Computer Vision

Machine Vision A: Computer Vision D: Adds vision and perception to machines

Pattern Recognition A: Computer Vision

SLR A: Computer Vision D: Single Lens Reflexive

Signal A: Computer Vision D: Conveys information or energy

Digital Image A: Signal D: 2D limited in spatial domain

Signal Types A: Signal

Digital A: Signal Types D: Discrete

Analog A: Signal Types D: Continuous

Colour Channel A: Digital Image T: BGR in Open CV T: Operations are done by channel

Image Transformations A: Digital Image

Geometric Transformations A: Signal Trans D: Matrix Multiplication, Linear Transformation

Affine Transformations A: Geometric Trans T: Scaling, Transformation, Rotation, Shearing T: Can be local or global T: $\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Interpolation A: Affine Trans E: Fill up with grey level images, like when the black-white image is rotate

K Nearest Neighbours Interpolation A: Interpolation T: Consider neighbours T: Find average of pixels, can be weighted average

Pixel Transformations A: Signal Transformations D: Element wise operations

Grayscale Transformation A: Signal Transformations D: Reduce the dynamic range of your image T: $(\frac{x - min(a)}{max(a) - min(a)}(newmax - newmin)) + newmin$

Brightness A: Digital Image D: Average greyscale

Contrast A: Grayscale Trans D: Dynamic Range of greyscale

Histogram A: Image Trans D: Breakdown of brightnesses of a gray channel

Histogram Equalisation A: Histogram D: Make peaks same height T: Affects shape of image

Histogram Color Reduction A: Histogram D: Simplify image to

Template Matching A: Histogram

Histogram Stretching Algorithm A: Histogram D: Make the dynamic range the full range

Convolution A: Image Trans D: If h is shift-invariant, then the T can be obtained from convolving x and impulse response h T: $M = floor(\frac{N + 2P - F}{S}) + 1$

Filter A: Convolution D: Also known as Impulse Response

Shift Invariant A: Convolution D: Same as Linear Transformation

Lowpass Filter A: Filter T: Filters out the high frequency E: Serves as averaging E: Easier to implement than a highpass filter

Highpass Filter A: Filter T: Filters out the low frequency T: Typically contains negative values T: If $sum = 0$, then results are dark (grey level decreases)

T: If $sum = 1$, then results are same brightness T: Noise is amplified E: Serves as difference E: Can identify lines and edges

Bandpass Filter A: FIlter T: Filters out frequencies outside a range

Gaussian Filter A: Filter T: Go for averaging, but the nearer pixels have higher weight

Padding A: Convolution E: Used to maintain the same dimension T: $P = \frac{F-1}{2}$

Stride A: Convolution T: Moves the filter with larger steps T: Can miss some pixels

Feature A: Convolution D: Feature is just the image itself

Convolution Types A: Convolution

Standard Convolution A: Convolution Types D: Result of convolution is larger T: $M = N + 2p$

Same Convolution A: Convolution D: Result of convolution is same size T: $M = N$

Valid Convolution A: Convolution D: Only the central part is retained T: $M - 2p$

Derivative A: Convolution

Derivative Types A: Derivative

First Derivative A: Derivative Types T: $\frac{\partial f}{\partial x} = f(x+1) - f(x)$ E: If there is an edge or line, there will be a large change

Second Derivative A: Derivative Types T: $\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$ E: No need to take absolute value E: It is possible to take the first derivative, then remove small values, then apply derivative again

Laplacian A: Derivative Types T: $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$ T: $[f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)] \Rightarrow \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

Sobel Filter A: Highpass E: Find the direction of the edge by finding the strength of the vertical and horizontal derivatives separately T: $G_x = a \star h_{sx}$ T: $G_y = a \star h_{sy}$ T: If $|G| > threshold$, then consider it T: $tan^{-1}(\frac{G_y}{G_x})$

Vertical Edge Detector A: Sobel Filter D: h_{sy}

Horizontal Edge Detector A: Sobel Filter D: h_{sx}

Unsharp Masking A: Lowpass Filter E: Perform low pass filtering, then subtract this from the original image, to get a fake highpass