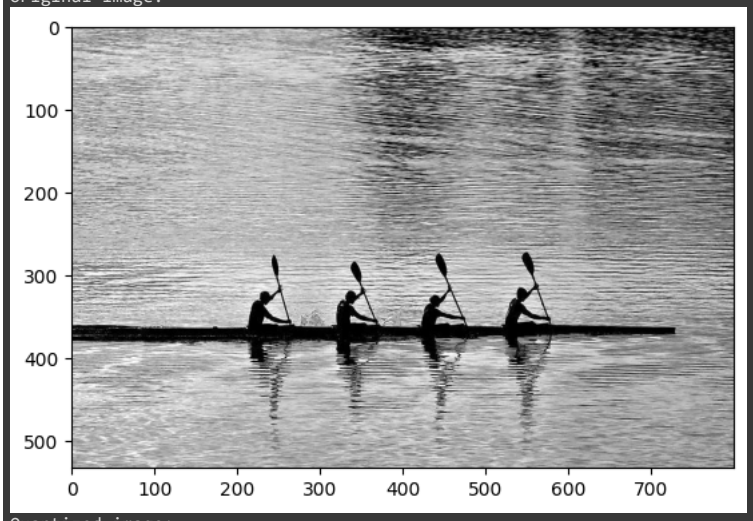
Quantization Techinque

1st Sample

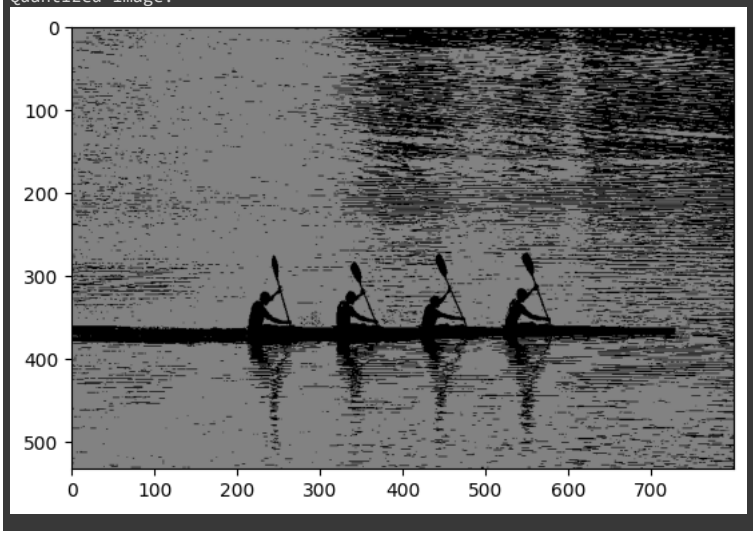
```
import numpy as np
from skimage import data
from matplotlib import pyplot as plt
from google.colab import files

uploaded = files.upload()
image_path = next(iter(uploaded.keys()))
image = io.imread(image_path)
print("Original image:")
plt.imshow(image)
plt.show()
ratio = 130
for i in range(image.shape[0]):
    for j in range(image.shape[1]):
        for k in range(image.shape[2]):
            image[i][j][k] = int(image[i][j][k] / ratio) * ratio
print("Quantized image:")
plt.imshow(image)
plt.show()
```

```
Choose Files No file chosen
• Image1.jpg(image/jpeg) - 118727 bytes, last modified: 1/22/2024 - 100% done
Saving Image1.jpg to Image1 (4).jpg
Original image:
```




```
Quantized image:
```



2nd Sample of Quantization Technique

```
import numpy as np
```

```
from skimage import data
from matplotlib import pyplot as plt
from google.colab import files

uploaded = files.upload()
image_path = next(iter(uploaded.keys()))
image = io.imread(image_path)
print("Original image:")
plt.imshow(image)
plt.show()
ratio = 130
for i in range(image.shape[0]):
    for j in range(image.shape[1]):
        for k in range(image.shape[2]):
            image[i][j][k] = int(image[i][j][k] / ratio) * ratio
print("Quantized image:")
plt.imshow(image)
plt.show()
```

Choose Files No file chosen

- **Image2.png**(image/png) - 110865 bytes, last modified: 1/22/2024 - 100% done

```
Saving Image2.png to Image2 (3).png
Original image:
```


```
Quantized image:
```

3rd Sample of Quantization Technique

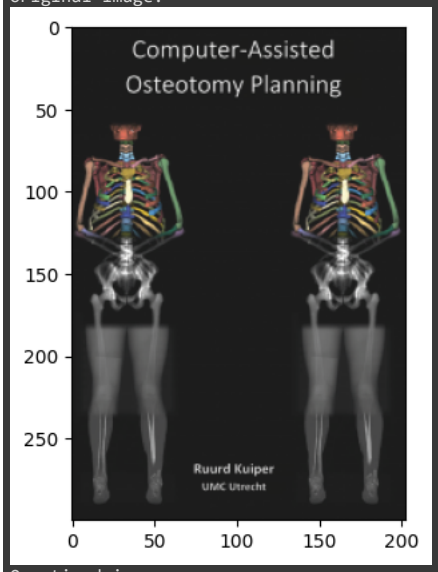
```
import numpy as np
from skimage import data
from matplotlib import pyplot as plt
from google.colab import files

uploaded = files.upload()
image_path = next(iter(uploaded.keys()))
image = io.imread(image_path)
print("Original image:")
plt.imshow(image)
plt.show()
ratio = 100
for i in range(image.shape[0]):
    for j in range(image.shape[1]):
        for k in range(image.shape[2]):
            image[i][j][k] = int(image[i][j][k] / ratio) * ratio
print("Quantized image:")
plt.imshow(image)
plt.show()
```

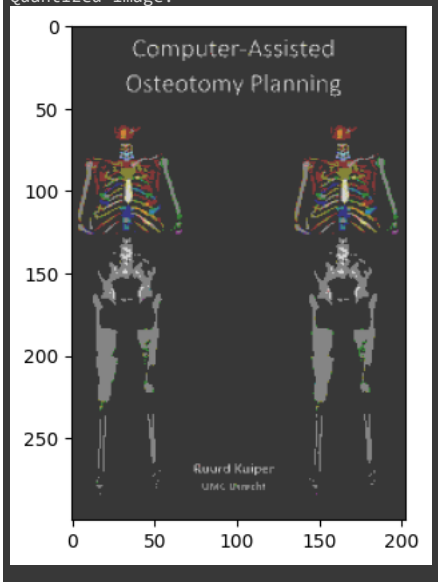
Choose Files No file chosen

- **image3.png**(image/png) - 64907 bytes, last modified: 1/22/2024 - 100% done

```
Saving image3.png to image3.png
Original image:
```




```
Quantized image:
```



4th Sample of Quantization Technique

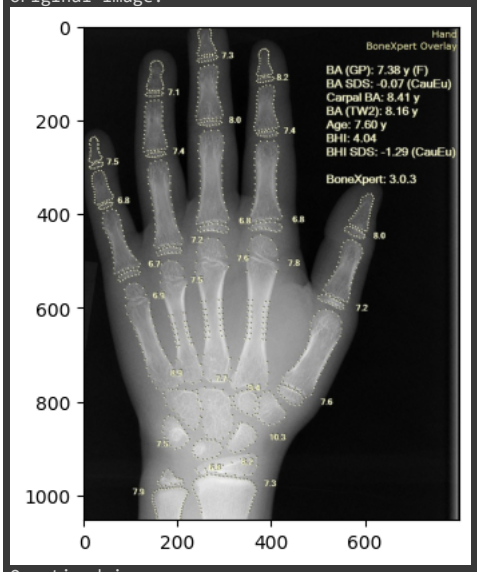
```
import numpy as np
from skimage import data
from matplotlib import pyplot as plt
from google.colab import files

uploaded = files.upload()
image_path = next(iter(uploaded.keys()))
image = io.imread(image_path)
print("Original image:")
plt.imshow(image)
plt.show()
ratio = 130
for i in range(image.shape[0]):
    for j in range(image.shape[1]):
        for k in range(image.shape[2]):
            image[i][j][k] = int(image[i][j][k] / ratio) * ratio
print("Quantized image:")
plt.imshow(image)
plt.show()
```

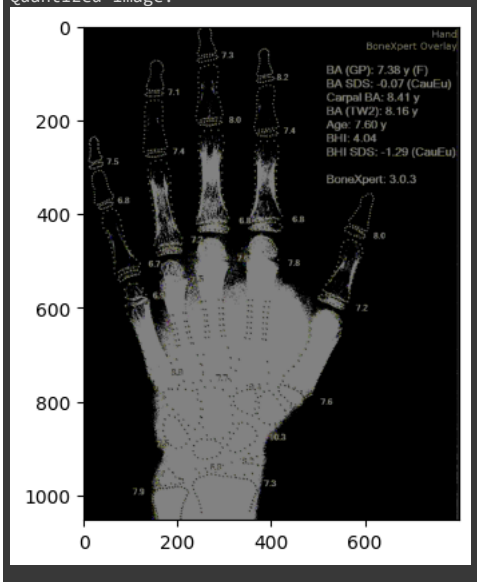
Choose Files No file chosen

- **image4.jpg**(image/jpeg) - 185755 bytes, last modified: 1/22/2024 - 100% done

```
Saving image4.jpg to image4.jpg
Original image:
```




```
Quantized image:
```

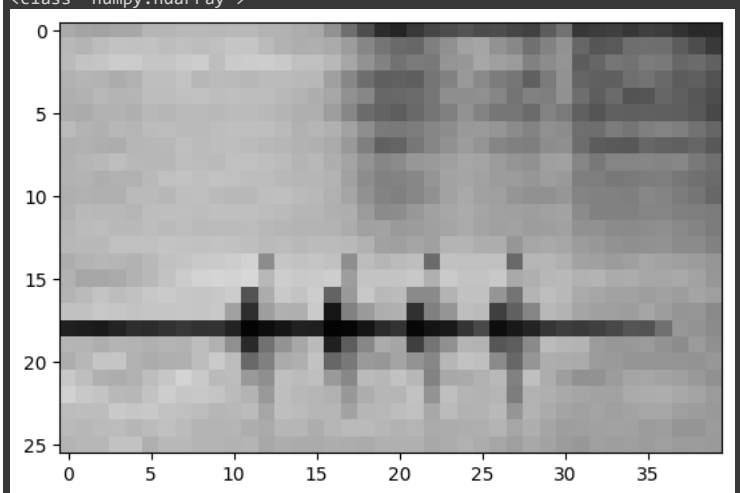


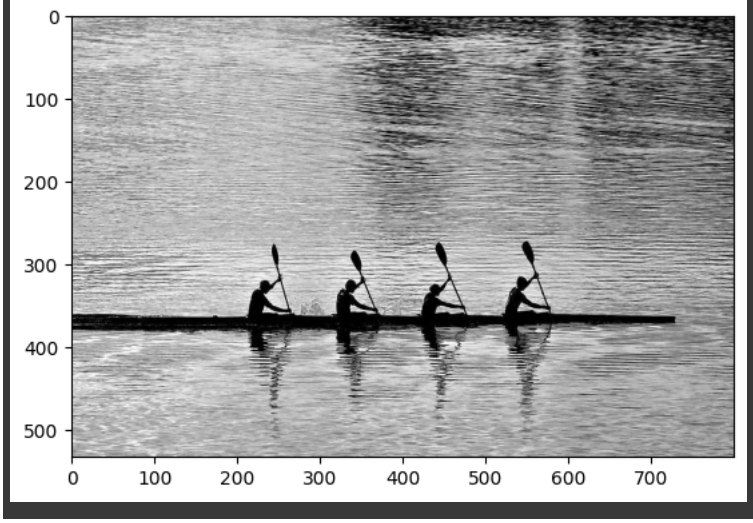
Sampling Techinque

1st Sample

```
import numpy as np
from skimage import data
from matplotlib import pyplot as plt
import imageio as img
uploaded = files.upload()
image_path = next(iter(uploaded.keys()))
image = io.imread(image_path)
print(image.shape)
print(type(image))
ratio=20
image1=np.zeros((int(image.shape[0]/ratio),
                 int(image.shape[1]/ratio),
                 image.shape[2]),dtype='float32')
for i in range(image1.shape[0]):
    for j in range(image1.shape[1]):
        for k in range(image1.shape[2]):
            delta=image[i*ratio:(i+1)*ratio,j*ratio:(j+1)*ratio,k]
            image1[i,j,k]=np.mean(delta)
plt.imshow(image1.astype('uint8'))
plt.show()
plt.imshow(image)
plt.show()
```

```
Choose Files No file chosen
• Image1.jpg(image/jpeg) - 118727 bytes, last modified: 1/22/2024 - 100% done
Saving Image1.jpg to Image1 (13).jpg
(533, 800, 3)
<class 'numpy.ndarray'>
```





2nd Sample of *Sampling* Technique

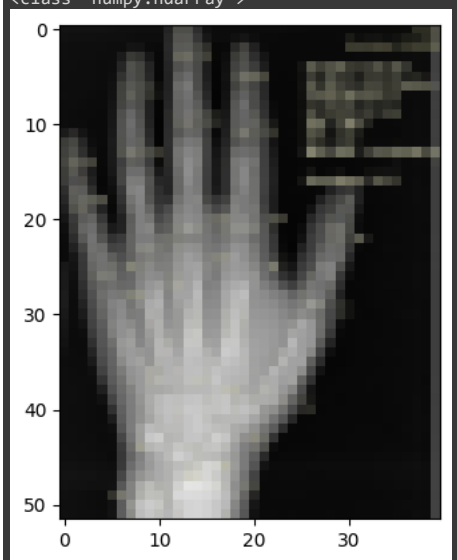
```
import numpy as np
from skimage import data
from matplotlib import pyplot as plt
import imageio as img
uploaded = files.upload()
```

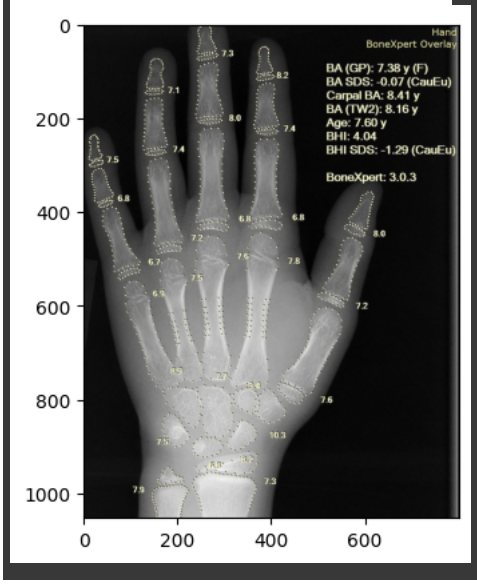
```
print(image.shape)
print(type(image))
ratio=20  #Here you can change the resolution of the image
image1=np.zeros((int(image.shape[0]/ratio),
                 int(image.shape[1]/ratio),
                 image.shape[2]),dtype='float32')
for i in range(image1.shape[0]):
    for j in range(image1.shape[1]):
        for k in range(image1.shape[2]):
            delta=image[i*ratio:(i+1)*ratio,j*ratio:(j+1)*ratio,k]
            image1[i,j,k]=np.mean(delta)
plt.imshow(image1.astype('uint8'))
plt.show()
plt.imshow(image)
plt.show()
```

Choose Files No file chosen

- **Image2.png**(image/png) - 110865 bytes, last modified: 1/22/2024 - 100% done

```
Saving Image2.png to Image2 (10).png
(1053, 800, 3)
<class 'numpy.ndarray'>
```

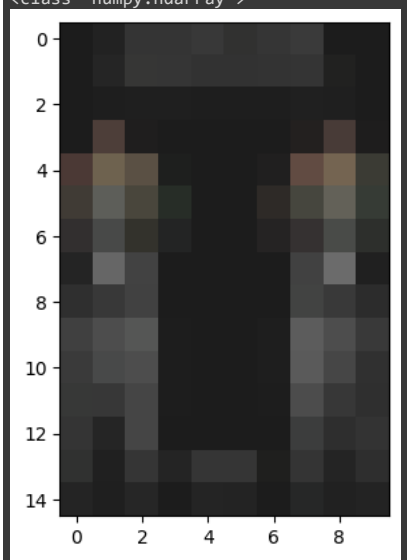


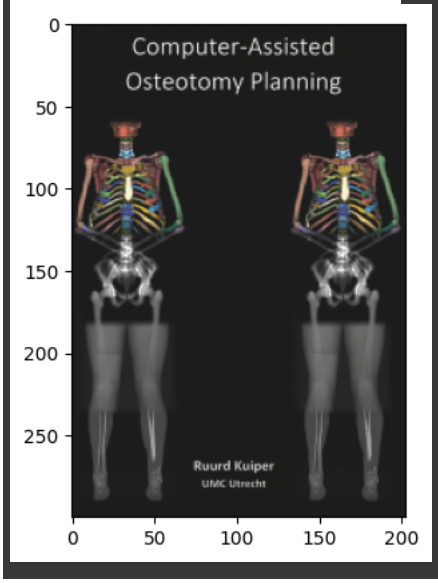


3rd Sample of Sampling Technique

```
import numpy as np
from skimage import data
from matplotlib import pyplot as plt
import imageio as img
uploaded = files.upload()
image_path = next(iter(uploaded.keys()))
image = io.imread(image_path)
print(image.shape)
print(type(image))
ratio=20
image1=np.zeros((int(image.shape[0]/ratio),
                 int(image.shape[1]/ratio),
                 image.shape[2]),dtype='float32')
for i in range(image1.shape[0]):
    for j in range(image1.shape[1]):
        for k in range(image1.shape[2]):
            delta=image[i*ratio:(i+1)*ratio,j*ratio:(j+1)*ratio,k]
            image1[i,j,k]=np.mean(delta)
plt.imshow(image1.astype('uint8'))
plt.show()
plt.imshow(image)
plt.show()
```

```
Choose Files No file chosen
• image3.png(image/png) - 64907 bytes, last modified: 1/22/2024 - 100% done
Saving image3.png to image3 (2).png
(300, 203, 4)
<class 'numpy.ndarray'>
```





4th Sample of Sampling Technique

```
import numpy as np
from skimage import data
from matplotlib import pyplot as plt
import imageio as img
uploaded = files.upload()
image_path = next(iter(uploaded.keys()))
image = io.imread(image_path)
print(image.shape)
print(type(image))
ratio=20
image1=np.zeros((int(image.shape[0]/ratio),
                 int(image.shape[1]/ratio),
                 image.shape[2]),dtype='float32')
for i in range(image1.shape[0]):
    for j in range(image1.shape[1]):
        for k in range(image1.shape[2]):
            delta=image[i*ratio:(i+1)*ratio,j*ratio:(j+1)*ratio,k]
            image1[i,j,k]=np.mean(delta)
plt.imshow(image1.astype('uint8'))
plt.show()
plt.imshow(image)
plt.show()
```

Choose Files No file chosen

- **image4.jpg**(image/jpeg) - 185755 bytes, last modified: 1/22/2024 - 100% done

```
Saving image4.jpg to image4 (3).jpg
(1053, 800, 3)
<class 'numpy.ndarray'>
```