

Matlab CVIP Toolbox Functions

The following functions are the *.m files in the Matlab CVIP Toolbox, organized into categories based on the type of operation. This list contains a brief description of the functions. For more details, including examples and syntax for each function, see the CVIP Toolbox Help pages.

Arithmetic and Logic

add_cvip - add two images, or add a constant to an image

and_cvip - perform a logical AND on two images

divide_cvip - divide one image by another, or divide an image by a constant

multiply_cvip - multiply two images, or multiply an image by a constant

not_cvip - perform a logical NOT on an image

or_cvip - perform a logical OR on two images

subtract_cvip - subtract one image from another, or subtract a constant from an image

xor_cvip - perform a logical XOR on two images

Band

assemble_bands_cvip - assembles 3-band image from 3 single-band images

extract_band_cvip - extracts one band from a 3-band image

Batch Processing

batch_cvip - script to allow for processing of multiple with a sequence of operations

batch_double_image_cvip - script to allow for processing of multiple with a sequence of operations, including two input image functions

Color

ccenhance_cvip - color contrast enhancement algorithm

cct2rgb_cvip - converts a color image from cylindrical coordinate to RGB color space

hsl2rgb_cvip - converts a color image from hsl color space back to RGB color space

hsv2rgb_cvip - converts a color image from hsv color space back to RGB color space

ipct_cvip - performs the inverse principal components transform in RGB-space

lab2rgb_cvip - converts a color image from CIE $L^*a^*b^*$ to RGB color space

luminance_cvip - performs color to luminance transform

lum_average_cvip - performs color to monochrome transform using average of all bands

luv2rgb_cvip - converts a color image from CIE $L^*u^*v^*$ to RGB color space

pct_cvip - performs the principal components transform in RGB-space

rgb2cct_cvip - converts a color image from RGB to cylindrical coordinate space

rgb2hsl_cvip – converts a color image from RGB to Hue/Saturation/Lightness color space
rgb2hsv_cvip – converts a color image from RGB to Hue/Saturation/Value color space
rgb2lab_cvip – converts a color image from RGB to CIE L*a*b* color space
rgb2luv_cvip – converts a color image from RGB to CIE L*u*v* space
rgb2sct_cvip – converts a color image from RGB to spherical coordinate space
rgb2xyz_cvip – converts a color image from RGB to CIE XYZ color space
sct2rgb_cvip – converts a color image from spherical coordinate space to RGB color space

Compression

bRLCdeco_cvip – bitplane run length coding (RLC) decompression
bRLCenco_cvip – bitplane RLC compression
btcdco_cvip – block truncation coding decompression, standard implementation
btcdco2_cvip – block truncation coding decompression 2, using CHAR (fast) implementation
btcdcol_cvip – block truncation coding decompression, column based implementation
btcdvec_cvip – block truncation coding decompression, vectorized implementation
btccco_cvip – block truncation coding compression, standard implementation
btccco2_cvip – block truncation coding compression 2, using CHAR (fast) implementation
btcccol_cvip – block truncation coding compression, column based implementation
btccvec_cvip – block truncation coding compression, vectorized implementation
kmeans_cvip – k-means algorithm, specify number of clusters and iterations
kmeans_plus_cvip – k-means++ algorithm, specify number of clusters and iterations
mBTCdeco_cvip – multilevel block truncation coding decompression
mBTCenco_cvip – multilevel block truncation coding compression
vipmread2_cvip – read vipm files, including compressed formats
vipmwrite2_cvip – write vipm files, including compressed formats

Conversion of Image Files

bin2graycode_cvip - converts natural binary code to gray code
data_type_cvip – data type conversion
graycode2bin_cvip – converts gray code to natural binary
halftone_cvip – convert image to binary with halftone technique
vipmread_cvip – read a CVIPtools vipm image file into Matlab
vipmread2_cvip – read a CVIPtools vipm image file into Matlab, including compressed files
vipmwrite_cvip – write a CVIPtools vipm image file to disk
vipmwrite2_cvip – write a CVIPtools vipm image file to disk, including compressed files

Edge/Line Detection

boiecox_ed_cvip – perform a Boie-Cox edge detection on the image
canny_ed_cvip – perform a Canny edge detection on the image
cerchar_ed_cvip - a spatial- multi spectral image edge detection filter
edge_link_cvip - links edge points into lines
frei_chen_ed_cvip – Frei-Chen filter, 9 basis image masks, edge/line subspace
harris_corner_cvip - a spatial (corner detection) filter
hough_inverse_cvip - performs an inverse Hough transform
hough_transform_cvip - performs a Hough transform
kirsch_ed_cvip - perform Kirsch edge detection
laplacian_ed_cvip – performs a Laplacian edge detection
laplacian_gauss_ed_cvip – computes the Laplacian of a Gaussian edge detector
marr_hildreth_ed_cvip – performs a Marr Hildreth edge detection on the image
moravec_corner_cvip - a spatial corner detecting filter
pratt_merit_cvip – calculates the Pratt Figure-of-Merit (FOM) for two binary images
prewitt_ed_cvip - performs Prewitt edge detection
pyramid_ed_cvip – perform a pyramid edge detection
roberts_ed_cvip – performs a Roberts edge detection
robinson_ed_cvip – perform a Robinson edge detection
shen_castan_ed_cvip – performs a Shen-Castan edge detection on the image
sobel_ed_cvip - performs Sobel edge detection

Geometry

checkboard_cvip – creates a checkerboard image
create_image_cvip – creates various geometric shapes
copy_paste_cvip - copies a subimage from one image and pastes it to another image
crop_cvip - crop a subimage from an image
enlarge_cvip - enlarges image to a user-specified size
fast_doublesize_cvip – enlarges the image to twice its original via a fast algorithm
rotate_cvip - rotate the given image by an angle specified by the user (Range 1 ~ 360 degrees)
shrink_cvip - shrinks the given image by a factor specified by the user (Range 0.1 ~ 1)
spatial_quant_cvip - quantize an image by one of the following methods: average, median, or decimation
translate_cvip - move the entire image horizontally and/or vertically; also used for cut-and-paste of subimage
zoom_cvip - zoom the given image by a factor specified by the user (Range 1 ~ 10)

Histogram

clahe_cvip – contrast limited adaptive histogram equalization

clahe_cvip_block – contrast limited adaptive histogram equalization, block processing

get_hist_image_cvip - generates a histogram Image from an image

gray_linear_cvip - graylevel linear modification

histeq_cvip - histogram equalization

hist_create_cvip – creates histogram as an array

hist_spec_cvip - perform histogram manipulation using formula specified in by character string(s) for the equation(s)

hist_shrink_cvip – shrink image data to specified range

hist_slide_cvip - histogram slide

hist_stretch_cvip - histogram stretch, specify range and percent clip both ends

local_histeq_cvip - local histogram equalization

unsharp_cvip - performs unsharp masking algorithm

Mapping

condremap_cvip - performs conditional remapping, user specified range and data type

data_type_cvip – data type conversion

logremap_cvip – performs a log remapping, as in Equation 5.2-25

relative_remap_cvip - remaps an image maintaining the relative range of each data band

remap_cvip - map image data into a specified range

Morphological

morphclose_cvip - perform grayscale morphological closing

morphdilate_cvip - perform grayscale morphological dilation

morpherode_cvip - perform grayscale morphological erosion

morphitermod_cvip – performs morphological iterative based modification as defined by a set of surrounds, a logical operation and the number of iterations

morphopen_cvip - perform grayscale morphological opening

morph_hitmiss_cvip – perform hitmiss operation

morph_thinning_cvip - performs thinning operation

morph_skeleton_cvip - perform skeletonization operation

Noise

gamma_noise_cvip - add gamma noise to an image

gaussian_noise_cvip - add gaussian noise

neg_exp_noise_cvip - add negative-exponential noise

rayleigh_noise_cvip - add Rayleigh noise

salt_pepper_noise_cvip - add speckle (salt-and-pepper) noise

uniform_noise_cvip - add uniform noise

Objective Fidelity Metrics

dice_cvip – calculates dice coefficient between two images

jaccard_cvip – calculates Jaccard index between two images

overlap_cvip – calculates overlap coefficients between two images

peak_snr_cvip – computes the peak signal-to-noise ratio between two images

rms_error_cvip – calculates the root-mean-squared-error between two images

snr_cvip – calculates the signal-to-noise ratio between two images

Pattern Classification: Algorithm Testing

leave_one_out_cvip – performs leave one out testing on a set of images

train_test_cvip – performs pattern classification testing with separate training and test sets of images

Pattern Classification: Classification Algorithms

k_nearest_neighbor_cvip – reads test and training files of feature vectors and creates output file for classification using k-nearest neighbor classification method

nearest_centroid_cvip - reads test and training files of feature vectors and creates output file for classification using nearest centroid classification method

nearest_centroid_set_cvip - reads test and training files of feature vectors and calculates centroids for each class

nearest_neighbor_cvip - reads test and training files of feature vectors and creates output file for classification using nearest neighbor classification method

Pattern Classification: Distance/Similarity Metrics

city_block_cvip – calculates the city block distance between two feature vectors

euclidean_distance_cvip – calculates the Euclidean distance between two feature vectors

maximum_value_metric_cvip – calculates the maximum value distance between two feature vectors

minkowski_distance_cvip – calculates the Minkowski distance between two feature vectors

pattern_city_block_cvip – takes two csv files as input, test and training set, then calculates the city block distance metric by comparing each vector in the test set to each in the training set

pattern_euclidean_cvip – takes two csv files as input, test and training set, then calculates the Euclidean distance metric by comparing each vector in the test set to each in the training set

pattern_maximum_cvip – takes two csv files as input, test and training set, then calculates the maximum value distance metric by comparing each vector in the test set to each in the training set

pattern_minkowski_cvip – takes two csv files as input, test and training set, then calculates the minkowski distance metric by comparing each vector in the test set to each in the training set

pattern_tanimoto_cvip – takes two csv files as input, test and training set, then calculates the Tanimoto similarity metric by comparing each vector in the test set to each in the training set

pattern_vector_inner_product_cvip – takes two csv files as input, test and training set, then calculates the normalized vector inner product metric by comparing each vector in the test set to each in the training set

tanimoto_metric_cvip – calculates the Tanimoto similarity metric between two feature vectors
vector_inner_product_cvip – calculates the *normalized* vector inner product similarity metric between two feature vectors

Pattern Classification: Feature Extraction

area_cvip – calculates the area in pixels of a binary object
aspect_cvip – finds the aspect ratio of a binary object
central_moments_cvip – returns the central moment of the order p and q for an object on the labeled image
centroid_cvip – finds the centroid of a binary object
cooccurrence_cvip – calculates the co-occurrence matrix for an object
euler_cvip – finds the Euler number of a binary object
feature_objects_cvip – extracts features from a group of objects in a single image and a single image mask for the objects, output is a csv file
feature_images_cvip – extracts features from a group of images, where each image contains one object of interest, and uses corresponding mask images, output is a csv file
hist_feature_cvip – calculates the 5 first order histogram features for an object
irregular_cvip – calculates the irregularity ratio of a binary object
label_cvip - labels objects based on 6-connectivity, NW/SE diagonal
orientation_cvip – calculates the axis of least second moment for a binary object
perimeter_cvip – calculates perimeter of a binary object
projection_cvip – extracts horizontal and vertical projections of a binary object
rst_invariant_cvip – calculates the 7 RST-invariant features defined in Table 6.1
texture_features_cvip – extracts 2nd-order histogram based texture features
thinness_cvip – calculate the thinness ratio of an object on the labeled image
spectral_features_cvip – extracts spectral features based on Fourier transform and ring and sector power

Pattern Classification: Normalization Methods

min_max_norm_cvip – applies min-max normalization to set of feature vectors in a matrix
pattern_min_max_norm_cvip – two csv files are input, training and test sets, and returns new csv files with the feature vectors normalized with min-max normalization
pattern_range_norm_cvip - two csv files are input, training and test sets, and returns new csv files with the feature vectors normalized with range normalization
pattern_snd_norm_cvip – two csv files are input, training and test sets, and returns new csv files with the feature vectors normalized with standard normal density (snd) normalization
pattern_softmax_norm_cvip – two csv files are input, training and test sets, and returns new csv files with the feature vectors normalized with softmax scaling normalization
pattern_unit_vector_norm_cvip – two csv files are input, training and test sets, and returns new csv files with the feature vectors normalized with unit vector normalization
range_norm_cvip – normalizes a set of feature vector values in a matrix based on the range of each feature

softmax_cvip – applies softmax normalization to set of feature vectors in a matrix
snd_norm_cvip – applies standard normal density normalization to a set of feature vectors in a matrix
unit_vector_norm_cvip – applies unit vector normalization to set of feature vectors in a matrix

Segmentation

autothreshold_cvip – perform auto single thresholding segmentation
dice_cvip – calculates dice coefficient between two images
fuzzyc_cvip – perform Fuzzy c-Means color segmentation
gray_quant_cvip – perform gray level quantization
hist_thresh_cvip – perform adaptive thresholding segmentation
igs_cvip – perform improved gray scale (IGS) quantization
jaccard_cvip – calculates Jaccard index between two images
kmeans_cvip – k-means algorithm, specify number of clusters and iterations
kmeans_plus_cvip – k-means++ algorithm, specify number of clusters and iterations
median_cut_cvip – perform median cut segmentation
otsu_cvip – perform Otsu thresholding segmentation
overlap_cvip – calculates overlap coefficients between two images
pct_median_cvip – perform PCT/median cut segmentation
sct_split_cvip – perform SCT/Center Split segmentation
split_merge_cvip – perform a split and merge segmentation
threshold_cvip – perform binary threshold on an image

Spatial Filters: Adaptive Filters

ace2_filter_cvip – adaptive contrast and enhancement filter
adaptive_contrast_cvip – adaptive contrast filter, adapts to local gray level statistics
adapt_median_filter_cvip – a ranked-order based adaptive median filter
ad_filter_cvip – anisotropic diffusion filter
clahe_cvip – contrast limited adaptive histogram equalization
clahe_cvip_block – contrast limited adaptive histogram equalization, block processing
exp_ace_filter_cvip – performs an exponential ACE filter
improved_mmse_filter_cvip – perform improved adaptive minimum mean squared error filter
kuwahara_filter_cvip – an edge preserving, smoothing filter
log_ace_filter_cvip – performs a log ACE filter operation
mmse_filter_cvip – minimum mean squared error restoration filter

Spatial Filters: Mean Filters

arithmetic_mean_cvip – perform an arithmetic mean averaging filter
contra_mean_cvip – perform a contra-harmonic mean filter
geometric_mean_cvip – performs a geometric mean filter
harmonic_mean_cvip – performs a harmonic mean filter
h_image_cvip – create a mask image according to the size and type
raster_deblur_mean_cvip – raster deblurring filter
yp_mean_cvip – performs a Yp mean filter

Spatial Filters: Miscellaneous

convolve_filter_cvip – convolves an image with a user specified convolution mask

difference_filter_cvip – performs a difference/emboss filter

h_image_cvip - create a mask image according to the size and type

Spatial Filters: Order Filters

alpha_filter_cvip - perform an alpha-trimmed mean filter

maximum_filter_cvip - performs a maximum filter

median_filter_cvip - performs a fast histogram-method median filter

midpoint_filter_cvip - performs a midpoint filter

minimum_filter_cvip - performs a minimum filter on an image

Transform

dct_cvip - perform block-wise discrete cosine transform

fft_cvip - performs fast Fourier transform

fft_mag_cvip - extract magnitude of Fourier spectrum

fft_phase_cvip – extract phase of Fourier spectrum

haar_cvip - perform forward Haar transform

idct_cvip – perform inverse discrete cosine transform

ifft_cvip – performs inverse Fourier transform

ihaar_cvip - perform inverse Haar transform

iwahad_cvip – perform inverse Walsh/Hadamard transform

iwavdaub4_cvip - perform inverse wavelet transform based on Daubechies wavelet

iwavhaar_cvip - perform inverse wavelet transform based on Haar wavelet

wavdaub4_cvip - perform forward wavelet transform based on Daubechies wavelet

wavhaar_cvip - perform forward wavelet transform based on Haar wavelet

walhad_cvip – perform forward Walsh/Hadamard transform

Transform Filters: Standard

butterworth_bandpass_cvip - perform Butterworth bandpass filter
butterworth_bandreject_cvip - perform Butterworth bandreject filter
butterworth_low_cvip - perform Butterworth lowpass filter
butterworth_high_cvip - perform Butterworth highpass filter
butterworth_h_cvip – create Butterworth mask image
h_image_cvip - create a mask image according to the size and type
highfreqemphasis_cvip - perform high frequency emphasis filter
homomorphic_cvip - performs homomorphic filtering on an input
ideal_bandpass_cvip - perform ideal bandpass filter
ideal_bandreject_cvip - perform ideal bandreject filter
ideal_low_cvip - perform ideal lowpass filter
ideal_h_cvip – create ideal mask image
ideal_high_cvip - perform ideal highpass filter

Transform Filters: Restoration

geometric_mean_xformfilter_cvip – performs the geometric mean restoration filter
inverse_xformfilter_cvip – performs the inverse restoration frequency domain filter
least_squares_filter_cvip - performs the least squares restoration filter
notch_filter_cvip – performs the notch frequency domain filter
parametric_wiener_filter_cvip - performs the parameter Wiener restoration filter
power_spect_eq_filter_cvip – power spectrum equalization frequency domain restoration filter
simple_wiener_filter_cvip - performs the simple Wiener restoration filter
wiener_filter_cvip – performs the Wiener restoration filter