COMP9517 Computer Vision

Assignment1

Zhiwei Wang

Z5135560

October 2019

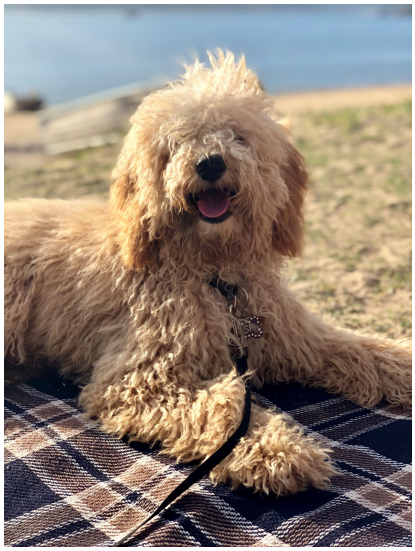
UNSW AUSTRALIA

Task 1

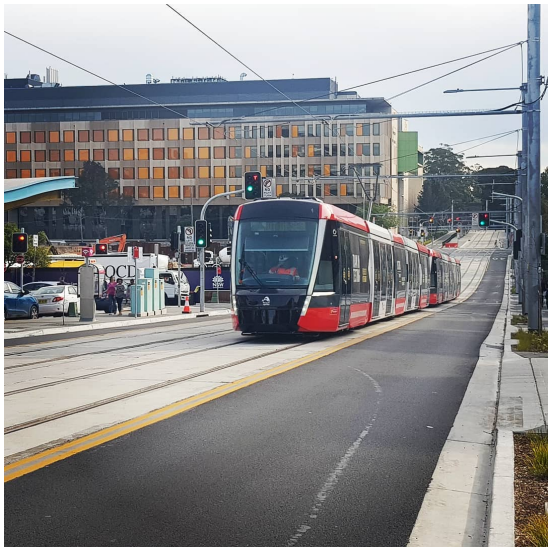
Starting with an input colour image (let us call this image *A*), i combine the three colour-bands into one band using the following equation:

$$I\,(x,y) = 0.299 * r(x,y) + 0.587 * g(x,y) + 0.114 * b(x,y)$$

Where *r*, *g* and *b* are colour-bands of image *I*.



(a)



(b)

(c)

(d)

Figure 1: (a) The original dog.jpg (b) The original light_rail.jpg
(c) The output of dog.jpg (d) The output of light_rail.jpg

The outputs of Task1 is shown in Figure1 which is (c) and (d). Obviously, they're all grey-level images. Losing color means losing a lot of important information. However, many features are still consistent with the original image, such as the appearance of dogs and trains.

Task 2

In task 2, based on the result of task 1(called I), a gray scale oil-painting image is created(called J). In this task, every pixel in the image is traversed. First of all, define an appropriate neighborhood around the current pixel. Secondly, use **np.histogram()** and **np.argmax()** to **find the most frequent** local pixel value in the neighborhood.

Here is a question that needs to be discussed in task2 and task3. It's the image padding. Pixels at the edge of the image sometimes fail to reach the desired size of the neighborhood. For example, The pixel whose coordinate is (0,0) has only 9 neighbors if the window size is 5. So, the function **np.pad()** is used to set all pixels outside the picture to 0 (according to the window size) to ensure that all pixels have enough neighbors. As a result, the edges of the output images for task 2 and task 3 may be black, because the value of brightness 0 is black.

(a) (b) (c)

(d) (e) (f)

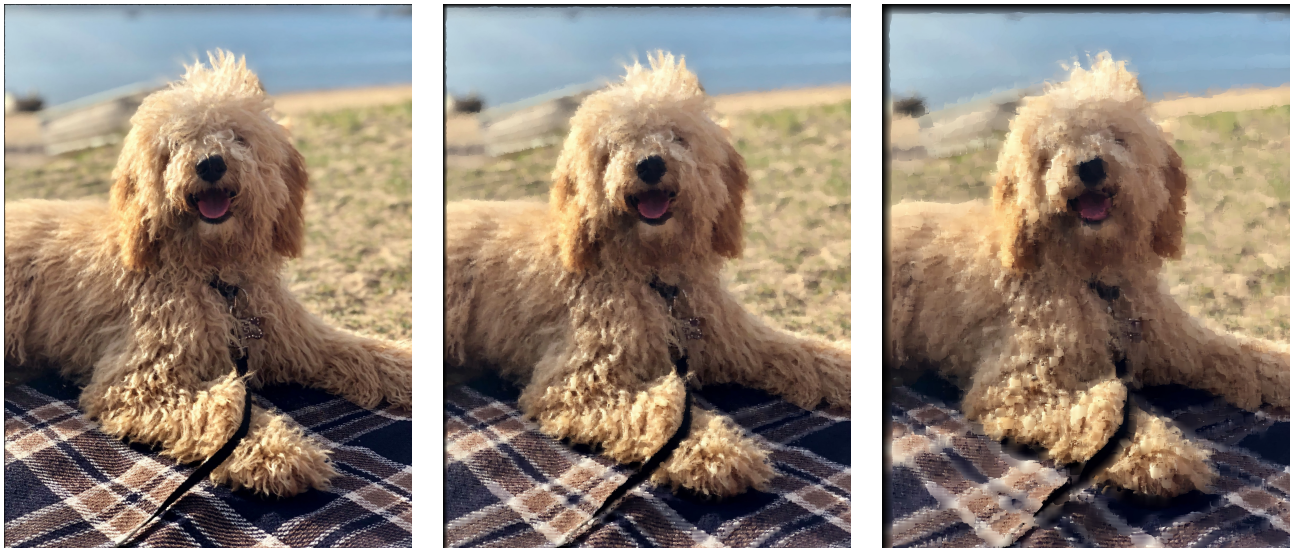
Figure 2: (a) dog: window size 7 (b) dog: window size 19 (c) dog: window size 41 (d) light rail: window size 7 (e) light rail: window size 19 (f) light rail: window size 41

Task 3

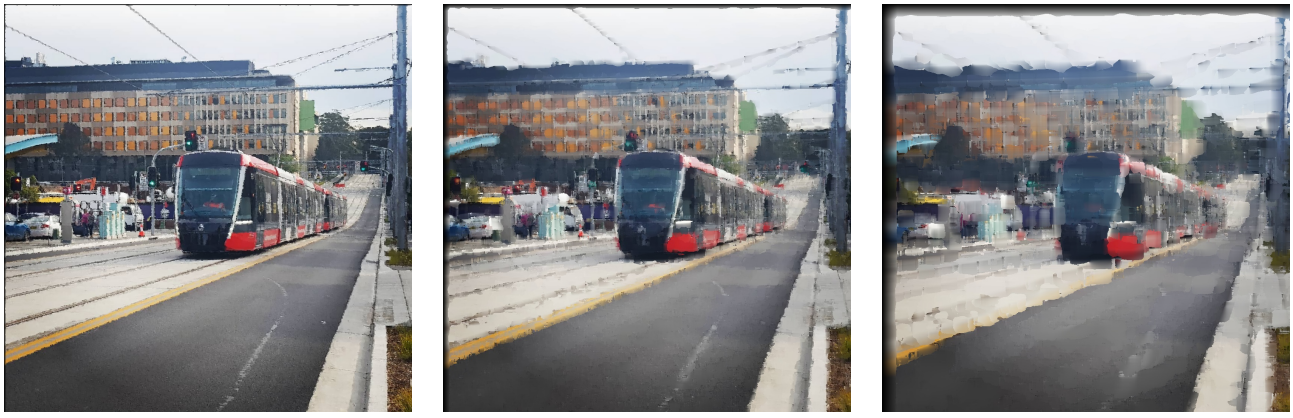
Task 3 will construct the final ‘oil painting effect’ image, based on the result of task 2(called J). All pixels in the original image are traversed.

At the beginning, pad the J according to the window size(mentioned in task2). Next, find all pixels which have the same value of current pixel in the neighborhood in the padded picture. **Calculate the mean** of these points in each band in the original picture. In order to make the coordinates of the padded J correspond to the original picture, the original picture is also filled.

Significantly ,the original picture is three-bands. So, function **cv2.split()** is used to split into three channels and pad them. Then **cv2.merge()** is used to merge them to a three-bands image.



(a) (b) (c)



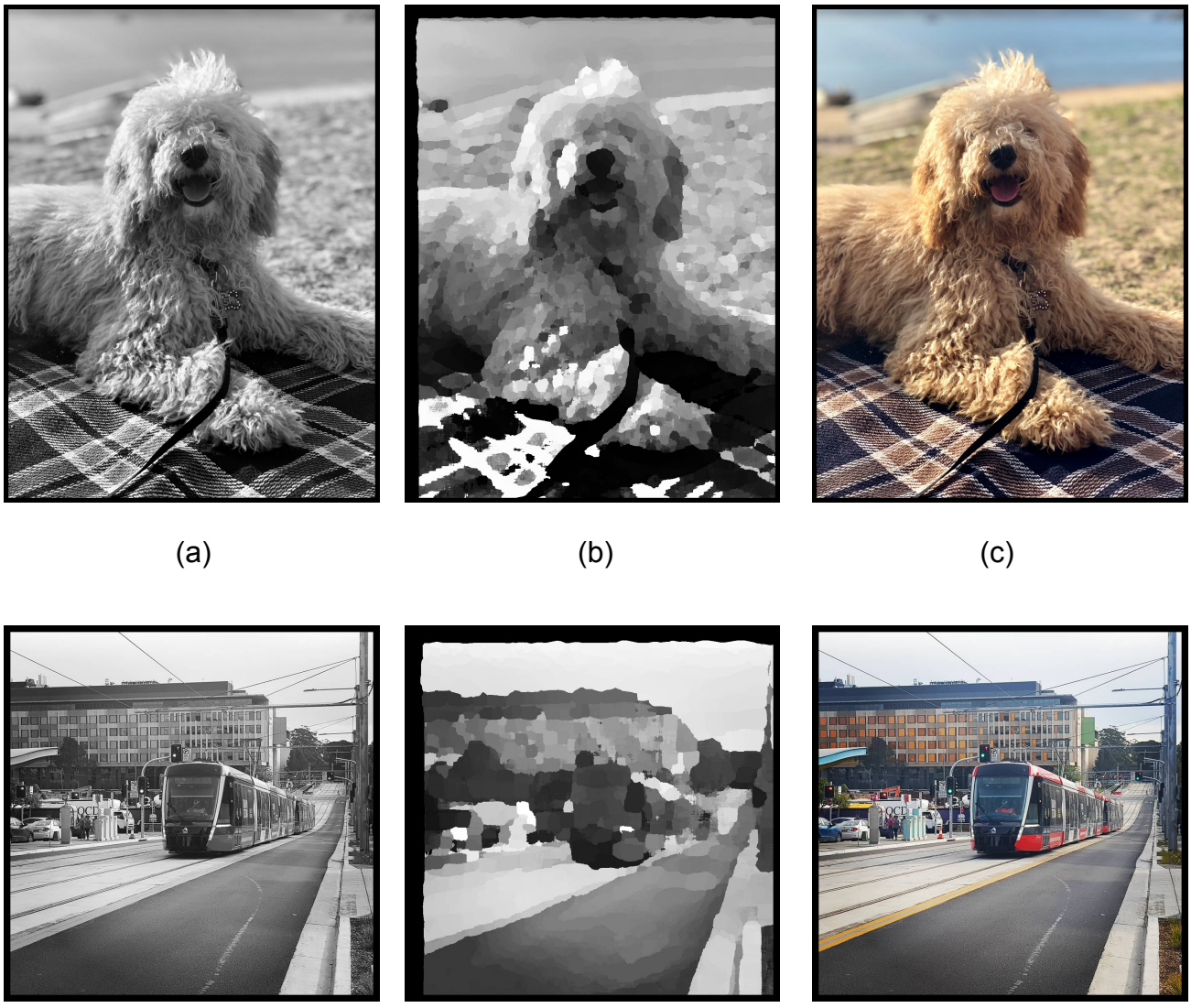
(d) (e) (f)

Figure 3: (a) dog: window size 7 (b) dog: window size 19 (c) dog: window size 41 (d) light rail: window size 7 (e) light rail: window size 19 (f) light rail: window size 41

Conclusion:

It is shown in figure2 and figure3 that oil painting effect becomes more obvious as the window size increases. Meanwhile, with the same window size, the effect of task2 was more obvious than that of task3.

Intermediate image:
window size: 41



(a) (b) (c)

(d) (e) (f)

Figure 3: (a) dog: padded I (b) dog: padded J (c) dog: padded original image (d) light rail: padded I (e) light rail: padded J (f) light rail: padded original image