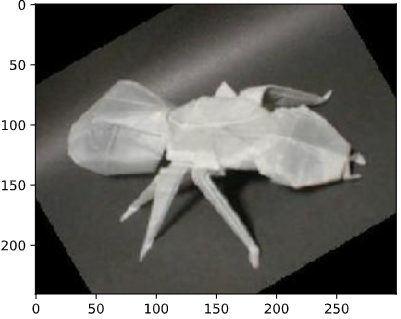
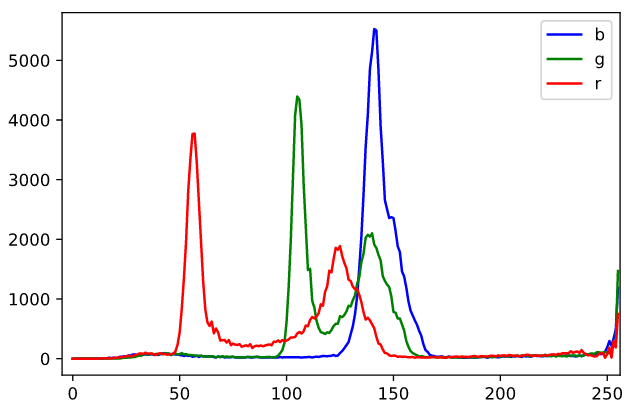
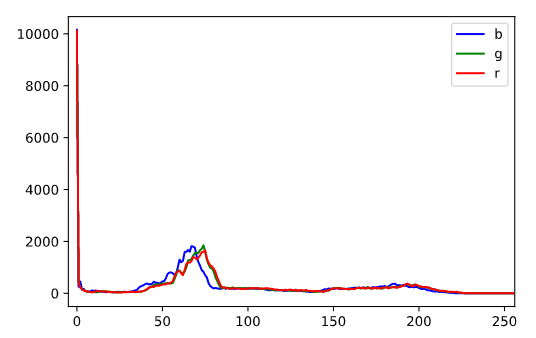
多媒體系統與應用 HW2

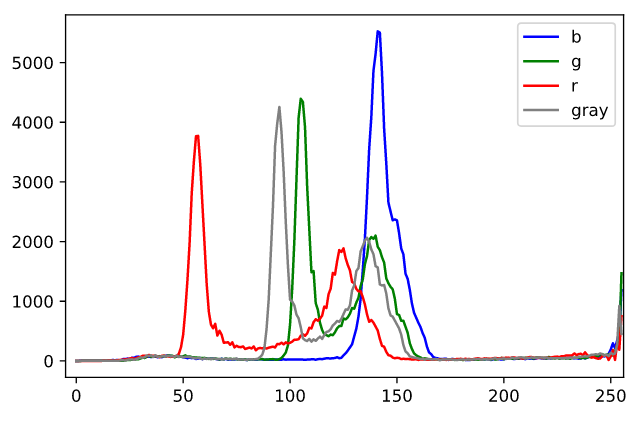
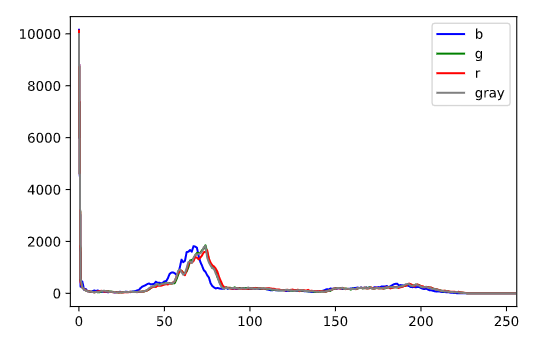
E14046648 機械108吳仰航

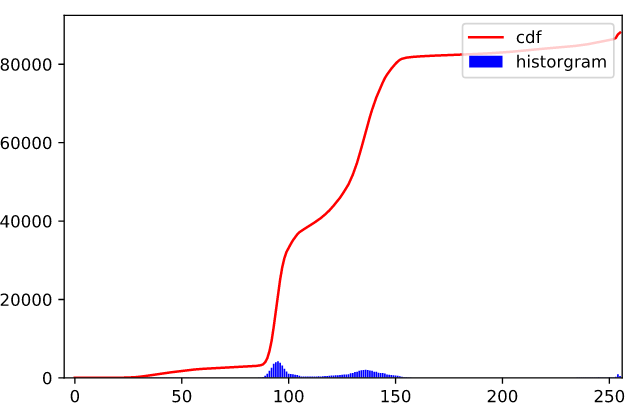
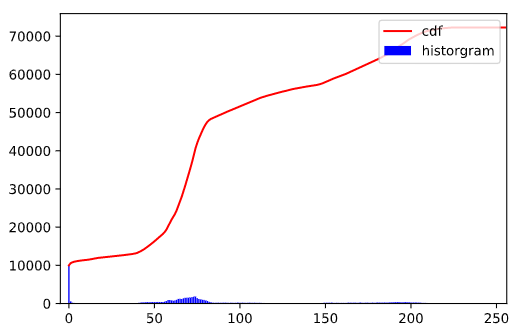
1. 首先透過opencv讀入需要之圖片

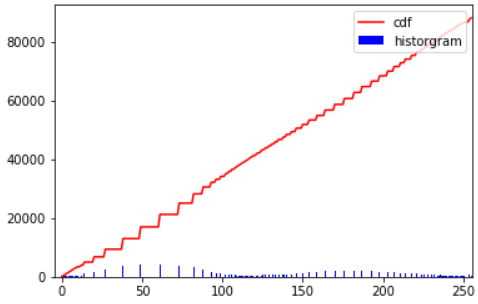
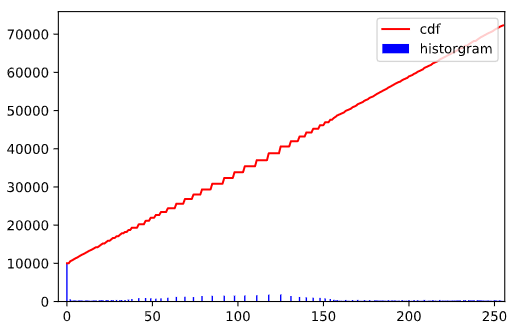


1. 利用show\_bgr\_his(img)化出該圖片rgb通道之histrogram趨勢，可以發現其螞蟻的rgb通道的特徵差不多，因此為了減少計算量只需要取一個通道即可，但也因為其分布趨勢一至，對於邊緣的偵測主要影響為光度(灰階)，如果今天是要找尋特定顏色的輪廓可以針對單一通道進行edge detection但今天是要對整幅圖片去進行邊緣偵測，因此將圖片轉為灰階。飛機圖片rgb分布比較離散，因此如果想針對藍天去做邊緣偵測也可以只取藍色通道的進行處理，不過我們是要找主體的，因此之後還是會轉為灰階。

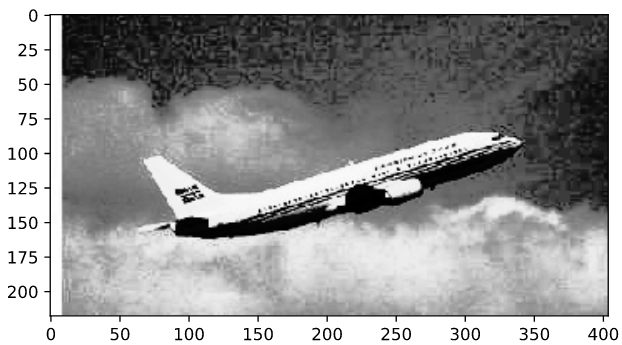
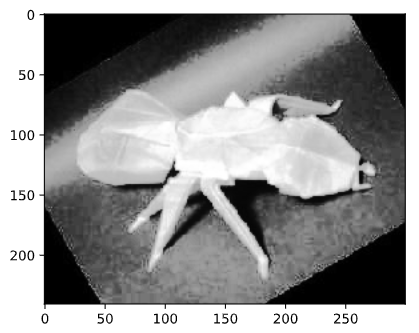


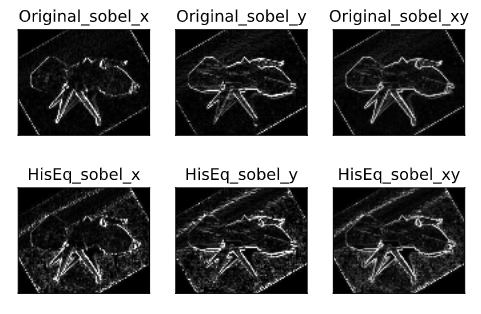
1. 轉完灰階之後依然可以發現，其灰階的histrogram趨勢與rgb一致，因此為了減少計算量之後都以灰階之圖片來進行處理。
2. 針對灰接圖片進行historgram與cdf做圖得到下圖。可以發現cdf的趨勢並不平緩，原因是個灰階值的出現機率不均，因此要透過historgram equalization來進行正規化。



1. historgram equalization可以透過此步驟當做圖片的預處理，因為可以將對比更加明顯，進而使得邊緣更容易被找尋出來。從結果也可以看出cdf這條趨勢線已經趨近於一條直線，代表著每一個gray level值出現的機率一樣大

同時從結果看起來也更加的具有對比度



1. 接下來針對做完eq的圖片進行sobel\_x與sobel\_y的運算，透過這兩個matrix針對圖片進行convolution可以找出該圖片在x,y方向的梯度，由於物建邊緣可能會是兩個不同的型態，因此在邊緣處梯度會最大。最後再將兩個方向的梯度進行加總，得到最後結果。

由以上結果可以看出，若試圖片有先過historgram\_eq則會出現更多的資訊，因為有先原本比較弱的邊緣透過預處理後變得比對性比較強，因此我們在透過sobel運算時，可以找出來一些比較細微的輪廓。

結論

1. 若是圖片想找全局觀的輪廓使用灰階是合理的選擇，但若是有特定的顏色需求可以透過改變處理的通道進行微調。
2. Sobel\_x sobel\_y是具有方向性的，如果今天想要找一個圖片輪廓分部偏向於左上右下，那可以改成sobel對角線的格式會有比較好的效果。
3. Historgram equalization可以進行更細緻的邊緣搜尋，但如果今天不想要有太多雜訊或是主體輪廓明顯，則不需要多做這個預處理反而可以得到更好的結果。

Bonus:

CannyThreshold(lowThreshold):

主要做了以下幾點運算

1. 減少噪音:

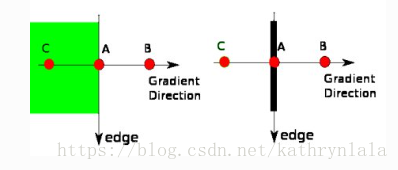
因為邊緣檢測對噪音非常敏感，所以實現邊緣檢測的第一步，是使用高斯濾波器對影象中的噪音進行移除。 高斯模糊:有點向convolution的概念，將一個pixel與週圍進行比較，若是該pixel的值特別突出，就會被周圍的值平滑化，針對胡椒鹽雜訊有很好的抑制效果，在處理螞蟻那張圖片實，如果把kernel size改成5\*5就有機會將背景紙張的邊際模糊掉，讓後續的edge detection可以找到螞蟻的完整輪廓。

1. Sobel\_xy:

同作業的sobel做法，用意在於找到xy方向梯度之變化，若是梯度變化大，則很有可能為物體的輪廓。

1. 非最大化抑制(nms):

上述獲取梯度的大小和方向之後，一張移除了不想要的畫素（這些畫素是指不組成邊的）全圖，為了得到這張全圖（為了移除這些畫素），對每一個畫素進行查驗，判斷它是否是梯度方向上的像零點的相鄰最大值(local maximum)。



上圖為例，梯度方向是正交與（is normal to )邊，點A在邊上（垂直方向），點B和點C在梯度方向上。所以點A就會被查驗是否和點B、點C相比較，判斷它是否是相鄰最大值(local maximum)。是的話，A被認為是下一個next stage;不是的話，則為0。 得到一個“thin edge".

1. 雙閥值:

進一步，來判斷上訴結果中，哪一個是真正的邊。為了實現這個，需要兩個minVal和maxVal的閾值。所有邊的強度梯度超過maxval確定為邊，低於minVal的確定不為邊（丟棄）。基於中間的判斷是否為邊是根據其連續性（connectivity）；如果它和"確定邊"的畫素相連，則為邊；如果和"不為邊"的畫素相連，則不為邊（丟棄)。 下圖中，邊A是在maxVal之上，是“確定邊”，雖然邊C在maxVal之下（minVal）之上，但它和邊A相連，所以認為它是有效邊，這樣我們就得到了一個完整的曲線。但是對於邊B，雖然和邊C在同一個區域，但它沒有和“確認邊”相連，故丟棄。 概念上像是os中需要做context switch，如果只設定一個單一threshold的化會context switch太頻繁，因此設定兩個threshold可以有效的減少switch，在影像這邊則可以減少一些比較細小的edge，只找到真的明顯的輪廓並同時保持其證確與完整性

