2. Object Labelling

2.1 Source Image Analysis

Figure 2.1 shows an image of a flock of birds flying in a cloudy sky. This task aims to label each bird with a unique colour. Forty-three distinct birds can be seen in this image. However, five pair of birds are overlapping with each other. As a result, those birds would be treated as one entity during labelling. We can separate the overlapping birds using image processing techniques like erosion or opening.

Figure 2.1: Image of a flock of flying birds

2.2 Strategy

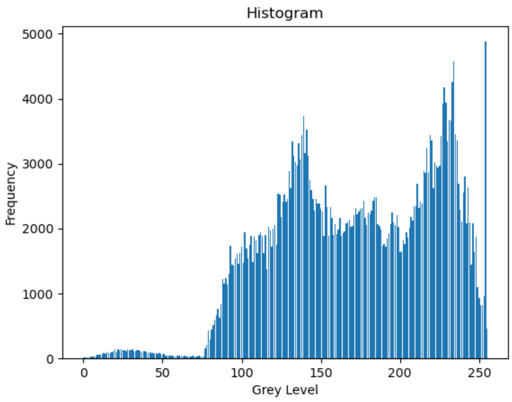
We need to obtain a binary thresholded image from the source image. Keeping the background pixels in the thresholded image as black and birds pixel as white would ease the further process of object (birds) detection and labelling. An algorithm is required to label each bird with a different grey-scale intensity. We can use this grey-labelled image to label birds in the source image with different colours.

1. Threshold source image using multiple image processing technique to obtain a binary image where the background pixels are black while the bird's pixel is white.
2. Generate a greyscale labelled image from the binary image obtained in the previous step. At the end of this step, each unconnected birds must be labelled with a unique intensity of grey.
3. Use the greyscale labelled image to label each bird in the source image with a unique colour.
4. Use the opening technique (with different parameters) on the result obtained at step 1 to separate the five overlapping birds pair.
5. Apply step 2 and 3 to reprocess the result obtained in step 4 to label birds.

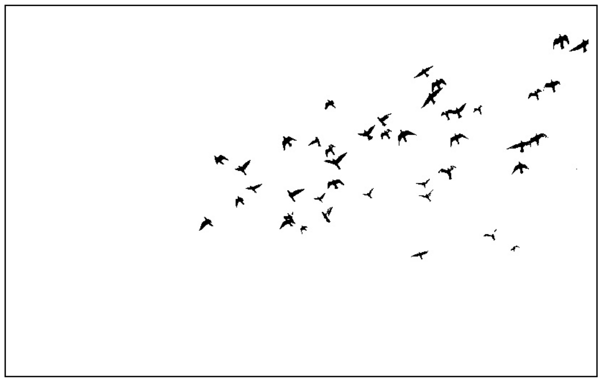
2.3 Implementation and Result

The following points describe the implementation and result of the strategy mentioned in the previous section.

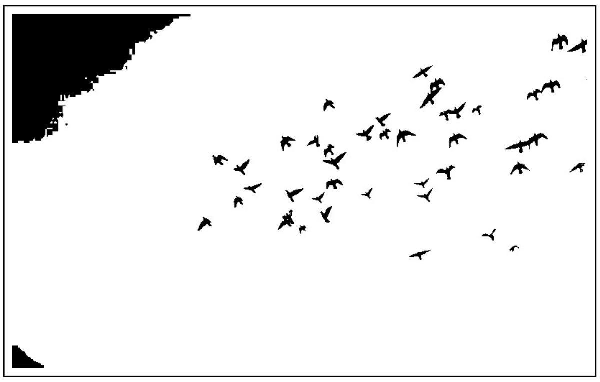
1. Threshold source image
   The histogram obtained for the source image is shown in figure 2.2(a). It shows that the maximum pixel intensities are in the range of 80 to 250. As the birds in this image have dark shades, we will manually threshold this image at T = 60, 100 and 140 to see which T-value produces the best result. Figure 2.2(b)(c)(d) shows source image thresholded at T = 60, 100 and 140. The left top region of figure 2.2(c)(d) is dark, which tells us that the pixel intensity of this region is similar to the pixel intensity of few birds. However, both of these figures show details of birds better than figure 2.2(b).



(a) Histogram of birds image



(b) T = 60

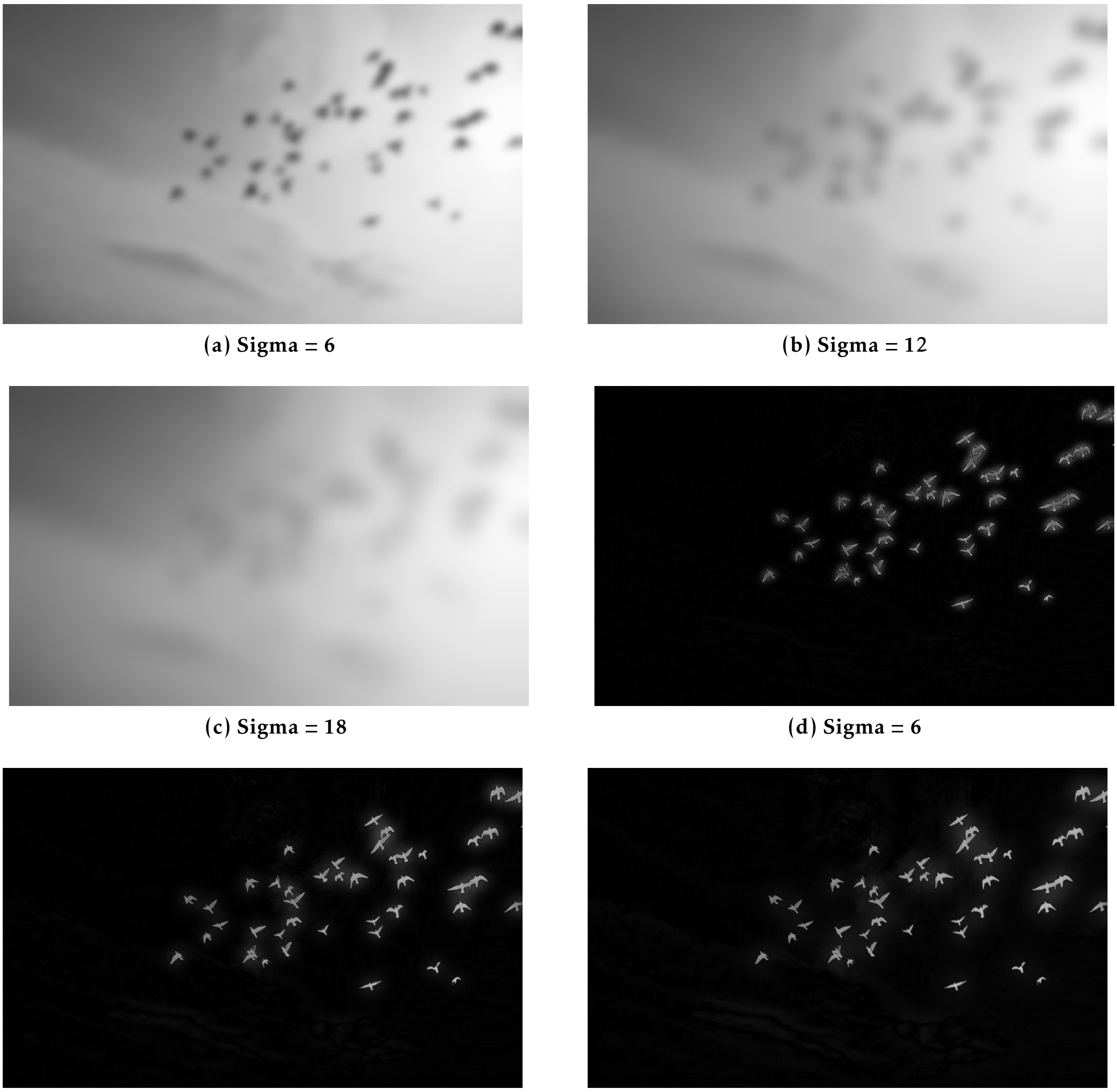


(c) T = 100

(d) T = 140

Figure 2.2: Birds image thresholded at different T-Values

Since the grey intensity of the left-top region of the image is similar to a few birds pixel intensity, smoothing out the image with the Gaussian technique and subtracting it from the original image should produce a better result. Figure 2.3(a)(b)(c) shows the Gaussian smoothed image at different standard deviations(sigma). Figure 2.3(d)(e)(f) shows the result after subtracting the Gaussian smoothed image from the source image. The image obtained at sigma = 18 (figure 2.3(f)) is better than the other two as it has comparatively less noise. The histogram of the image obtained from figure 2.3(f) is shown in figure 2.4(a). Otsu's method is used on this image to get the final thresholded image shown in figure 2.4(b). The T-Value obtained from the Otsu method here is 73.



(a) Sigma = 6

(b) Sigma = 12

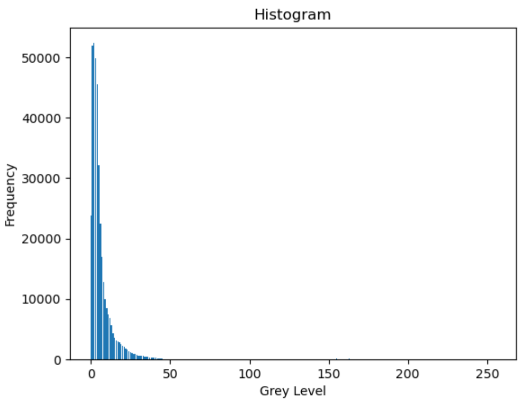
(c) Sigma = 18

(d) Sigma = 6

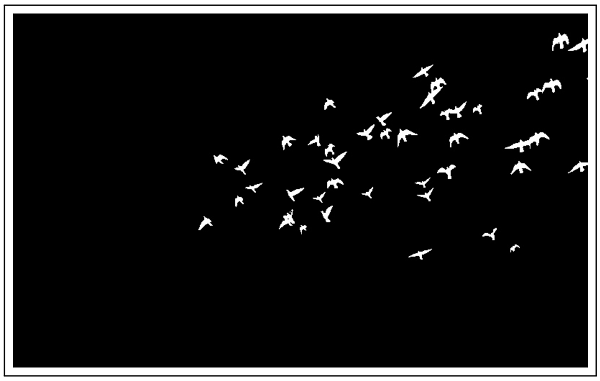
(e) Sigma = 12

(f) Sigma = 18

Figure 2.3: Birds image smoothed with Gaussian and subtracted from source



(a) Histogram of birds image shown in fig 2.3(f)



(b) Thresholded image obtained from image shown in fig 2.3(f) using Otsu's method

Figure 2.4: Final thresholded image of birds

2. Generate greyscale labelled image: The algorithm used to label each pixel in the thresholded image shown in figure 2.4(b) is described below. Also, the result obtained is shown in figure 2.5(a). It shows that all unconnected birds (Count = 39) are labelled with a unique grey intensity.
   (a) Load a binary image containing the object and background as a greyscale image. In our case, the object is denoted by white pixels, and the background is denoted by black pixels.
   (b) Loop through all the pixels of the loaded image from left to right and top to bottom. If a black pixel is encountered, then ignore it and move forward in the loop, else go to the next step.
   (c) If a white pixel is encountered, then check the neighbourhood pixels. There exist eight neighbours of any pixel, which can be represented using direction convention. The pixel above can be called a north pixel. The pixel diagonally above and to the right can be called as North-east pixel, as shown in figure 2.5.
   (d) If the neighbouring pixels do not have any label, i.e. they are all black or white pixels, assign a new label to the current pixel.
   (e) If the neighbouring pixels have a label, i.e. they have a grey intensity, assign the minimum neighbour label to the current pixel. The rationale behind assigning the minimum neighbour label is discussed in section 2.4.
   (f) Count the labels assigned after iterating through all the pixels of the source image.
   (g) Iterate all the image pixels multiple times using the logic given in steps (b) to (f) till the minimum count of labels is obtained. The final image obtained (figure 2.6(a)) will have each unconnected object labelled with a unique grey intensity.

NORTH-WEST PIXEL	NORTH PIXEL	NORTH-EAST PIXEL
WEST PIXEL	**CURRENT PIXEL**	EAST PIXEL
SOUTH-WEST PIXEL	SOUTH PIXEL	SOUTH-EAST PIXEL

Figure 2.5: Image showing Neighbour Pixels

3. Colour label birds in the source image using the grey-labelled image: The algorithm used to label each pixel of the source image is described below. Also, the result obtained after colour labelling is shown in figure 2.6(b). It shows that all unconnected birds (Count = 39) are labelled with a unique colour.

   (a) Load the source image where the objects are required to be labelled with a unique colour. Let us call it imgSource. Also, load the greyscale labelled image obtained in the previous section. Let us call it imglabelled.

   (b) Prepare a dictionary mapping different grey-intensities found in imglabelled to a unique BGR colour.

   (c) Loop through all the pixels of imgSource from left to right and top to bottom. Use imglabelled to identify pixels position having different greyscale intensities (excluding black pixels). Assign a colour to relevant pixels in imgSource using the dictionary prepared in the last step.

(a) Birds image labelled with different intensities of grey

(b) Birds image (Count = 39) labelled with different colours

Figure 2.6: Birds labelled with different colours

4. Use the opening technique to separate connected pair of birds: The image shown in figure 2.4(b) is opened with different parameters like the size of the kernel, number of iterations. Figure 2.7(a) shows the best result obtained after performing the opening operation on the final thresholded image shown in figure 2.4(b). Only one pair is separated after this operation. Opening operation tends to separate objects which are connected by a smaller number of pixels. If a large number of pixels would connect the objects, this operation will change the shape of all the objects of an image (and the result becomes unusable). In the source image, only one pair of overlapping birds seems to be connected by few pixels. Hence, only that pair (encircled in green in fig 2.7(b)) can be separated using the opening method and labelled with different colours.

(a) Result obtained by performing opening on image shown in fig 2.4(b)

(b) Birds labelled with different colours after separating overlapping pairs

Figure 2.7: Result obtained after performing opening operation

2.4 Discussion

Smoothing the source image and subtracting it from the original image gives a result which normalises the colour intensities across the whole image. Thresholding operation performed after that correctly segments the object and background. This chapter used this technique to get a binary segmentation of the object/background present in the source image.

We used an algorithm(discussed in section 2.3.2) to label each bird in the thresholded image with a unique grey intensity. In that algorithm, we tend to assign a minimum neighbour pixel label to the current processing pixel. The rationale behind using this logic is that when we iterate the pixels of an image from left to right and top to bottom and assign a new label to the first white pixel of an object, then all the object's pixels should be equalized with that label. In our case, we started assigning the label starting from label 1 to 255 (minimum to maximum). Hence, the label assigned to the first encountered pixel of any object will be the minimum label of that object. **A screenshot of a bird's pixels labelled after the first iteration of the algorithm is shown in figure 2.8. The first labelled pixel of this bird is encircled in green. Note that this is the minimum value assigned among all the labelled pixels of this bird.** An interesting thing to note here is that the count of birds reported by the algorithm in figure 2.6(b) is 39 instead of 38. **The algorithm code is detecting an extra bird(encircled in green in figure 2.6(b)) due to the inclusion of a bird's feather (which is out of the image).**

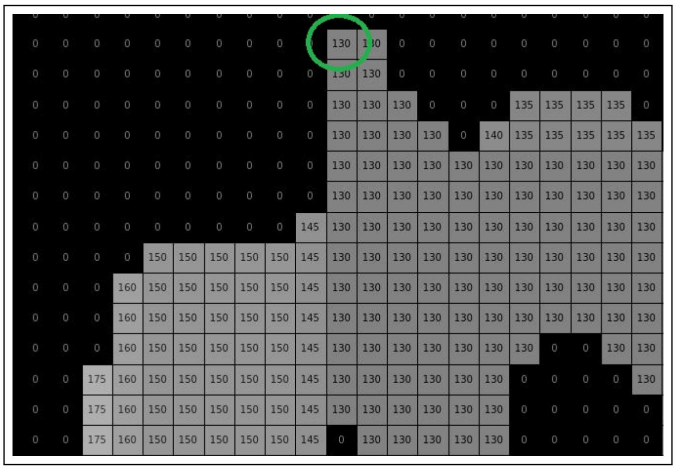


Figure 2.8: Image showing pixel's labelled after first iteration of the labelling algorithm

The erosion/opening operation used in computer vision can not separate and maintain the shape of two objects if a large number of pixels connects them. These operations will change the shape of the object if repeated multiple times. **Our experiment found out that the bird's shape turned to circular dots after performing erosion(or opening) iteratively.** Hence, the birds that overlapped significantly cannot be separated using this method.

2.5 Conclusion

The total bird count in the source image is 43. If we count the overlapping birds as a single object, then the count reduces to 38. We got a decent result in figure 2.6(b), where all connected objects (birds) are labelled with a unique colour. The result obtained after opening operation (shown in figure 2.7(b)) to separate the birds justify our findings of the erosion/opening operation used in computer vision.