Original image:



**Fig-1: image 1**



**Fig-2: image 2**

After gray filtering:





Edge detection using different algorithms:

**1.Sobel:**



**Fig-3: sobel using 0.1 threshold**



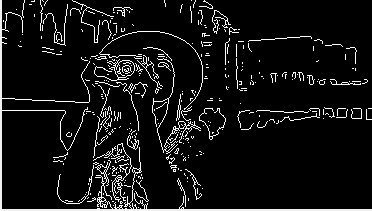
**Fig-4:sobel using 0.2 threshold**



**Fig-4: sobel using 0.3 threshold**

Even though the image loses a lot of data, for 0.1 threshold value, the algorithm generates a very clear edge detection and with the increase of threshold the image becomes more blurry so in this case 0.1 threshold gives better result.

**2.Prewitt:**



**Fig-5: Prewitt for 0.1 threshold**



**fig-6: Prewitt for 0.2 threshold**



**fig-7: Prewitt for 0.3 threshold**

Even though the image loses a lot of data, for 0.1 threshold value, the algorithm generates a very clear edge detection and with the increase of threshold the image becomes more blurry so in this case 0.1 threshold gives better result. It gives similar output as sobel.

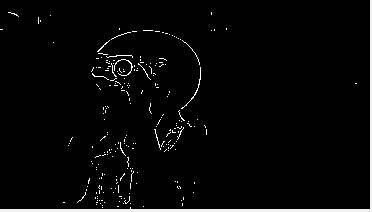
**3.Roberts:**



**Fig-8: Roberts for 0.1 threshold**



**Fig-9: Roberts for 0.2 threshold**



**Fig-10: Roberts for 0.3 threshold**

Even though the image loses a lot of data, for 0.1 threshold value, the algorithm generates a very clear edge detection and with the increase of threshold the image becomes more blurry so in this case 0.1 threshold gives better result. It gives a little bit better output for the same thresholds of the previous two algorithms.

**4.log:**



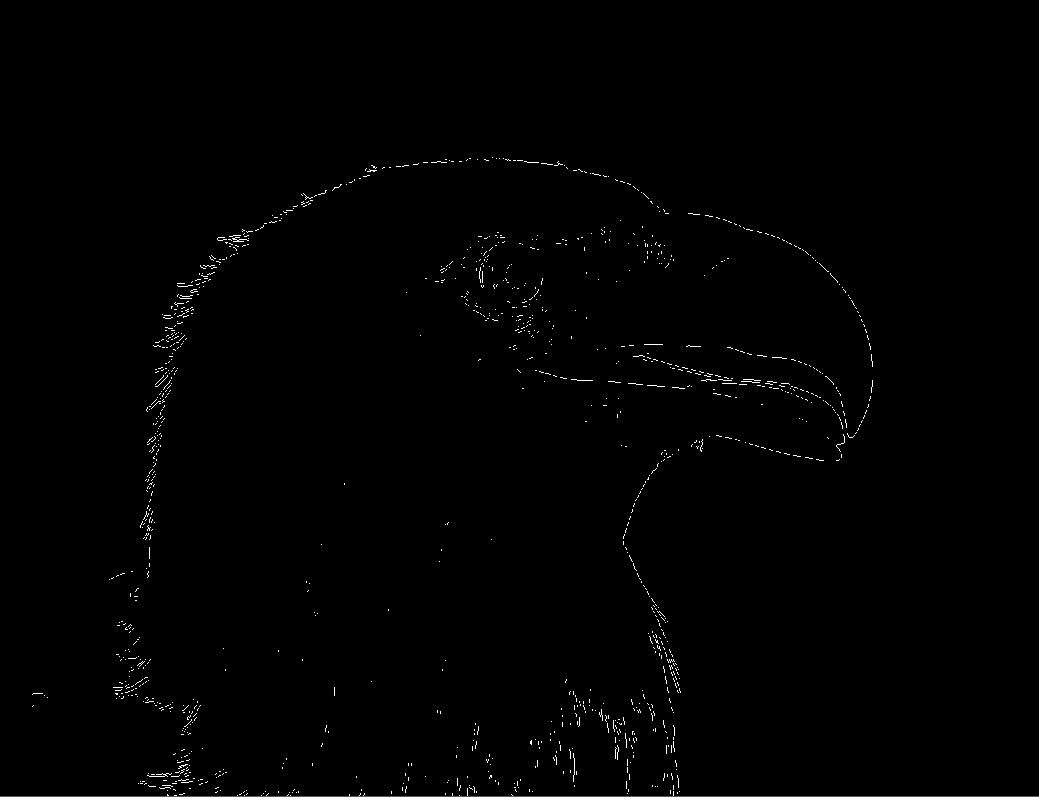
**Fig-11: log for 0.01**



**Fig-12: log for 0.02**

The image loses a lot of data, for 0.01 threshold value, the algorithm generates a very clear edge detection and with the increase of threshold the image becomes more blurry so in this case 0.1 threshold gives better result. For the same threshold as the previous algorithms, log can not detect any edge and gives a black screen.

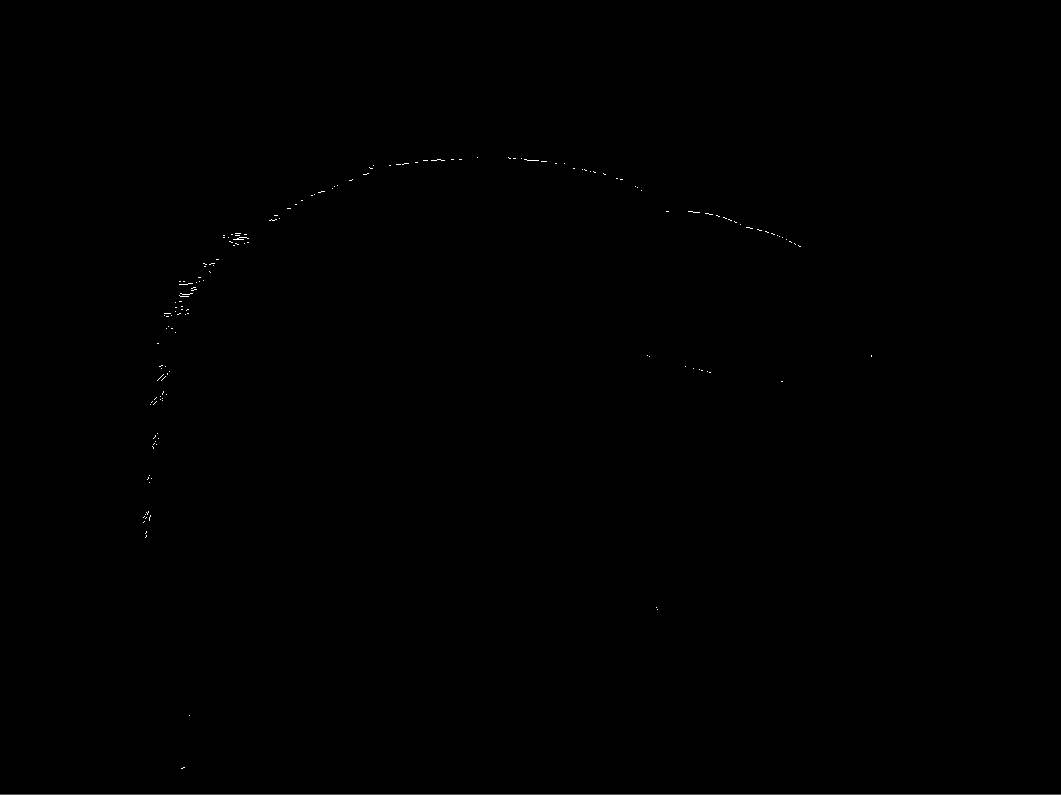
**5.Zero-Crossing:**



**Fig-13: Zero-Crossing for 0.01**



**Fig-14: Zero-Crossing for 0.02**



**Fig-15: Zero-Crossing for 0.03**

The image loses a lot of data, for 0.01 threshold value, the algorithm generates a very clear edge detection and with the increase of threshold the image becomes more blurry so in this case 0.01 threshold gives better result. It gives similar result as log.

**6.Canny:**



**Fig-16: Canny for 0.01 & 3 threshold**



**Fig-17: Canny for 0.01 & 5 threshold**



**Fig-16: Canny for 0.01 & 7 threshold**

since canny uses two thresholds, it gives much better output than any other algorithms. For the value of h=3, the image gives too much detail, with h=5, it removes some unnecessary details and with h=7 it removes almost all the unnecessary detail for this image.

The auto threshold for this algorithms are:

Sobel = 0.0624

Prewitt= 0.0609

Roberts = 0.0651

Log = 0.0021

Zero cross = 0.0021

Canny = 0.0188 ~ 0.0469