```python
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
from skimage.metrics import
mean_squared_error,peak_signal_noise_ratio,structural_similarity
```

```python
img1=cv2.imread("/content/Screenshot (2).png")
img2=cv2.imread("/content/Screenshot (3).png")
```

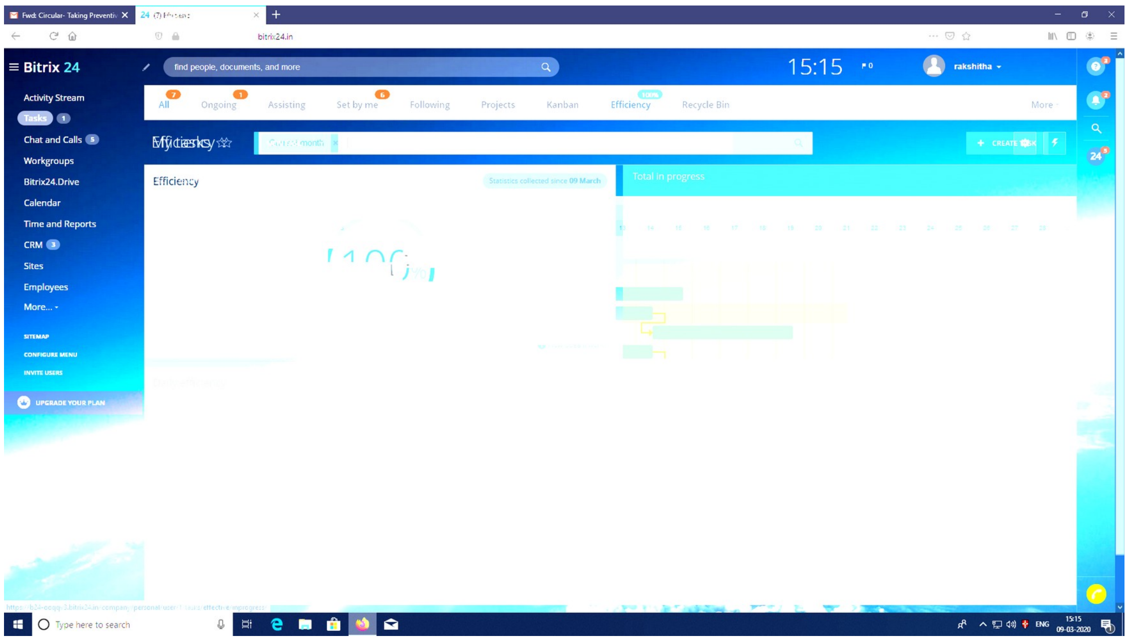
```python
img1.shape
```

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

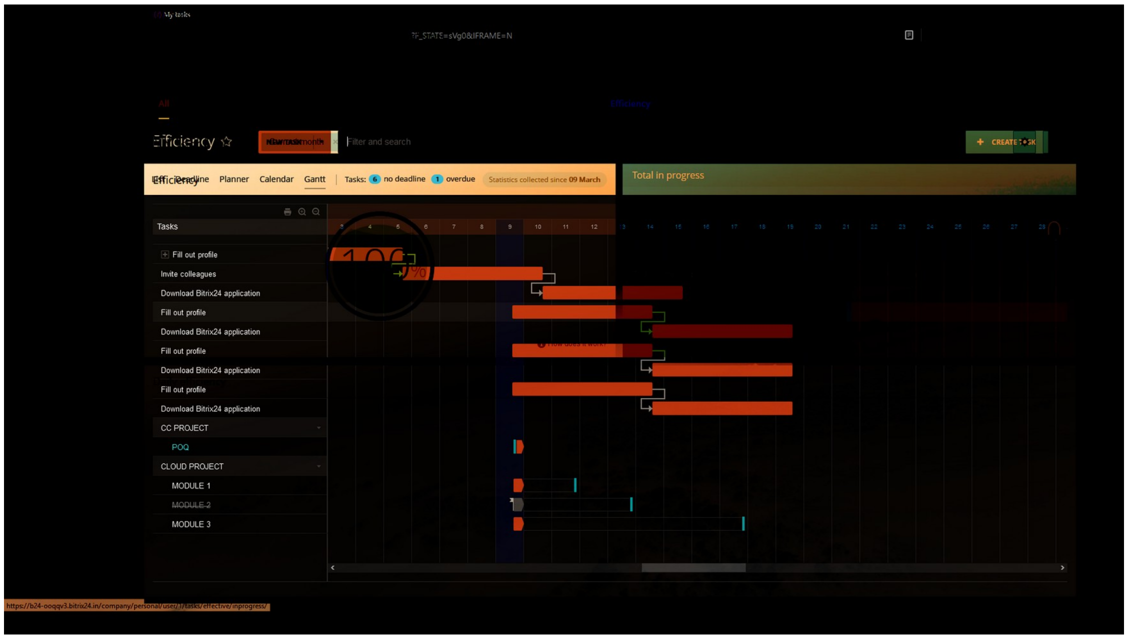
```python
img2.shape
```

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

```python
img3=cv2.add(img1,img2)
cv2_imshow(img3)
```

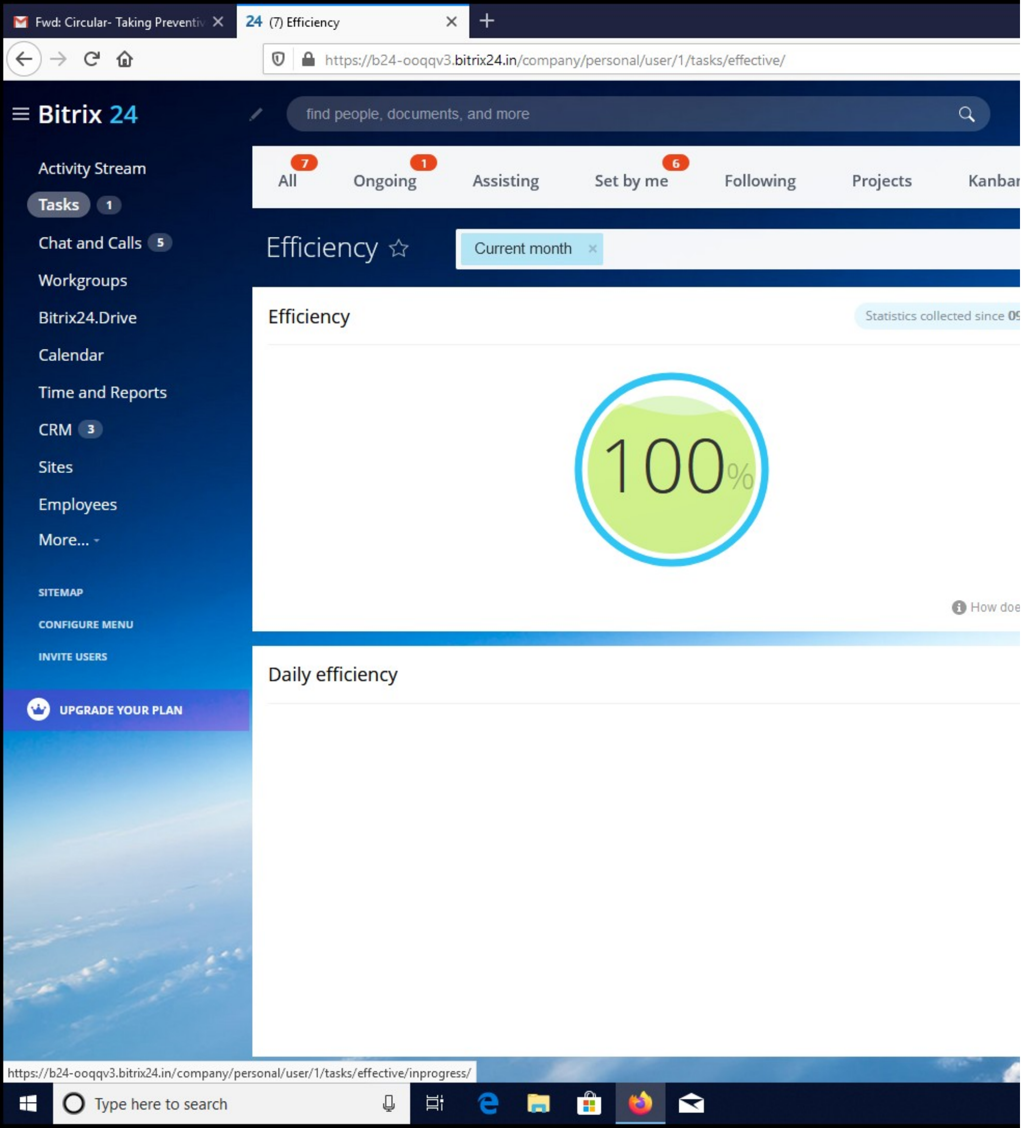


```python
img4=cv2.subtract(img1,img2)
cv2_imshow(img4)
```


Efficiency
Filter and search
CREATE TASK
Deadline Planner Calendar Gantt
Tasks: 6 no deadline 1 overdue
Statistics collected since 09 March
Total in progress
Tasks
Fill out profile
Invite colleagues
Download Bitrix24 application
Fill out profile
Download Bitrix24 application
Fill out profile
Download Bitrix24 application
Fill out profile
Download Bitrix24 application
CC PROJECT
POQ
CLOUD PROJECT
MODULE 1
~~MODULE 2~~
MODULE 3
https://b24-ooqqv3.bitrix24.in/company/personal/user/1/tasks/effective/inprogress/

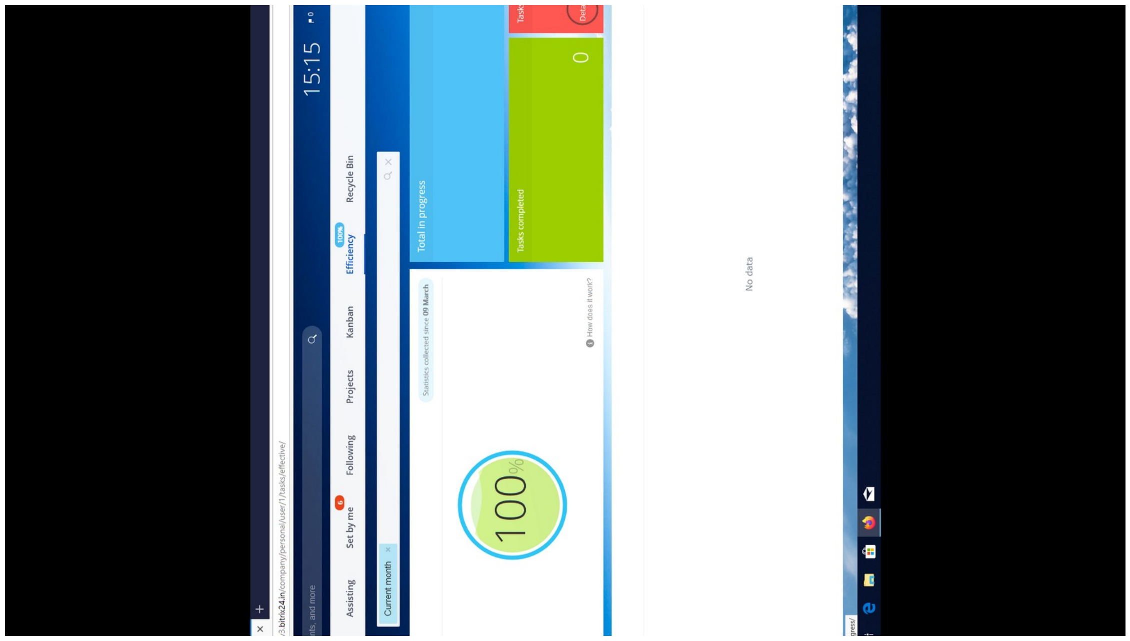
```
M = np.float32([[1,0,100],[0,1,50]])

dst = cv2.warpAffine(img1,M,(1080,1920))
cv2_imshow(dst)
```

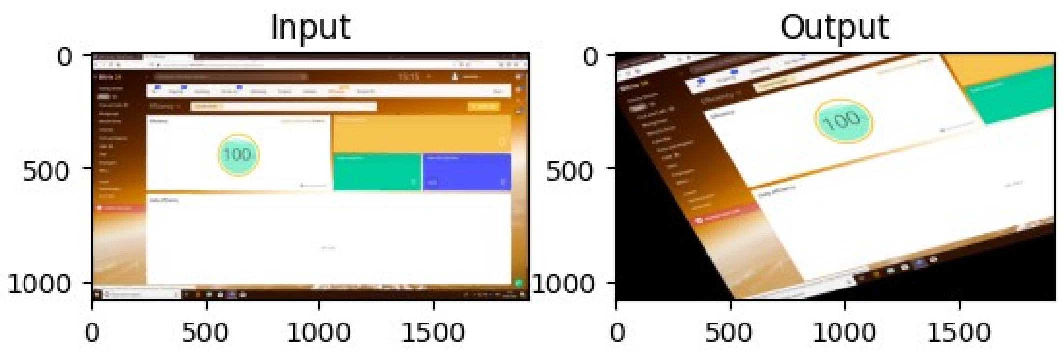


Fwd: Circular- Taking Preventiv
(7) Efficiency
https://b24-ooqqv3.bitrix24.in/company/personal/user/1/tasks/effective/
Bitrix 24
find people, documents, and more
Activity Stream
Tasks 1
Chat and Calls 5
Workgroups
Bitrix24.Drive
Calendar
Time and Reports
CRM 3
Sites
Employees
More...
SITEMAP
CONFIGURE MENU
INVITE USERS
UPGRADE YOUR PLAN
All 7
Ongoing 1
Assisting
Set by me 6
Following
Projects
Kanbar
Efficiency
Current month
Efficiency
Statistics collected since 09
100%
How doe
Daily efficiency
https://b24-ooqqv3.bitrix24.in/company/personal/user/1/tasks/effective/inprogress/
Type here to search

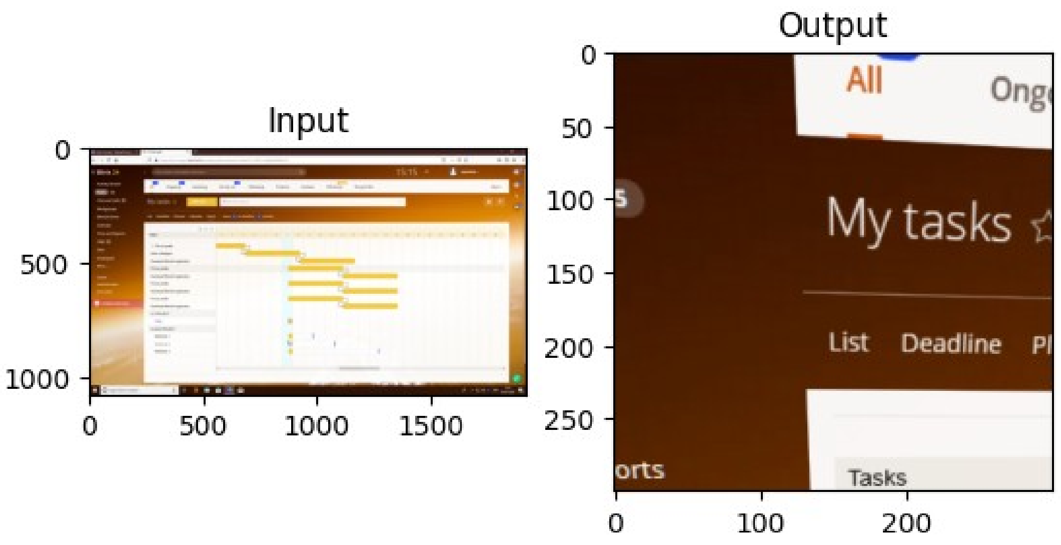
```
rows,cols,cha=img1.shape
M = cv2.getRotationMatrix2D(((cols-1)/2.0,(rows-1)/2.0),90,1)
dst = cv2.warpAffine(img1,M,(cols,rows))
cv2_imshow(dst)
```



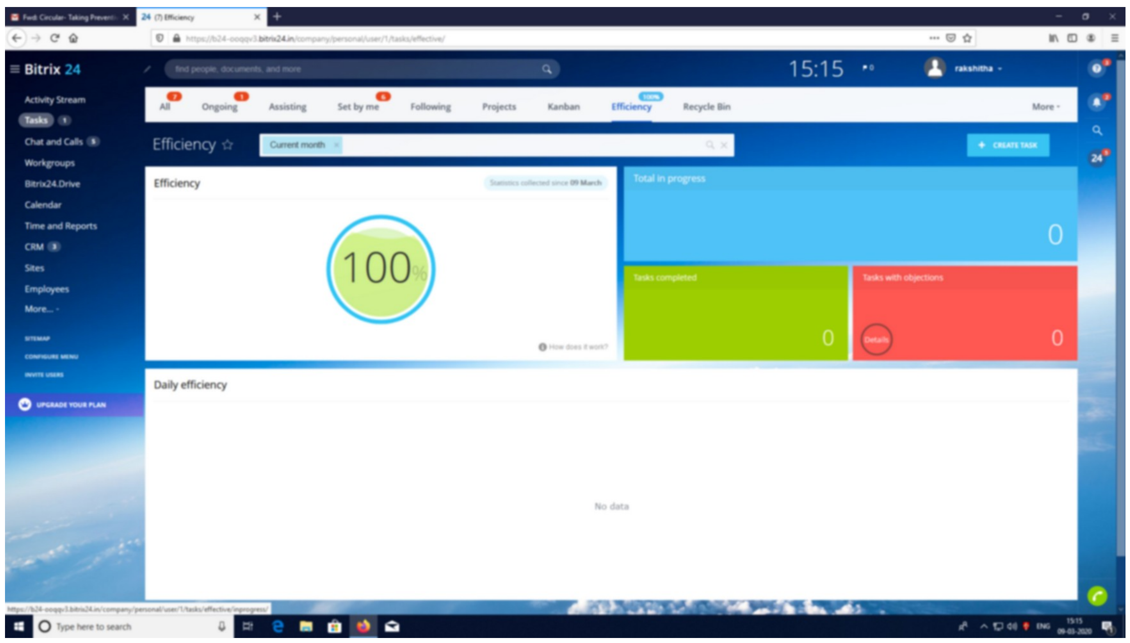
```
rows,cols,ch = img1.shape
pts1 = np.float32([[50,50],[200,50],[50,200]])
pts2 = np.float32([[10,100],[200,50],[100,250]])
M = cv2.getAffineTransform(pts1,pts2)
dst = cv2.warpAffine(img1,M,(cols,rows))
plt.subplot(121),plt.imshow(img1),plt.title('Input')
plt.subplot(122),plt.imshow(dst),plt.title('Output')
plt.show()
```



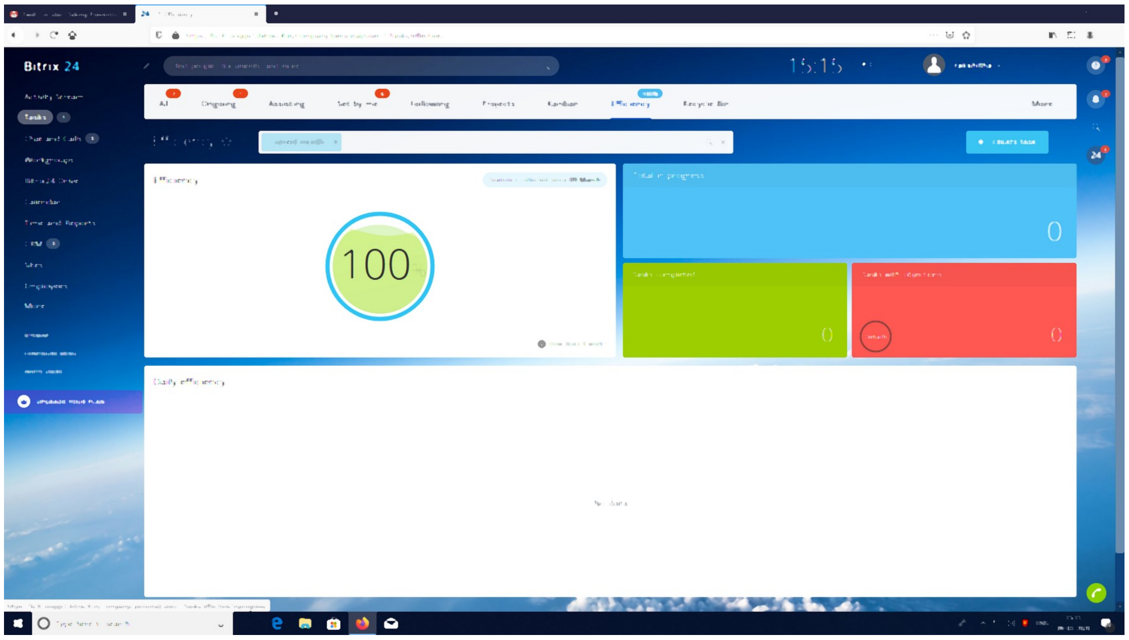
```
pts1 = np.float32([[156,165],[368,152],[128,387],[389,390]])
pts2 = np.float32([[0,0],[300,0],[0,300],[300,300]])
M = cv2.getPerspectiveTransform(pts1,pts2)
dst = cv2.warpPerspective(img2,M,(300,300))
plt.subplot(121),plt.imshow(img2),plt.title('Input')
plt.subplot(122),plt.imshow(dst),plt.title('Output')
plt.show()
```



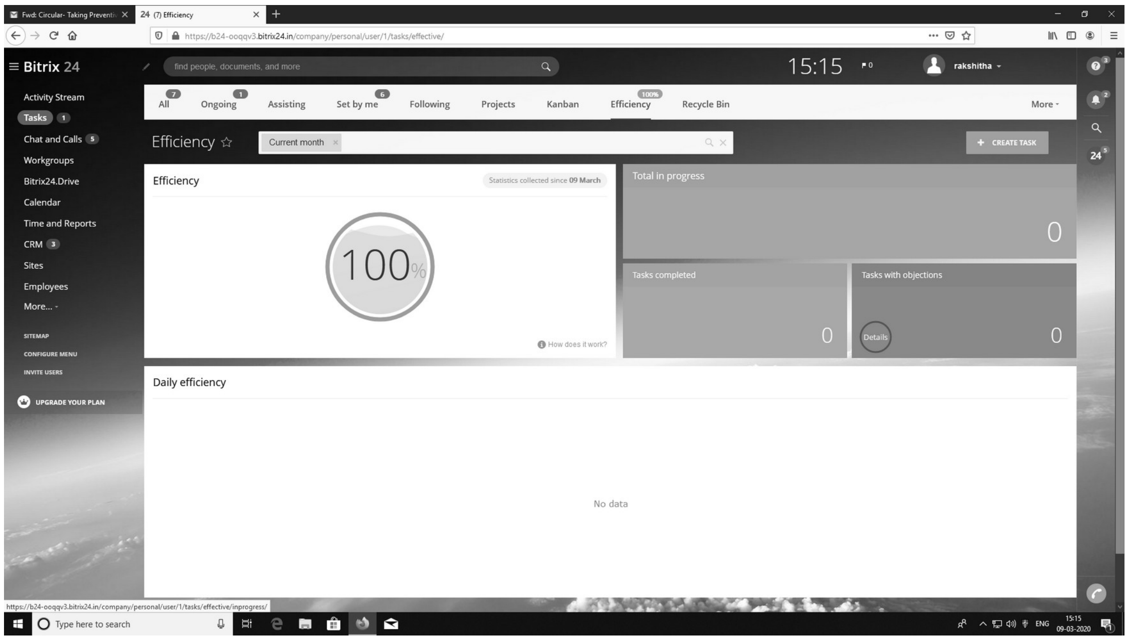
```
Gaussian = cv2.GaussianBlur(img1, (7, 7), 0)
cv2_imshow( Gaussian)
```



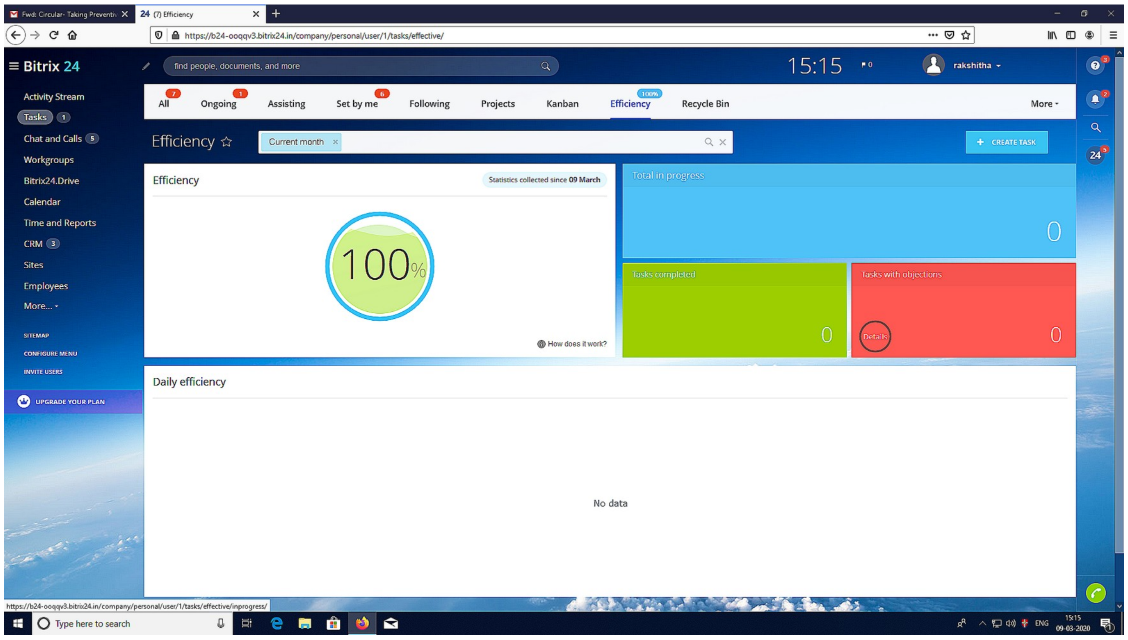
```
median = cv2.medianBlur(img1, 5)
cv2_imshow(median)
```



```
gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
cv2_imshow(gray)
```

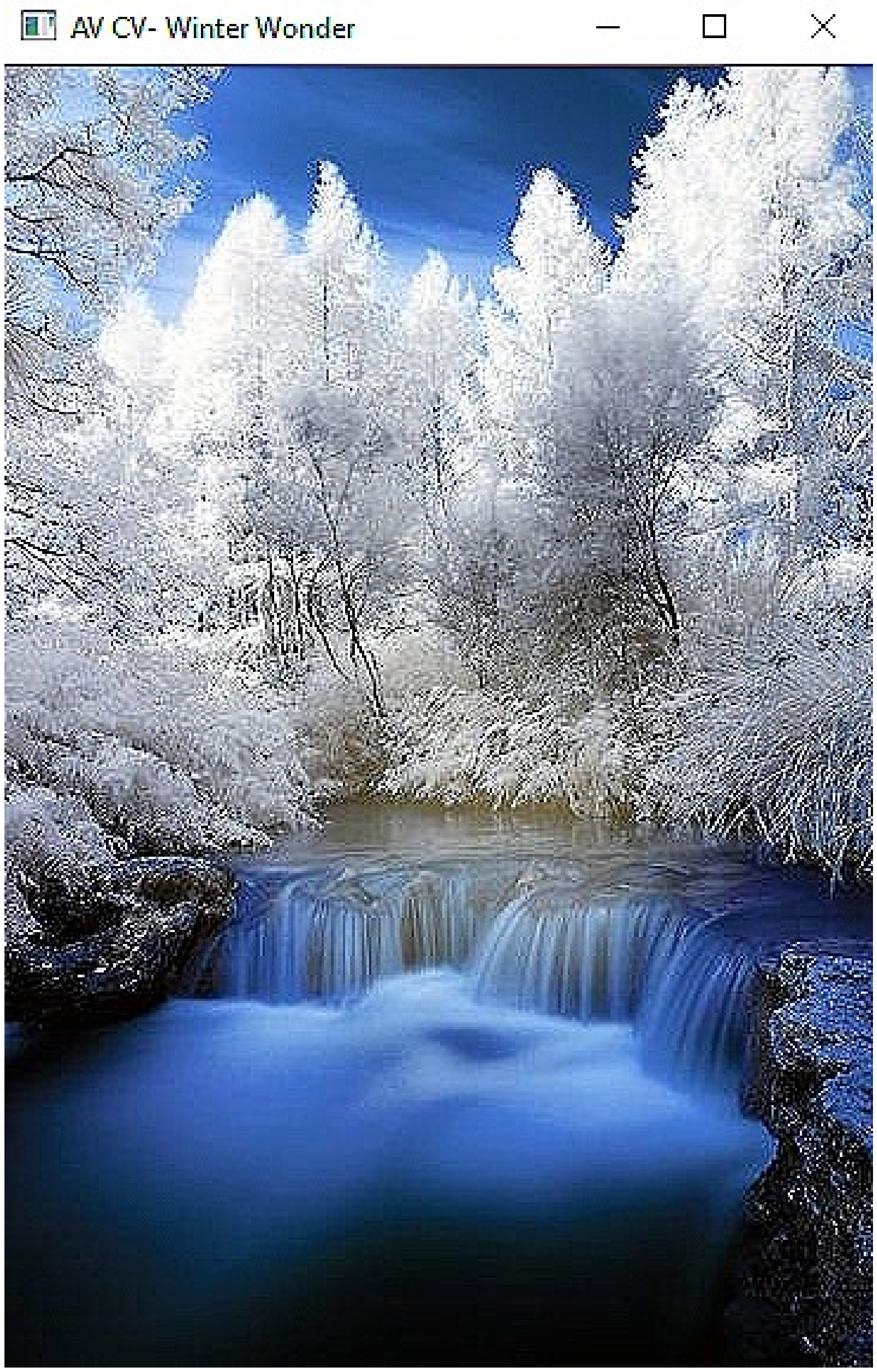


```
kernel = np.array([[0, -1, 0],
                   [-1, 5,-1],
                   [0, -1, 0]])
image_sharp = cv2.filter2D(src=img1, ddepth=-1, kernel=kernel)
cv2_imshow(image_sharp)
```



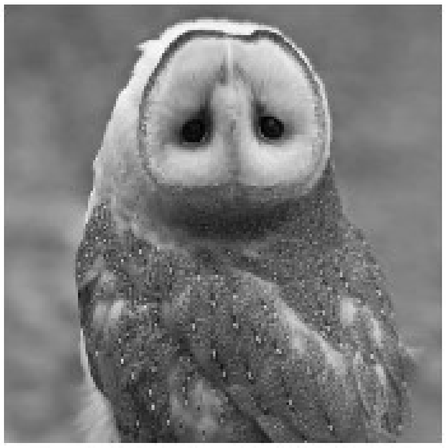
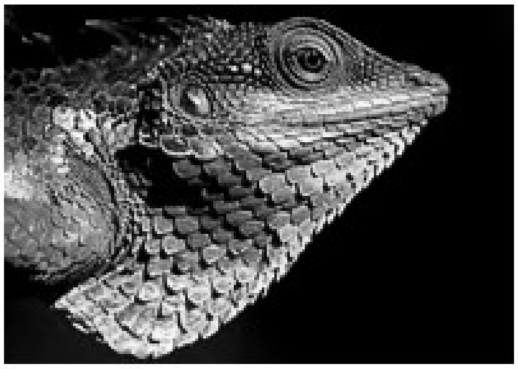
```
capimg=cv2.imread("/content/91190Capture.png")

kernel = np.array([[0, -1, 0],
                   [-1, 5,-1],
                   [0, -1, 0]])
image_sharp = cv2.filter2D(src=capimg, ddepth=-1, kernel=kernel)
cv2_imshow(image_sharp)
```



```
imga1=cv2.imread("/content/Screenshot-from-2022-07-28-16-15-43.png")
cv2_imshow(imga1)
```

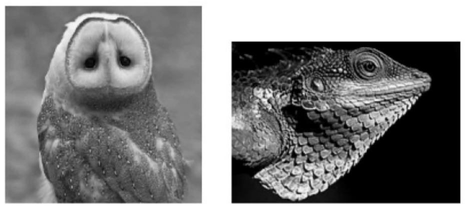
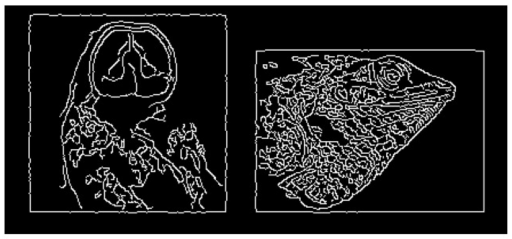
```
gray1=cv2.cvtColor(imga1,cv2.COLOR_BGR2GRAY)
cv2_imshow(gray1)
```

```
blurred = cv2.GaussianBlur(gray1, (5, 5), 0)
# Perform the canny operator
canny = cv2.Canny(blurred, 30, 150)

fig,ax =  plt.subplots(1,2,figsize=(18, 18))
ax[0].imshow(gray1,cmap='gray')
ax[1].imshow(canny,cmap='gray')
ax[0].axis('off')
ax[1].axis('off')
```

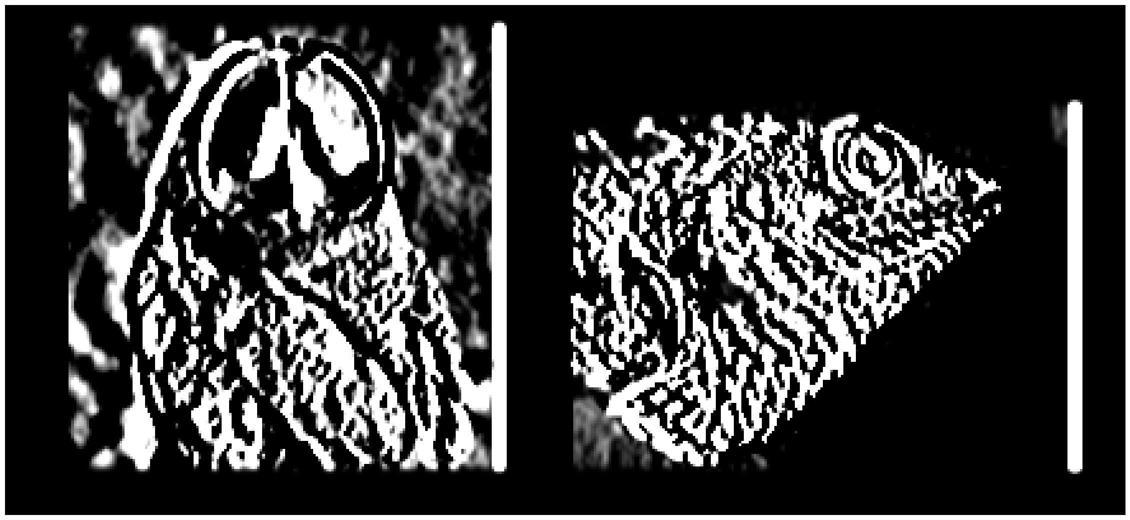
```
(-0.5, 439.5, 199.5, -0.5)
```

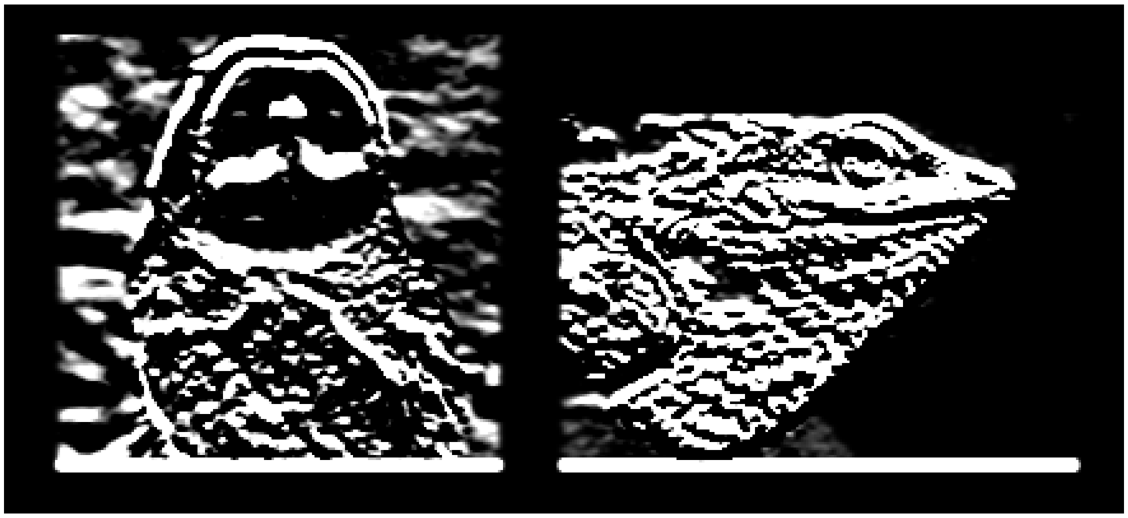
```
# Convert to graycsale
img_gray = cv2.cvtColor(imga1, cv2.COLOR_BGR2GRAY)
# Blur the image for better edge detection
img_blur = cv2.GaussianBlur(img_gray, (3,3), 0)

sobelx = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=0,
ksize=5) # Sobel Edge Detection on the X axis
sobely = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=0, dy=1,
ksize=5) # Sobel Edge Detection on the Y axis
sobelxy = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=1,
ksize=5)

cv2_imshow( sobelx);
```



```
cv2_imshow( sobely);
```



```
cv2_imshow( sobelxy);
```