

```
In [1]: pip install opencv-python
```

```
Requirement already satisfied: opencv-python in c:\users\my pc\anaconda3\lib\site-packages (4.9.0.80)
Requirement already satisfied: numpy>=1.17.0 in c:\users\my pc\anaconda3\lib\site-packages (from opencv-python) (1.23.5)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 23.1.2 -> 23.3.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
In [2]: import cv2
```

```
C:\Users\my pc\anaconda3\lib\site-packages\numpy\_distributor_init.py:30: UserWarning:
loaded more than 1 DLL from .libs:
C:\Users\my pc\anaconda3\lib\site-packages\numpy\.libs\libopenblas.FB5AE2TYXYH2IJRDKGDG
Q3XBKLTF43H.gfortran-win_amd64.dll
C:\Users\my pc\anaconda3\lib\site-packages\numpy\.libs\libopenblas64__v0.3.23-gcc_10_3_
0.dll
    warnings.warn("loaded more than 1 DLL from .libs:")
```

```
In [3]: import matplotlib.pyplot as plt
import numpy as np
```

```
In [4]: img1 = cv2.imread("DataS/OpenCV_DATA/dog_backpack.webp")
img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
img2 = cv2.imread("DataS/OpenCV_DATA/watermark_no_copy.png")
img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)
```

```
In [5]: img1.shape
```

```
Out[5]: (252, 276, 3)
```

```
In [6]: img2.shape
```

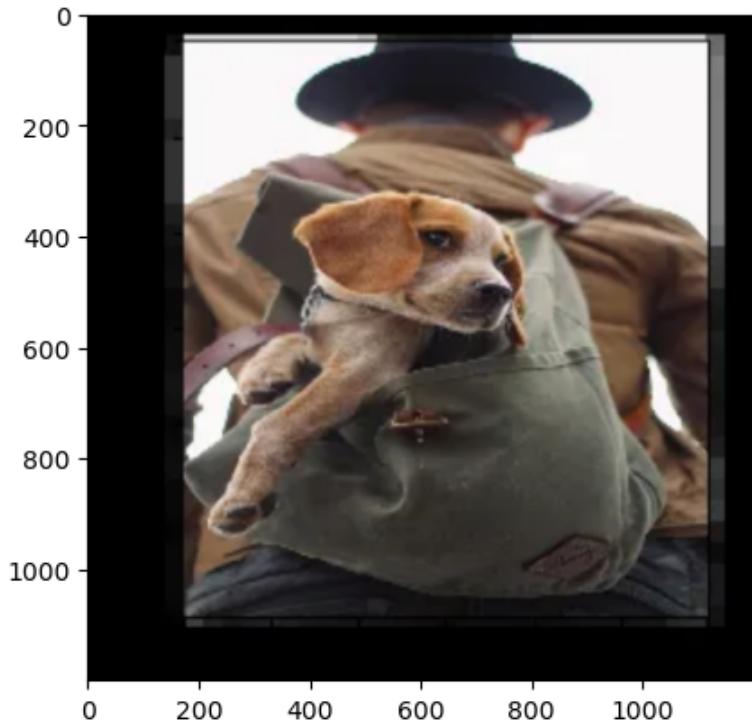
```
Out[6]: (252, 276, 3)
```

```
In [7]: # Blending the image to same size :
```

```
img1 = cv2.resize(img1,(1200,1200))
img2 = cv2.resize(img2,(1200,1200))
```

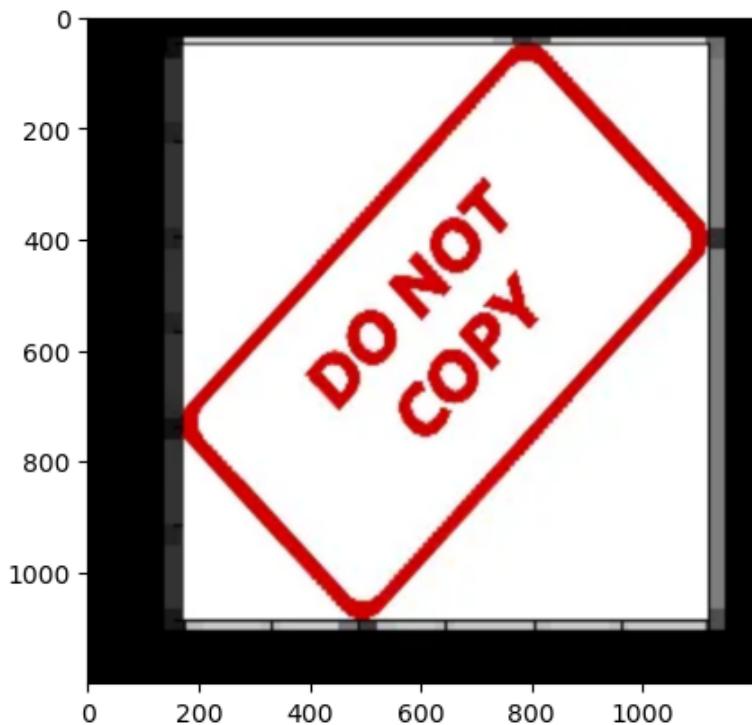
```
In [8]: plt.imshow(img1)
```

```
Out[8]: <matplotlib.image.AxesImage at 0x26ddaaec3a0>
```



```
In [9]: plt.imshow(img2)
```

```
Out[9]: <matplotlib.image.AxesImage at 0x26ddaf9ceb0>
```



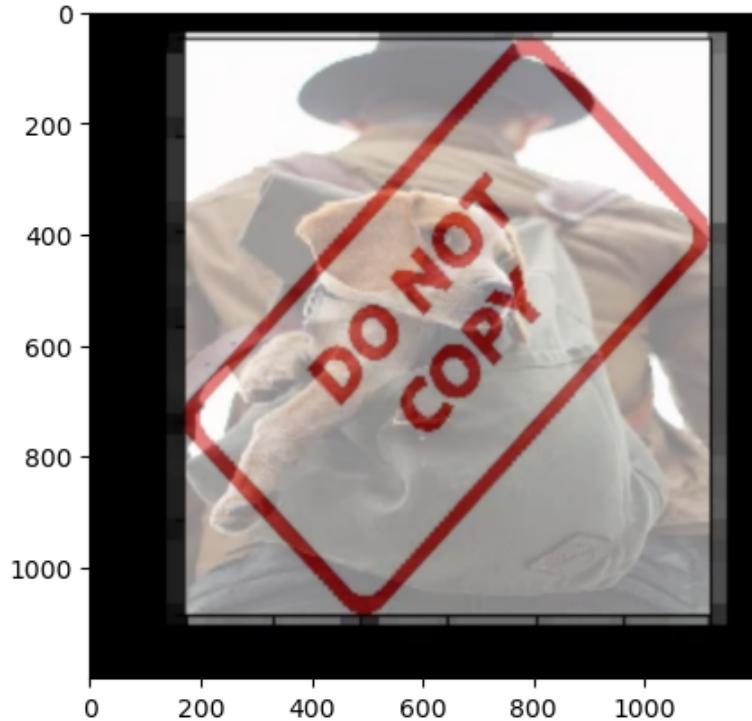
```
In [10]: img1.shape
```

```
Out[10]: (1200, 1200, 3)
```

```
In [11]: blended = cv2.addWeighted(src1 = img1, alpha = 0.5, src2 = img2, beta = 0.5, gamma = 0.5)
```

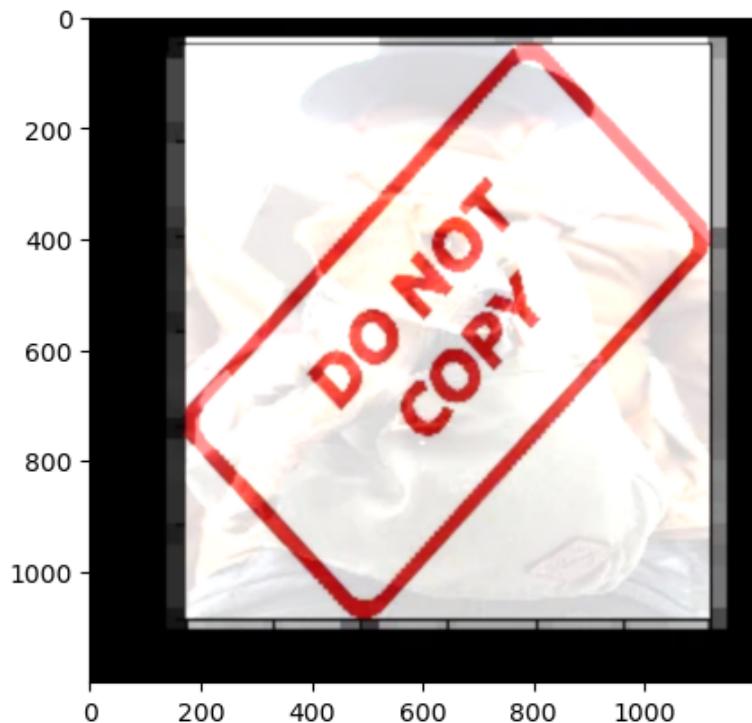
```
In [12]: plt.imshow(blended)
```

```
Out[12]: <matplotlib.image.AxesImage at 0x26ddb012550>
```



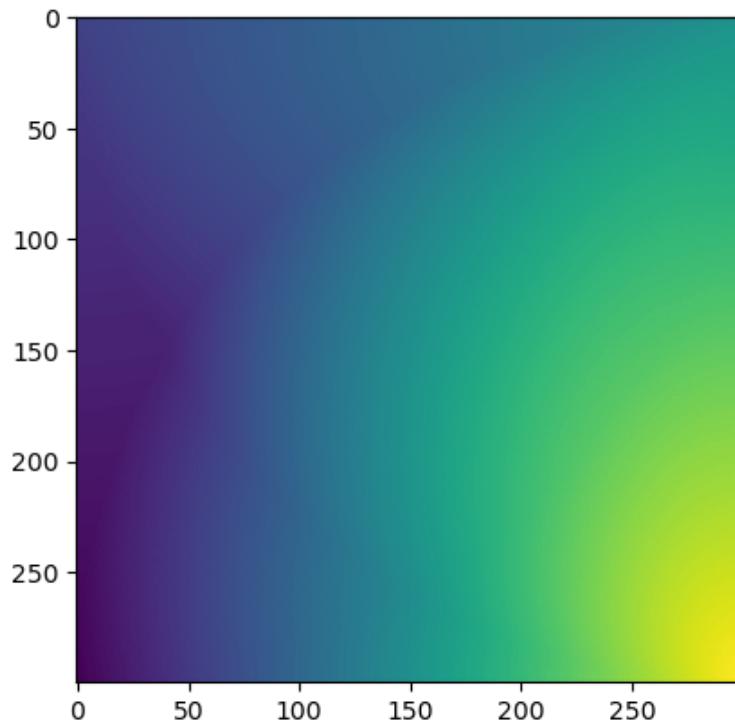
```
In [19]: blended = cv2.addWeighted(src1 = img1, alpha = 0.6, src2 = img2, beta = 0.8, gamma = 0.5)  
plt.imshow(blended)
```

```
Out[19]: <matplotlib.image.AxesImage at 0x26ddd650c40>
```



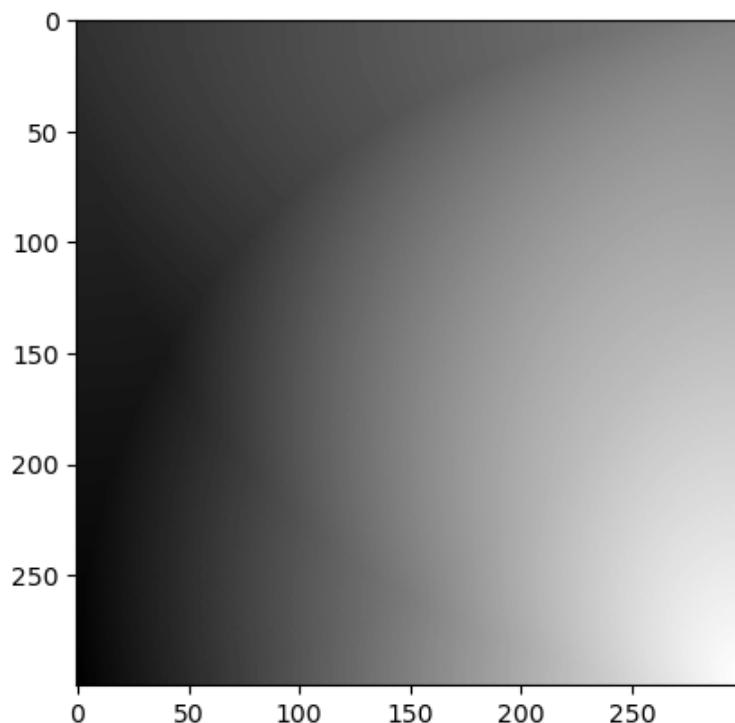
```
In [32]: img = cv2.imread("DataS/OpenCV_DATA/rainbow.jpg",0)
plt.imshow(img)
```

Out[32]: <matplotlib.image.AxesImage at 0x26dddfaca2b0>



```
In [33]: plt.imshow(img,cmap ='gray')
```

Out[33]: <matplotlib.image.AxesImage at 0x26ddfb29160>



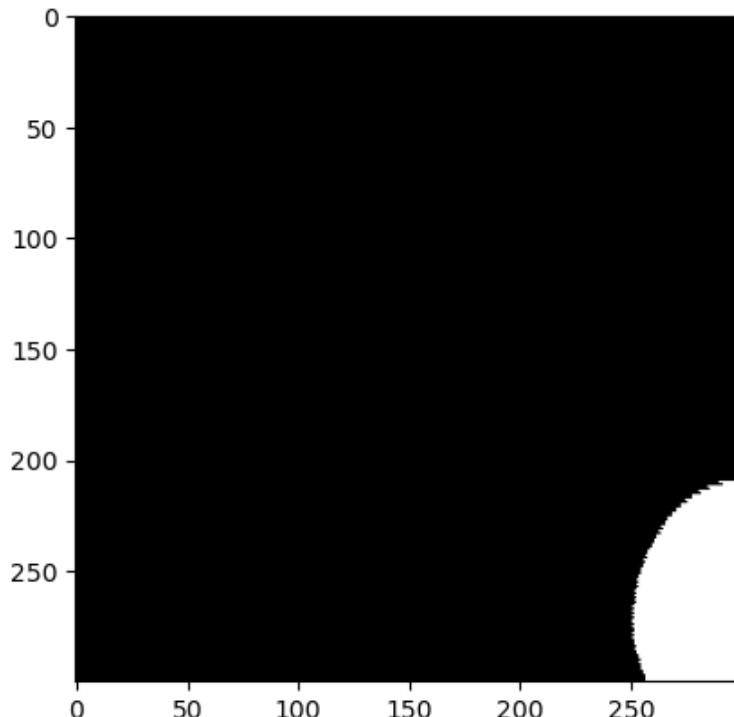
```
In [34]: ret,thresh1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
```

```
In [35]: ret
```

```
Out[35]: 127.0
```

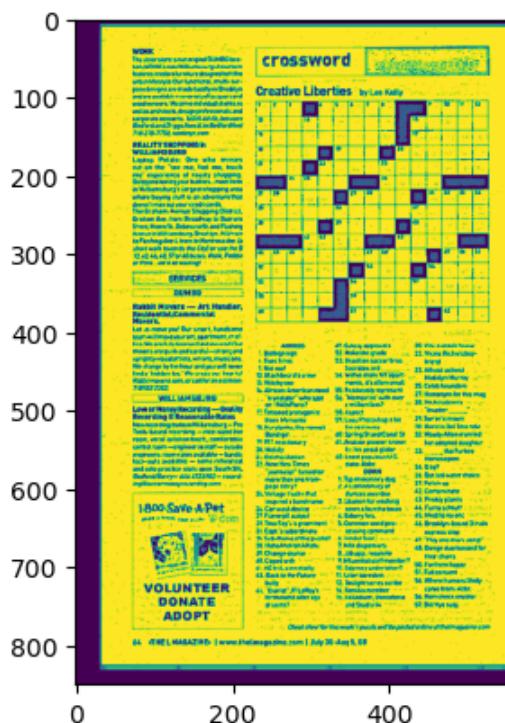
```
In [36]: plt.imshow(thresh1, cmap = 'gray')
```

```
Out[36]: <matplotlib.image.AxesImage at 0x26de0cdb1f0>
```



```
In [101]: img3 = cv2.imread("DataS/OpenCV_DATA/crossword.webp",0)
plt.imshow(img3)
```

```
Out[101]: <matplotlib.image.AxesImage at 0x26de2aa4d30>
```



```
In [102]: def show_pic(img3):
    fig = plt.figure(figsize = (15,15))
    ax = fig.add_subplot(111)
    ax.imshow(img3,cmap = 'gray')
```

```
In [103]: show_pic(img3)
```



```
In [113]: def load_img():
    img = cv2.imread("DataS/OpenCV_DATA/bricks.webp").astype(np.float32)/255
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    return(img)
```

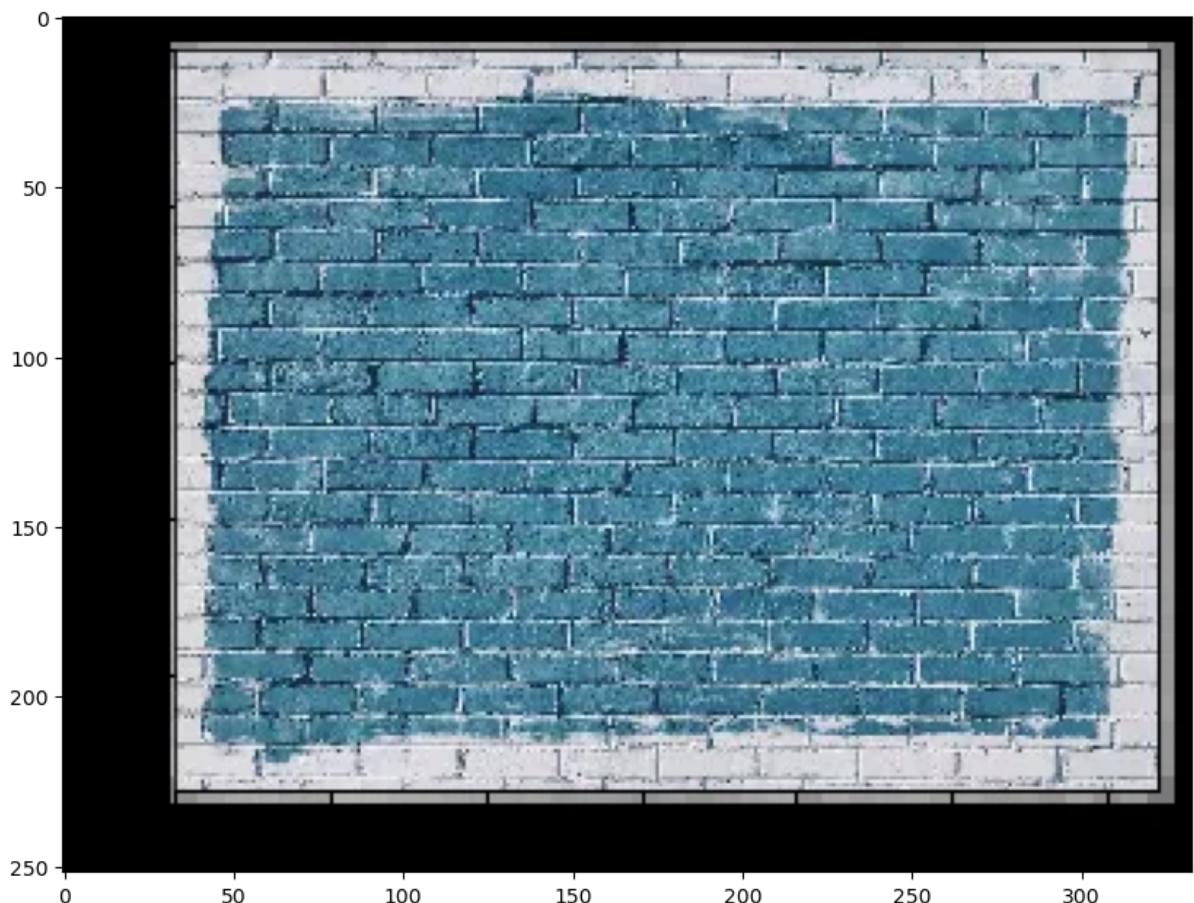
```
In [105]: img = cv2.resize(img,(1200,1200))
```

```
In [106]: img.shape
```

```
Out[106]: (1200, 1200, 3)
```

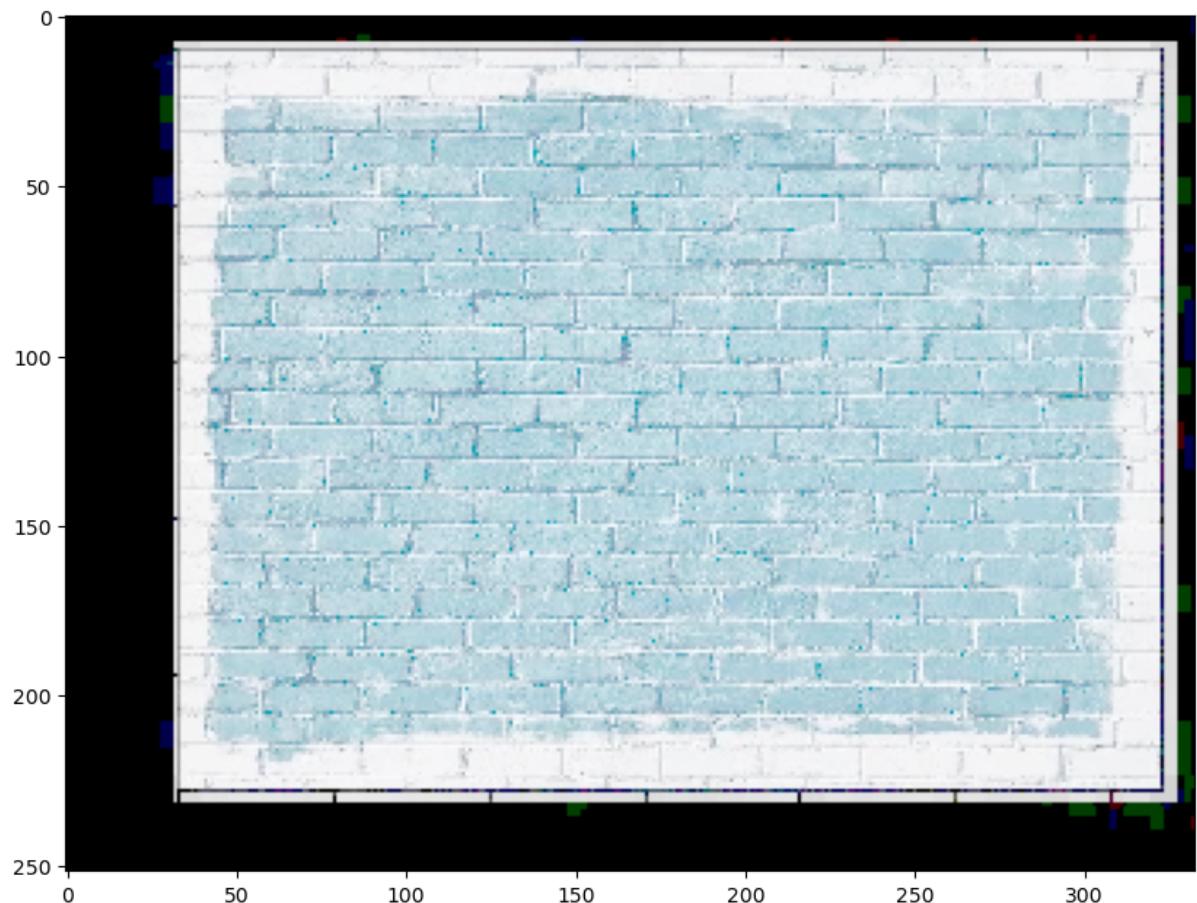
```
In [114]: def show_pic(img):
    fig = plt.figure(figsize = (10,10))
    ax = fig.add_subplot(111)
    ax.imshow(img)
```

```
In [115]: i = load_img()
show_pic(i)
```

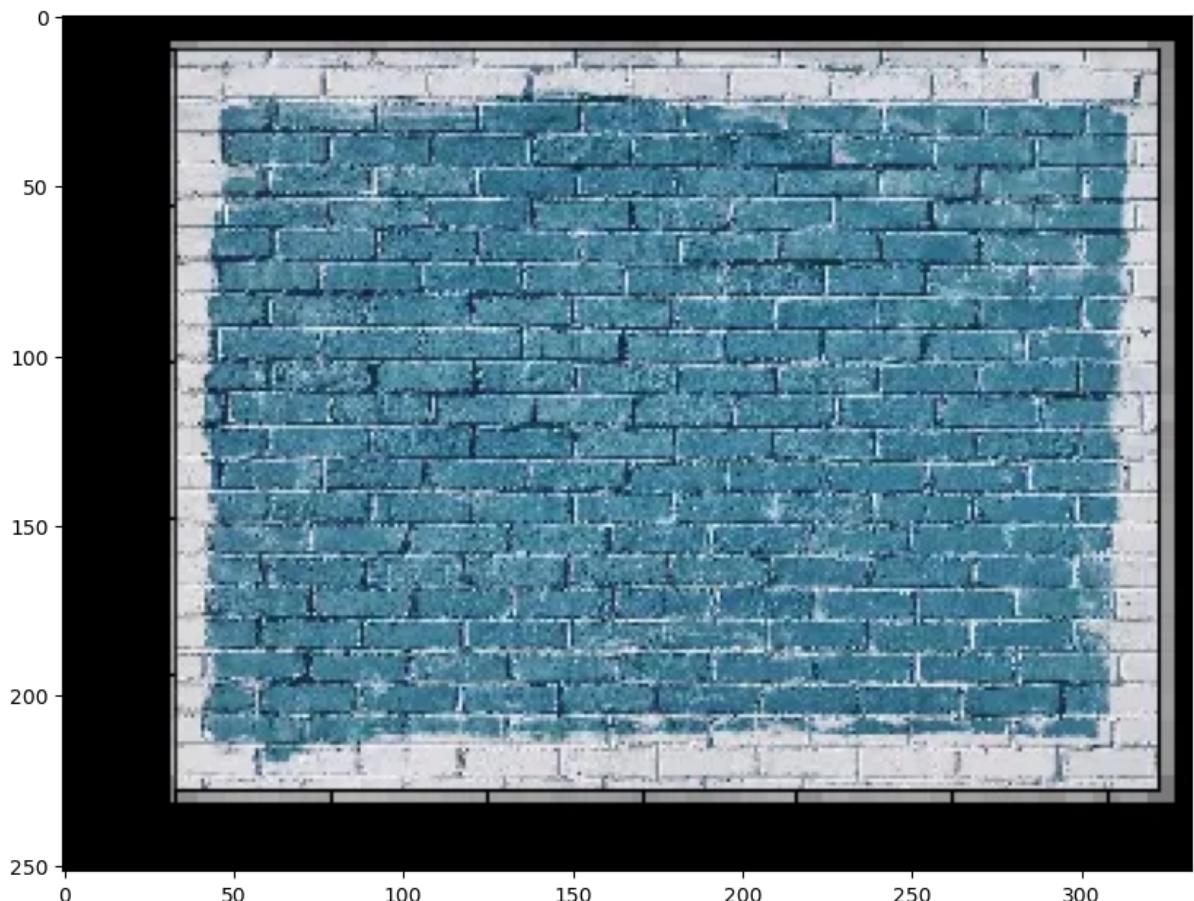


```
In [109]: gamma = 1/4
```

```
In [110]: result = np.power(i,gamma)  
show_pic(result)
```



```
In [116]: img = load_img()
font = cv2.FONT_HERSHEY_COMPLEX
cv2.putText(img, text = "Bricks" , org = (10,600), fontFace=font, fontScale=10, color=(255,0,0),
            thickness = 4)
show_pic(img)
```



```
In [124]: cap = cv2.VideoCapture(0)
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

while True:
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow('frame', gray)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

```
In [*]: cap = cv2.VideoCapture(0)
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

while True:
    ret,frame = cap.read()
    #gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow('frame',frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

In []:

In []: