Queen Mary
University of London

Introduction to Computer Vision

Coursework

Submission 2

Your name Yu-Jhen, Hsu

Student number 190279963

Question 4(a)

I_t

I_{t+1}

Motion field of I_{t+1}

Question 4(b)

I_{t+1}

P_{t+1}

Your comments

The prediction of the next frame is acceptable. Because of the limit of the search area, the image will not have much difference compare with frame t. However, the prediction can be improved by taking the relationship between speed, shape, size, and distance.

Firstly, the speed in different position of image influence the accuracy of prediction. It is seen that the movement of the car is more than the search area we defined, especially when the car is closer to the camera. The search area is 20 pixels, and the block area is 16 pixels, meaning the estimated motion vector will fall in the range from negative two and positive two. The speed of the car should be considered when doing block matching. The speed of cars being closer to the camera will be much significant compared with those being farther to the camera. Therefore, the block matching might be improved by considering the speed changed related to the distance of the camera.

Another issue related to block matching is ignoring the effect of the change of shape and size when objects are moving. When the object moves further, the size of the car is smaller, and some features are lost. When doing the block matching, it might be unable to find the same area and search for possible movement.

However, block matching might be useful when the objects move between left and right and assuming that the object's speed, shape and size will only slightly be changed over than time. In that case, the block matching might able to provide a higher accurate predicted image.

Question 4(c)

P_{t+1}

Block size = 4x4

P_{t+1}

Block size = 8x8

P_{t+1}

Block size = 16x16

Your comments:

Because the black area is cover by the original frame, the predicted result will look awkward when the block area becomes smaller. Because some parts of several blocks are overlap and the blank space are filled in by original frame, the picture will use more pixels from the original image and object will look weird. Although the predicted picture looks worse when decreasing the block size, the motion vector will increase and provide more details on how the object moves in both the object and its shadow. When the bock size is equal to the search area, nothing happens because it only searches for the original area.

Decreasing the block size might be able to deal with the problem of the relationship between size and shape, and the distance from the camera. Because the smaller size of the block only contains a smaller number of features, significant features might remain when the object is further from the camera. However, it might need to find a better way to fill in the blank area to give a more reasonable predicted image.

Question 4(d)

P_{t+1}

Window size = 8x8

P_{t+1}

Window size = 16x16

P_{t+1}

Window size = 32x32

Your comments:

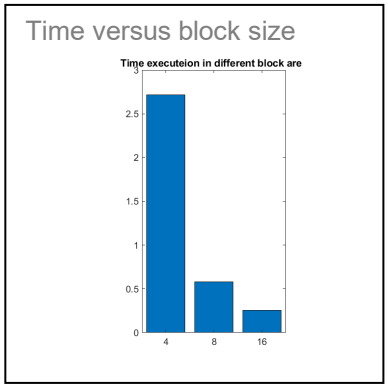
There are two issues is as same as question 4(c), which is search area equal to block area and the better methods to fill in the blank area.

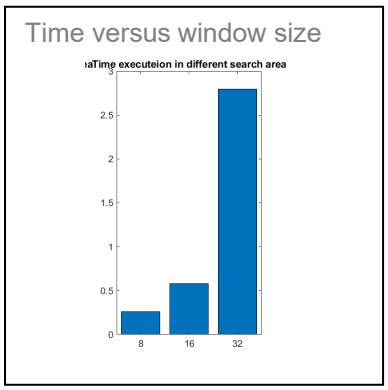
When increasing the search area, it can solve the problem related to the speed. Because the search area is more significant, meaning it can compare with the area with is further than the smaller search area. However, because of the increasing the search area, the minimal cost might choose the other area, which has a similar colour with original frame and result in the inaccuracy in some blocks.

By doing the question c and d, when doing to block matching, it will need to define the block area, which can improve by defining object rather than a small block to increase the accuracy and giving larger search area when the object is closer the camera.

Question 4(e)

Plot graphs:





Your comments:

The time complexity of the block search is O(n^4) because there are four loops inside the function. When increasing the block size, the number of motion vector and calculation time decreases. However, the execute time will increase when increase the search area, because a block will need to compare with more neighbour blocks. The decrease and increase rates are exponential. Therefore, expanding the search area will require a much larger time compare with the previous search area. However, decreasing the block size will only decrease significantly when the block size is large.

This method is using pixel difference, meaning calculating the MSE based on different pixels value. It can also be achieved by using the colour distribution similarity. However, the time complexity of block search will increase to O(n^6), because creating the colour distribution will require extra two for loop to calculate the histogram in each block.

It seems that the full search will require plenty of time. It might be a better idea to use another method such as logarithmic, n-step and conjugate search to decrease the time complexity and enhance the performance.

Question 5(a)

Original frames:

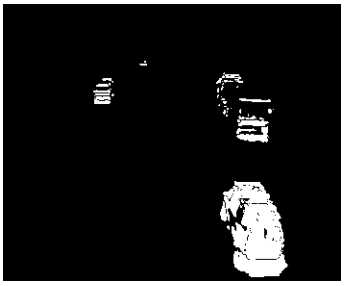
Reference frame

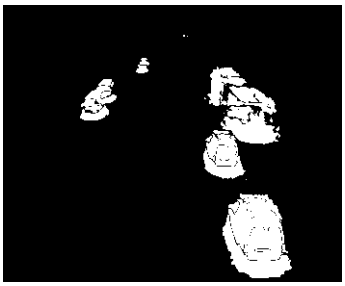
Selected frame 1

Selected frame 2

Frame differencing:

Threshold results:





Your comments:

Making the frame difference needs to be careful about the type of data type. Because some pixels subtraction might less than 0, it needs to change to double type and absolute the subtraction result to get the correct answer.

The image needs to blur before doing to pixel subtraction to reduce the influence of noise. By applying the threshold, it will provide a more precise region. The level of the threshold will influence the area being remained. If the threshold is more significant, more pixels are eliminated and result in less information in the resulting image. In the other way, if the threshold is too small, the noise will not be able to be excluded and result in the picture contains noises. Therefore, threshold plays an essential role in frame differencing.

Question 5(b)

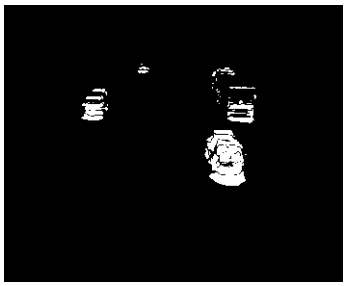
Original frame:

Selected frame 1

Selected frame 2

Frame differencing:

Threshold results:



Your comments for 5a,5b:

The result of using the previous frame as the reference frame is much better than the one using the first frame as the reference frame. Because using the first frame as the reference frame will result in duplicating the object number when that objects in the reference frame are moving to the other position or disappear from the frame. Therefore, using the previous frame as the reference frame can provide much accurate result.

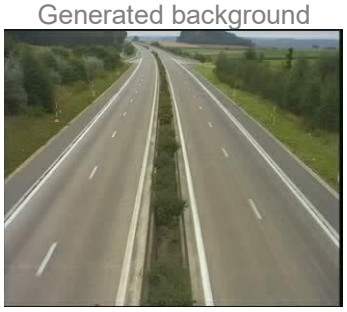
Although using the previous frame as reference frame provide a more accurate result, the objects being further from the camera are ignored in the result because the movement of the objects might less than 1 pixels and be recognised as background. The objects being closer to the camera have better regions as having significant movement. Because of the projection from the 3D world to the 2D image, the objects which are closer to the camera will have more significant movement compared with the object being further than the camera in the same speed.

The middle part of the objects is excluded because the pixel value does not change in those positions. Therefore, some objects might be counted twice because the object is separated into two parts.

Some methods can set the threshold by itself, meaning that it can reduce the influence of human but increasing the calculation time also. Therefore, the function picks up the threshold manually to decrease the calculation time.

Question 5(c)

Generated background

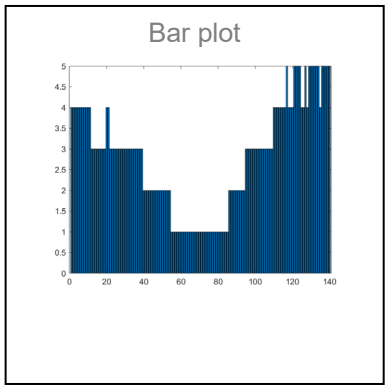


Your comments:

There are two ways to the generated background, being mean filter or median filter. The mean filter can reduce the memory requirement but might influence by the noise. The median filter will reduce the influence of the noise but requires higher memory requirement.

This background is generated by the mode filter which using the same concept of the median filter. This method assumes that the pixel value in a specific position is the value which appears most frequently. To achieve this method, it will need to have the space to store the frequency of value appears in a specific position during all the frame. Therefore, it will require more space to store in memory.

Question 5(d)



Your comments:

To count the object number in a specific frame, it will need to use several techniques, being dilation, erosion, and connected component algorithm, to produce a better region to count the number of objects. The purpose of dilation and erosion is to fill in the gaps using dilation and delete the border pixels which are generated by dilation using erosion. In the next step, the connected component algorithm is used to count the number of objects in a specific frame. The labels having less than 30 components will be eliminated to reduce the influence of noises. Although it can reduce the impact of the noise, it might also remove the objects which are farther than the camera and only occupy a small number of pixels. Because the function does not exclude the shadow regions, two nearby objects will be recognized as the same object.

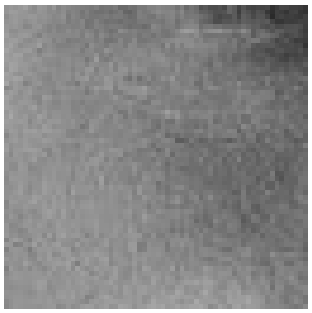
The result is robust when the objects are not closed to each other and close to the camera. The function can count closer objects better than further objects. Because the function is trying to exclude the areas which are generated by noise, it will show the unstable counting when the object is far away from the camera. Therefore, there is a trade-off between count every possible object and removing the influence of noise. Moreover, when the object close to each other, the shadow will result in two objects being recognized the same objects. This problem might be able to solve by including shadow detection.

Question 6(a)

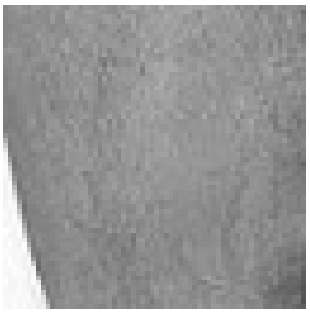
Three non-consecutive windows

W1

W2

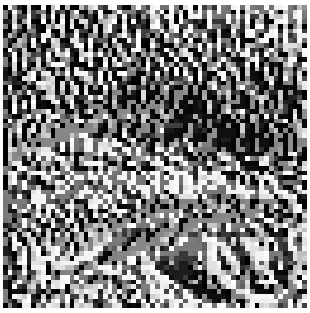


W3

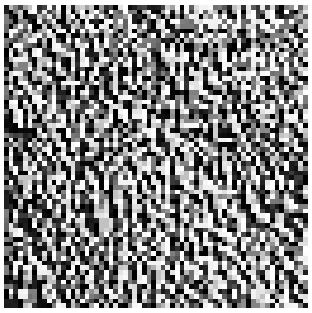


LBP of windows

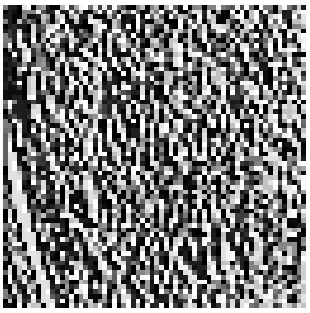
LBP1



LBP2

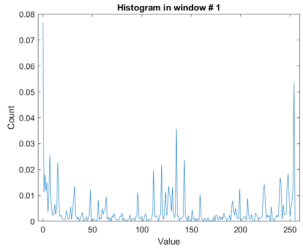


LBP3

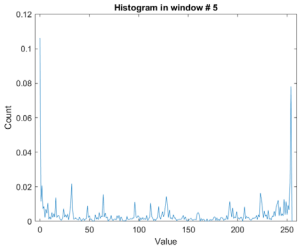


Histograms of LBPs

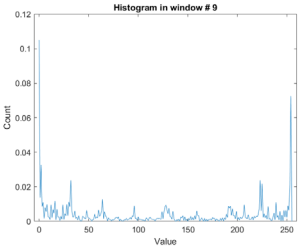
H1



H2



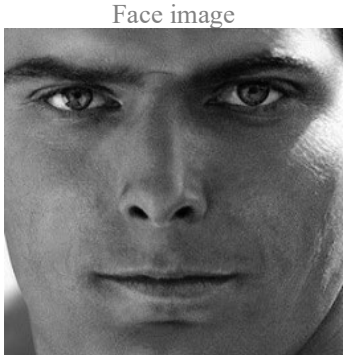
H3



Question 6(b)

Two example images:

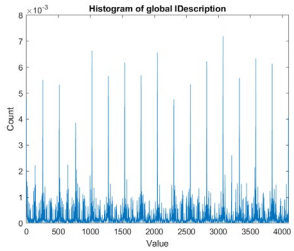
Face image



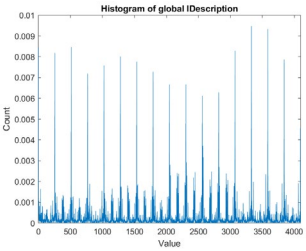
Car image

Descriptors:

Face descriptor



Car descriptor



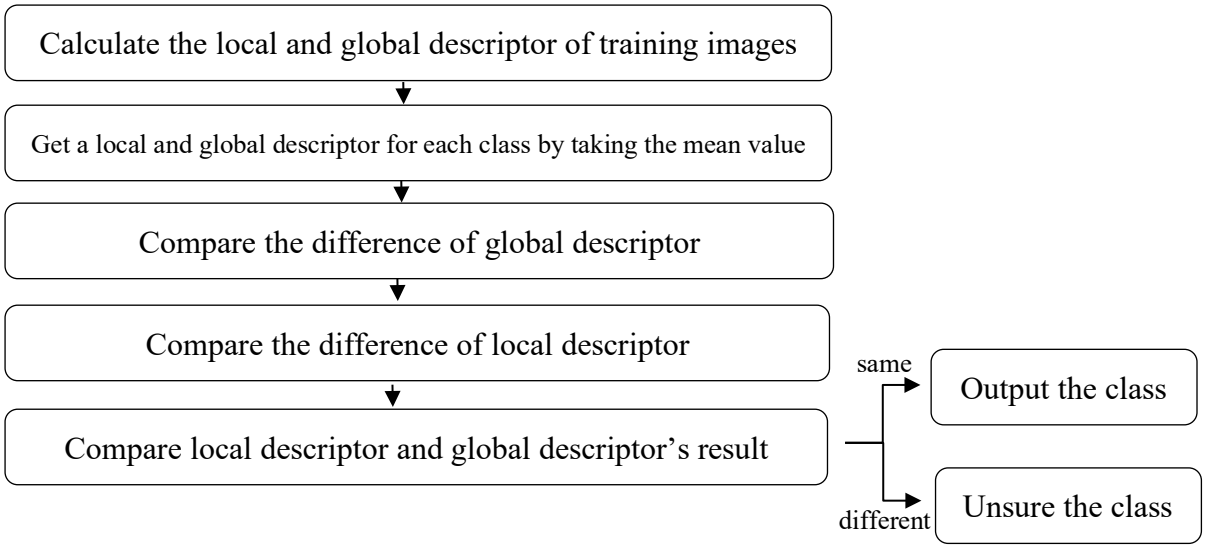
Your comments:

The mirroring method needs to be done to calculate the LBP value in the border. The border pixels will be excluded without mirroring method because the LBP is comparing with the eight neighbours and the border does not have enough neighbours and conquer the problem of index out of bound in the matrix.

Each LBP windows show the different feature of the specific part of the picture and have the unique histogram. By cutting the LBP into several windows, it can have more features to use in the classification. If the pictures only generate one LBP window, it will only have 256 features. However, when the window sizes increase, it creates window number multiply by 256 features which allow improving the accuracy in classification. This improvement is because the number of features can solve the problem of under-fitting.

Question 6(c)

Block diagram of classification process



Your comments:

Using the simple image as the classifier might suffer from the problem of position and using accessories. For example, the face-2 image has much longer eyelashes and misclassify when using it as the classifier. When using the face-2 image, the global descriptor will prefer to classify image to the car, and the local descriptor will be closer to half number of matching windows. The other problem is that the position of features might result in a different result. For example, the face-1 has more area in the forehead compare with face2 and face3. Therefore, the data pre-processing might be necessary when using this technique.

Using the global descriptor that is the mean of the same class's image can able to classify the image correctly. Because the difference between car and face MSE is small, it is possible not a very significant method. By including the component of the local descriptor, it can address this problem because the local descriptor will count the similarity based on different blocks. By combining local and global descriptor, the classification is robust.

Question 6(d)

Your comments:

Decreasing the window size will result in increasing the size of the global descriptor. The MSE will decrease, but the calculation time and memory requirement will increase. Because the feature is only focused on small regions, it contains less information in a specific position. It can predict better if the position of feature is identical in every image. Therefore, it requires more time to do data pre-processing. If the position of features, such as eyes and nose, are in the identical in a different image, decreasing the window size will provide a more robust result in the local descriptor.

The calculation time and memory requirement increase because of the increase in window number. The function will need to do the frequency function in each window and result in the increase in calculation time. The rise of window number means the computer requires more space to store the information.

Question 6(e)

Your comments:

Increasing the window size will result in decreasing the size of the global descriptor. The MSE will increase, but the calculation time and memory requirement will decrease. The MSE will increase because each window contains more information. Any missing features in that area will heavily increase the MSE. However, because each window provides more information, the local descriptor can have a better result. For example, the classification of the test image has a 4/4 similarity while having a 54/64 similarity when decreasing the window size. Increasing the window size does not require the precision of the position of specific feature and can reduce the time in data pre-processing.

Question 6(f)

Your comments

Because the object's position, shape and size might change during the time, it will need to change the way to get the detected area inside each frame. It might be done by changing the area to do the detection in the frame. Therefore, the function will add two parameters – the search area and search size.

The search area defines the area to do the LBP, and the search size determines the size of the search area. Because the global descriptor is done and window number is defined, the function needs to make sure the window number in classifier and search size is the same. When the window number is the same, the matrix subtraction can be done, and MSE can be calculated and finish the classification process. The size of the search area will decrease when the function does not find the target class at this level and until the region of the search area is less than a specific value.

This method has a drawback that will require significant time in doing the calculation. The possible improvement is using the local descriptor. By using the local descriptor as feature bags, if the specific blocks in the search area contain lots of similar features, it might mean that the object is located near this area and can shrink the search area.