Assignment 3

Special Topics in Computer Vision

Gagandeep Singh: 2016037

Question 1

The following are three different ways to subtract the background when the camera is stationary:

1. For a stationary camera, we could take two consecutive frames and take the absolute difference of the two images. The output would be an image with the background subtracted. To get better results, we could blur it and create a threshold. This method would only work if the objects in the foreground are moving
2. One way to subtract the background is to consider a frame initially where there might not have been any foreground or moving items. This frame could act as the base for our foreground detection. For any frame thereafter, we can subtract the initially selected frame from the current frame, thus deleting the background and detecting the foreground for us.
3. Another way to detect the foreground is by taking the mean of pixels of each frame prior to a particular frame at time T. The mean/median/mode of all the frames uptil time T-1 would be regarded as our background. This background could be subtracted from the frame/image at time T and a threshold could be applied to get the foreground.

Question 2

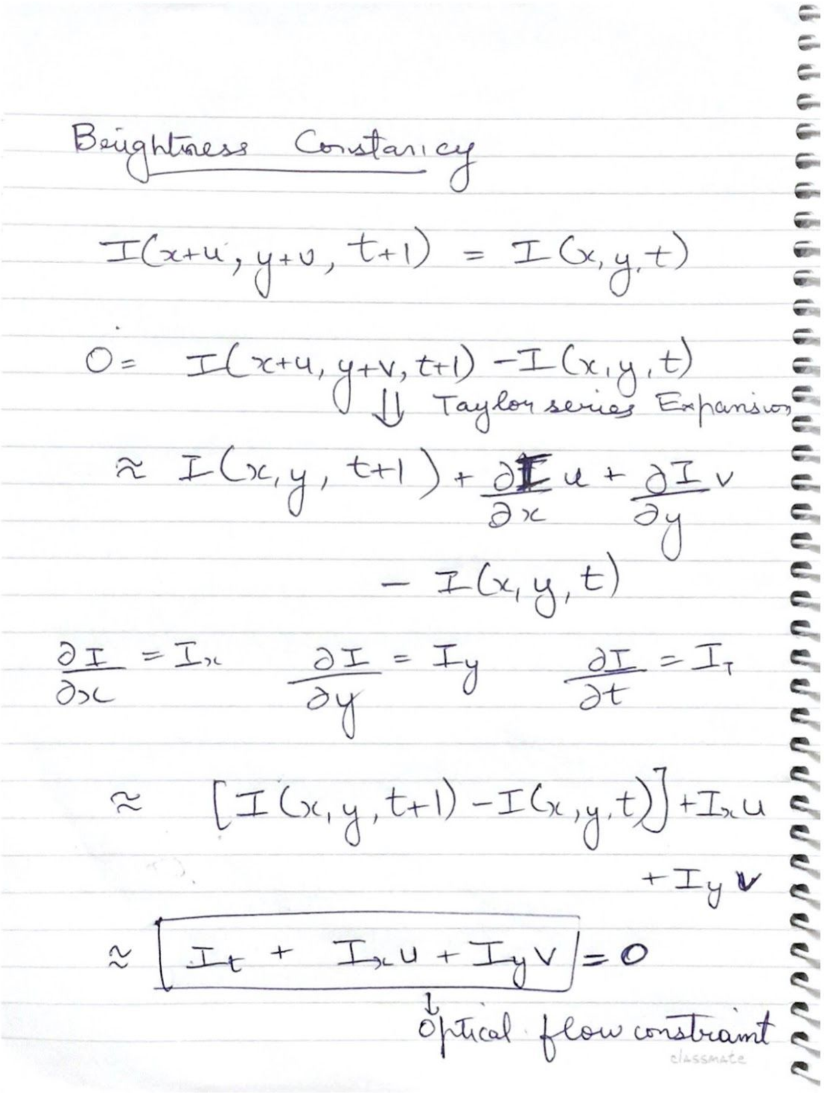
Motion History Image(MHI) is a way to detect motion in a series of images. This technique uses temporal changes of the series of images/frames are captured into a single image template which signifies how recently the motion has occured at that particular pixel.
MHI computes a single static and bidimensional map that integrates both the spatial location as well as the temporal history of motion in the object and thus the spatial and temporal resolutions get retained.
Using MHI, we can depict the movement of an object in a single image as well as it is shown in the video counterpart.
The final MHI image gives a grayscale image as the output. The parts which are the brightest on the grayscale are the ones which have moved the most recently.

Few Applications:

1. Motion History Images are used to represent motion sequences in a compact manner.
2. Used to detect diseases in a human body by capturing images and detecting movements.
3. To visualize the cerebral blood flow changes during occlusion

Question 3

Brightness Constancy: In optical flow, if the some object is moving and the pixel in each frame changes its position, the projection of a pixel looks the same in every frame, i.e. , a pixel will look exactly the same in every frame of the video.



Brightness Constancy

$$I(x+u, y+v, t+1) = I(x,y,t)$$

$$0 = I(x+u, y+v, t+1) - I(x,y,t)$$

⇓ Taylor series Expansion

$$\approx I(x,y,t+1) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v - I(x,y,t)$$

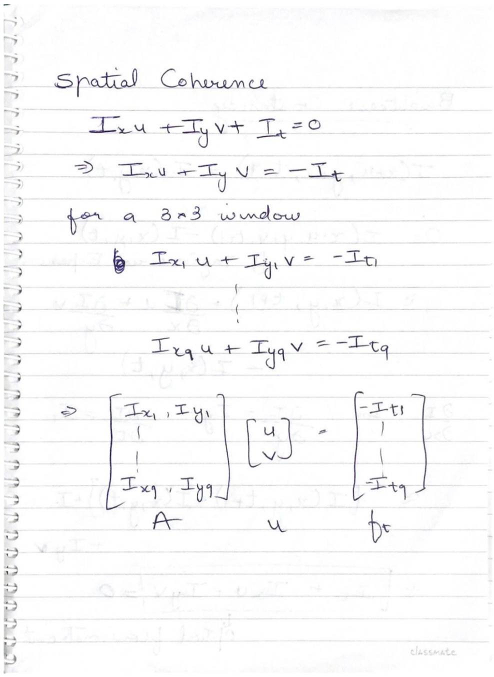
$$\frac{\partial I}{\partial x} = I_x \qquad \frac{\partial I}{\partial y} = I_y \qquad \frac{\partial I}{\partial t} = I_T$$

$$\approx [I(x,y,t+1) - I(x,y,t)] + I_x u + I_y v$$

$$\approx \boxed{I_t + I_x u + I_y v} = 0$$

Optical flow constraint

Spatial Coherence: Spatial coherence states that every point in a frame moves similar to how its neighbours move while scrolling from frame to frame.



Spatial Coherence

$$I_x u + I_y v + I_t = 0$$

$$\Rightarrow I_x u + I_y v = -I_t$$

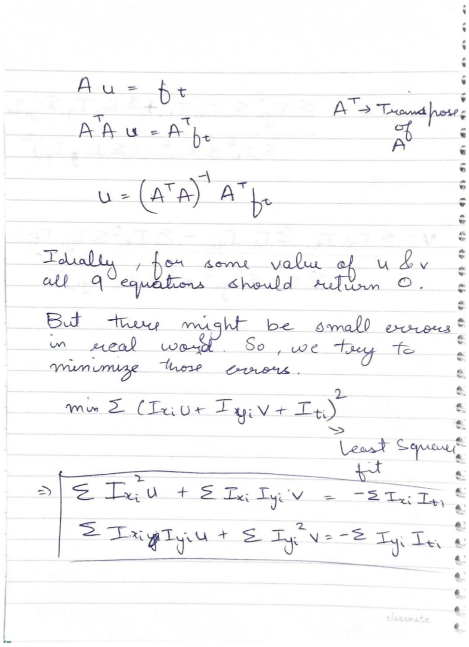
for a 3×3 window

$$I_{x_1} u + I_{y_1} v = -I_{t_1}$$

$$\vdots$$

$$I_{x_9} u + I_{y_9} v = -I_{t_9}$$

$$\Rightarrow \underbrace{\begin{bmatrix} I_{x_1}, I_{y_1} \\ \vdots \\ I_{x_9}, I_{y_9} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{u} = \underbrace{\begin{bmatrix} -I_{t_1} \\ \vdots \\ -I_{t_9} \end{bmatrix}}_{b_t}$$



$$A u = b_t$$

$$A^T A u = A^T b_t$$

$A^T \rightarrow$ Transpose of A

$$u = (A^T A)^{-1} A^T b_t$$

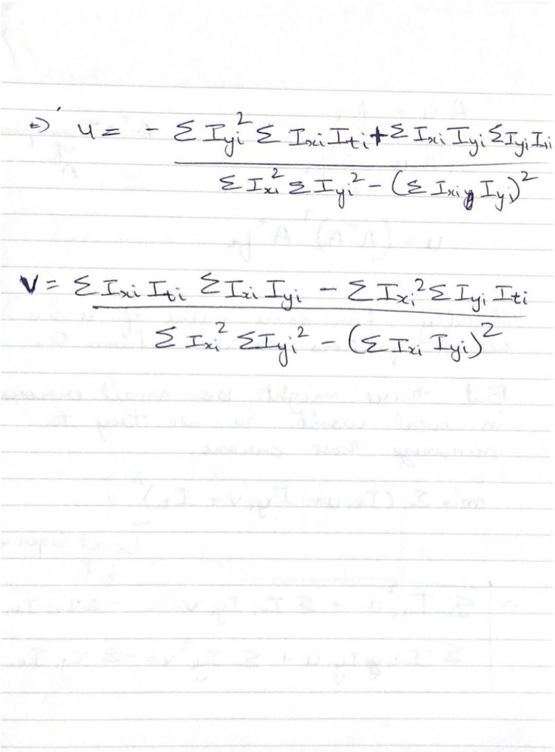
Ideally, for some value of u & v all 9 equations should return 0.

But there might be small errors in real world. So, we try to minimize those errors.

$$\min \sum (I_{xi} u + I_{yi} v + I_{ti})^2$$

→ Least Squares fit

$$\Rightarrow \boxed{\begin{aligned} &\sum I_{xi}^2 u + \sum I_{xi} I_{yi} v = -\sum I_{xi} I_{ti} \\ &\sum I_{xi} I_{yi} u + \sum I_{yi}^2 v = -\sum I_{yi} I_{ti} \end{aligned}}$$



$$\Rightarrow u = \frac{-\sum I_{yi}^2 \sum I_{xi} I_{ti} + \sum I_{xi} I_{yi} \sum I_{yi} I_{ti}}{\sum I_{xi}^2 \sum I_{yi}^2 - (\sum I_{xi} I_{yi})^2}$$

$$v = \frac{\sum I_{xi} I_{ti} \sum I_{xi} I_{yi} - \sum I_{xi}^2 \sum I_{yi} I_{ti}}{\sum I_{xi}^2 \sum I_{yi}^2 - (\sum I_{xi} I_{yi})^2}$$

Question 4

Image Segmentation is a way to transform an image into segments which could be easily analysed and processed for further evaluation. Image segmentation helps in assigning labels to each pixel in the image. Pixels with similar characteristics are assigned a similar label.
The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image.
Image segmentation is used in a variety of fields including object detection, medical imaging, traffic control systems, information retrieval from images etc.
There are few different ways to segment images into cluster:

1. **K-means Clustering**: K-means clustering is an iterative method to cluster an image into K different segments. Initially K random pixels are taken as cluster centers the the following process is continued until convergence.
    a. Assign each pixel in the image to one cluster center which that pixel is closest to.
    b. Take the mean of each cluster. The mean of each cluster are the new cluster centers.
    c. Repeat this process until convergence. You can also put a threshold on it, such that is the cluster changes with value less than the threshold, the process end.

2. **Thresholding**: Thresholding is a process in which we turn a grayscale image into a binary image. One of the most popular ways of thresholding is Otsu's Threshold.
    Otsu Threshold:
    1. We create a histogram of the gray scale image.
    2. We divide the grayscale image into 2 groups. Group 1 being black and group 2 being white(0-255).
    3. We consider each integer from 0-255 and divide the histogram into two groups at that point and calculate the in class variance of the groups and add them.
    4. At whichever point the in class variance sum is minimum, we consider that as our threshold.

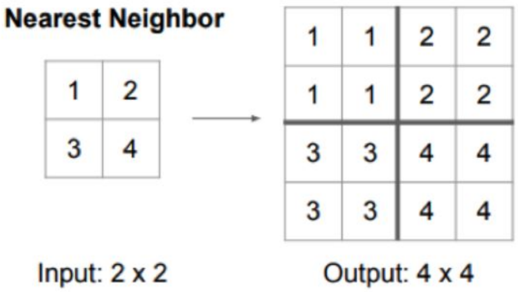
Question 5

There are various upsampling strategies used in convolutional neural networks to accomplish semantic segmentation:

1. **Nearest Neighbour**: In nearest neighbour upsampling in CNN, we multiply the number of rows and columns (size) of the grid by the stride we consider,

and for each pixel in the input downsampled grid, we duplicate the pixel value in its neighbourhood.

The following image depicts the nearest neighbour upsampling.

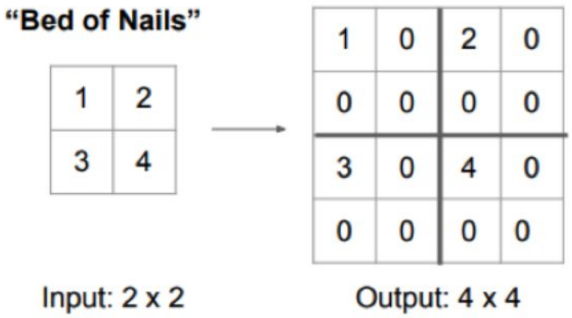


The input is a downsampled 2x2 grid which when upsampled with a stride of 2 gives the above output.

2. **Bed Of Nails upsampling**: It is similar to nearest neighbour upsampling strategy, the only difference being,instead of duplicating the value in the nearest neighbours, we assign all neighbours as 0 and the top left corner as the value of the pixel.

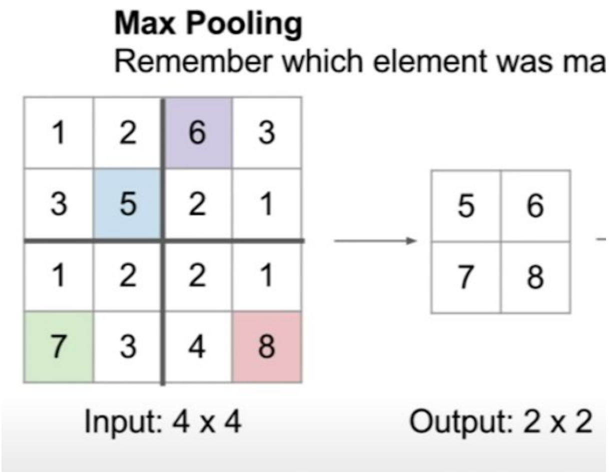
   For example, if we consider a stride of value **a** , then of each pixel in the downsampled grid, we create and **a x a** grid with its top left value as the value of the pixel and all other values as 0, thus creating an upsampled grid.

   The following image depicts bed of nails upsampling:



   The above example shows an input of 2x2 grid with a stride of 2. When trying to upsample the input grid with bed of nails, it gives a 4x4 grid as the output.

3. **Max Unpooling**: In the initial layers of the CNN, we downsample using the max pooling. In max pooling, we consider a stride of **a** and for each block of a x a , we consider the location of the maximum pixel value in the grid.



Max Pooling

Remember which element was ma

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

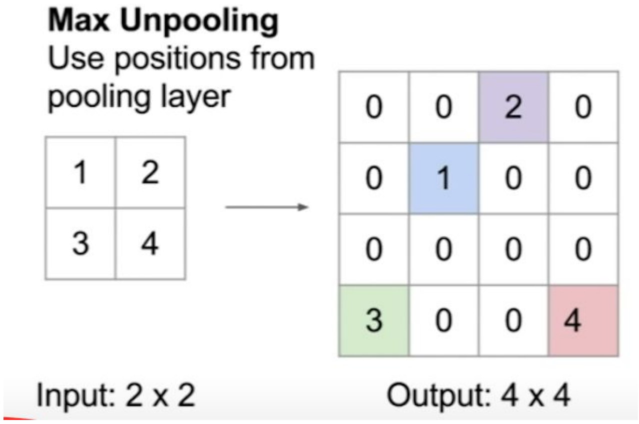
→

5	6
7	8

Input: 4 x 4

Output: 2 x 2

For each corresponding layer at the end CNN, in which we upsample, we consider the location stored for each a x a grid in the corresponding downsample and put the value of the pixels at that particular location and fill all other values in the sub-grid as 0. The following image depicts the max unpooling to upsample the grid.



Max Unpooling

Use positions from pooling layer

1	2
3	4

→

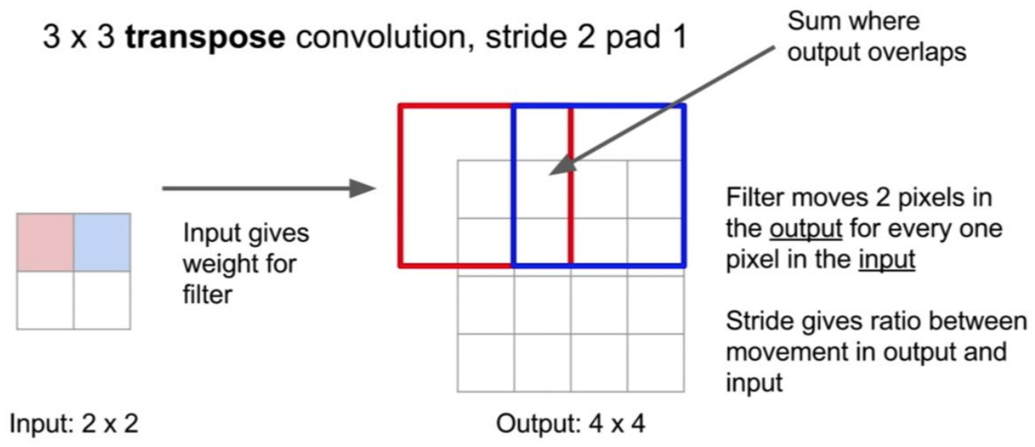
0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

Input: 2 x 2

Output: 4 x 4

4. **Transpose Upsampling**: Unlike other upsampling techniques, transpose upsampling is a learnable upsampling strategy. For a downsampled input, we consider a stride and a 3 x 3 convolution and for each pixel in the input grid, we multiply it with the layer filter creating the pixel values for the upsampled image. The input grid acts as a weight for the layer filter.
   Wherever the

   The following is an example of transpose upsampling with a stride of 2.



3 x 3 **transpose** convolution, stride 2 pad 1

Sum where output overlaps

Input gives weight for filter

Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

Input: 2 x 2

Output: 4 x 4

The above example shows the upsampling with stride of 2 and a 3 x 3 convolution.

Wherever the 2 different output maps overlap, we sum the values in those pixels while upsampling.