

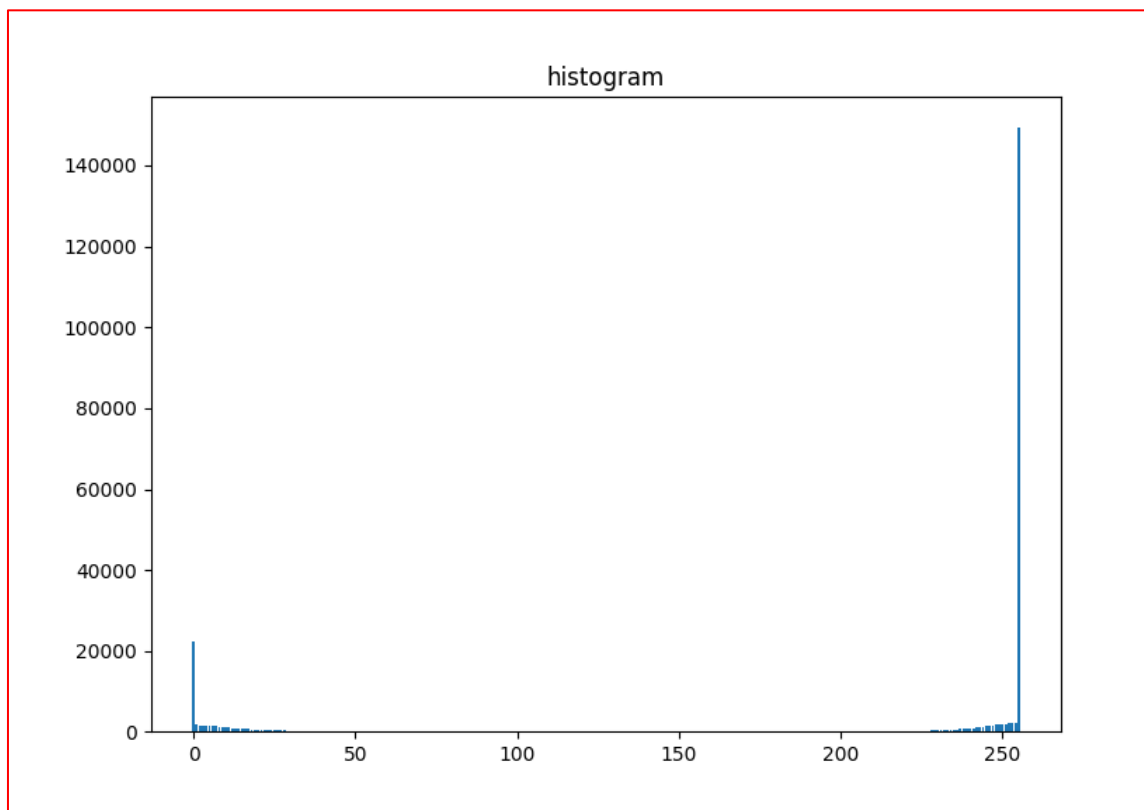
## Lab 8 – Statystyki obrazu, histogram i filtrowanie przez wyrównanie histogramu

### Zadanie 2

Statystyki:

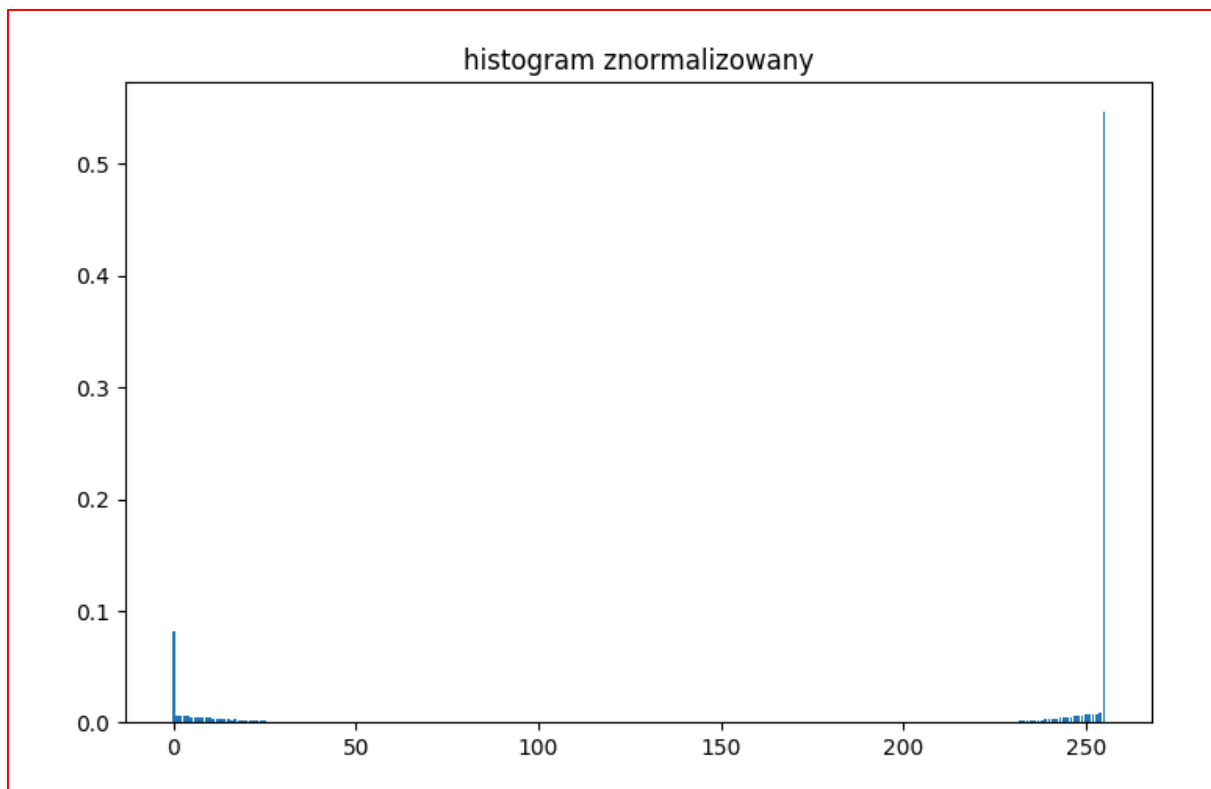
```
extrema [(0, 255)]  
count [273564]  
mean [188.49828193768187]  
median [255]  
stddev [99.55570192800243]
```

Na moim obrazie minimalną wartością piksela jest 0, a maksymalną 255 – zawiera więc wartości z obu końców skali. Mean informuje, że średnia wartość piksela na moim obrazie to 188. Mediana równa 255 pokazuje, że ponad połowa pikseli obrazu musi mieć wartość 255 (jest to środkowa wartość niemalejąco uporządkowanej listy wartości pikseli). Odchylenie standardowe jest duże, bo mam na obrazie dużo pikseli bliskich 0 jak i dużo bliskich 255.

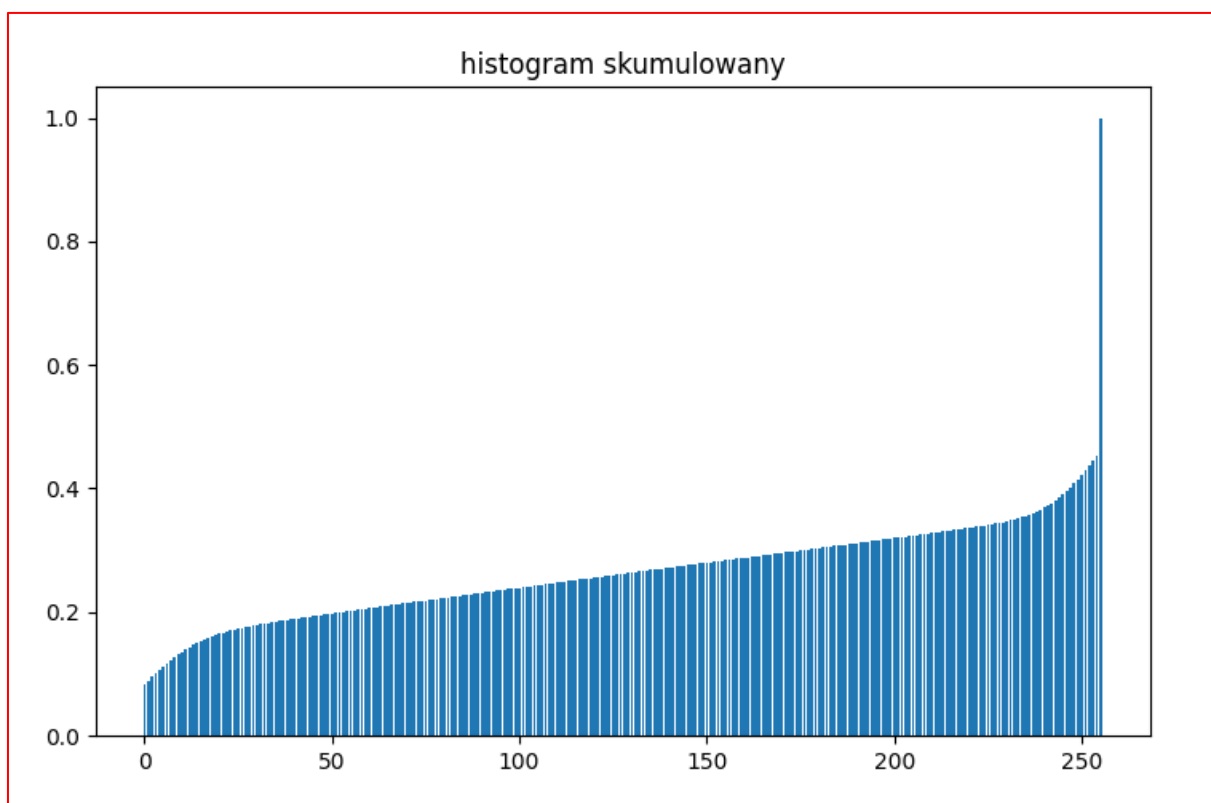


Na histogramie widać, że najmniejsza i największa wartość to odpowiednio 0 i 255. Ciężko z wykresu określić średnią, ale na pewno jest bliżej 255 niż 0. Ponieważ słupek w 255 jest największy ze wszystkich to widać, że znajduje się w nim mediana.

### Zadanie 3

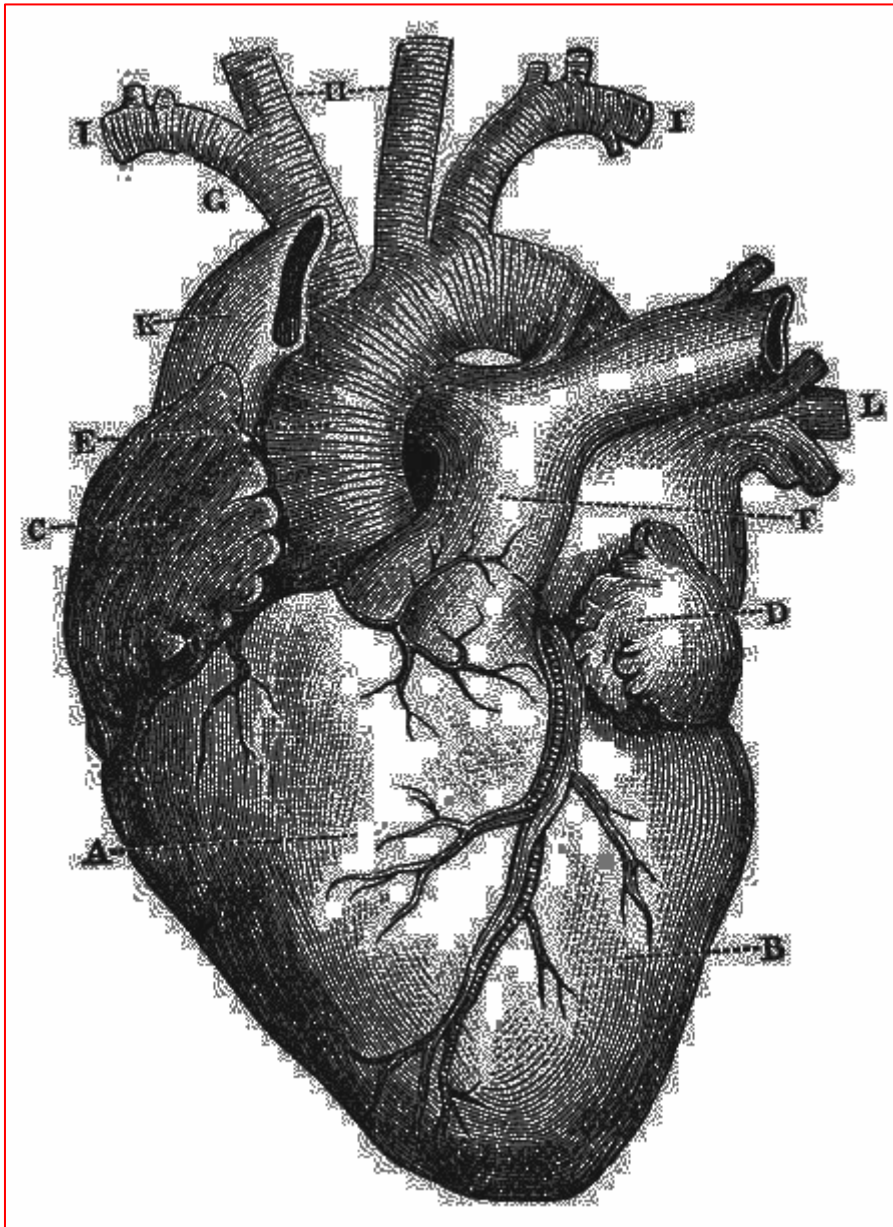


### Zadanie 4



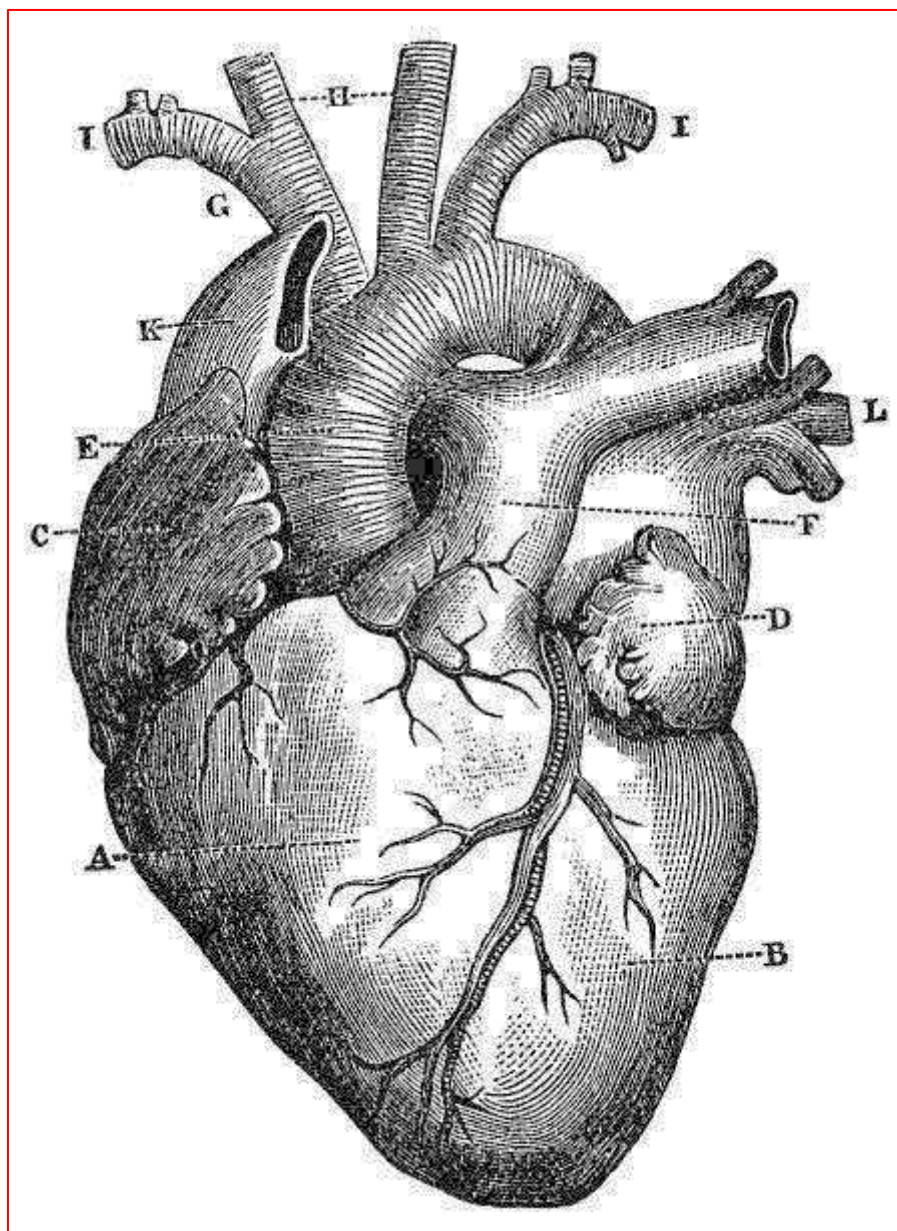
### Zadanie 5

obraz: equalized.png



## Zadanie 6

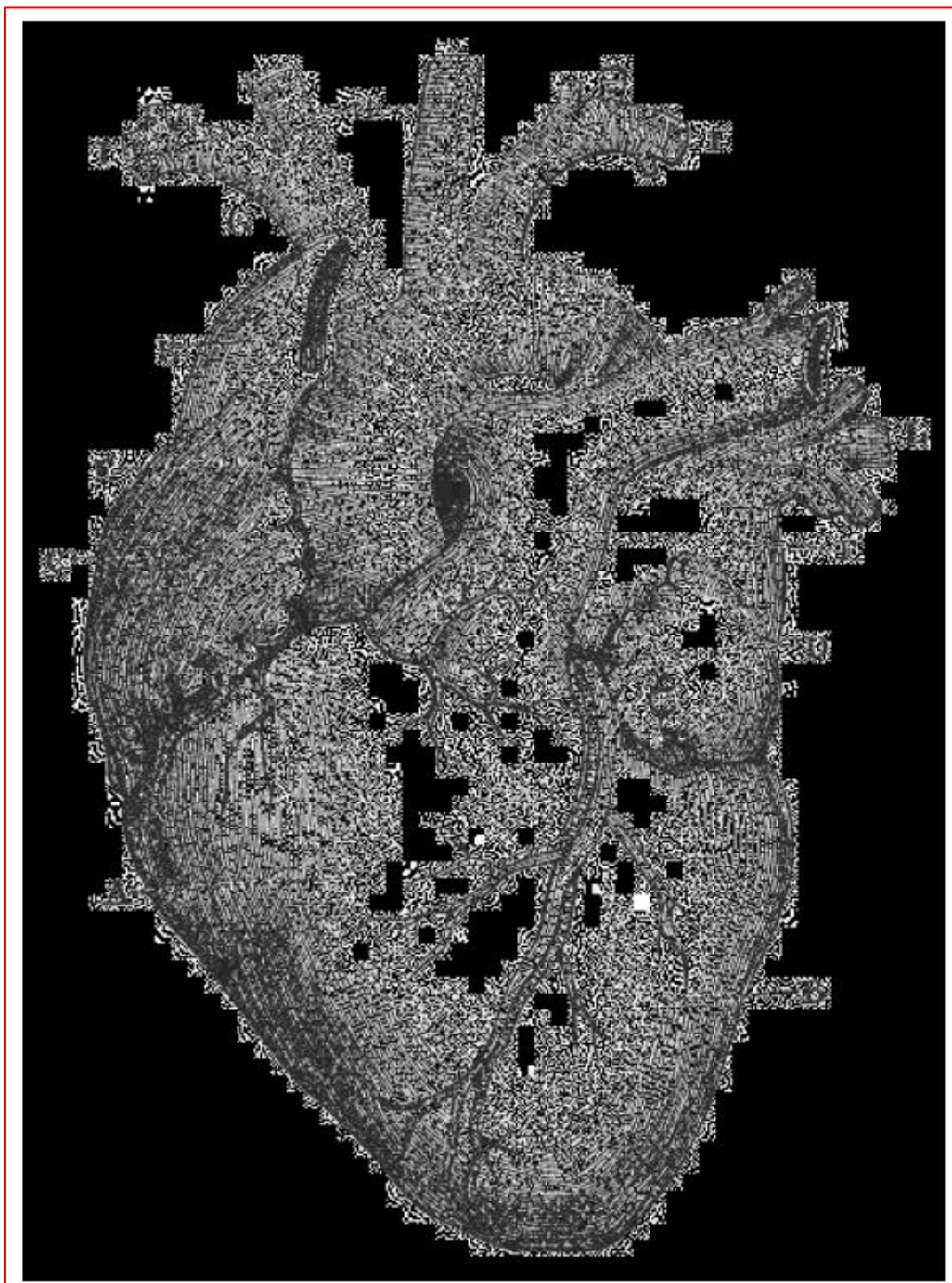
obraz: equalized1.png



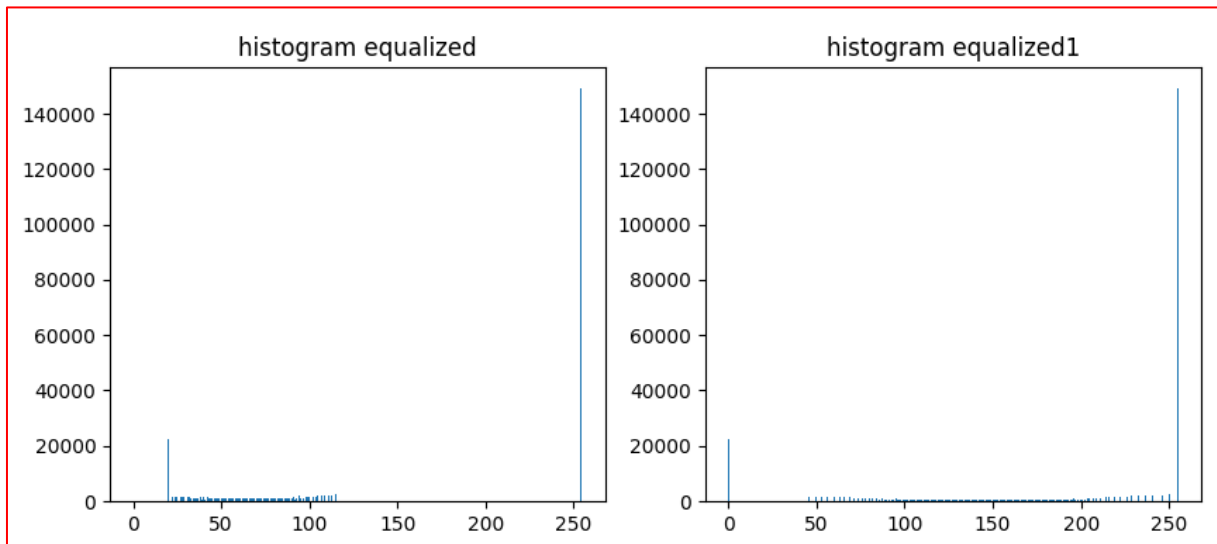


### 6.1 porównanie metodą ImageChops:

Widać, że obrazy się różnią, szczególnie wokół linii ilustracji serca



## 6.2 histogramy obu obrazów:



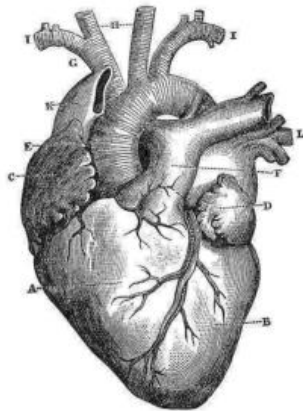
## 6.3 statystyki obu obrazów:

```
extrema [(20, 254)]
count [273564]
mean [165.71216241903176]
median [254]
stddev [99.057590456447]
*****
extrema [(0, 255)]
count [273564]
mean [194.883628693834]
median [255]
stddev [84.8353772589209]
```

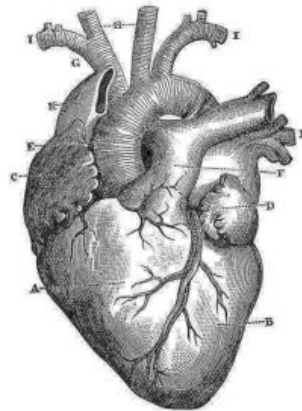
Piksele obrazu wykonanego funkcjami nie osiągają aż tak skrajnych wartości jak te w obrazie wykonanym metodą `ImageOps.equalize()`, przez co pierwsza średnia oraz mediana są niższe.

## Zadanie 7

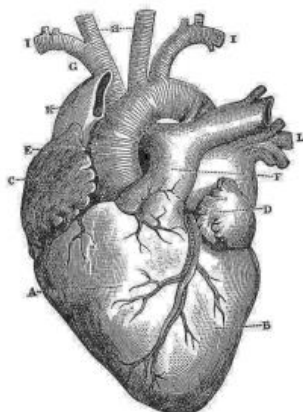
obraz początkowy



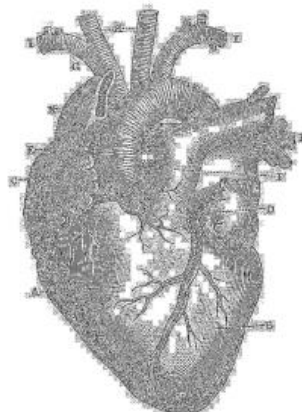
DETAIL



SHARPEN



CONTOUR



Najlepszy efekt daje filtr SHARPEN – wokół ilustracji jest najmniej fallout, nie rozmywa konturów.