

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

$\text{floor}(\text{kernel size} / 2)$

$[1, 1, 1]$ 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 = $\sqrt{gx^2 + gy^2}$

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)

$$h(i, j) \cdot f(x - i, y - j)$$

Correlation

$$h(i, j) \cdot f(x + i, y + j)$$

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

Intuition of Sobel filter

$$g_x(x, y) = f(x + 1, y) - f(x - 1, y)$$

correlate $[1, 0, 1]$ with the image

$$[-1, 0, 1]$$

$$[-2, 0, 2] = 2[-1, 0, 1] \text{ (put more weights at center)}$$

$$[-1, 0, -1]$$

Combine the both result using the horizontal and vertical Sobel 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

$$g_{xx}(x, y) \approx f(x-1, y) - 2 \cdot f(x, y) + f(x+1, y)$$

$$g_{yy}(x, y) \approx f(x, y-1) - 2 \cdot f(x, y) + f(x, y+1)$$

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

$g_{xx}(x, y) \approx f(x-1, y) - 2f(x, y) + f(x+1, y)$ represents $\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$

$g_{yy}(x, y) \approx f(x, y-1) - 2f(x, y) + f(x, y+1)$ represents $\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$

Hysteresis

pixels below some value 0

hysteresis 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: $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$ cross correlation $\begin{bmatrix} 3 & 2 & 1 \end{bmatrix}$ is not the same as $\begin{bmatrix} 3 & 2 & 1 \end{bmatrix}$ cross correlation $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$

$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$ cross convolution $\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$ is $\begin{bmatrix} 3 & 2 & 1 \end{bmatrix}$ 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 ways having to do with interpolation (bilinear, linearly solve between 2 neighbours) or 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 (division can have problems)

blurring in itself is not lossy only downsampling

for every single element, composed of 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
texton histogram as a feature for segmentation will not give a clear segmen-
tation, problematic edges, results in a pixelated image for the texture
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

L7 notes

Some terms

SSD

low error will look dark, high error will look bright

shape is going to affect the ellipse, size does not matters

$H = \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix}$ gradient is x direction is 0

$H = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}$ 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

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 characteristic

Harris operator is more efficient compared to eigen value decomposition

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)