Utility Functions:

1. cv::imread(imgpath) :

It reads the image given in the imagepath and returns an image.
Example :

```
cv::Mat img  = cv::imread('\home\souro\index.png');
```

2. cv::imwrite(storagepath , img) :

It write or store the image in the specified path and return void.
Example :

```
cv::imwrite('\home\souro\images\sample.png',img);
```

3. cv::imshow(windowname, img) :

It show the image in a popup window .
Example :

```
cv::imshow('sample',img);
```

4. cv::waitKey(int delay):

If it use in a program It waits for the delay time (in milisecond) mention in the function to perform next instruction in the program If not any key press from user. If its value set to 0 it waits for infinite times if not any key press from user.
Example :

```
cv::waitkey(1000);
```

5. cv::Mat image(image_size, Mat::depth , value):
Mat:: depth

`CV_8U` - 8-bit unsigned integers (`0..255`)

`CV_8S` - 8-bit signed integers (`-128..127`)

`CV_16U` - 16-bit unsigned integers (`0..65535`)

`CV_16S` - 16-bit signed integers (`-32768..32767`)

`CV_32S` - 32-bit signed integers (`-2147483648..2147483647`)

`CV_32F` - 32-bit floating-point numbers (`-FLT_MAX..FLT_MAX, INF, NAN`)

`CV_64F` - 64-bit floating-point numbers (`-DBL_MAX..DBL_MAX, INF, NAN`)

example: cv::Mat img(imgThresh.size(), CV_8UC3, BLACK_RGB);

6. clone():
It is a function of a image object and clones the image to another image.
Example :

```
cv::Mat imgThreshCopy = imgThresh.clone();
```

7.empty():
It is a function of a image object which checks whether the image is empty or not.
Example :

```
if (imgOriginalScene.empty()) {
std::cout << "error: image not read from file\n\n";
}
```

8. copyTo(resultImg):

It is a function of a image object which copy the image into a resulted image.

Example:

```
cv::Mat3b res(img.rows, img.cols, cv::Vec3b(0,0,0));
// Copy images in correct position
findcharImg.copyTo(res(cv::Rect(0, 0, findcharImg.cols, findcharImg.rows)));
imgContours.copyTo(res(cv::Rect(findcharImg.cols, 0, imgContours.cols,
imgContours.rows)));
```

9. cv::addWeighted(src1,alpha,src2,beta,gamma,dst):

It blends two images src1 with alpha proportions and src2 with beta proportions and result image in dst .

Example:

$$dst = \alpha \cdot src1 + \beta \cdot src2 + \gamma$$

```
cv::addWeighted( src1, alpha, src2, beta, 0.0, dst);
```

10. cv::cvtColor(src,dst, conversion_type)

It convert the color of the source image and produce the destination image using conversion type.

Example:

```
cv::cvtColor(imgOriginalScene, imgHSV, cv::COLOR_BGR2HSV);
cv::cvtColor(imgOriginalScene, imgGRAY, cv::COLOR_BGR2GRAY);
```

11.cv::split(src,dst):

It splits an image into a vector of images.

Example:

```
std::vector<cv::Mat> vectorOfHSVImages;
cv::split(imgHSV, vectorOfHSVImages);
imgValue = vectorOfHSVImages[2];
```

12. cv::equalizeHist(src, dst):

It is the function for histogram equalisation and converts source image to destination image.

Important points:

1. if all pixel of r,g,b of an image having same value then the color image converted to grayscale image and the saturation of the image become 0.

```
for(int i = 0; i<imgContours.rows; i++)
{
    for(int j = 0;j<imgContours.cols;j++)
    {
        imgContours.at<cv::Vec3b>(i,j)[0] =imgOriginalScene.at<cv::Vec3b>(i,j)
[0]*0.29+imgOriginalScene.at<cv::Vec3b>(i,j)[1]*0.55+imgOriginalScene.at<cv::Vec3b>(i,j)
[2]*0.16;
        imgContours.at<cv::Vec3b>(i,j)[1] =imgOriginalScene.at<cv::Vec3b>(i,j)
[0]*0.29+imgOriginalScene.at<cv::Vec3b>(i,j)[1]*0.55+imgOriginalScene.at<cv::Vec3b>(i,j)
[2]*0.16;
        imgContours.at<cv::Vec3b>(i,j)[2] =imgOriginalScene.at<cv::Vec3b>(i,j)
[0]*0.29+imgOriginalScene.at<cv::Vec3b>(i,j)[1]*0.55+imgOriginalScene.at<cv::Vec3b>(i,j)
[2]*0.16;
    }
```

}

2. In document Image analysis the image is a type of binary image where the white background is represented by 0 and the black writings or foreground represented by 1.