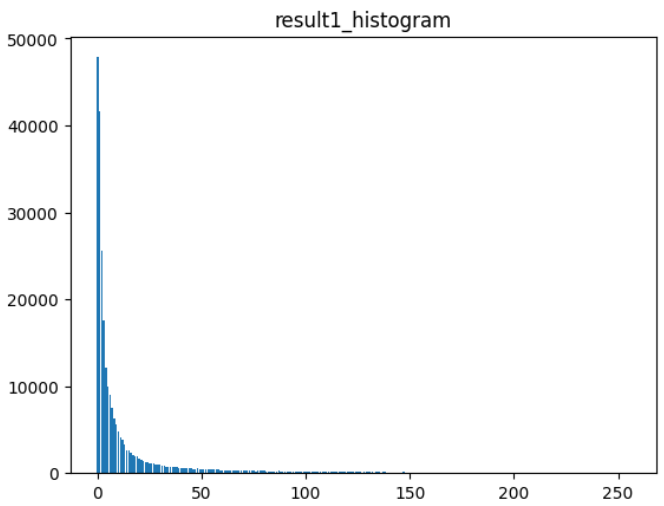
DIP Report HW2

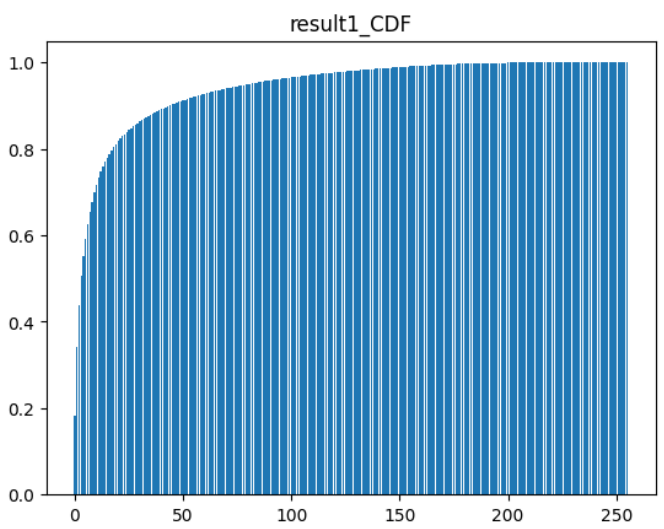
- **<u>Problem 1: EDGE DETECTION</u>**

  The sheep in sample1.png does not know how to perform edge detection algorithms, please help her to accomplish the following tasks.

  1. (a) (10 pt) Apply Sobel edge detection to sample1.png. Output the gradient image and its corresponding edge map as result1.png and result2.png, respectively. Please also describe how you select the threshold and how it affects the result.

     - Ans: 在做完gradient後，畫出該圖gradient descent的直方圖，並計算gradient descent的PDF,CDF後畫出CDF直方圖，可根據圖片判斷當初多少percentage的點在edge上，再利用CDF計算指定的percentage應在哪個gradient之intensity，即可指定gradient中大於哪個intentsity可被判斷為edge。當threshold設的低的話，將會有比較多的點被選為edge點，當threshold設定的過高時，可能會有edge因為gradient不達threshold而被篩掉。

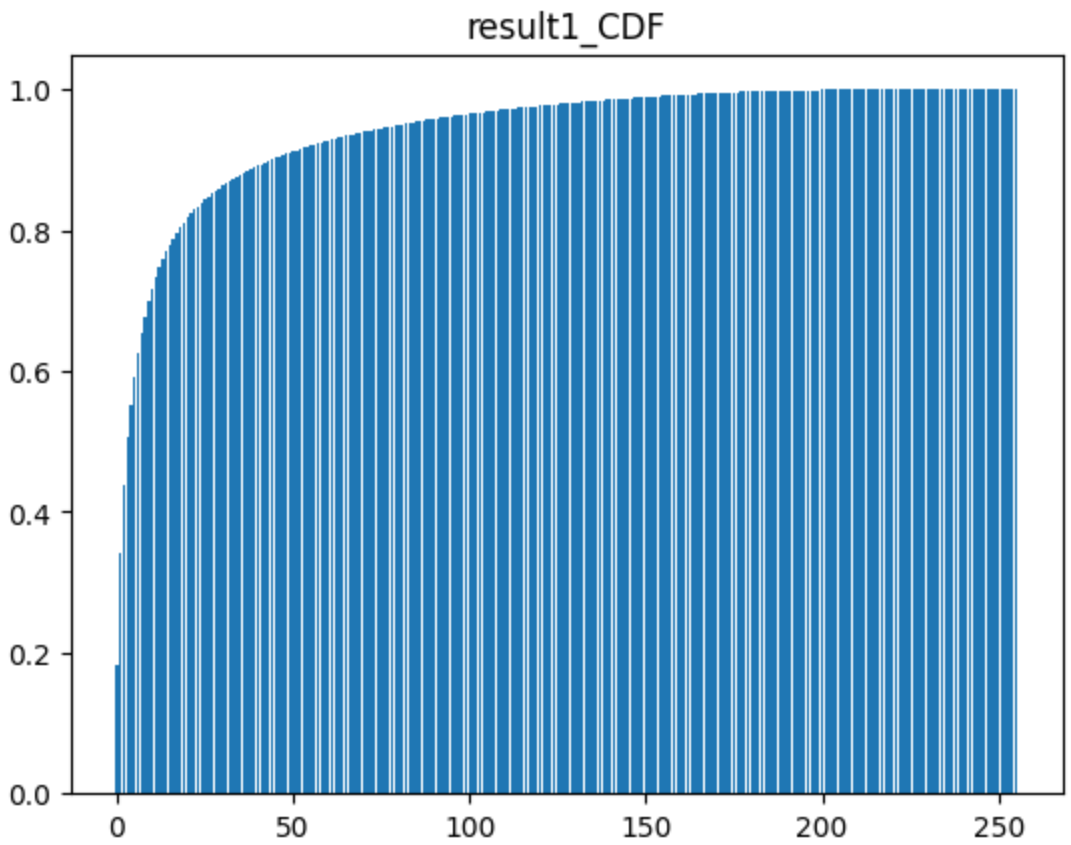




- Original Image (sample1)

- Output Image
    1. result1

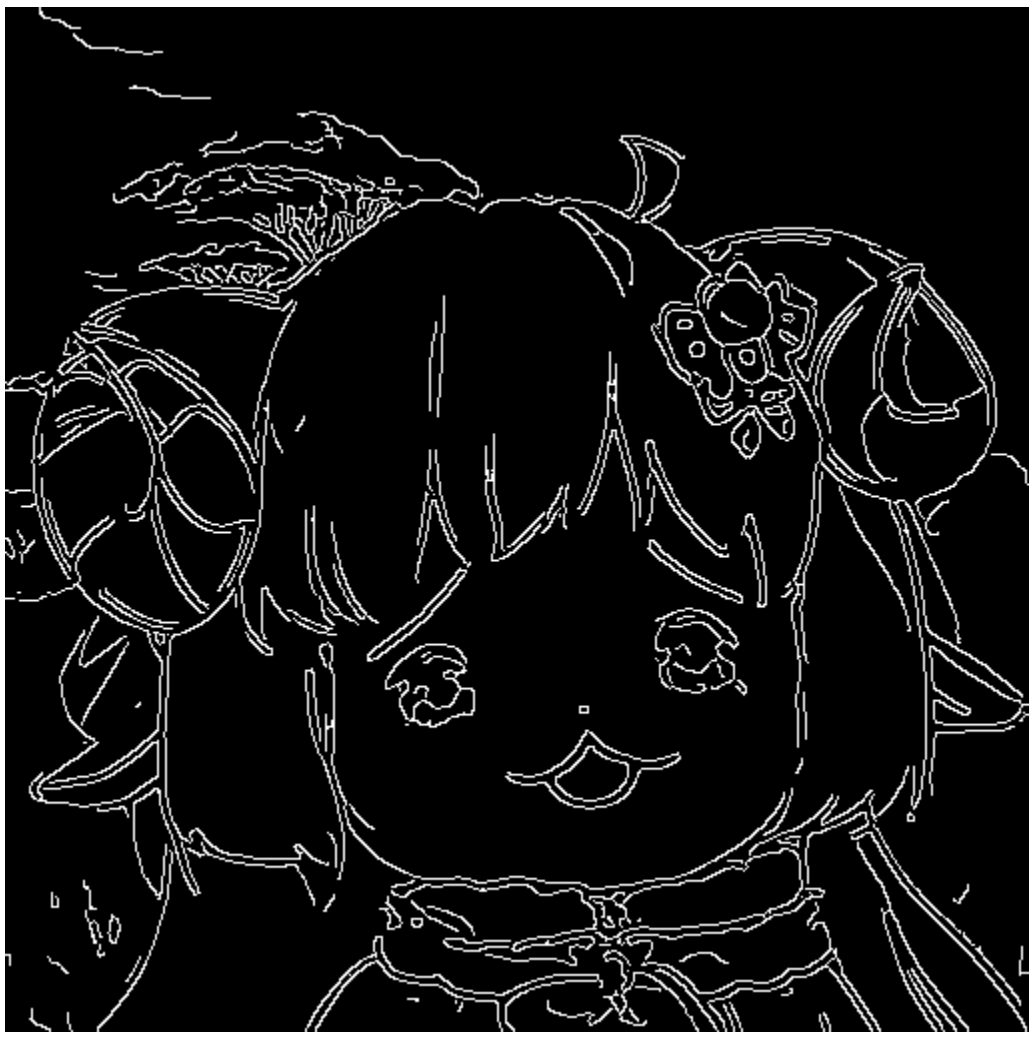
    2. result2



1. (b) (20 pt) Perform Canny edge detection on sample1.png and output the edge map as result3.png. Please also describe how you select the parameters and how they affect the result.

    - Ans:做完non maximal suppression後我依舊使用計算gradient CDF的方式，再設定大於多少gradient percentage應確定為edge(過 high threshold)，介於哪個percentage間應為edge candidate(介於high threshold和low threshold之間)，若high threshold設定很高但low threshold設定過低，仍會使edge和大量candidate相連，造成後續做connected component labeling時仍有很多candidate被選為edge。在選擇時就看想要很明確乾淨的edge,還是細節盡量顯現的edge，在對畫面中這些edge應佔多少percentage來做選擇。在做connected component labeling時，我使用8-neighbor connectivity來完成，我發現相較於4-neighbor connectivity其線段會比較完整，可能跟我選的high/low threshold的percentage相當接近有關。
    - Original Image (sample1)

- Output Image (reuslt3)



2. (c) (10 pt) Use the Laplacian of Gaussian edge detection to generate the edge map of sample1.png and output it as result4.png. Compare result2.png result3.png and result4.png and discuss on these three results.

- Ans:我自己最喜歡Canny的結果，其線條很乾淨，雖然因為人物是漫畫已有似乎已有描邊，因此會時常有兩條edge平行的情形，但是明顯是三張當中線條最乾淨準確的，且細節（後方的樹的樹枝細節、雲）都有描繪。其次是Sobel，操作簡單且我覺得效果不錯，才是Laplacian of Gaussian edge detection，雖然能避開threshold的問題，但是其線段有些邊緣會過多被描繪，且斷斷續續，肉眼看下來不是那麼舒服。
- Original Image (sample1)

- Output Image
    1. result2 (Sobel Edge Detection)

    2. result3 (Canny)

3. result4 (Laplacian of Gaussian edge detection)

3. (d) (10 pt) Perform edge crispening on sample2.png and output the result as result5.png. What difference can you observe from sample2.png and result5.png? Please specify the parameters you choose and discuss how they affect the result.

   - Ans:我選擇使用gaussian filter來作為low pass filter，且設定c=3/5

     function:G=c/(2*c-1)*img-(1-c)/(2*c-1)*after_low_pass_img
     觀察為當c設定小的時候，銳化效果明顯，反之c值設定較大時，銳化效果較為低
     猜測可能是，c值越小時會使得影像中較多區塊符合條件而被銳化，而c較大時可能能讓較多區域保持原樣。

   - Original Image (sample2)

- Output Image (result5) c=3/5

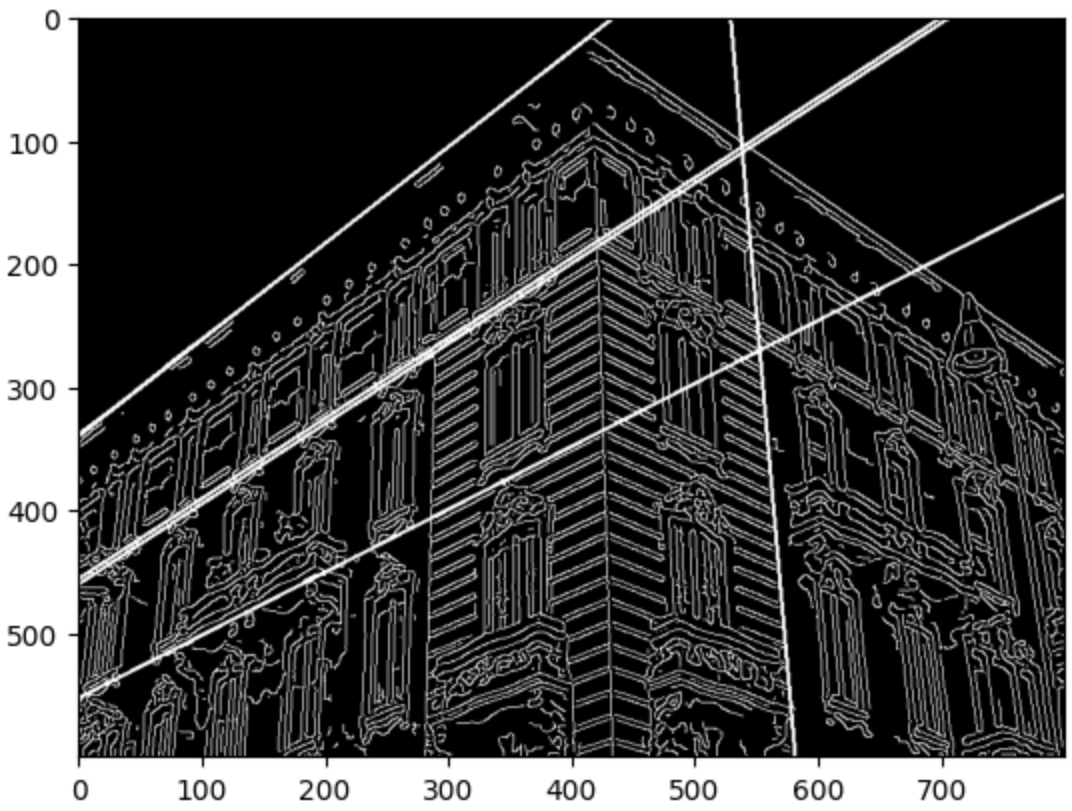
c值較大

4. (e) (Bonus) Perform Canny edge detection on result5.png and output the edge map as result6.png. Then apply the Hough transform to result6.png and output the Hough space as result7.png. What lines can you detect by this method?
    - Ans:
    - Original Image

    - Output Image

result6

result7



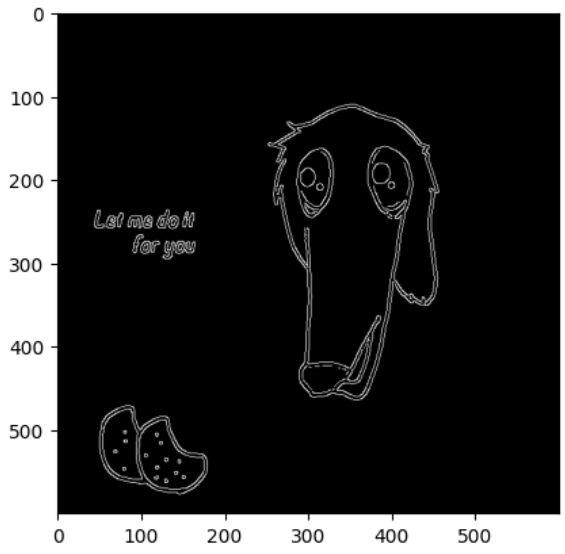
- **Problem 2: GEOMETRICAL MODIFICATION**

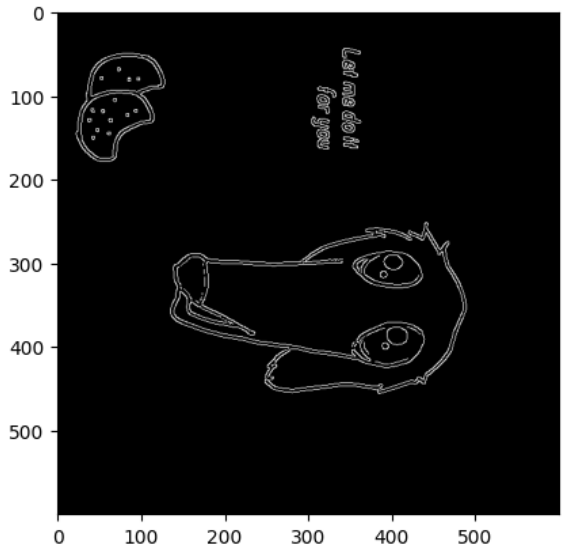
  Please design several geometrical modification algorithms to meet the following requirements. Your results may not be exactly the same as the sample images. Just try to create the effects as closely as possible.

  1. (a) (25 pt) The Borzoi wants to help you to get the potato chips. Please design an algorithm to make sample3.png become sample4.png. Output the result as result8.png with the same dimension as sample3.png. Please describe your method and implementation details clearly. (hint: you may perform rotation, scaling, translation, etc.)

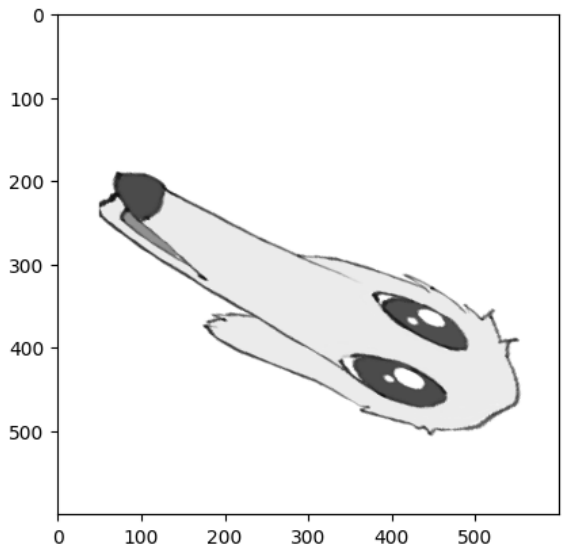
     - Ans:我先對該圖進行canny後得到edge map再對其最右邊的component（從image to cartesian coordinate後應在圖片的最下方後）利用類似掃描的方式 (cartesian coordinate中下方往上掃，訂出最上端點和最下端點，再由此區間中左往右掃訂出左右端點)即可定位出包含狗狗的最小長方形邊框

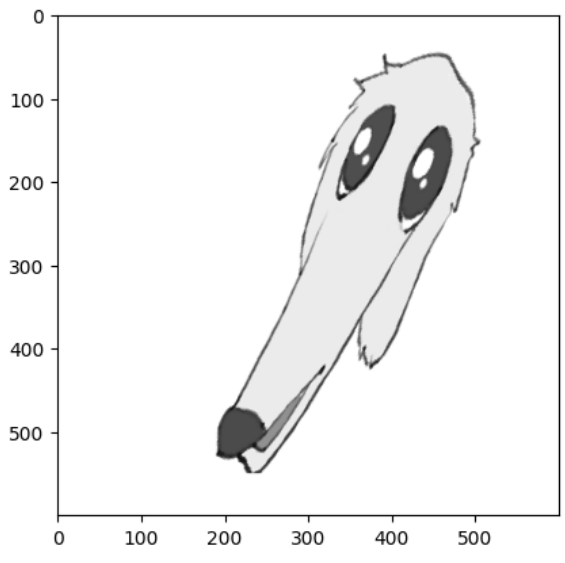
       再定位出洋芋片的最下緣(cartesian coordinate)後計算狗狗的scale比例，scale多少長度可到達原圖洋芋片右方）



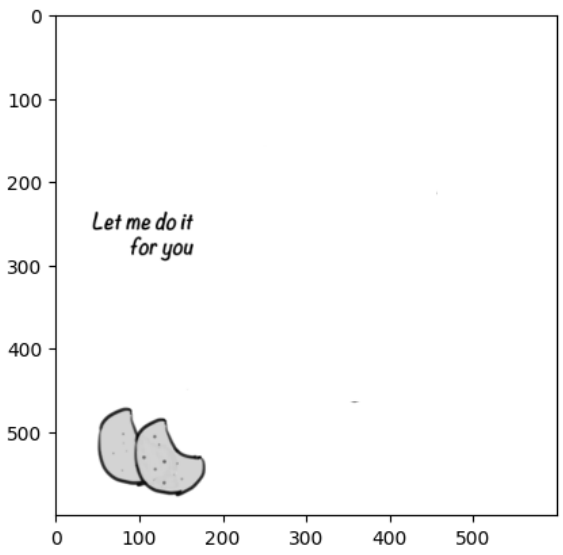


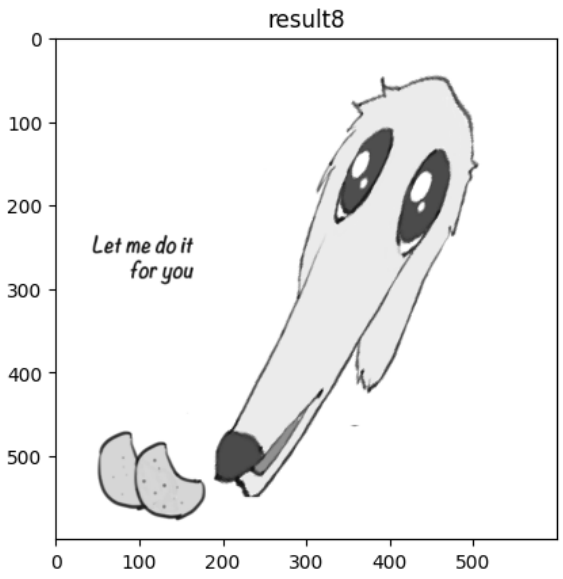
接著先做transform，使的原點移動到相對原圖是框住狗狗方框左下角的位置(因要對原點做後續cartesian coordinate下的scale,rotation)，再依據剛剛計算的scale比例來scale並rotate，做完後再將原點transform到前面計算的洋芋片最右緣，後作coordinate transform回image coordinate





接著再將原圖中狗狗的部份做清除（將前面定位好的方框框中intensity=255)，後將兩圖每個pixel做比較，選擇兩圖pixel值中最小為新圖中該pixel之intensity值，完成狗與洋芋片合併。





- Original Image
    1. sample3

2. sample4

- Output Image (result8)

2. (b) (25 pt) I made my own Popcat picture, although there’s no cat in it. By observing the effect shown in sample6.png, please design an algorithm to make sample5.png look like it as much as possible and save the output as result9.png. Please describe the details of your method and also provide some discussions on the designed method, the result, and the difference between result9.png and sample6.png, etc.
    - Ans:
    - Original Image
    - Output Image