

https://www.youtube.com/watch?v=5-jR3_demMM

<https://theailearner.com/tag/skeletonization-opencv/>

برای حل این سوال باید ابتدا تصویر را باینری کنیم و بعد فرمول زیر را روی آن اجرا کنیم:

$$S(A) = \bigcup_{k=0}^K S_k(A)$$

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

$$A \ominus kB = ((A \ominus B) \ominus B) \ominus \dots$$

$$K = \max\{k | (A \ominus kB) \neq \emptyset\}$$

این فرمول به ما میگوید که برای بدست آوردن اسکلت تصویر باید سایش تصویر را با عنصر ساختاری بدست بیاوریم و آن را ذخیره کنیم بعد عملگر باز با همان عنصر ساختاری را روی آن اعمال کنیم. سپس حاصل عملگر باز را از آن مقدار ذخیره شده کم کنیم. دفعه ی بعد همین مرحله ی سایش را به جای خود تصویر و عنصر ساختاری روی جواب سایش مرحله ی قبل و عنصر ساختاری انجام دهیم.

با هر بار تکرار باید بین جواب های قبلی و جواب دفعه ی جدید اجتماع بگیریم. تعداد تکرار مراحل تا زمانی است که پاسخ عملگر سایش تهی

نباشد.

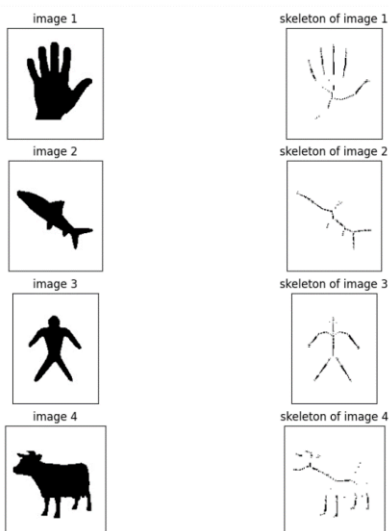
```
def get_skeleton(image):
    """
    Finds the skeleton of the input image.

    Parameters:
        image (numpy.ndarray): The input image.

    Returns:
        numpy.ndarray: The skeleton image.
        numpy.ndarray: The parameters required for reconstructing image
    """

    res = image.copy()
    params = []

    #Write your code here
    ret, thresh = cv2.threshold(res, 230, 255, cv2.THRESH_BINARY)
    img = cv2.bitwise_not(thresh)
    res = np.zeros((img.shape))
    SE = np.ones((3,3))
    Opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, SE)
    Sk = img - Opening
    res = Sk + res
    while cv2.countNonZero(Opening) != 0:
        img = cv2.erode(img, SE)
        params.append(Sk)
        Opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, SE)
        Sk = img - Opening
        res = Sk + res
    return 1 - res, params
```



اسکلت های بدست آمده به صورت زیر خواهد بود:

(ب)

باید از فرمول زیر استفاده کنیم:

$$K = \max\{k | (A \ominus kB) \neq \emptyset\}$$

$$A = \bigcup_{k=0}^K S_k(A) \oplus kB$$

این فرمول به ما میگوید که برای بدست آوردن تصویر اصلی از اسکلت آن، اسکلت داده شده را با عنصر ساختاری گسترش میدهیم. باید حاصل را ذخیره کنیم و در هر مرحله حاصل بدست آمده از مرحله ی قبل را با عنصر ساختاری گسترش بدهیم. همچنین باید حاصل تمام مراحل را با هم اجتماع بگیریم تا کل تصویر بدست بیاید. تعداد مراحل انجام عملیات هم به همان تعداد انجام مراحل تولید اسکلت از تصویر است. یعنی تعداد دفعاتی که میتوان تصویر را با عنصر ساختاری کاهش داد طوری که تهی نشود.

کد آن به صورت زیر خواهد بود:

```
def recons_skeleton(image, params):
    """
    Finds the original image from the skeleton.

    Parameters:
        image (numpy.ndarray): The skeleton of image.
        params (numpy.ndarray): The parameters required for reconstructing image

    Returns:
        numpy.ndarray: The original image.
    """

    res = image.copy()

    #Write your code here
    k = len(params)
    res = np.zeros((image.shape))
    SE = np.ones((3,3))

    for i in range(k):
        Sk = params[i]
        dilate = cv2.erode(cv2.bitwise_not(Sk), SE)
        for j in range(i - 1):
            dilate = cv2.erode(dilate, SE)
        res = dilate + res
    maxVal = np.max(res)
    ret, res = cv2.threshold(res, maxVal-1, 1, cv2.THRESH_BINARY)
    return res
```

skeleton of image 1



image 1



skeleton of image 2

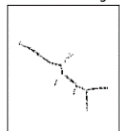


image 2



skeleton of image 3



image 3



skeleton of image 4



image 4



نتیجه ی آن به شکل مقابل خواهد بود: