         tr("Open Image"),
                                         NULL,
                                         tr("Images (*.png *.xpm *.jpg)"));

cv::Mat image = cv::imread(imagePath.toStdString());

// New Added
cv::rectangle(image, Point(0,0), Point(100,100), Scalar(255,0,0), 5);
// //////////

cv::imshow("Window Name", image);
```

Thank you ~~

BIOPHOTONICS & BIOIMAGING LABORATORY