

## Common Object Module (COM) Functions – cviptools.dll

The following functions are in the dynamically linked library, *cviptools.dll*. They are all based on the common object module interface (COM), and can be used directly by those familiar with this interface. The COM functions are essentially the CVIPtools C functions with a COM wrapper on top. These functions are in the *CVIPTools.cpp* file in the *\$CVIPtoolsHOME\CVIPCOM* directory.

Note that the CVIPtools installation program performs system registration for *cviptools.dll*, which is required to run CVIPtools.exe. If any changes are made to *cviptools.dll*, the user needs to perform system registration. This can be done in a DOS shell with this command: *regsvr32 cviptools.dll*.

**Ace2\_Filter**([in] long\* image,[in] long wsize,[in]double alpha, [in] double beta,[out,retval] long\* Result\_Image);  
**Acuity\_Nightvision\_Filter**([in] long\* image,[in] long reason, [in] long threshold, [in] long choice,[out,retval] long\* Result\_Image);  
**Adapt\_Median\_Filter**([in] long\* image,[in] long mask\_size,[out,retval] long\* Result\_Image);  
**AD\_Filter**([in] long\* image,[in] long iteration\_num,[in] long speed,[in] long edge\_height,[out,retval] long\* Result\_Image);  
**Adaptive\_Contrast\_Filter**([in] long\* image,[in] double k1, [in] double k2,[in] long kernel\_size,[in] double min\_gain, [in] double max\_gain,[out,retval] long\* Result\_Image);  
**Add\_Image**([in] long\* input\_im1,[in] long\* input\_im2, [out,retval] long\* result\_im);  
**Alpha\_Filter**([in] long\* image,[in] long mask\_size, [in] long p,[out,retval] long\* Result\_Image);  
**And\_Image**([in] long\* input\_im1,[in] long\* input\_im2, [out,retval] long\* result\_im);  
**Area**([in] long\* image,[in] int r, [in] int c, [out,retval] long\* Result);  
**Aspect**([in] long\* image,[in] int r, [in] int c, [out,retval] long\* Result);  
**Assemble\_Bands**([in] long\* image,[out,retval] long\* Result\_Image);  
**Bilinear\_Interp**([in] long\* image, [in] float factor,[out,retval] long\* Result\_Image);  
**Auto\_Threshold\_Segment**([in] long\* inputImage, [in] float limit, [out,retval] long\* Result\_Image);  
**Bintocvip**([ in ] BSTR \*File\_Name, [in] long data\_bands, [in] long color\_order, [in] long interleaved, [in] long height, [in] int width, [in] long verbose, [out, retval] long\* Result\_Image);  
**Bit\_Compress**([ in ] BSTR \*File\_Name, [in] long\* Input\_image, [in]long sect);  
**Bit-Decompress**([ in ] BSTR \*File\_Name, [out,retval] long\* Result\_Image);

**Boiecox\_Filter**([in] long\* image,[in] double var,[in] long do\_threshold, [in] long do\_hyst, [in] long thin, [in] double high\_threshold, [in] double low\_threshold, [out] long \*imageThreshold, [out] long \*imageHyst, [out,retval] long\* Result\_Image);  
**Btc\_Compress**([ in ] BSTR \*File\_Name, [in] long\* Input\_image, [in]long blocksize);  
**Btc-Decompression**([ in ] BSTR \*File\_Name, [out,retval] long\* Result\_Image);  
**Btc2\_Compress**([ in ] BSTR \*File\_Name, [in] long\* Input\_image, [in]long blocksize);  
**Btc2-Decompress**([ in ] BSTR \*File\_Name, [out,retval] long\* Result\_Image);  
**Btc3\_Compress**([ in ] BSTR \*File\_Name, [in] long\* Input\_image, [in]long blocksize);  
**Btc3-Decompress**([ in ] BSTR \*File\_Name, [out,retval] long\* Result\_Image);  
**Butterworth\_Band\_Pass**([in] long\* image,[in] long block\_size, [in] long dc, [in] long inner, [in] long outer, [in] long order,[out,retval] long\* Result\_Image);  
**Butterworth\_Band\_Reject**([in] long\* image,[in] long block\_size, [in] long dc, [in] long inner, [in] long outer, [in] long order,[out,retval] long\* Result\_Image);  
**Butterworth\_High**([in] long\* image,[in] long block\_size, [in] long dc, [in] long cutoff, [in] long order,[out,retval] long\* Result\_Image);  
**Butterworth\_Low**([in] long\* image,[in] long block\_size, [in] long dc, [in] long cutoff, [in] long order,[out,retval] long\* Result\_Image);  
**Canny\_Filter**([in] long \*image, [in] double low, [in] double high, [in] double var, [in] long \*nonmax\_mag, [in] long \*nonmax\_dir, [out,retval] long \*Result\_Image);  
**Cast\_Image**([in] long\* image,[in] long dtype);  
**CentroID**([in] long\* input\_im1, [in] long r, [in] long c ,[out,retval] VARIANT \*result\_array);  
**Cerchar\_Filter**([in] long \*inputImage, [out,retval] long \*Result\_image);  
**Check\_Bin**([in] long\* image,[out,retval] long\* Result);  
**Check\_xform\_history**([in] long\* image,[in] int filter, [out,retval] long\* Result);  
**CLAHE**([in] long\* image,[in] int size, [in] double clippercent [in] int mb,[out,retval] long\* Result\_Image);  
**CLAHE**([in] long\* image,[in] int size, [in] double clippercent [in] int mb,[out,retval] long\* Result\_Image);  
**Dice\_Coef**([in] long\* input\_im1,[in] long\* input\_im2, [out,retval] VARIANT\* error\_values);  
**Close\_Console**();  
**Close\_Console**();  
**ColorXform**([in] long\* image,[in]long newcspace, [in]double \*norm, [in]double\*refwhite,[in] long dir, [out,retval] long\* Result\_Image);  
**CondRemap\_Image**([in] long \* image, long dtype, int min, int max,[out,retval] long \*Result\_Image);  
**Contra\_Filter**([in] long\* image,[in] long mask\_size, [in] long p,[out,retval] long\* Result\_Image);  
**Copy\_Paste**([in] long\* src\_image, [in] long\* dest\_image, [in] int start\_r, [in] int start\_c, [in] int height, [in] int width, [in] int dest\_r, [in] int dest\_c, [in] long transparent,[out,retval] long\* Result\_Image);

**Create\_Black**([in] int width, [in] int height,[out,retval] long\* Result\_Image);  
**Create\_Checkboard**([in] int width, [in]int height,[in]int firstx,[in]int firsty,[in]int blockx,[in]int blocky,[out,retval] long\* Result\_Image);  
**Create\_Circle**([in] int width, [in] int height,[in]int centerx,[in]int centery,[in]int radius,[out,retval] long\* Result\_Image);  
**Create\_Cosine**([in] int img\_size, [in] int frequency, [in] int choice,[out,retval] long\* Result\_Image);  
**Create\_Degenerate\_Circle**([in] int width, [in] int height,[in]int centerx,[in]int centery,[in]int radius1,[in]int radius2,[out,retval] long\* Result\_Image);  
**Create\_Ellipse**([in] int width, [in] int height,[in]int centerx,[in]int centery,[in]int hor\_length,[in]int ver\_length,[out,retval] long\* Result\_Image);  
**Create\_Line**([in] int width, [in] int height, [in ] int tlx, [in] int tly, [in] int brx, [in ] int bry,[out,retval] long\* Result\_Image);  
**Create\_Rectangle**([in ] int width, [in] int height, [in] int tlx, [in] int tly, [in] int sqwidth, [in] int sqheight,[out,retval] long\* Result\_Image);  
**Create\_Sine**([in] int img\_size, [in] int frequency, [in] int choice,[out,retval] long\* Result\_Image);  
**Create\_Squarewave**([in] int img\_size, [in] int frequency, [in] int choice,[out,retval] long\* Result\_Image);  
**Crop**([in] long\* image, [in] int row\_offset, [in] int col\_offset, [in] int rows, [in] int cols,[out,retval] long\* Result\_Image);  
**CVIP\_OUTPUT**([out, retval] BSTR \* sMessage);  
**CVIPhalfitone**([in] long\* image, [ in ] int halftone, [ in ] int maxval, [ in ] float fthreshval, [ in ] long retain\_image, [ in ] long verbose,[out,retval] long\* Result\_Image);  
**Cvptoccc**([ in ] BSTR \*File\_Name, [in] long maxcolor,[in] long verbose, [in] long dermvis,[in] long\* cvip\_Image);  
**Cvptoepts**([ in ] BSTR \*File\_Name,[in] long\* cvip\_Image, [in] double scale\_x, [in] double scale\_y, [in] long band, [in] long verbose);  
**Cvptogif**([ in ] BSTR \*File\_Name,[in] long\* cvip\_Image, [in] long interlace, [in] long verbose);  
**Cvptoiris**([ in ] BSTR \*File\_Name,[in] long\* cvip\_Image,[in] long verbose,[in] long prt\_type);  
**Cvptoitex**([ in ] BSTR \*File\_Name,[in] long\* cvip\_Image, [ in ] BSTR \*comment,[in] long verbose);  
**Cvptojpg**([ in ] BSTR \*File\_Name, [in] long\* Input\_image,[in] int quality, [in] long grayscale, [in] long optimize, [in] int smooth, [in] long verbose, [in]BSTR \*qtablesFile);  
**Cvptojp2**([ in ] BSTR \*File\_Name, [in] int rate, [in] int quality, [in] int umberofoylayers, [in] int block\_size);  
**Cvptopnm**([ in ] BSTR \*File\_Name,[in] long\* cvip\_Image, [in] long verbose);  
**Cvptopng**([ in] long\* Input\_image, [ in] BSTR \*File\_Name);  
**Cvptoras**([ in ] BSTR \*File\_Name,[in] long\* cvip\_Image, [in] long pr\_type, [in] long verbose);

**Cvptotiff**([ in ] BSTR \*File\_Name,[in] long\* cvip\_Image, [in] long compression, [in] long fillorder, [in] long g3options, [in] long predictor, [in] long rowsperstrip, [in] long verbose);  
**Cvptovip**([ in ] BSTR \*File\_Name,[in] long\* cvip\_Image, [in] long verbose,[in] long save\_history,[in] long is\_compressed);  
**Date\_Range**([in] long\* input\_im1, [out,retval] VARIANT \*result\_array);  
**Dct\_Transform**([in] long\* image,[in] long block\_size,[out,retval] long\* Result\_Image);  
**Delete\_Image**([in] long\* image);  
**Divide\_Image**([in] long\* input\_im1,[in] long\* input\_im2,[ in ]long zero2num, [out,retval] long\* result\_im);  
**Dpc\_Compress**([ in ] BSTR \*File\_Name, [in] long\* Input\_image, [in]float ratio,[in]int bit\_length, [in] int clipping, [in]int direction, [in]int origin, [in]float deviation);  
**Dpc-Decompress**([ in ] BSTR \*File\_Name, [out,retval] long\* Result\_Image);  
**Draw\_mesh**([in] long\* image,[in] long\* pmesh);  
**Duplicate\_Image**([in] long\* image,[out,retval] long\* Result\_Image);  
**DynRLC\_Compression**([in] BSTR\* File\_Name, [in] long \*inputImage, [in] long WindowSize, [out, retval] long \* result);  
**DynRLC\_deCompression**([in] BSTR\* File\_Name, [out, retval] long \*Result\_Image);  
**Edge\_Detect\_Filter**([in] long\* image, [in] long program, [in] long mask\_choice, [in] long mask\_size, [in] long keep\_dc, [in] long threshold,[in] long threshold1, [in] long thresh,[in] long thr,[out,retval] long\* Result\_Image);  
**Edge\_Link\_Filter**([in] long\* image,[in] long connection,[out,retval] long\* Result\_Image);  
**Enlarge**([in] long\* image, [in] int row, [in] int col,[out,retval] long\* Result\_Image);  
**Epstocvip**([ in ] BSTR \*File\_Name,[in] long verbose);  
**Euler**([in] long\* image,[in] long row, [in] long col, [out,retval] long \*Result\_Image);  
**Exp\_Ace\_Filter**([in] long\* image,[in] long wsize,[in]double alpha, [in] double beta,[out,retval] long\* Result\_Image);  
**Extract\_Band**([in] long\* image,[in] int bandno, [out,retval] long\* Result\_Image);  
**Fft\_Phase**([in] long\* image,[in] long remap\_norm, [in] double k,[out,retval] long\* Result\_Image);  
**Fft\_Transform**([in] long\* image,[in] long block\_size,[out,retval] long\* Result\_Image);  
**File\_To\_Mesh**([ in ] BSTR \*File\_Name, [out, retval] long\* Result\_Image);  
**Fractal\_Compression**([in] long \*inputImage, [in] BSTR \*File\_Name, [in] double Tolerate, [in] long min\_part, [in] long max\_part, [in] long dom\_type, [in] long dom\_step, [in] long c1, [in] long c2, [in] long s\_bits, [in] long o\_bits, [out, retval] long \* result);  
**Fractal\_deCompression**([in] BSTR \*File\_Name, [out, retval] long \* Result\_Image);  
**Fuzzyc\_Segment**([in] long\* image,[in] double variance, [out,retval] long\* Result\_Image);  
**Gamma\_Noise**([in] long\* image,[in] double var, [in] int alpha,[out,retval] long\* Result\_Image);  
**Gaussian\_Noise**([in] long\* image,[in] double var, [in] double mean,[out,retval] long\* Result\_Image);  
**Geometric\_Filter**([in] long\* image,[in] long mask\_size,[out,retval] long\* Result\_Image);

**Geometric\_Mean**([in] long\* image,[in] long\* image1,[in] long\* image2,[in] long\* image3,[in] double gamma,[in] double alpha,[in] long choice,[in] long cutoff,[out,retval] long\* Result\_Image);  
**get\_hist\_real**([in] long\* image,[out,retval] long\* Result\_Image);  
**Get\_Histogram\_Image**([in] long\* image,[out,retval] long\* Result\_Image);  
**Get\_max\_min\_value**([in] long\* image,[in] double\* Result);  
**GetDataFormat\_Image**([in] long\* image,[out,retval] long\* Result);  
**GetDataType\_Image**([in] long\* image,[out,retval] long\* Result);  
**GetImageInfo**([in] long\* orig\_im, [out,retval] VARIANT \*result\_array);  
**Getlast\_Hist**([in] long\* input\_im, [in] long \* program,[in] int nprogs, [out, retval] long\* Result);  
**GetNoOfBands\_Image**([in] long\* image,[out,retval] long\* Result);  
**GetNoOfCols\_Image**([in] long\* image,[out,retval] long\* Result);  
**GetNoOfRows\_Image**([in] long\* image,[out,retval] long\* Result);  
**giftocvip**([ in ] BSTR \*File\_Name,[in] long\* cvip\_Image, [in] long imageNumber,[in] long showmessage);  
**Glr\_Compress**([ in ] BSTR \*File\_Name, [in] long\* Input\_image, [in]long win);  
**Glr-Decompress**([ in ] BSTR \*File\_Name, [out,retval] long\* Result\_Image);  
**Gray\_Binary**([in] long\* image,[in] int direction,[out,retval] long\* Result\_Image);  
**Gray\_Linear**([in] long\* image, [in] double start, [in] double end,[in] double s\_gray, [in] double slope, [in] int change, [in] int band,[out,retval] long\* Result\_Image);  
**Gray\_Multiplication**([in] long\* image,[in] float ratio,[in] long options, [out,retval] long\* Result\_Image);  
**Gray\_Multiply**([in] long\* image,[in] float ratio,[out,retval] long\* Result\_Image);  
**Gray\_Multiply2**([in] long\* image,[in] float ratio,[out,retval] long\* Result\_Image);  
**Gray\_Quant\_Segment**([in] long\* image,[in] long num\_bits,[out,retval] long\* Result\_Image);  
**Graylevel\_Quant**([in] long\* image,[in] long num\_bits, [in] long choice, [out,retval] long\* Result\_Image);  
**Graylevel\_Remap**([in] long\* inputImage,[in] long bandR, [in] long bandG, [in] long bandB, [out,retval] long \*Result\_Image);  
**GVFSnake\_Segment**([in] long\* image, [in] long num\_iter, [in] long alpha, [in] long beta, [in] long kappa, [out,retval] long\* Result\_Image);  
**H\_image**([in ] long type, [in ] long height, [in ] long width,[out,retval] long\* Result\_Image);  
**Haar\_Transform**([in] long\* image,[in] long param1,[in] long param2, [out,retval] long\* Result\_Image);  
**Harmonic\_Filter**([in] long\* image,[in] long mask\_size,[out,retval] long\* Result\_Image);  
**Harris\_Corner\_Filter**([in] long \*inputImage, [in] float alpha, [in] int threshold, [in] int border, [in] float dmin, [out,retval] long \*Result\_Image);  
**High\_Freq\_Emphasis**([in] long\* image,[in] long block\_size, [in] long dc, [in] long Cutoff, [in] double alfa, [in] long order,[out,retval] long\* Result\_Image);  
**Highboost**([in] long \*inputImage, [in] long MaskSize, [in] long CenterValue, [in] long

ifAdd2Origin, [out, retval] long \* Result\_Image);  
**HighFreq\_Emphasis**([in] long \* inputImage, [in] long TransformMethod, [in] long CutoffFreq, [in] long FilterOrder, [in] double OffSet, [in] long KeepDC, [out, retval] long \* Result\_Image);  
**Highpass\_Spatial**([in] long \*inputImage, [in] long ifAdd2Origin, [out, retval] long \* Result\_Image);  
**Hist\_Feature**([in] long\* orig\_im, [in] long\* labeled\_im, [in] long r, [in] long c, [out,retval] VARIANT \*result\_array);  
**Hist\_Slide**([in] long\* image, [in] int slide, [out,retval] long\* Result\_Image);  
**Hist\_Spec**([in] long \* InputImage, [in] BSTR \*bandR, [in] BSTR \*bandG, [in] BSTR \*bandB, [out, retval] long\* Result\_Image);  
**Hist\_Stretch**([in] long\* image, [in] int low\_limit, [in] int high\_limit, [in] float low\_clip, [in] float high\_clip, [out,retval] long\* Result\_Image);  
**Hist\_Thresh\_Segment**([in] long\* image, [out,retval] long\* Result\_Image);  
**Histeq**([in] long\* image, [in] int mb, [out,retval] long\* Result\_Image);  
**Histogram\_Spec**([in] long \* InputImage, [in] double\* sped\_in, [out, retval] long\* Result\_Image);  
**History\_Add**([in] long\* input\_im, [in] long\* input\_history);  
**History\_Check**([in] long\* image, [in] int program, [out,retval] long\* Result);  
**History\_Copy**([in] long\* input\_from, [in] long\* input\_to);  
**history\_create**([in] long prog, [in] long type, [in] float value, [out, retval] long\* Result);  
**History\_Get**([in] long\* image, [in] int program, [out,retval] long\* Result);  
**History\_get\_data**([in] long\* history, [in] int program, [out,retval] long\* Result);  
**Homomorphic**([in] long\* image, [in] double upper, [in] double lower, [in] long cutoff, [in] long ifAdd2Origin, [out,retval] long\* Result\_Image);  
**Horizontal\_Flip**([in] long\* image, [out,retval] long\* Result\_Image);  
**Hough\_Filter**([in] long \*image, [in] BSTR \* name\_in, [in] BSTR \* degree\_string\_in, [in] long threshold, [in] long connection, [in] long interactive, [in] long delta\_length, [in] long segment\_length, [out, retval] long\* Result\_Image);  
**Huf\_Compress**([ in ] BSTR \*File\_Name, [in] long\* Input\_image);  
**Huf-Decompress**([ in ] BSTR \*File\_Name, [out,retval] long\* Result\_Image);  
**Idct\_Transform**([in] long\* image, [in] long block\_size, [out,retval] long\* Result\_Image);  
**Ideal\_Band\_Pass**([in] long\* image, [in] long block\_size, [in] long dc, [in] long inner, [in] long outer, [out,retval] long\* Result\_Image);  
**Ideal\_Band\_Reject**([in] long\* image, [in] long block\_size, [in] long dc, [in] long inner, [in] long outer, [out,retval] long\* Result\_Image);  
**Ideal\_High**([in] long\* image, [in] long block\_size, [in] long dc, [in] long cutoff, [out,retval] long\* Result\_Image);  
**Ideal\_Low**([in] long\* image, [in] long block\_size, [in] long dc, [in] long cutoff, [out,retval] long\* Result\_Image);  
**Ifft\_Transform**([in] long\* image, [in] long block\_size, [out,retval] long\* Result\_Image);  
**Igs\_Segment**([in] long\* image, [in] long gray\_level, [out,retval] long\* Result\_Image);



**Improved\_Mmse\_Filter**([in] Long\* inputImage, [in] float threshval, [in] int kernel\_size, [in] float noise\_var, [out,retval] Long\* Result\_Image);  
**Input\_Mesh**([in] long \* mesh\_array,[out, retval] long\* Result\_Image);  
**Intensity\_Slicing**([in] long \*inputImage,[in] long \*lookupTable, [in] long options, [out,retval] long \*Result\_Image);  
**Inverse\_Xformfilter**([in] long\* image,[in] long\* image1,[in] long choice,[in] double cutoff,[out,retval] long\* Result\_Image);  
**Ipct**([in] long\* image, [in] long is\_mask, [out,retval] long\* Result\_Image);  
**Irregular**([in] long\* image,[in] int r, [in] int c, [out,retval] long\* Result);  
**Jaccard\_Index**([in] long\* input\_im1,[in] long\* input\_im2, [out,retval] VARIANT\* error\_values);  
**Jpg\_Compress**([ in ] BSTR \*File\_Name, [in] long\* Input\_image,[in ] int quality, [in] long grayscale, [in] long optimize, [in] int smooth, [in] long verbose, [in]BSTR \*qtablesFile);  
**Jpg-Decompress**([ in ] BSTR \*File\_Name, int colors, [in] long blocksmooth, [in] long grayscale, [in] long nodither, [in] long verbose, [out,retval] long\* Result\_Image);  
**Jpgtovicp**([ in ] BSTR \*File\_Name, [ in ] int colors, [ in ] long blocksmooth, [ in ] long grayscale, [ in ] long nodither, [ in ] long verbose,[out, retval] long\* Result\_Image);  
**Jp2tovicp**([ in ] BSTR \*File\_Name, [out,retval] long\* Result\_Image);  
**Jp2\_Compress**[ in ] BSTR \*File\_Name, [ in ] long\* Input\_image, [in ] int rate, [in] int quality, [in] int numberoflayers, [in] int block\_size );  
**Jp2\_DeCompress**([ in ] BSTR \*File\_Name, [out,retval] long\* Result\_Image);  
**K\_Nearest\_Neighbor**([in] BSTR\* TestFile, [in] BSTR\* TrainingFile, [in] BSTR\* OutputFile, [in] int normalize, [in] int distance\_similarity, [in] int k, [in] float s\_min, [in]float s\_max, [in] float r\_softmax, [in] float r\_minkowski, [out,retval] long \*ErrorCode);  
**Kirsch\_Filter**([in] long\* image, [in] long\* dirImage, [in] long mask\_choice, [in] long mask\_size, [in] long keep\_dc, [in] long threshold,[out,retval] long\* Result\_Image);  
**Kuwahara\_Filter** ([in] long \*image, [in] long mask\_size, [out, retval] long \*Result\_Image);  
**Label**([in] long\* image,[out,retval] long\* Result\_Image);  
**Laws\_texture**([in] long\* input\_im1, [in] long\* input\_im2, [in] long r, [in] long c , [in] int window\_size, [in] int vector1,[in] int vector2, [out,retval] VARIANT \*result\_array);  
**Least\_Squares**([in] long\* image,[in] long\* image1,[in] long\* image2,[in] double gamma,[in] long choice,[in] long cutoff,[out,retval] long\* Result\_Image);  
**Local\_Histeq**([in] long\* image,[in] int size, [in] int mb,[out,retval] long\* Result\_Image);  
**Log\_Ace\_Filter**([in] long\* image,[in] long wsize,[in]double alpha, [in] double beta,[out,retval] long\* Result\_Image);  
**Log\_Remap**([in] long\* image,[in] long band, [out,retval] long\* Result\_Image);  
**LogMap\_Image**([in] long\* image, int band,[out,retval] long\* Result\_Image);  
**Lum\_Average**([in] long\* image,[out,retval] long\* Result\_Image);

**Luminance\_Image**([in] long\* image,[out,retval] long\* Result\_Image);  
**Make\_Histogram**([in]double\* sped\_in,[in]long image\_format,[in]long color\_format,[in] long bands,[out, retval] long\* Result\_Image);  
**Marr\_Hildreth\_Filter**([in] long\* image, [in] float sigma, [in] int threshold, [out,retval] long\* Result\_Image);  
**Maximum\_Filter**([in] long\* image,[in] long mask\_size,[out,retval] long\* Result\_Image);  
**Mean\_Filter**([in] long\* image,[in] long mask\_size,[out,retval] long\* Result\_Image);  
**Median\_Cut\_Segment**([in] long\* image,[in] long newcolors, [in] long is\_bg, [in] long r\_bg, [in] long g\_bg, [in] long b\_bg,[out,retval] long\* Result\_Image);  
**Median\_Filter**([in] long\* image,[in] long mask\_size,[out,retval] long\* Result\_Image);  
**Mesh\_To\_File**([ in ] BSTR \*File\_Name, [in] long\* pmesh);  
**Mesh\_Warping**([in] long\* image,[in] long\* pmesh, [in] long method,[out,retval] long\* Result\_Image);  
**Mesh\_WarpingRI**([in] long\* image,[in] long\* pmesh, [in] long method, [in] long zero\_out,[out,retval] long\* Result\_Image);  
**Midpoint\_Filter**([in] long\* image,[in] long mask\_size,[out,retval] long\* Result\_Image);  
**Minimum\_Filter**([in] long\* image,[in] long mask\_size,[out,retval] long\* Result\_Image);  
**Mmse\_Filter**([in] long\* image,[in] double noise\_var, [in] long kernel\_size,[out,retval] long\* Result\_Image);  
**Moravec\_Corner\_Filter**([in] long \*inputImage, [in] float threshval, [out,retval] long \*Result\_Image);  
**MorphClose**([in] long\* image,int k\_type, int ksize, int height,int width,[out,retval] long\* Result\_Image);  
**MorphDilate**([in] long\* image, int k\_type, int ksize, int height,int width,[out,retval] long\* Result\_Image);  
**MorphErode**([in] long\* image,int k\_type, int ksize, int height,int width,[out,retval] long\* Result\_Image);  
**Morph\_Hitmiss**([in] long\* inputImage, [in] long MaskSize, char\* HitmissFilter, [out, retval] long\* Result\_Image);  
**Morpho**([in] long\* input\_im1, [ in ] BSTR \*File\_Name, [in] long rotate,[in] long boolFUNC,[in] long connectedness, [in] long no\_of\_iter, [in] long fields, [out,retval] long\* Result\_Image);  
**Morpho\_com**([in] long\* image, [ in ] BSTR \*sur\_set, [in] long p1,[in] long p2, [in] long p3,[in] long p4, [in] long p5, [out,retval] long\* Result\_Image);  
**MorphOpen**([in] long\* image,int k\_type, int ksize, int height,int width,[out,retval] long\* Result\_Image);  
**Morph\_Skeleton**([in] Long\* inputImage, [in] Long FilterDimension, [in] char\* SkeletonMatrix, [in] Long Iterations,[in] Long four\_eight\_mask, [in] Long method, [out,retval] Long\* Result\_Image);  
**Morph\_Thinning**([in] Long\* inputImage,[in] Long MaskSize,[in] char\* ThinningFilter,



[out,retval] Long\* Result\_Image);

**Multi\_Resolution\_Segment**([in] long\* image,[in] long choice, [in] double param1, [in] double param2, [in] long run\_PCT, [out,retval] long\* Result\_Image);

**Multiply\_Image**([in] long\* input\_im1,[in] long\* input\_im2, [out,retval] long\* result\_im);

**Nearest\_Centroid**([in] BSTR\* TestFile, [in] BSTR\* TrainingFile, [in] BSTR\* OutputFile, [in] int normalize, [in] int distance\_similarity, [in] float s\_min, [in]float s\_max, [in] float r\_softmax, [in] float r\_minkowski, [out,retval] long \*ErrorCode);

**Nearest\_Neighbor**([in] BSTR\* TestFile, [in] BSTR\* TrainingFile, [in] BSTR\* OutputFile, [in] int normalize, [in] int distance\_similarity, [in] float s\_min, [in]float s\_max, [in] float r\_softmax, [in] float r\_minkowski, [out,retval] long \*ErrorCode)

**Neg\_Exp\_Noise**([in] long\* image,[in] double var, [out,retval] long\* Result\_Image);

**New\_Image**(long image\_format, long color\_space, int bands, int heigth, int width, long data\_type, long data\_format,[out,retval] long\* Result\_Image);

**Nonfft\_Xformfilter**([in] long\* image,[in] long block\_size, [in] long dc, [in] long filtertype, [in] long p1, [in] double p2, [in] long order,[out,retval] long\* Result\_Image);

**Not\_Image**([in] long\* image,[out,retval] long\* Result\_Image);

**Notch**([ in ] BSTR \*File\_Name, [in] long x, [in] long y, [in] long radius,[in] long\* image,[in] long number, [in] long interactive, [out,retval] long\* Result\_Image);

**Object\_Crop**([in] long\* image, [in] long \* coordinates ,[in] long format, [in] long R, [in] long G,[in] long B,[out, retval] long\* Result\_Image);

**Open\_Consol\_Redirect\_Output**();

**Or\_Image**([in] long\* input\_im1,[in] long\* input\_im2, [out,retval] long\* result\_im);

**Orientation**([in] long\* image,[in] int r, [in] int c, [out,retval] long\* Result);

**Otsu\_Segment** ([in] long \*image, [out, retval] long \*Result\_Image);

**Overlap\_Coef**([in] long\* input\_im1,[in] long\* input\_im2, [out,retval] VARIANT\* error\_values);

**Param\_AD\_Filter**([in] long\* image,[in] long iteration\_num,[in] long speed,[in] long edge\_height,[in] long blk\_size,[out,retval] long\* Result\_Image);

**Parametric\_Wiener**([in] long\* image,[in] long\* image1,[in] long\* image2,[in] long\* image3,[in] double gamma,[in] long choice,[in] long cutoff,[out,retval] long\* Result\_Image);

**Pct**([in] long\* image,[in] long is\_mask,[in] double \*maskP,[out,retval] long\* Result\_Image);

**Pct\_Color**([in] long\* image,[in] long is\_mask,[in] double \*maskP,[in] long choice, [out,retval] long\* Result\_Image);

**Pct\_Median\_Segment**([in] long\* image,[in] long colors,[out,retval] long\* Result\_Image);

**Perimeter**([in] long\* image,[in] int r, [in] int c, [out,retval] long\* Result);

**Power\_Spect\_Eq**([in] long\* image,[in] long\* image1,[in] long\* image2,[in] long\* image3,[in] long choice,[in] long cutoff,[out,retval] long\* Result\_Image);

**Pratt\_Merit**([in] long \*image1, [in] long \*image2, [in] double scale\_factor, [out,retval] double \*Result);

**Prewitt\_Filter**([in] long\* image, [in] long\* dirImage, [in] long mask\_choice, [in] long mask\_size, [in] long keep\_dc, [in] long threshold,[out,retval] long\* Result\_Image);

**Projection**([in] long\* input\_im1, [in] int r, [in] int c ,[in] int height, [in] int width,[out,retval] VARIANT \*result\_array);

**Pseudo\_Remap**([in] long\* image,[in] long \* lookupTable,[out,retval] long\* Result\_Image);

**Pseudocol\_Freq**([in] long\* image, [in] int inner, [in] int outer, [in] int blow,[in] int bband, [in] int bhigh, [out,retval] long\* Result\_Image);

**Raster\_Deblur\_Filter**([in] long\* image,[out,retval] long\* Result\_Image);

**Rayleigh\_Noise**([in] long\* image,[in] double var, [out,retval] long\* Result\_Image);

**Read\_Image**([ in ] BSTR \*File\_Name, [out, retval] long\* Result\_Image);

**Remap\_Image**([in] long\* image,long dtype, long dmin, long dmax,[out,retval] long\* Result\_Image);

**REMAP**([in] long\* image,long dtype, long dmin, long dmax,[out,retval] long\* Result\_Image);

**Rms\_Error**([in] long\* input\_im1,[in] long\* input\_im2, [out,retval] VARIANT \*result\_array);

**Robinson\_Filter**([in] long\* image, [in] long\* dirImage, [in] long mask\_choice, [in] long mask\_size, [in] long keep\_dc, [in] long threshold,[out,retval] long\* Result\_Image);

**Rotate**([in] long\* image,[in] float degrees,[out,retval] long\* Result\_Image);

**Rst\_invariant**([in] long\* input\_im1, [in] long r, [in] long c ,[out,retval] VARIANT \*result\_array);

**Save\_Compressed\_Data**([in] long\* input\_im, [in ] BSTR\*File\_Name);

**Sct\_Split\_Segment**([in] long\* image,[in] long A\_split, [in] long B\_split,[out,retval] long\* Result\_Image);

**Set\_Console**([in] long Handle);

**Sharpen\_I**([in] long \*inputImage, [in] long ifRemap, [in] long MaskChoice, [in] long MaskSize, [in] float LowClip, [in] float HighClip, [in] long ifAdd2Origin, [out, retval] long \* Result\_Image);

**Sharpen\_II**([in] long \*inputImage, [in] long ifAdd2Origin, [out, retval] long \* Result\_Image);

**Shen\_Castan\_Filter**([in] long \*Image,[in] double b, [in] long window\_size, [in] double low\_threshold, [in] double high\_threshold, [in] long thin\_factor, [in] long \*ZeroInterImage, [out,retval] long \*Result\_Image);

**Show\_Image**([ in ] int dc, [ in ] int x, [ in ]int y, [ in ] long\* image);

**Show\_Image\_Ex**([ in ] int dc,[in] long\* mem\_dc, [ in ] int x, [ in ]int y, [ in ] long\* image);

**Shrink**([in] long\* image,[in] float factor ,[out,retval] long\* Result\_Image);

**Simple\_Wiener**([in] long\* image,[in] long\* image1,[in] double k,[out,retval] long\* Result\_Image);

**Single\_Filter**([in] long\* image, [in] double sx, [in] double sy, [in] long xcen, [in] long ycen,[in] double rot, [in] double beta, [in] long N, [in] double \*h, [in] long choice, [out,retval] long\* Result\_Image);

**Smooth\_Filter**([in] long\* image,[in] long kernel,[out,retval] long\* Result\_Image);

**Snr**([in] long\* input\_im1,[in] long\* input\_im2, [out,retval] VARIANT \*result\_array);

**Sobel\_Filter**([in] long\* image, [in] long\* dirImage, [in] long mask\_choice, [in] long mask\_size, [in] long keep\_dc, [in] long threshold,[out,retval] long\* Result\_Image);

**Spatial\_Quant**([in] long\* image,[in] int row,[in] int col,[in] int method,[out,retval] long\* Result\_Image);

**Spec\_Hist\_Image**([in] long \*InputImage, [in] BSTR \* bandR, [in] BSTR \* bandG, [in] BSTR \* bandB, [out,retval] long \*Result\_Image);

**Specify\_Filter**([in] long\* image,[in] long mask\_height,[in] long mask\_width,[in] double \*maskP, [in] int normalization, [out,retval] long\* Result\_Image);

**Speckle\_Noise**([in] long\* image,[in] double psalt, [in] double ppepper,[out,retval] long\* Result\_Image);

**Spectral\_Feature**(long \*input\_im1, long \*input\_im2, [in] long no\_of\_bands, [in] long no\_of\_sectors,[in] long r,[in] long c,[out,retval] VARIANT \*result\_array);

**Split\_Merge\_Segment**([in] long\* image,[in] long level, [in] long choice,[in] double parameter0,[in] double parameter1,[in] long Run\_PCT, [out,retval] long\* Result\_Image);

**Subtract\_Image**([in] long\* input\_im1,[in] long\* input\_im2, [out,retval] long\* result\_im);

**TextureFeature**([in] long\* orig\_im, [in] long\* labeled\_im, [in] long r, [in] long c , [in] long distance , [in] long hex\_equiv ,[out,retval] VARIANT \*result\_array);

**TextureFeature2**([in] long\* inputImage, [in] long\* labeledImage, [in] long band, [in] long r, [in] long c, [in] long distance, [in] long energy, [in] long inertia, [in] long correlation, [in] long invDiff, [in] long entropy, [in] long zero\_rc, [out,retval] VARIANT\* result);

**Thinness**([in] long\* image,[in] int r, [in] int c, [out,retval] double\* Result);

**Threshold\_Segment**([in] long\* image,[in] long threshval, [in] long thresh\_inbyte, [out,retval] long\* Result\_Image);

**Tifftocvip**([ in ] BSTR \*File\_Name, [in ] int verbose, [out, retval] long\* Result\_Image);

**Tile\_by\_name**([ in ] BSTR \*File\_Name);

**Transform\_Compression**([in] long \*image, [in] BSTR \* filename, [in] long color\_space, [in] long xform, [in] long WaveletBasis, [in] long subimage\_size, [in] long quant, [in] long JPEG\_Q\_Table, [in] long coding, [in] long data\_type, [in] long remap\_type, [in] long KeepDC, [out, retval] long \*Result\_Image);

**Transform\_Sharpen**([in] long \* inputImage, [in] long TransformMethod, [in] long CutoffFreq, [in] long FilterOrder, [in] long Offset, [in] long KeepDC, [in] long Add2Origin, [out, retval] long \* Result\_Image);

**Transform\_Smoothing**([in] long \* inputImage, [in] long TransformMethod, [in] long CutoffFreq, [in] long FilterOrder, [in] long Offset, [in] long KeepDC, [out, retval] long \* Result\_Image);

**Translate**([in] long\* image, [in] long do\_wrap, [in] int y\_off, [in] int x\_off,[in] int y\_mount,[in] int x\_mount,[in] int y\_slide, [in] int x\_slide, [in] float fill\_out,[out,retval] long\* Result\_Image);

**Uniform\_Noise**([in] long\* image,[in] double var, [in] double mean,[out,retval] long\* Result\_Image);  
**Unsharp\_Filter**([in] long\* image, [in] long lower, [in] long upper,[in] double low\_clip, [in] double high\_clip,[out,retval] long\* Result\_Image);  
**Vq\_Compress**([in] long \*inputImage,[in] BSTR \*File\_Name,[in] long cdbook\_in\_file,[in] long fixed\_codebook, [in] double in\_error\_thres,[in] BSTR\* cdbook\_file,[in] long in\_no\_of\_entries,[in] long in\_row\_vector,[in] long in\_col\_vector, [out, retval] long \* result);  
**Vq-Decompress**([in] BSTR \*File\_Name, [out, retval] long \*Result\_Image);  
**Walhad\_Transform**([in] long\* image,[in] long param1,[in] long param2, [out,retval] long\* Result\_Image);  
**Watershed\_Segment**([in] long \* inputImage, [in] int choice, [in] float threshold, [out,retval] long \* Result\_Image);  
**Wavdaub4\_Transform**([in] long\* image,[in] long param1,[in] long param2, [out,retval] long\* Result\_Image);  
**Wavhaar\_Transform**([in] long\* image,[in] long param1,[in] long param2, [out,retval] long\* Result\_Image);  
**Wiener**([in] long\* image,[in] long\* image1,[in] long\* image2,[in] long\* image3,[in] long choice,[in] long cutoff,[out,retval] long\* Result\_Image);  
**Write\_Image**([in] long\* input\_im,[in] BSTR\*File\_Name, [in] long retain\_image, [in] long set\_up, [in]long new\_format, [in] long showmessages);  
**Xor\_Image**([in] long\* input\_im1,[in] long\* input\_im2, [out,retval] long\* result\_im);  
**Xvq\_Compress**([in] long\* image,[in] long xform, [in] long scheme, [in] BSTR \*filename, [in] long file\_type, [in] long remap\_type, [in] long dc,[out,retval] long\* Result\_Image);  
**Xvq-Decompress**([in] BSTR \*filename, [out, retval] long \*Result\_Image);  
**Ypmean\_Filter**([in] long\* image,[in] long mask\_size, [in] long p,[out,retval] long\* Result\_Image);  
**Zon\_Compress**([ in ] BSTR \*File\_Name, [in] long\* Input\_image, [in]int block\_size,[in]int choice, [in ]int mask\_type, [in]float compress\_ratio);  
**Zon-Decompress**([ in ] BSTR \*File\_Name, [out,retval] long\* Result\_Image);  
**Zon2\_Compress**([ in ] BSTR \*File\_Name, [in] long\* Input\_image, [in]int block\_size,[in]int choice, [in ]int mask\_type, [in]float compress\_ratio);  
**Zon2-Decompress**([ in ] BSTR \*File\_Name, [out,retval] long\* Result\_Image);  
**Zoom**([in] long\* image, [ in ] int quadrant, [ in ] int X, [ in ] int Y, [ in ] int dx, [ in ] int dy, [ in ] float temp\_factor,[out,retval] long\* Result\_Image);  
**Zvl\_Compress**([ in ] BSTR \*File\_Name, [in] long\* Input\_image);  
**Zvl-Decompress**([ in ] BSTR \*File\_Name, [out,retval] long\* Result\_Image);