Harshit Singh

K/K2

60009200063

AIM: To Perform Basic Image Processing Operations in Python

PROBLEM STATEMENT:

Read an Image

Display an Image

Convert in Grey Scale

Crop an Image

Arithmetic Operations

Logical Operations

THEORY:

Libraries used:

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

Features:

- OpenCV is an open-source library, so it is free to use and can be easily customized to meet specific needs.
- OpenCV provides a comprehensive set of algorithms for image processing and computer vision, including object detection, recognition, tracking, and segmentation.
- Python is a popular programming language for scientific computing, and OpenCV's Python interface makes it easy to integrate computer vision algorithms with other Python libraries like NumPy, SciPy, and Matplotlib.
- OpenCV's Python bindings support both Python 2 and Python 3, making it accessible to a wide range of users.
- OpenCV is cross-platform, so code written in Python can be easily ported to other platforms like Linux, Windows, and macOS.

- OpenCV has a large and active community, so there are plenty of resources available for learning and troubleshooting, including documentation, tutorials, and forums.

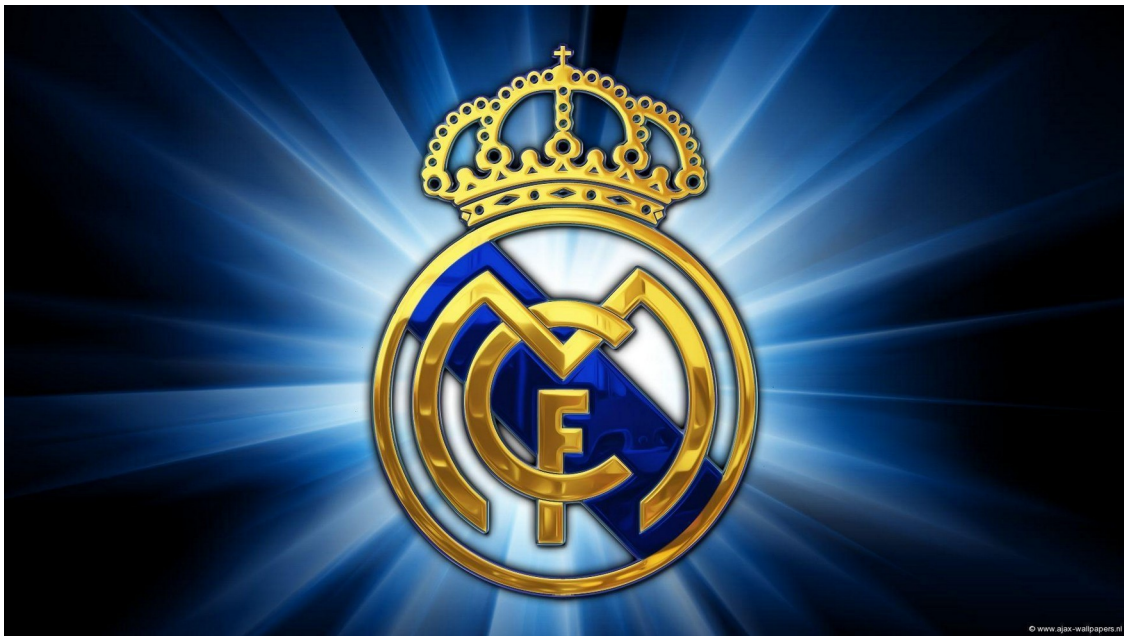
Limitations:

- Speed: OpenCV in Python can be slower compared to other languages like C++ because of the GIL (Global Interpreter Lock) in Python. This can limit the performance of real-time applications.
- Memory management: Memory management in Python can be problematic, especially when working with large datasets. This can lead to slower processing times and memory leaks.
- Limited machine learning capabilities: Although OpenCV has some machine learning capabilities, it may not be as robust as other popular machine learning libraries like TensorFlow and PyTorch.
- Limited support for deep learning: OpenCV has limited support for deep learning, which may be a disadvantage for complex image processing tasks.
- Limited support for GPU processing: OpenCV in Python may not provide support for GPU processing, which may limit the speed of certain applications.
- Limited documentation: OpenCV documentation in Python may not be as extensive as in other languages, which may make it challenging for beginners to learn and use the library effectively.

RESULT:

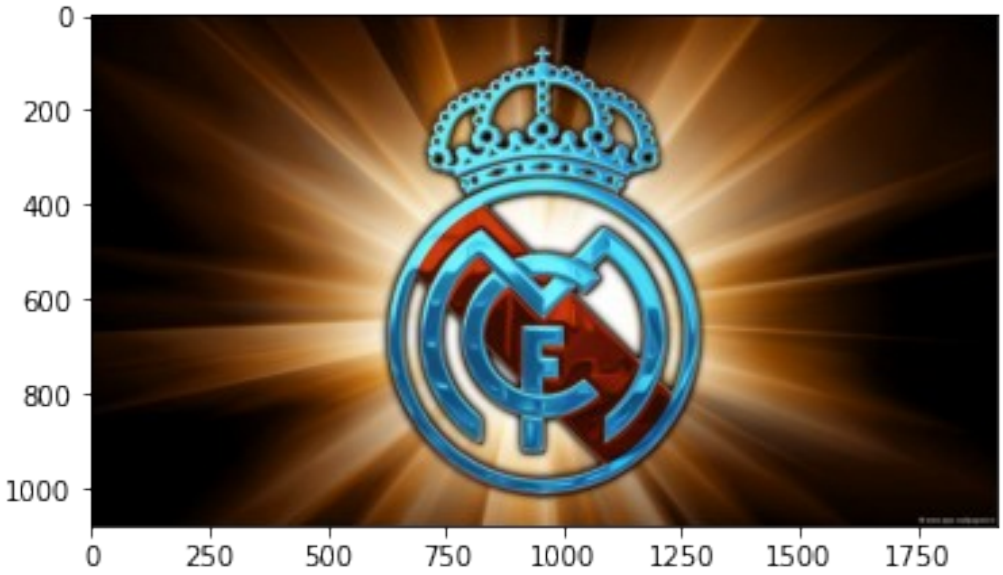
Basic Image Processing

```
import cv2
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt

img = cv2.imread("/content/rma.jpg", cv2.IMREAD_COLOR)

img.shape

(1080, 1920, 3)

cv2_imshow(img)
```



Displaying the Image using Matplotlib

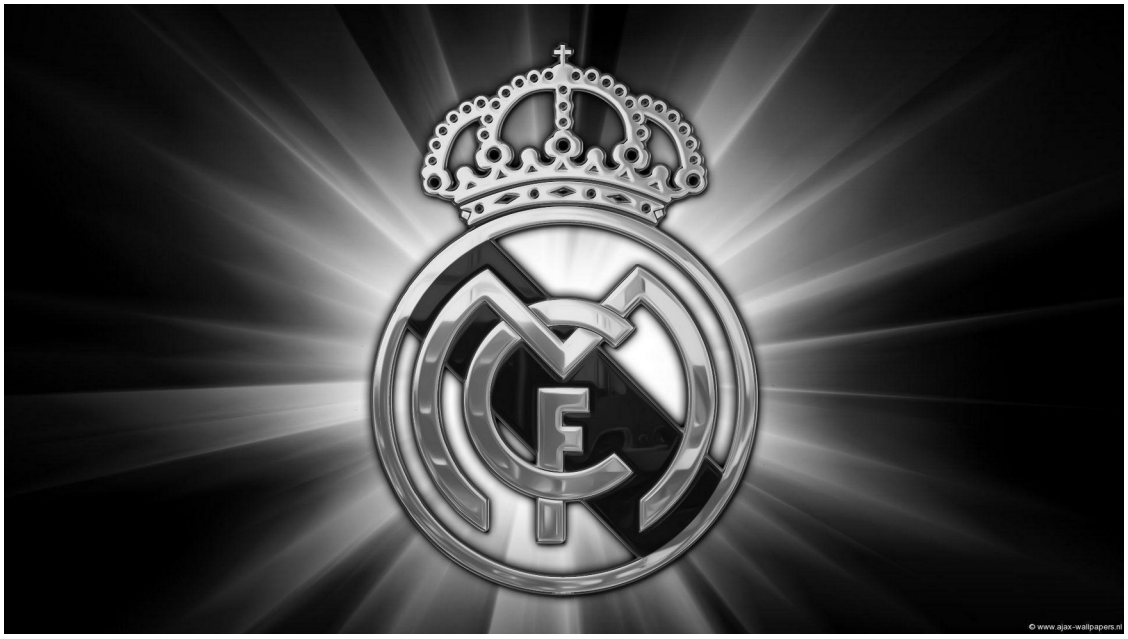
```
plt.imshow(img)

<matplotlib.image.AxesImage at 0x7ff8987407f0>
```



Grey Scale of image

```
grey_image = cv2.imread("/content/rma.jpg", 0)
```

```
cv2_imshow(grey_image)
```



Crop the Image

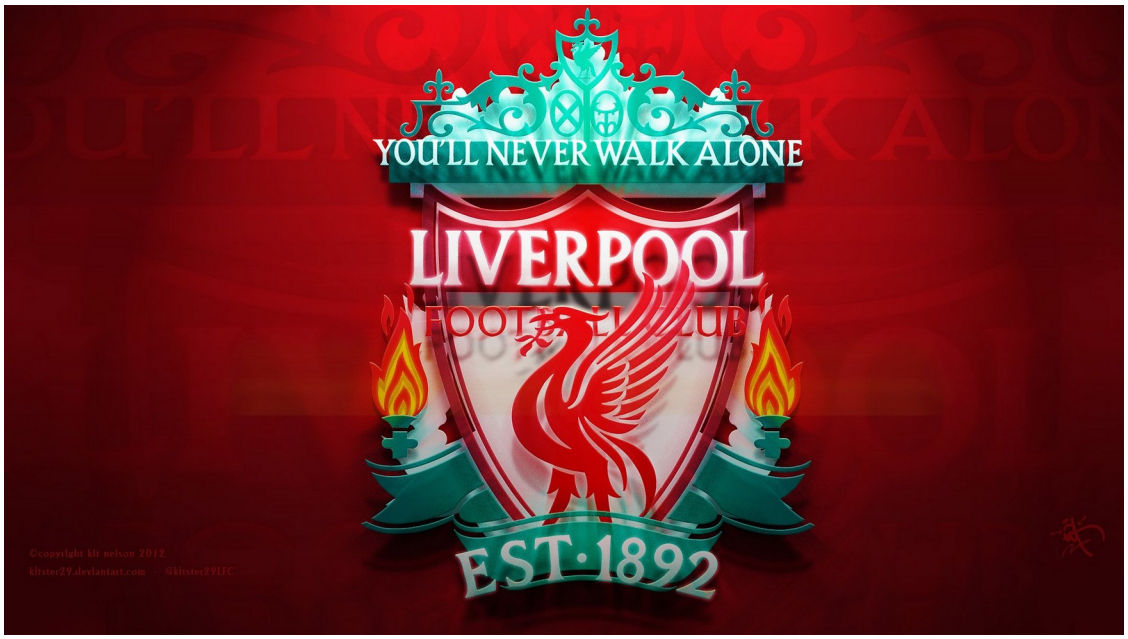
```
img.shape
```

```
(1080, 1920, 3)
```

```
crop = img[300:900, 100:800]
cv2_imshow(crop)
```

Arithmetic Operations on 2 images

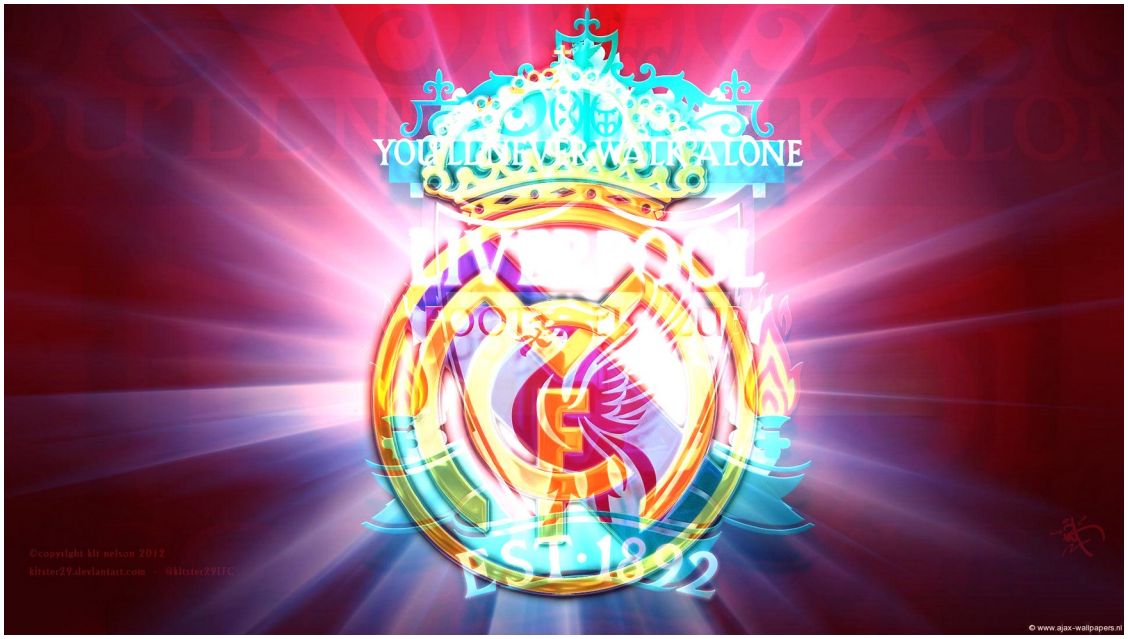
```
img2 = cv2.imread("/content/liverpool.jpg", cv2.IMREAD_COLOR)
cv2_imshow(img2)
```




```
img2.shape
```

```
(1080, 1920, 3)
```

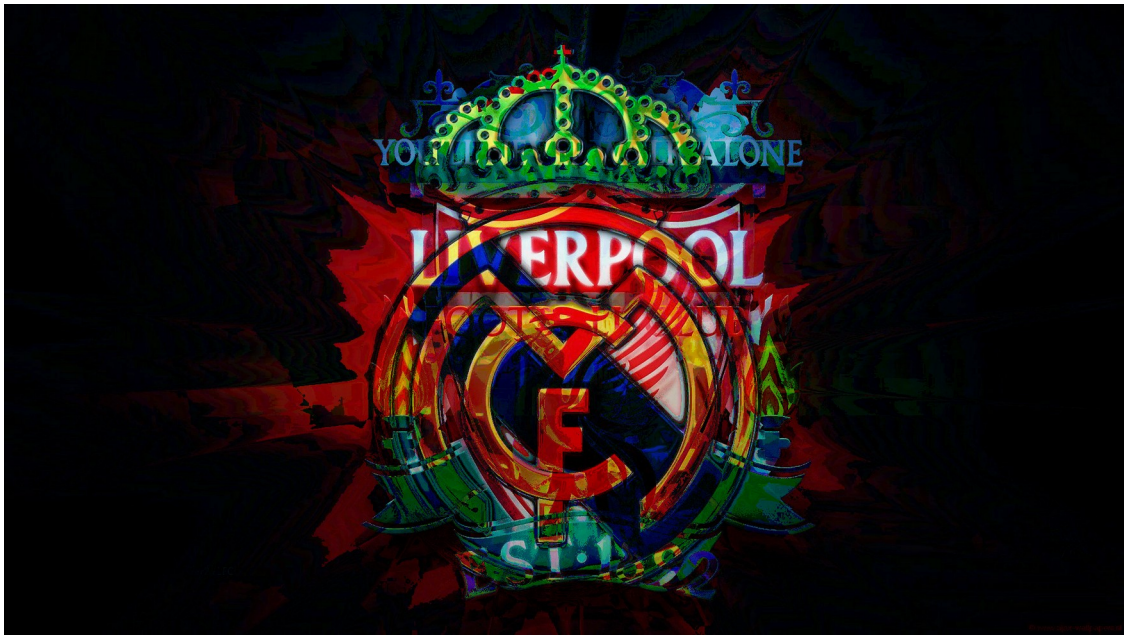
```
new_img = cv2.add(img, img2)
cv2_imshow(new_img)
```



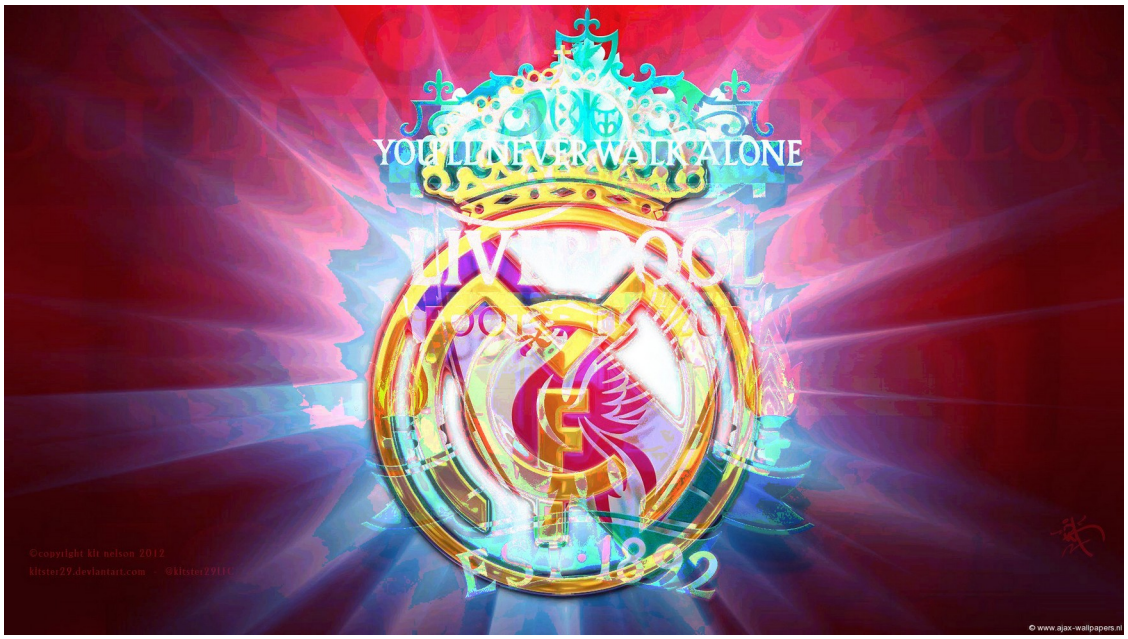

```
new_img = cv2.subtract(img, img2)
cv2_imshow(new_img)
```

Logical Operation on 2 Images

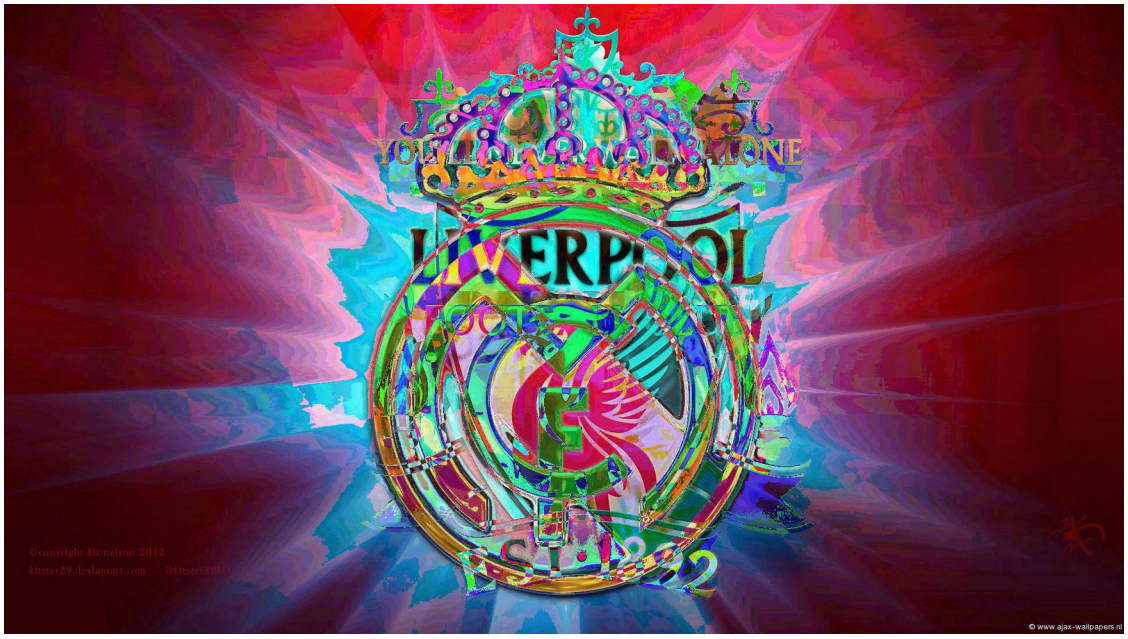
```
new_img = cv2.bitwise_and(img, img2)
cv2_imshow(new_img)
```



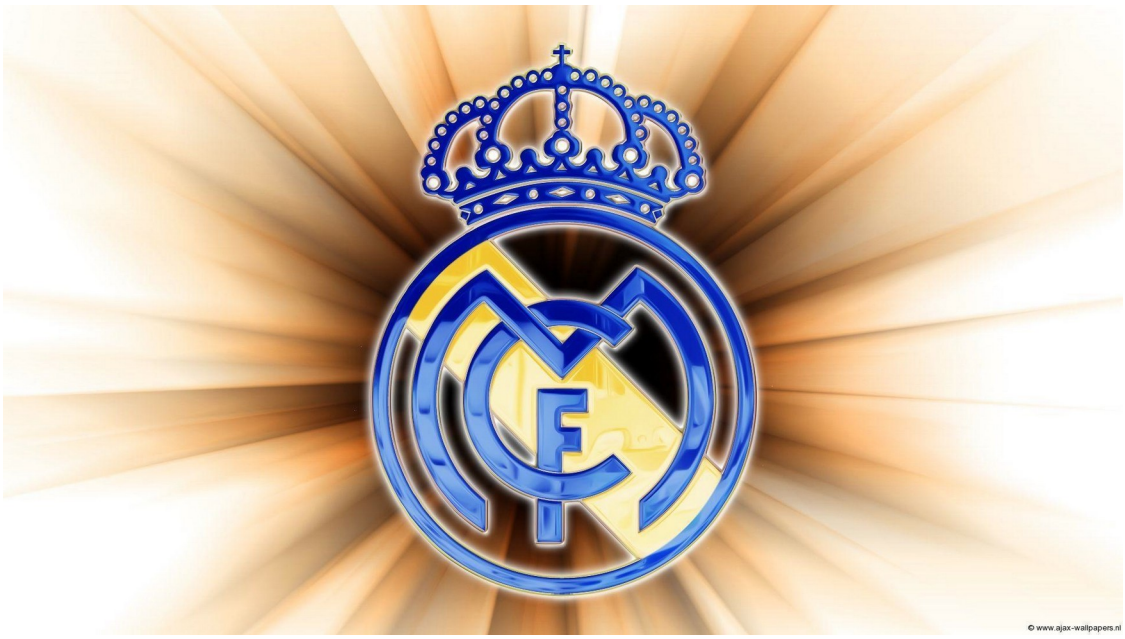
```
new_img = cv2.bitwise_or(img, img2)
cv2_imshow(new_img)
```




```
new_img = cv2.bitwise_xor(img, img2)
cv2_imshow(new_img)
```



```
new_img = cv2.bitwise_not(img)
cv2_imshow(new_img)
```




```
new_img = cv2.bitwise_not(img2, mask=None)
cv2_imshow(new_img)
```