Assignment 2 – Basic Segmentation

Task 1.1

The technique used in this task revolves around selecting a threshold for the blue channel and eliminating/ not selecting any pixel from the original image that has a blue channel value higher than the selected threshold.

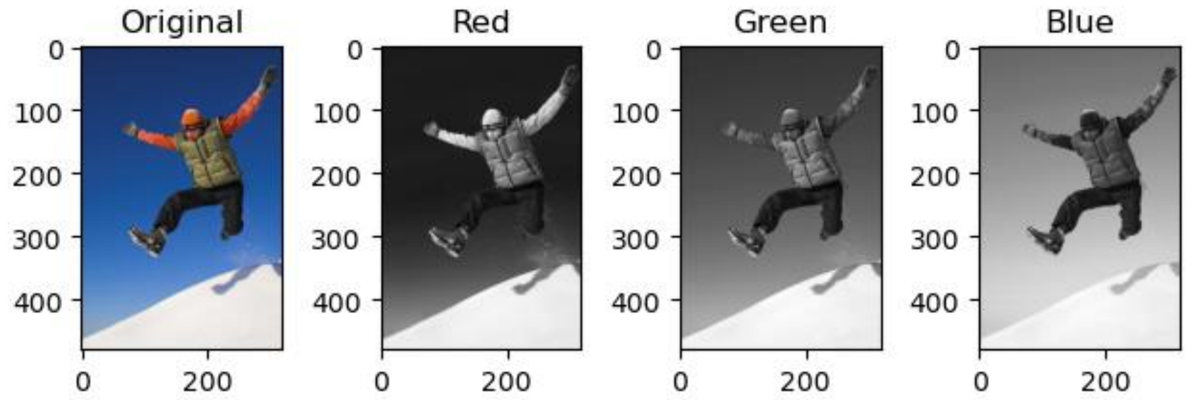

Figure 1 Segmentation in RGB channels using a grayscale.

In the picture above, we have separated the original image into its red, green and blue components and displayed each one using a grayscale, where areas that have a higher value of a channel will be closer to white while areas with a lower value of a channel will be closer to black.

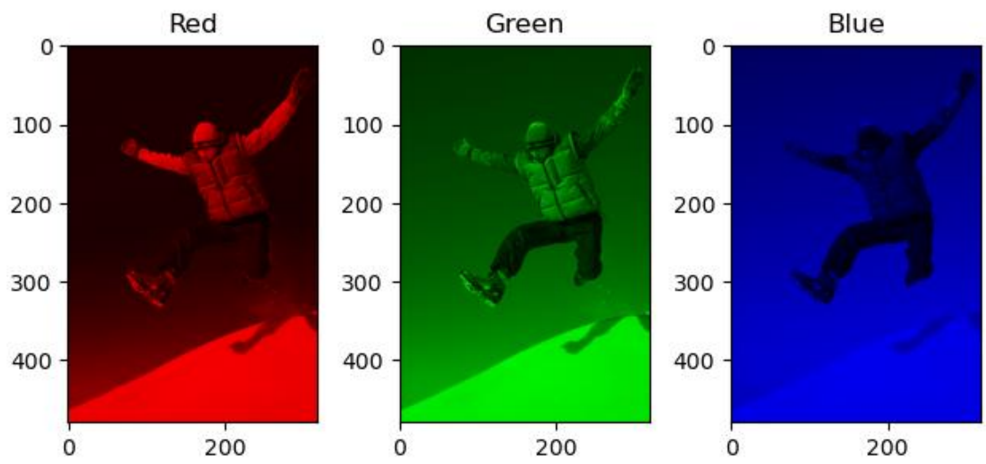

Figure 2 Segmentation in RGB channels

This picture represents the same segmentation as presented in Figure 1, where for each channel we have zeroed the values of the other channels. For instance, int the Red plot, the blue and green component of each pixel has been changed to zero, presenting only the red component of the original image.

Both segmentations allowed us to analyse how each channel is present in the original picture and start drawing conclusions on we will be able to remove the blue background.

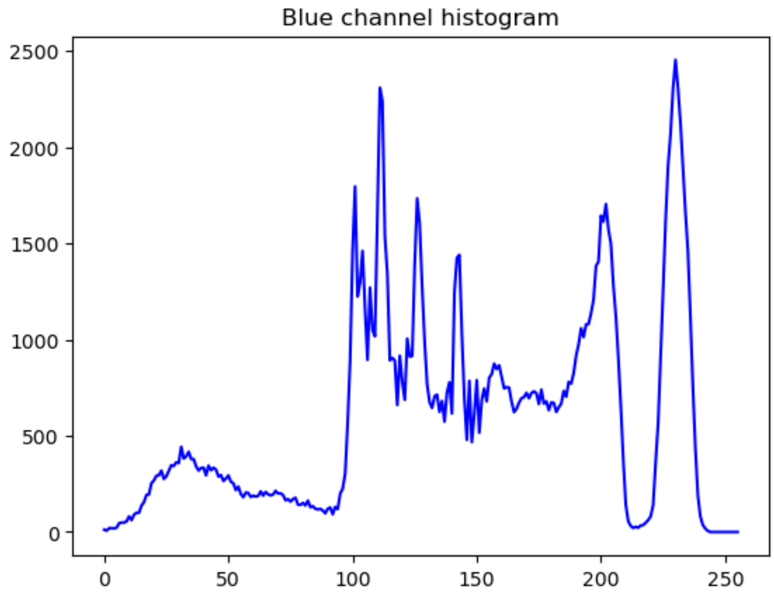

Figure 3 Blue channel histogram

Using the blue channel histogram, we can conclude that there are mainly to areas that have a high blue channel intensity, from around 90 to 210 and around 215 to 245. This analysis is useful when deciding the threshold values needed to remove the blue background.

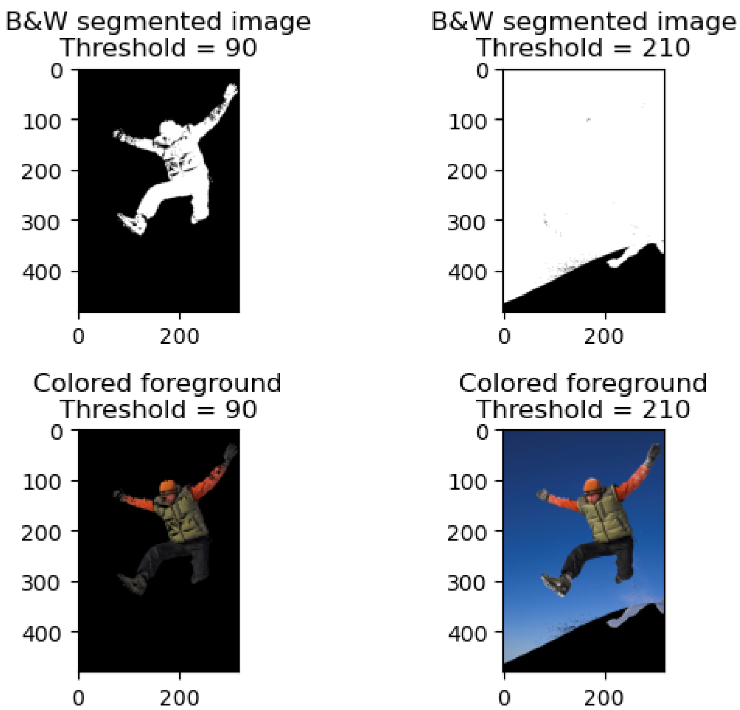

Figure 4 Removing background with different thresholds

Using the values taken from the blue channel histogram, we can see the difference between using the values 90 and 210 as thresholds. Using 90 as the threshold, the background as well as the snow has been removed while using 210 as the threshold only the snow has been removed, which allow us to conclude that the area from the blue channel histogram between 215 and 245 represents the snow.

Using the same method for the other images, we start by observing the segmentation in red, green and blue channels of the next image.

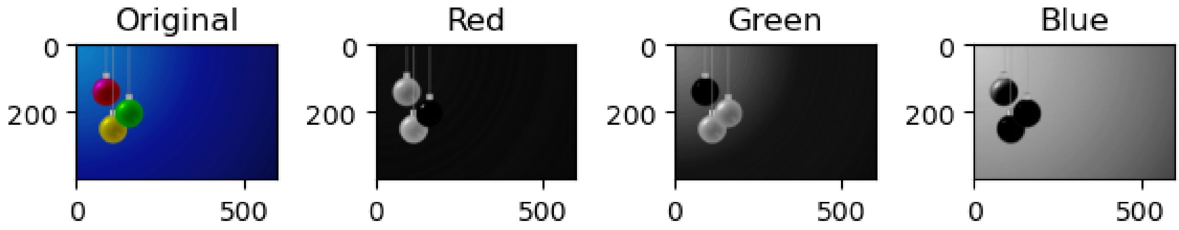

Figure 5 Segmentation in RGB channels using a grayscale.

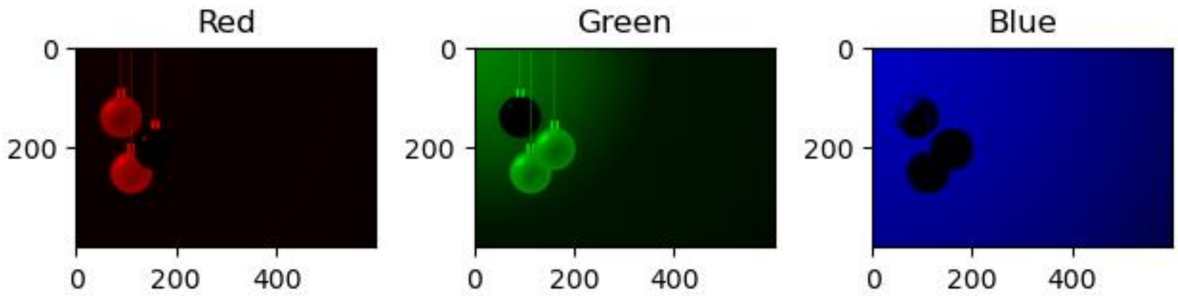

Figure 6 Segmentation in RGB channels

Observing the original image in Figure 5, we want to keep the three Christmas balls and their wires while removing everything else. It is noticeable that the two lower balls have very little blue intensity, with the highest one showing some blue levels in the top left area, which may be a problem when removing the background.

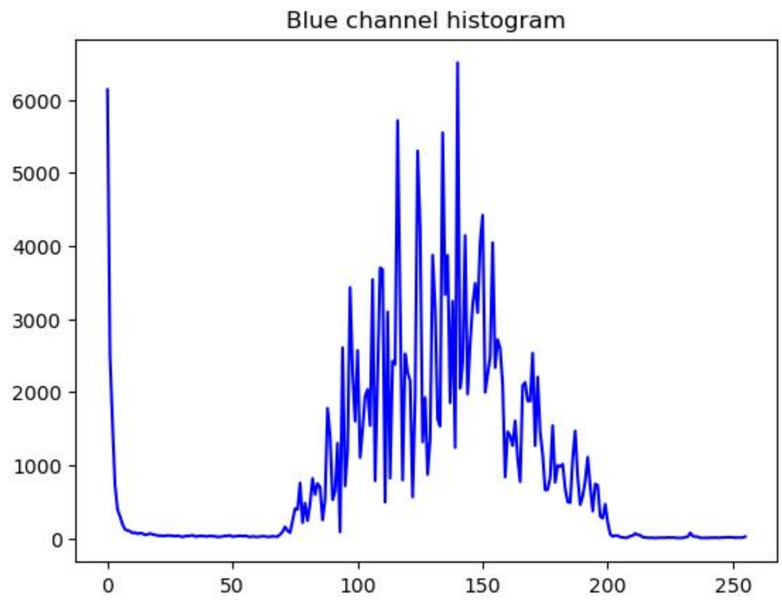

Figure 7 Blue channel histogram

Unlike the first image blue channel histogram, in this case, we see only one area ranging from around 75 to 200 where the blue channel has high intensity.

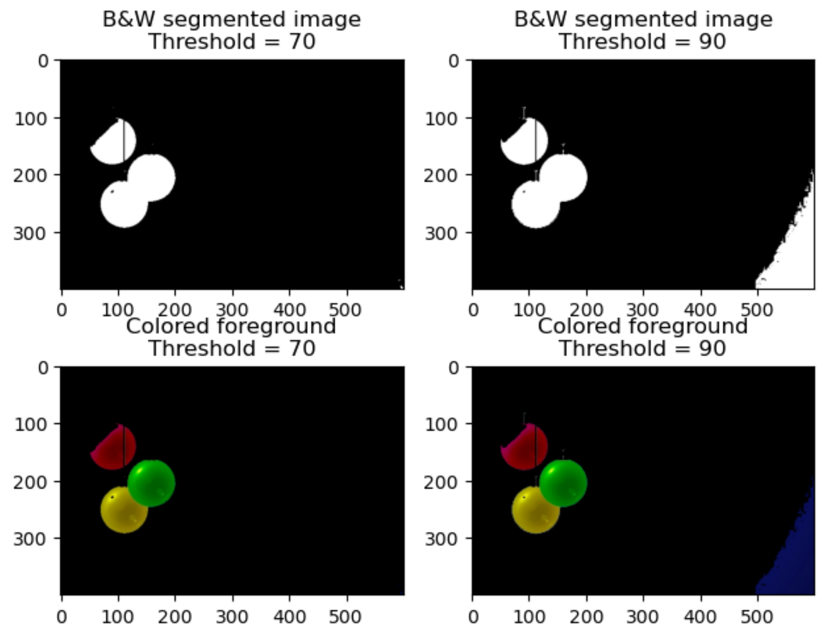

Figure 8 Removing background with different thresholds

Using the value 70 as the threshold, we are able to remove the background but as expected the top left corner of the highest ball was also removed. We attempted to increase the value used as threshold, but the results present on the two pictures on the right clearly show that is had a negative impact in the removal of the background, while not improving the area we wanted to capture.

Analysing the last picture, we can once again start by analysing the segmentation in the red, green and blue channels.

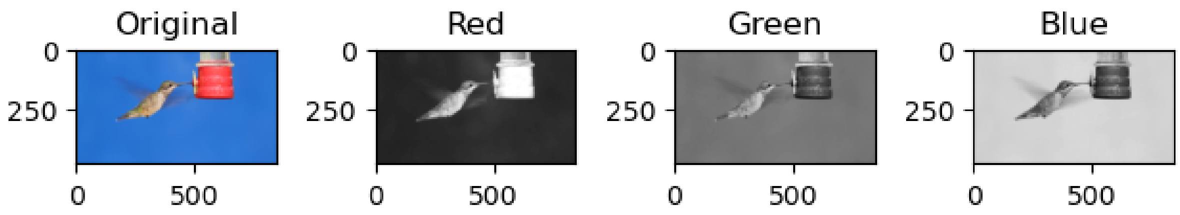

Figure 9 Segmentation in RGB channels using a grayscale

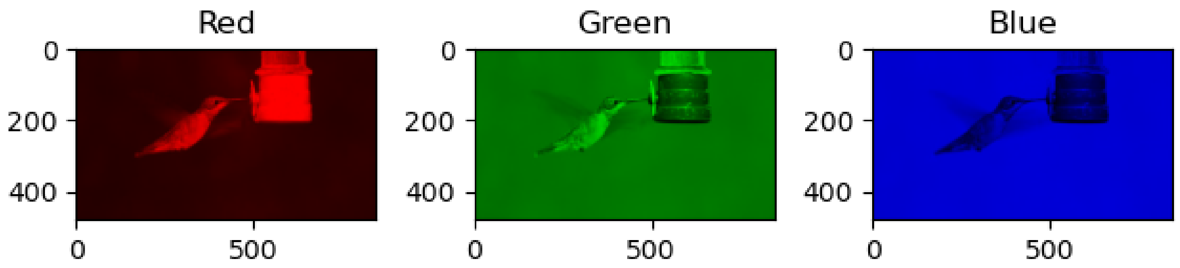

Figure 10 Segmentation in RGB channels

Notice the wings of the bird, which appear transparent/like a shadow due to their movement. Capturing them without also capturing the blue background is expected to be the hardest challenge for this image.

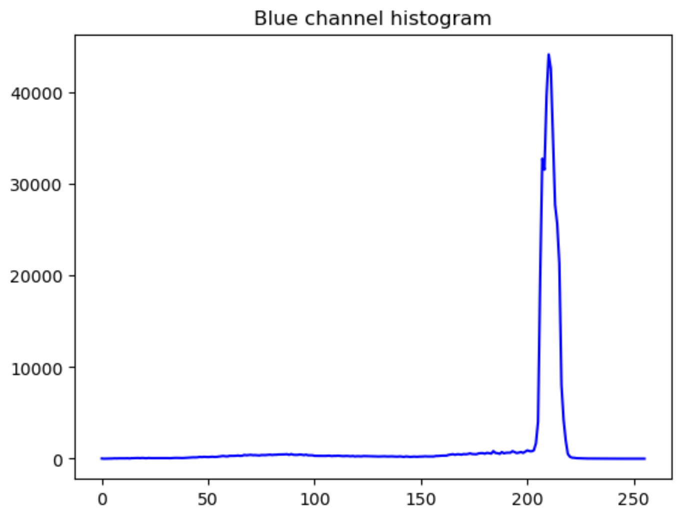

Figure 11 Blue channel histogram

Analysing the blue channel histogram, there is clearly one area of high blues intensity, starting around 200 to 225, but there is also a slight gradual increase in the blue intensity from 150 to 200.

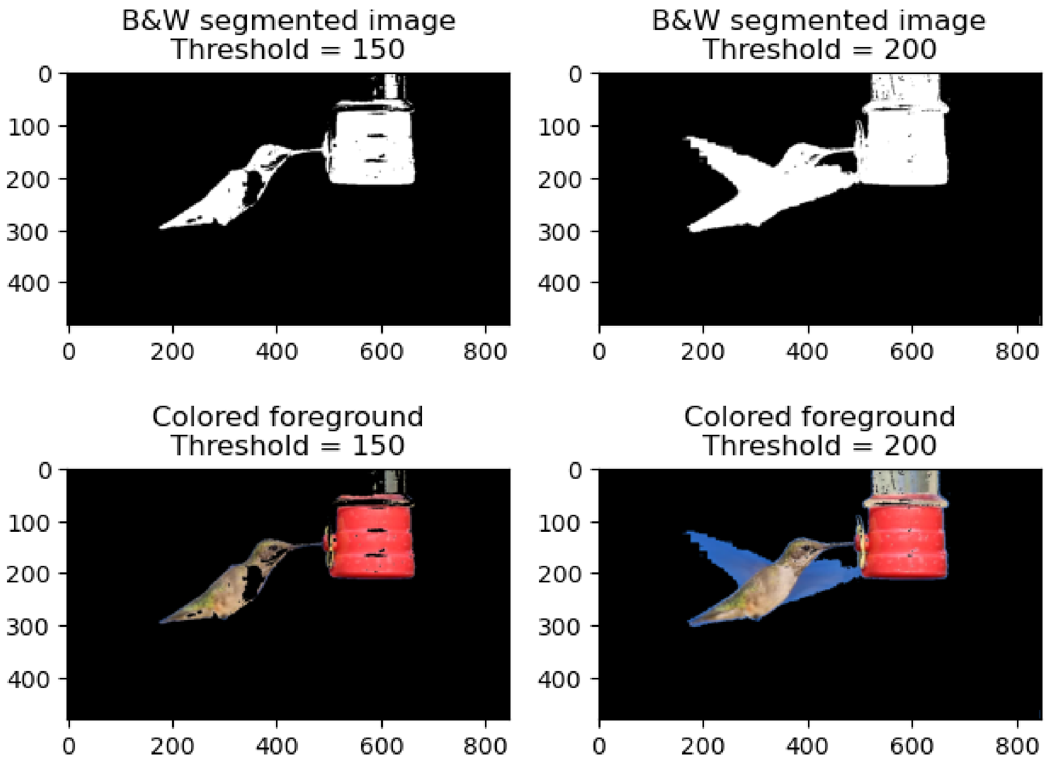

Figure 12 Removing background with different thresholds.

Using the values taken from the histogram, we can see that a threshold of 150 captures the bird as well as the red object but fails to capture the wings of the bird, as expected. Increasing the threshold to 200, the wings are now captured but so has been the blue background int the wings region.

Could there be low blue value in zones of the background? Could there be high blue values in some parts of the foreground objects? Is it always true that a pixel with a high value in the B component is always blue?

1. Yes, a low value blue zone will appear in the back due to the way the image is segmented, only the blue component of the RGB colour is considered, we selected a threshold and removed all pixels where the blue value is above that threshold.
2. No, for the inverse of the previous reason.
3. No, eg a white pixel, a white pixel in RGB is (255, 255, 255) which is not blue but has a maximum blue value.

To conclude, this segmentation technique yields good results when the foreground does not have or has little blue in it. When the blue colour is present in the foreground, this technique fails to identify and capture it while efficiently removing the background, producing errors in the final image.

Task 1.2

For the second task in this notebook, we started by splitting the image into its three RGB components the output is the same as in task 1.1.

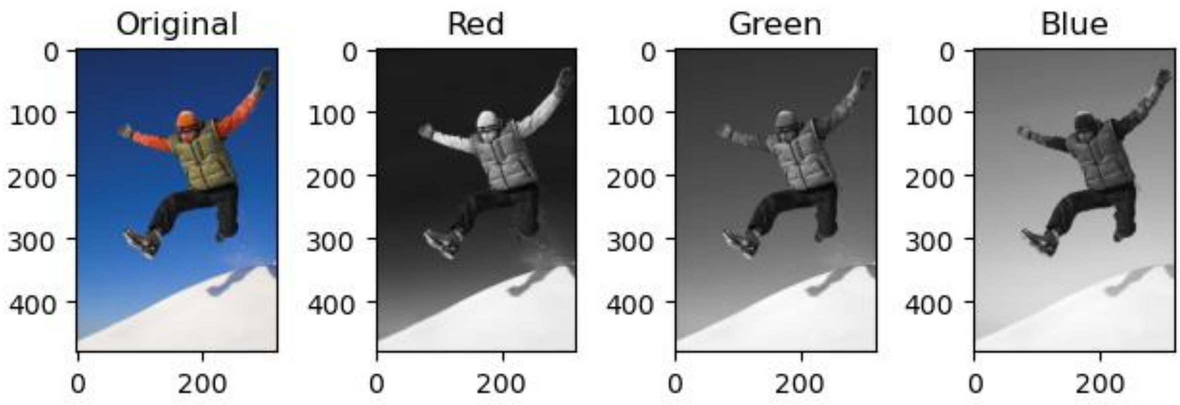

Figure 13 Segmentation in RGB channels using a grayscale.

Next, we had to compute the "blueness" factor of each pixel in the image given by the following formula: $blueness = B - \max(R, G)$. If a pixel had a blueness value of 255 it meant that that pixel is pure blue and can be represented as (255, 0, 0) in RGB, oh the other hand if a pixel had a blueness value of -255 it meant that it either had a max RGB red or green (or even both at the same time) value and a 0 on the blue scale.

Then, to better view how high (or low depending on how you analyse it) the blueness of each pixel is, we decided on two different threshold values as the "cut off" point, the values used were 25 and 40, these values were determined by an analysis of the frequency of the blueness values of the image.

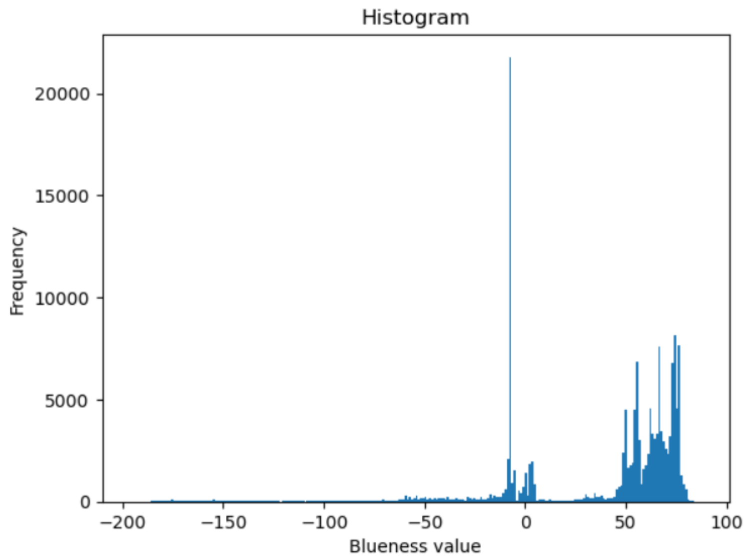

Figure 14 histogram with the frequency of each blueness values

The last coding step in this exercise was to "recolour" the image in a way that only the pixels above the threshold were visible, similar to task 1.1, with this in mind we painted over the pixels in the original image, with the pixels above the threshold being white and the ones underneath being black.

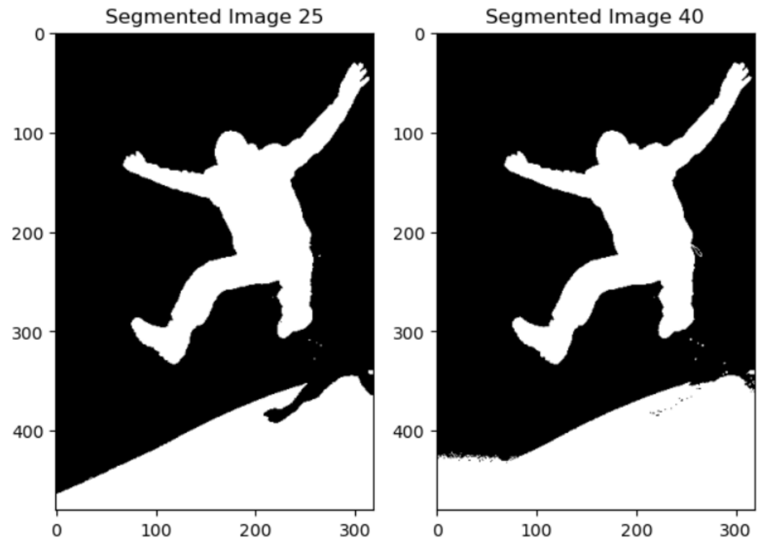

Figure 15 segmenting the image with different thresholds

As an additional exercise we decided to see how the blueness thresholds we chose would affect the base image, how much of the background versus the person in the centre would change.

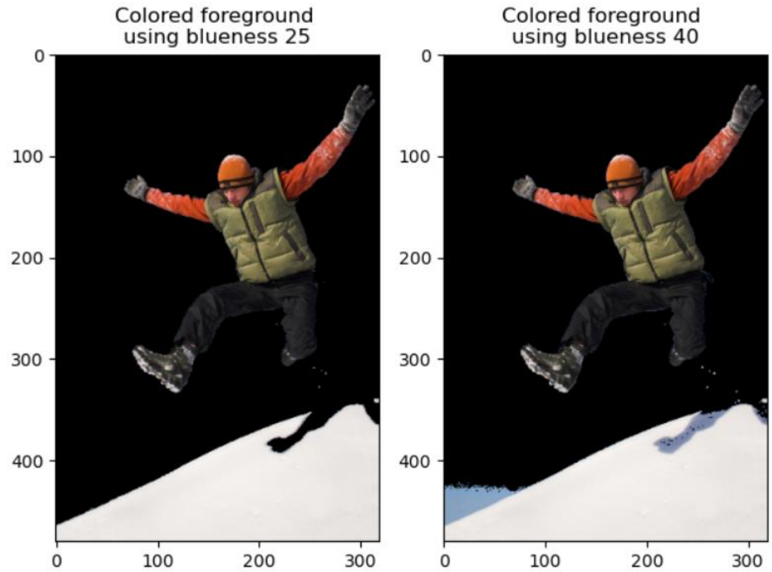

Figure 16 colouring the image with different thresholds.

Overall, the second approach much provided **better results**. This was expected because of the way the blueness factor was calculated.

Task 2

This task required us to essentially “cut” a part of an image and “paste” it into another one. This were the two images we chose:

Figure 17 images used for task 2.

There were multiple options to do this task, the simplest, and the one we used, is to use a simple blueness method we implemented in the previous tasks on the first image and paint over the second image with the selected part, this is the result with the addWeighted() function:

Figure 18 joined image with addWeighted().

We also decided to try and experiment with manually colouring the image, which provided a more concise result:

Figure 18 joined image manually coloured.