```
!pip install opencv-python-headless
```

```
Requirement already satisfied: opencv-python-headless in /usr/local/lib/python3.11/dist-packages (4.11.0.86)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.11/dist-packages (from opencv-python-headless) (2.0.2)
```

```python
import cv2
```

```python
import numpy as np
```

```python
from google.colab.patches import cv2_imshow
```

```python
from google.colab import files
```

```python
from PIL import Image
```

```python
import io
```

```python
#Upload an image
uploaded = files.upload()
```

```
Choose files  2025-04-04.png
    • 2025-04-04.png(image/png) - 2497718 bytes, last modified: 04/04/2025 - 100% done
    Saving 2025-04-04.png to 2025-04-04 (1).png
```

```python
#Load and resize the image
for fn in uploaded.keys():
  image_path = fn
```

```python
img = cv2.imread(image_path)
img = cv2.resize(img, (600,600))
```

```python
#Convert to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Start coding or generate with AI.

```python
#Apply median blur
gray_blur = cv2.medianBlur(gray, 5)
```

```python
#Detect edges using adaptive thresholding
edges = cv2.adaptiveThreshold(
    gray_blur, 255,
    cv2.ADAPTIVE_THRESH_MEAN_C,
    cv2.THRESH_BINARY,
    blockSize = 9,
    C=9
)
```

```python
#Apply bilateral filter for smooth colour
colour = cv2.bilateralFilter(img, d=9, sigmaColor=300, sigmaSpace=300)
```
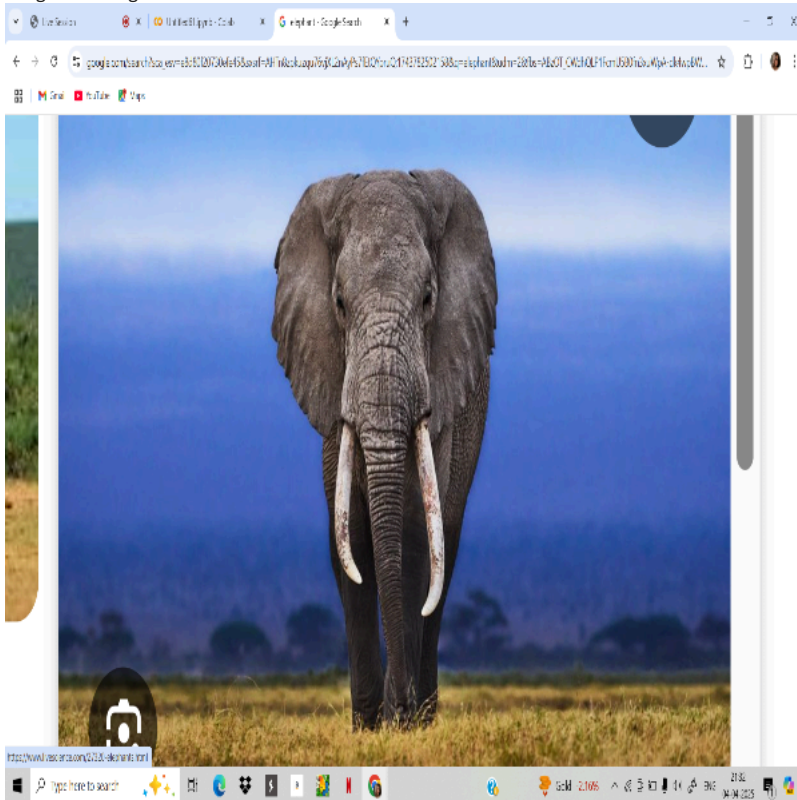
```python
#Combine color image with edge mark
cartoon = cv2.bitwise_and(colour, colour, mask=edges)
```
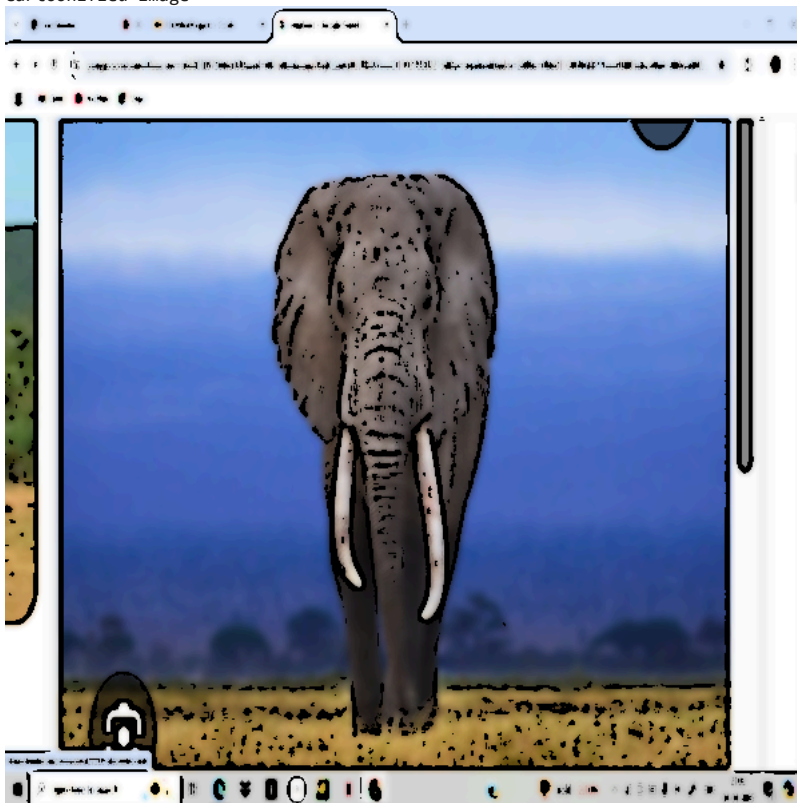
```python
#Show original and cartoonified images
print("Original image:")
cv2_imshow(img)
```

Original image:


```
print("Cartoonified image")
cv2_imshow(cartoon)
```

Cartoonified image


```
#Save and download the cartoon image
cv2.imwrite("Cartoonified_image.jpg", cartoon)
files.download("Cartoonified_image.jpg")
```

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.