L3 notes

Some terms

d(h * f) h convolve with f
Know what the math equations are doing and apply them
Pad the sides with 0 and shift the kernel across the numbers
floor(kernel size / 2)

$[1, 1, 1]$ symmetric
Linear (shift invariant) = output Is scaled accordingly

Correlation filter to shift pixels in image

Normalised = summed up add to 1
Effects are negligible for large images

Gaussian kernel depends more on the sigma

Salt and pepper = isolated pixels that are problematic
Smooth away the impulse and spread it all around
Impulse is salt without the pepper

Higher signal amplify both the signals and noise

Histogram stretching adjust contrast

Bitmap save the pixels independently

Laplacian

the process to get the residual is [upsample (fill with 0s) $\rightarrow$ blur $\rightarrow$ subtract from original]
(the reason for the blur is to "spread out" the values over the filled-in black pixels)

Q: Does spatial quantization mean the quantisation of values of x and y? i.e. x and y can only take discrete integer values?
A: spatial quantization basically happens because we cant represent continuous x and y coordinates in a discrete system

Q: It seems that you always regard intensity as integers, is it a routine for this course?
A: Yes and no

Q: does adjusting abstract and brightness simply refer to the linear mapping or also include the non-linear gamma mapping
A: Non-linear mapping also just the contrast
Photography operation we dont use stretching (we use images that stretch over the entire range)

Q: In practice, is convolution used more than correlation?
A: Convolution

Q: Is reconstructing the original image using laplacian pyramid 100% lossless?
A: Up to the difference of quantisation

Q: Can we compensate motion blur by finding the motion blur kernel and use the inverse kernel to demotion blur?
A: Yes, finding the kernel is always hard

Q: Is convolution invertible? In a sense that let's say we have a convolved image, is there always a filter such that we can recover the original image using convolution? e,g, undoing blur convolution using a specific sharpening convolution
A: If we throw away half of the pixels

Q: Does laplacian pyramids take 2* space than gaussian pyramids simply because laplacian pyramids have 2 sets of the pyramids (gaussian + residual)?
A: Same amount of space

L4 notes

Some terms

mean filter does not remove all noise and blurs the image only isolated values 0-255

median filter remove all noise and blurs the image slightly arrange in increasing starting from 0

difference filter sum up all the values in row same value for the entire row

each pixel has a tangent plane

kernel size controls the strength of the filter

border problem output is reduced in size for the pixels along the edge

the image patch and the template always contain positive numbers, cos $\Theta \in [0, 1]$, i.e., the output of normalized cross-correlation is normalized to the interval [0,1], where 0 means no similarity and 1 means a complete similarity.

region of interest can benefit template matching

image has discrete representation we need approximation, positive gradient value when the image change from dark to bright and negative when reversed image

Image sharpening g(x,y)=f(x,y)-f(x,y) $\circ$ h(x,y)

Magnitude = $\sqrt{(gx2 + gy2)}$

Approximated magnitude = |gx | + |gy |
non maxim suppression if edge has a magnitude too small connected to another pixel above a threshold, dont prune

extract edges to detect start and end edges , useful for 3d to know the shape and geometry
why? resilient and lighting and color useful for recognition

Template matching

This object is now the template (kernel) and by correlating an image with this template, the output image indicates where the object is. Each pixel in the output image now holds a value, which states the similarity between the template and an image patch (with the same size as the template) centered at this particular pixel position. The brighter a value, the higher the similarity.

Purely correlation since strongest response is when original image exactly matches output
what happens when u zoom the baby's face 2x (refer to image)?

Template is not symmetrical

Neighborhood processing

Neighbor pixels play a role when determining the output value of a pixel

Convolution vs Correlation

Convolution (rotated 180 degrees)
h(i, j) · f (x - i, y - j)

Correlation
h(i, j) · f (x + i, y + j)

To check if same result, the kernel before and after 180 degrees are same

Intuition of Sobel filter
gx (x, y) f (x + 1, y) f (x - 1, y)
correlate [1, 0, 1] with the image
$[-1, 0, 1]$
$[-2, 0, 2] = 2[-1, 0, 1]$ (put more weights at center)
$[-1, 0, -1]$

Combine the both result using the horizontal and vertical Sorbel to get

the final edge image
3x3 kernel is used as we need to include the neighbours as single row or single column kernel is sensitive to noise

Derivatives

does not matter the order of the derivative and gaussian blur
gaussian filter to remove noise before laplacian applied

vertical shows pixel of image row
horizontal shows the gradient value

f(x) grey level value
f'(x) gradient value
$[1, -2, 1]$
f"(x) gradient of the gradient

gxx (x, y) ≈ f (x - 1, y) 2 $\cdot f(x, y) + f(x + 1, y)$
$gyy(x, y)$≈ f (x, y - 1) 2 $\cdot f(x, y) + f(x, y + 1)$

First order derivative

Sobel is a combination of derivative kernel
Sobel results in wide edges as it is first order
First order derivative (thicker edges)
Roberts, Prewitt, Sobel filter

Second order derivative

Second order derivative can know the exact edge and detect 1 px thin edge
Smoothing is needed
DoG (difference in gaussian)
laplacian is the second order derivative (most sensitive to noise)
-1 white 0 grey 1 black

To approximate the 2nd order derivatives
gxx(x, y) $\approx$f (x1, y)2f (x, y)+f (x+1, y) represents [1;-2;1]
gyy(x, y) $\approx$f (x, y1)2f (x, y)+f (x, y+1) represents [1 -2 1]

Hysterisis

pixels below some value 0
hysterisis join the strong and weak pixels
thresholding depends on the image

cv2.canny is interpolated version
random forest is a learned model from human markups

gradient shift dark to light and light to dark

is the center a local maximum or not if yes keep it as part of the edge else discard it

Q: would there be multiple local maxes after performing the non-maximum suppression?
A: no multiple local maxes

Q: is horizontal gradient detection always from left to right? similar for vertical case.
A: left to right

Q: Could you explain the intuition behind why cross correlation isn't commutative/associative, even tho convolution is just flipping the kernel?
A: [1, 2, 3] cross correlation [3, 2, 1] is not the same as [3, 2, 1] cross correlation [1, 2, 3]

$[1, 2, 3]$ cross convolution [0, 1, 0] is [3, 2, 1] so it is not identity

Q: From slide 9, how is the 1D derivative filter constructed from the finite differences discrete version equation with h = 2?
A: 2 elements we are using

Q: When convolution/cross-correlation is mentioned, is full padding assumed by default?
A: is not full padding, lecture examples are only full padding

Q: is padding to just inserting zero rows and zero columns or that plus blurring/interpolation? It seems to have been used both wayshaving to do nn interpolation (bilinear, linearly solve between 2 neighbours)0 interspersed in rows and columns
blurring to reduce noise (to avoid enhancing the noise)

Q: I think it was mentioned last lecture that convolutions are generally invertible. How do we reconcile this with the notion that blurring is lossy?
A: convolution is multiplicity in the inverse domain (do division can have problems)

blurring in itself is not lossy only downsampling
for every single elements, composed on several pixels, mixture of weights, same kernel applied to next kernel blurring + downsampling makes it lossy

factor all of this in the system of equations should be able to recover the image

Q: why not take the residual of the original versus upsampled of previous image
A: remove some values these are the values we want to preserve in the residual else will keep some of the detailing

L6 notes

Some terms

bandwidth is too narrow not very representative
we don't know what is the correct/ incorrect value for bandwidth
sum of all the gaussian forms the new curve
dirac delta - bandwidth is very small
number of textons can be smaller/larger than the filter banks
euclidean distance between histogram ends up compensating for each other
What are textures used for?
cue to tell us on the underlying 3d structure
scale of the envelope, single or many wavelengths

running across all the images where each image is a pixel
run the cluster where every single pixel and image is a datapoint replace all pixels in the email with the texton id, compute the histogram based on the samples, each of the texton image is represented with a histogram
texton id is per pixel
x and y then the texton would not be purely texture

texture is independent feature to the segment it belongs to, we don't have to do segmentation to find the segment in the first place
the responses is for the whole filter bank
Do k-means twice
first k means to form texton dictionary
second k means NN search (can also apply mean-shift)

histogram is 100 dimension vector also
Feature Representation
Filter bank Response
Texton histogram

Note: Texton histogram as a feature for segmentation will not give a clear segmentation, problematic edges, results in a pixelated image for the texture (not localised)
Good localization, pixel from the 1 pixel or all over is the same
CNN is aggregation of many filters, take a feature response and apply another kernel to it

blurring each channel independently (2d blurring), now we are blurring the filter response
we have consider the location of the pixel for segmentation
region is bounded by two separate histograms, then u would have separate results
Q: Are we able to somehow use a filter to extract coloraturas as a filter inside the filter bank? So that we can run clustering algo on both texture and colour at the same time?
A: you can mix the colors if you want, create a gabor filter for each channel

Q: can I clarify whether textons are generated based on clustering done over all filter results within a filterbank? Or is it done using only some sort of 'most differentiating' filter result?
A: filter bank only to apply to all images
apply 2d convolution
Q: how does using the same filter bank ensure the bins are the same
A: having very different feature response
1 image per type then maybe
Q: Why 3 clusters?
A: there is nothing to distinguish the skin from the background

Quiz 1

Red and green considered equally, but blue is not considered at all
Maximum shades of grey will not be the shade

L7 notes

Some terms

SSD
keypoints are also corners or interest points
low error will look dark, high error will look bright
shape is going to affect the ellipse, size does not matters
H = [0 0 ; 0 C] gradient is x direction is 0
H = [A 0 ; 0 0] gradient is y direction is 0

greedy approach non-maximum suppression has keypoints approximately

in the same location as it looks for areas that has high contrast
automatic scale selection, harris is equivariant to changes of scale

Weights of derivative

gaussian before we do the summation for the weights, more weight at the center, less weights at the edge

Instead of applying the effects, we are looking at two viewpoints under two different lighting conditions

highest response has a signal has characteristic scale that has the same as gaussian, width of signal corresponds in signal charasteristic

Harris operator is more efficient compared to eigen value decomposition
If the region is unusual then it is difficult for matching, approximated distinctiveness with SSD error
Linearize with 2nd moment matrix
Quantify distinctiveness with eigenvalues of H
R is the cornerness is defined as
R = det(H) - k $trace^2$(H)

Values can be obtained from the matrix itself

LoG finds blobs (keypoint detector that tries to find roughly circular regions) maximum scale is already built-in (maximum size of the kernel wrt to image)

Auto correlation: Create a template of the patch, shift it around where I use the local patch
Q: Is template the benchmark for comparison? A: strongest peak where it matches to itself Refer to template matching

No matter where u shift the template, will get a strong response at the sky
Third dimension is the intensity
We need 3d How do we analyse the points without looking at 3d

λ_{min} and λ_{max} are perpendicular to each other

SSD error E(u, v)
Error will be 0 or greater
H is always positive definite
Semi definite where there is 0 case, flat region only in synthetic images
Cross section is approximately circular (a lot of change in all directions)
Direction of the fastest change perpendicular to the slowest change will be approximately equal
Which direction I change in it is going to be different
Contrast causes the threshold
λ_{min} would be small as $(\lambda_{min})^{-\frac{1}{2}}$ where there is no change on the straight line

Q: Clustering the responses of the textons
A: The response are the data points, each feature response is 1 dimension, each data point cluster in 38 dimensions
Dimensionality: Refer to L6 supplements
One blue dot is 1 pixel, x location is 1 feature value and y location is 1 feature value
Eigenvalue decomposition of pixel then u would get every pixel also
axis length is very long

Q: Texture and template matching are they the same?
A: Correlate the long, what is the pattern vs the individual image. Detecting many texture in 1 image, so the feature response will be different in different part of images. Only a particular filter is tuned to that signature response. Normally we don't have to hand-create the filters. Deep learning to learn whole bunch of features, or gabor filter, can tune the filters. Over many dimensions can get a unique response to that example.

Q: Filter bank to find the size
A: Large enough filter bank or some way to count

Binary classification if the sheet is defect or not

Q: Why is the skin and background cannot be segmented?
A: Texture of the background is very large, approximate scale of the marble is on par with the arm. Doing agnostic segmentation, we don't model the content in the image, only based on filter response. There is no reason why

it should end up in certain segmentations.
Clustering is done over all of the pixels over all of the images. Creating the local histogram is trial and error

L8 notes

Some terms

SIFT - scaling iterative feature response
Pixel wise difference (gradients) - more sensitive to relative location but sensitive to deformations
Color histogram - 3d histogram
What size of the cells do we want, fine grids, overlap etc Bin sizes - more finely tuned, a small shift will affect the histogram
Compare histogram - chi square
Orientation normalization

Averaging

Bimodal distribution average end up with dominant orientation with no gradient

MOPS or GIST is no longer used as a descriptor

Mode - SIFT

SIFT - understand the implication of doing this
Can be combined with other keypoint detector
as the threshold is larger, it removes more keypoints, depends on illumination difference

poorly localised if it sits on an edge and not a corner
corners have high curvature, edges have low curvature

needs to be clear where the keypoint is matched to when moving in a straight line

ratio of eigenvalues to be below the edge thresh high threshold more tolerant of edges
smaller bar to be corner, longer bar is edge
white on black and black on white difference
black bar cannot be keypoints as we also have the surrounding neighbors

We don't assign gradients in original orientation - dominant orientation
Taking the original histogram with some wrapping operation, form of normalisation wrt dominant orientation
Normalize to unit length
If it exceeds the threshold set the threshold value, rectify and assign it. Then renormalize.
Clamping is to encourage invariance to small changes to illumination, over count in certain orientations.
can use in the dense form where it can use in any rotation
High robust

Feature matching

Image transformation

Filtering
Only changes on the actual image intensities
G(x) = h { F(x) Warping
G(x) = F { h(x) changes the location of the pixels

2d projective transformation
affine: parallel lines are still preserved projective: also known as a homography, try to solve for the parameters for a projective transformation
Not an arbitrary matrix with 9 degrees of freedom (3x3)
Solve for H using direct linear transform

SVD

$A = U \sum V^T$
V is a singular matrix
singular values found on diagonal values of sigma

columns of V are single vectors
N number of correspondenses
Normalization such that the centroid is (0, 0) of the similarity transformation is a whitening step
RANSAC deal with outliers, sample from the set of noisy observations
Even though we can sample more, we usually s to be equal to the number of the outliers