

Aryan Shirke 21bs111

The headers, prototypes, and usage examples for some of the common mathematical morphology functions available in OpenCV:

### 1. Erosion:

- Header: `#include <imgproc.hpp>`
- Prototype: `void cv::erode(InputArray src, OutputArray dst, InputArray kernel, Point anchor=Point(-1,-1), int iterations=1, int borderType=BORDER_CONSTANT, const Scalar& borderValue=morphologyDefaultBorderValue())`
- Usage Example:

```
cpp
Mat src = imread("input.jpg", IMREAD_GRAYSCALE);
Mat dst;
Mat kernel = getStructuringElement(MORPH_RECT,
Size(3,3));
erode(src, dst, kernel);
```

### 2. Dilation:

- Header: `#include <imgproc.hpp>`
- Prototype: `void cv::dilate(InputArray src, OutputArray dst, InputArray kernel, Point anchor=Point(-1,-1), int iterations=1, int`

Aryan Shirke 21bs111

```
borderType=BORDER_CONSTANT, const Scalar&  
borderValue=morphologyDefaultBorderValue())
```

- Usage Example:

```
cpp  
Mat src = imread("input.jpg", IMREAD_GRAYSCALE);  
Mat dst;  
Mat kernel = getStructuringElement(MORPH_RECT,  
Size(3,3));  
dilate(src, dst, kernel);
```

3. Opening:

- Header: `#include imgproc.hpp`

- Prototype: `void cv::morphologyEx(InputArray src,  
OutputArray dst, int op, InputArray kernel, Point  
anchor=Point(-1,-1), int iterations=1, int`

```
borderType=BORDER_CONSTANT, const Scalar&  
borderValue=morphologyDefaultBorderValue())
```

- Usage Example:

```
cpp  
Mat src = imread("input.jpg", IMREAD_GRAYSCALE);  
Mat dst;  
Mat kernel = getStructuringElement(MORPH_RECT,  
Size(3,3));  
morphologyEx(src, dst, MORPH_OPEN, kernel);
```

Aryan Shirke 21bs111

#### 4. Closing:

- Header: `#include imgproc.hpp`
- Prototype: `void cv::morphologyEx(InputArray src, OutputArray dst, int op, InputArray kernel, Point anchor=Point(-1,-1), int iterations=1, int borderType=BORDER_CONSTANT, const Scalar& borderValue=morphologyDefaultBorderValue())`
- Usage Example:

```
cpp
Mat src = imread("input.jpg", IMREAD_GRAYSCALE);
Mat dst;
Mat kernel = getStructuringElement(MORPH_RECT,
Size(3, 3));
morphologyEx(src, dst, MORPH_CLOSE, kernel);
```

#### 5. Thinning:

OpenCV does not provide a built-in function for thinning. You may need to implement thinning algorithms yourself or use third-party libraries.

#### 6. Thickening:

OpenCV does not provide a built-in function for thickening.

Aryan Shirke 2165111

It's usually performed by dilation operation, but no specific function for thickening is available in OpenCV.