**AIRBOTS - VSAAS**

This document specifies all functionalities to be included in VSAAS project - phase 1 and also some brief description about the functions and parameters to be passed and to be returned. Each and every functionality should be specified under different group.

**IMAGE\_UTILITY functionalities:**

The functionalities specified under this group will be used for doing operations in images or frames.

**command [options] image1 [options] image2 [options] output\_image**

1. **IMAGE TRANSFORMATION**

1.CROP

Input arguments:

image - Image from which the area has to be cropped

x1, y1 - Initial x and y of the point of the area to be cropped

w, h - Width and Height of the cropping area.

name - If the cropped image is to saved, then name should be the file to be saved

No Name - then it will return the image

Command line:

1. **convert -crop 640x620+0+0 <input file> <output file>**
2. **convert <input file> -gravity East -crop 400x400+0+0 <output file>**
3. **mogrify -chop 0x45+0+0 -gravity South \*.jpg**

* **convert/mogrify** is used to convert between image formats as well as resize an image, blur, crop, despeckle, dither, draw on, flip, join, re-sample, and much more.
* [Convert/mogrify command](https://www.imagemagick.org/script/convert.php)
* **-crop** is used to cut out one or more regions of the image.(“-crop” by default is relative to the top-left corner of the image, it can be relatively changed using **“-gravity”.** The -gravity command can take all cardinal directions including NorthWest, NorthEast, SouthWest, SouthEast and center.)
* **-chop** is used to remove the pixels size specified in the parameters

2.RESIZE

Input arguments:

* image - Image which has to be resized
* new\_width, new\_height - New Width and Height in which the image has to be resized
* name - If the resized image is to saved, then name should be the file to e saved
* No Name - then it will return the image

Command line:

1. **convert <input file> -resize 64x64 <output file>**
2. To resize an image down to a width of 1024 pixels:**mogrify -geometry 1024x <in/output file>**
3. To resize an image down to a height of 800 pixels:**mogrify -geometry x800 <in/output file>**
4. To resize an image such that the longest side (either width or height) is 800 pixels:**mogrify -geometry 800x800 <in/output file>**
5. To resize an image down by half: **mogrify -resize 50% <in/output file>**

* **“-resize”** will fit the image into the requested size.

3. TILE

Input arguments:

* **“-montage”** command is designed to produce an array of thumbnail images.
* **“-geometry”** setting defines the size of the individual thumbnail images, and the spacing between them.

Command line:

1. **montage <input file> <input file><input file> <input file> <outputfile>**
2. To tile a set of images:**montage <input file> <outputfile>**
3. To tile a set of 7 images, arranged in a single row: **montage <input file> -tile 7x1 <outputfile>**
4. To specify that only a spacing of 2 pixels be used between images in the above setting: **montage <input file> -tile 5x2 -geometry +2+2 <outputfile>**
5. To specify crop, zoom and resize then command: **convert <input file> -gravity center -crop 90x90+0+0 +repage -resize "120x90^" -gravity center -crop 120x90+0+0 +repage <outputfile>**

4. JOIN

Input arguments:

* **“append”** is used to join images side-by-side or one below another into a single row or column.

Command line:

1. To join images horizontally side-by-side into a row: **convert <input file> +append <outputfile>**
2. To join images vertically one below another into a column: **convert <input file> -append <outputfile>**

**II. SPECIAL EFFECTS:**

1.BLUR

Input arguments:

* **-channel** setting is used for the color channels where the four colors of the image were blurred, leaving the transparency or alpha channel of the image as is.

Command line:

1. **convert <input file> -channel RGBA -blur 0x5 <output file>**
2. **convert <input file> -channel RGBA -blur 0x5 -alpha off <output file>**

2. SHARPENING

Input arguments:

* **“-sharpen”** spreads the pixels out and will make the corners of the image thick.
* **“-unsharp”** makes the corners of the image unthick.
* Parameters RadiusxSigma+Amount+Threshold
* [Parameter Description](https://even.li/imagemagick-sharp-web-sized-photographs/)
* **“-compose”** is used for merging two or more images in a variety of ways
* **"-animate"** is used to generate aFlicker of two similar images/dissimilar images

Command line:

**1. convert -sharpen 1.2x1.2+5+0 <input file> <output file>**

**2. convert -unsharp 1.4x1.4+5+0 <input file> <output file>**

**3. composite -compose difference <input file> <input file> <output file>**

**4. convert <input file> <input file> -scale 400% miff:-|\animate -delay 50 -loop0-**

**5. display <output file>**

3.SHADOW

Input arguments:

* **“-shadow”** operator will take an image and convert it into a shadow image that can then be positioned under the original image at given offset.
* Parameters used are OpacityxSigma +x +y

Command line:

1. **convert <input file> \( +clone -background navy -shadow 80x3+5+5 \) +swap \ -background none -layers merge +repage <output file>**
2. For images which are already scaled and compressed: **convert <input file> -bordercolor white -border 13 \( +clone -background black -shadow 80x3+2+2 \) +swap -background white -layers merge +repage <output file>**
3. For creating thumbnails: **convert <input file> -thumbnail 200x200 -bordercolor white -border 6 \( +clone -background black -shadow 80x3+2+2 \) +swap -background white -layers merge +repage <output file>**
4. For raw images: **convert <input file> -scale 600x400 -quality 86 -strip -bordercolor white -border 13 \( +clone -background black -shadow 80x3+2+2 \) +swap -background white -layers merge +repage <output file>**

4.BRIGHTNESS and CONTRAST

Input arguments:

* **“-brightness-contrast”**adjustments on images is done to make the intensity of the image brighter or darker.
* Parameters are brightness x contrast
* + is used to increase the intensity value and - is used to decrease the intensity value.

Command line:

1. **convert -brightness-contrast 50x20 <input file> <output file>**

**III. File Conversion**

* To get a complete listing of which image formats that are supported on your system, type **identify -list format**
* To determine the colorspace of your image and to display the detailed description of the image, use this command:**identify -verbose <input file>**
* [**Pseudo formats**](https://www.imagemagick.org/script/formats.php#pseudo)
* [**Built\_in\_Images**](https://www.imagemagick.org/script/formats.php#builtin-images)
* To see what type and fonts are currently available to your version run the following commands:
  + **convert -list type**
  + **convert -list font**
* **convert <input file> <output file>**
* **convert <input file> -resize 120x120 <output file>**
* To convert a PDF with a single page into a JPG: **convert <pdf file> <jpg file>**
* insert copyright notice: **convert -fill red -draw 'text 20 20 "© 2017 example.com"' <input file> <output file>**

**IV.TRANSPARENCY**

1. GRAYSCALE and TRANSPARENCY \ HUE / SATURATION/ LUMINOSITY

* **Grayscale** converts an image to black and white

Command line:

1. **convert -type Grayscale <input file> <output file>**
2. create color image with transparent background: **convert <input file>-transparent white <output file>**
3. change to gray with transparent background: **convert <input file> -colorspace gray/-monochrome <output file>**
4. change to separate gray channels: **convert <input file> -separate <output file>**
5. **convert eye.gif storm.gif -negate +append cmd\_negate.gif**
6. **convert <input file> -colorspace HSL -channel Saturation -negate -evaluate multiply 0.8 -negate -colorspace RGB <output file>**

2. GAMMA CONVERSION

* For converting an image into a linear color colorspace, RGB or -gamma 1/2.2 = 0.454545 use the following commands:

Command line:

1. **convert <input file> -colorspace RGB <output file>**
2. **convert <input file> -gamma 0.454545 <output file>**

3. NOISE REDUCTION

Input arguments:

* - resize to 320\*240 (i would like fairly large and detailed thumbs)
* - apply noise filter
* - do a subtle unsharp mask
* - save to png and jpg

Command line:

1. **convert <input file> -resize 320x240+0+0 -noise 5 -median 5 -unsharp 5 -normalize -write <output file> <output file>**

4. VIBRANCE

* **Vibrance is excellent for making nonlinear color adjustments. It analyzes the color in an image, and rather than affecting all the colors in a uniform way, it treats different colors in different ways. For example, if you increase the Vibrance slider, the bright and highly saturated colors will remain relatively unmodified, whereas the less-saturated colors will become more colorful—in essence, brighter and more varied. If you decrease the Vibrance slider, the weaker colors fade away and only the most prominent colors remain.**

V. IMAGE ORIENTATION

1. ROTATE

Input argument:

image - Image to get rotated

angle - rotation angle

name - name to save the rotated video, else return the rotated image

Command line:

1. To rotate 90 degrees clockwise and write output as new file:

**convert -rotate "90" <input file> <output file>**

**2.** To rotate 45 degrees counter-clockwise and overwrite the input file:

**mogrify -rotate "-45" <in/output file>**

2. FLIP/FLOP

Input arguments:

* left/right flip
* up/down flip

Command line:

1. **convert -flop <input file> <output file>**
2. **convert -flip <input file> <output file>**

VI. TEXT HANDLING

Input arguments:

* To list the fonts ImageMagick knows about, use this command:**identify -list type**

Command line:

## Simple Text Label: **convert -background lightblue -fill blue \-font Courier -pointsize 72 label:Anthony \<output file>**

## Text Attributes effect: **convert -background white -fill dodgerblue -font Courier \ -strokewidth 2 -stroke blue -undercolor lightblue \ -size 165x70 -gravity center label:Anthony <output file>**

**VIDEO\_UTILITY functionalities:**

1. VIDEO TO FRAMES:

Input arguments:

* -i <input filename>, -vf<filter string>
* Output one image every second: **ffmpeg -i <input video file> -vf fps=1 <output%d file>**
* Output one image every minute: **ffmpeg -i <input video file> -vf fps=1/600 <output%d file>**

1. FRAMES TO VIDEO:

* Conversion of images into video:  **ffmpeg -framerate 1/3 -pattern\_type glob -i <input image file> <output video file>**

1. VIDEO TO GIF:

* **ffmpeg -i <input video file>-vf scale=320:-1 -r 10 -f image2pipe -vcodec ppm - | convert -delay 5 -loop 0 - <output gif file>**

1. VIDEO SPLIT

* The command to split the first 5 seconds would be:**ffmpeg -i <input video file>-vcodec copy -acodec copy -ss 00:00:00 -t 00:00:05 <output video file>**
* **ffmpeg -i yourvideoname.mp4 -t 00:00:59 -c copy part1.mp4 -ss 00:00:59 -codec copy part2.mp4**

1. VIDEO CONCATENATION

Input arguments

* Yum install **gpac** (library for MP4Box)
* Yum install **mkvtoolnix** ( library for mkvmerge)
* For available formats supported by **ffmpeg**: **ffmpeg -formats**

Commandline

1. **MP4Box -cat 1.mp4 -cat 1.mp4 -new mergedFile.mp4**
2. **sox short1.wav short2.wav short3.wav long.wav**
3. **mkvmerge -o outfile.mp4 1.mp4 2.mp4**
4. VIDEO WATERMARKING

Command line:

1. **ffmpeg -i Office.mp4 -i s3bubble-logo-white.png -filter\_complex "overlay=10:10" watermarked.mp4**
2. VIDEO CONVERSION

Input arguments:

* target pal-dvd : Output format
* ps 2000000000 maximum size for the output file
* -aspect 16:9 : Widescreen.
* -ar to specify the audio sample rate (kHz) and -ac to specify the number of channels (1 for mono, 2 for stereo).
* FFmpeg can place the mov atom at the beginning with the -movflags faststart options

Command line:

1. **ffmpeg -i input.mp4 output.webm**
2. **ffmpeg -i input.mp4 output.mkv**
3. **ffmpeg -i input.mp4 -c:v vp9 -c:a libvorbis output.mkv**
4. **ffmpeg -i yourvideoname.mp4 -vn -ab 128 outputaudio.mp3**
5. **ffmpeg -i 1.mp4 -c:v libx264 output\_filename.avi**
6. **ffmpeg -i video.mpg -ab 26k -f flv video1.flv**
7. **ffmpeg -i video.avi -target pal-dvd -ps 2000000000 -aspect 16:9 video.mpeg**
8. **ffmpeg -i video.flv -ar 44100 -ac 2 -vcodec copy video.mp4**
9. **ffmpeg -i video.flv -movflags faststart video.mp4**
10. **ffmpeg -i <input avi file> -c:v libx264 -crf 19 -preset slow -c:a aac -strict experimental -b:a 192k -ac 2 <output mp4 file>**
11. EXTRACT/MUTE AUDIO:

Input arguments:

* **-an** is used to remove the audio from the video file.
* **vn**: helps to disable video recording during the conversion.
* **ar**: helps you set audio sampling rate in Hz.
* **ab**: set the audio bitrate.
* **ac**: to set the number of audio channels.
* **-f**: format.
* **-r** specifies the frame rate (fps).
* **-s:** switch is used to resize a video while maintaining the aspect ratio.
* **-crf:** controls the quality.
* prefer -c:a aac -strict experimental instead of libvo\_aacenc

Command line:

1. **ffmpeg -i yourvideoname.mp4 -an mutevideo.mp4**
2. **ffmpeg -i video1.avi -vn -ar 44100 -ac 2 -ab 192 -f mp3 audio3.mp3**

9**.** Cut Video File into a Smaller Clip:

Input arguments:

* -ss defines the starting time stamp (here starting time is the 45th second) and -t tells the total time duration for the clip. So, -t 40 means 40 second duration.

Command line:

1. **ffmpeg -i <input file> -ss 00:00:45 -codec copy -t 40 <output file>**

10. Cropping an Audio File:

Input arguments:

* -ss is the staring time and -t is the duration of the cropped file.

Command line:

1. **ffmpeg -ss 00:00:15 -t 45 -i sampleaudio.mp3 croppedaudio.mp3**

**STREAMING functionalities:**

1. **VOD streaming:**
   1. **Ffmpeg:**

**server::ffmpeg -re -i <input file> -f mpegts udp:0.0.0.0:1234**

- ffmpeg is about calling ffmpeg.exe

- re tells to read input at native frame rate

- i input file

- f is the container format

- mpegts is MPEG TS

**client::ffplay udp://localhost:1234**

* 1. **Vlc:**

**Stream:**

Stream can be done by using ffmpeg (same command as the above)

**vlc client :: vlc udp://@:1234**

* 1. **Wowza:**

**Tested using the trail version of wowza**

Follow the steps listed to stream the VOD on the wowza website.

* Its allows us to successfully stream the static video to a mobile(android,ios), and embed it into the web-browser.
* There is a developer version available which allows us to customise the wowza XML file

**Config for XML file:**

This section shows you how to configure the vod application by editing the Application.xml configuration file in a text editor.

Note: If you created your application by following the previous section, you can skip this section. If you make changes to Application.xml and you're using Wowza Streaming Engine™ software, any supported settings will be displayed in the manager the next time it's started. Wowza Media Server™ software doesn't support Wowza Streaming Engine Manager, so you must edit Application.xml in a text editor if you're running Wowza Media Server.

1. Create the application folder [install-dir]/applications/vod.
2. Create the configuration folder [install-dir]/conf/vod and copy [install-dir]/conf/Application.xml to this new folder.
3. Edit the newly copied Application.xml file and make the following changes (some of these settings may already be present):
4. 1. Set the Streams/StreamType property to:
   2. <StreamType>default</StreamType>
   3. Set the HTTPStreamers property to:
   4. <HTTPStreamers>cupertinostreaming,smoothstreaming,sanjosestreaming,mpegdashstreaming</HTTPStreamers>
   5. Set the RTP/Authentication/PlayMethod to:
   6. <PlayMethod>none</PlayMethod>
5. Start the Wowza media server.

**2.LIVE STREAMING:**

**a.Wowza:**

**3.SAVING A STREAMED FILE:**

**ffmpeg -i <INPUT FILE> -f mpegts udp:0.0.0.0:1234 -f mpegts <OUTPUT FILE>**

**BENCHMARKING functionalities:**

Benchmarking evaluates the responsiveness and the stability of a software application in terms of its load capacity, behavior under stress conditions, reliability, scalability, etc

**On Development Conditions:**

1. Unit Testing
2. Integration Testing
3. Functional Testing
4. Algorithm Performance Testing
5. System Testing
6. Stress Testing
7. Performance Testing
8. Usability Testing
9. Acceptance Testing
10. Regression Testing
11. Beta Testing

**On Deployment Conditions:**

Benchmark testing involue performance testing metrics which every software application that belongs to organization should pass. the baseline performance metrics values should meet the standards at least,to process further.

**Initial Testing**

1. Maximum Volume of request per unit time to the application from end user
2. Service level agreement test
3. Permitted delay in response from application server test
4. Load condition test

**Web Server Performance Metrics**

1. Busy and Idle Threads
2. Throughput
3. Bandwidth requirements

**App Server Performance Metrics**

1. Load Distribution
2. CPU Hotspots
3. Worker Threads
4. Memory Issues