

Audion Image Tools — v1 Functional Spec

Status note: this is an early functional spec. Current user-facing behavior, GUI module order, crop PDF options, and release rules are documented in `README_RU.md`, `docs/USER_GUIDE_RU.md`, `docs/RELEASE_GUIDE_RU.md`, and `AGENTS.md`.

Project goal

Audion Image Tools is a portable Python-based image processing toolkit focused on:

- reliable batch conversion to **JPG** and **PNG**
- safe normalization of mixed image sources
- resolution/aspect-ratio fitting for screens and documents
- color/profile normalization for practical real-world archives
- predictable output naming and packaging for large batches

Primary user reality:

- most common work is **JPG/PNG**
- mixed archives may include HEIC/HEIF, TIFF, GIF, WebP, PSD, SVG, and camera RAW
- processing must be safe, explicit, and reproducible

Core design principle

Do **not** treat all formats as equally reliable.

The product should classify formats into 3 support tiers:

1. Core / stable

Use the main pipeline directly.

- BMP
- GIF
- JPG / JPEG
- PNG
- TGA
- TIFF / TIF
- WebP
- AVIF

2. Extended adapters

Supported through dedicated adapters or plugins.

- HEIC / HEIF
- SVG / SVGZ
- CR2 / CR3 / DNG

3. Best-effort / limited

Open when possible, but do not promise identical behavior to core formats.

- PSD
- WMF
- IFF
- XIF

Recommended library stack

Core

- Pillow
- Pillow ImageOps
- Pillow ImageCms

Extended

- pillow-heif for HEIC/HEIF
- CairoSVG for SVG/SVGZ rasterization
- rawpy for CR2/CR3/DNG

Optional fallback layer

- ImageMagick adapter for difficult legacy formats and edge cases

Input format support policy

Target input list

Requested list:

- *.bmp
- *.gif
- *.jpg
- *.jpeg
- *.iff
- *.png
- *.psd
- *.heic
- *.tga
- *.tif
- *.tiff
- *.xif
- *.wmf
- *.cr2
- *.cr3
- *.dng
- *.svg
- *.svgz
- *.heif

- *.avif
- *.webp

Practical recommendation for v1

Fully target first

- BMP
- GIF
- JPG / JPEG
- PNG
- TGA
- TIFF / TIF
- WebP
- AVIF
- HEIC / HEIF
- SVG / SVGZ
- CR2 / CR3 / DNG

Accept as best-effort

- PSD
- WMF
- IFF
- XIF

Main processing families

1. Convert to JPG / PNG

Purpose

Batch-convert mixed inputs to standard delivery formats.

Output formats

- JPG
- PNG

Rules

- do not change pixel dimensions unless explicitly requested
- do not change DPI unless explicitly requested
- preserve orientation correctly
- safely flatten unsupported alpha when exporting to JPG

JPG quality presets

- 60
- 75
- 90

PNG compression policy

PNG should remain visually lossless.

Use UI presets such as:

- Fast
- Balanced
- Maximum compression

Internally these should map to compression settings, not fake visual-quality percentages.

Recommended default:

- PNG = **Balanced**
-

2. DPI / resolution metadata adjustment

Purpose

Change image DPI metadata without changing pixel resolution.

Example:

- 300 x 300 DPI
- 600 x 600 DPI

Modes

- set DPI only
- preserve pixel size
- preserve color profile policy

Important note

This operation changes **print/display metadata**, not actual pixel dimensions.

Recommended presets

- 72 DPI
- 96 DPI
- 150 DPI
- 300 DPI
- 600 DPI
- custom

3. Screen fit / output resolution fit

Purpose

Fit images to screen-oriented targets.

Targets

- 1080p = 1920x1080
- 1440p = 2560x1440
- 2160p = 3840x2160

Modes

- **cover/crop** — fill target completely and crop excess
- **contain** — fit inside target without crop
- **pad** — fit inside target and add background padding
- **fit by height** — set target height and crop/extend width policy as chosen
- **fit by width** — optional extra mode for completeness

Orientation handling

- landscape target
- portrait target
- auto-detect based on source

4. Aspect ratio fitting

Purpose

Fit to standard aspect ratios while optionally preserving or changing resolution.

Target aspect groups

Screen

- 16:9 horizontal
- 16:9 vertical

Paper

- A4 horizontal ratio
- A4 vertical ratio
- A3 horizontal ratio

- A3 vertical ratio

Modes

- crop to ratio only, keep source resolution
- crop + resize to target resolution
- contain inside ratio box
- pad into ratio box

5. Normalize / Fix Image

This should be one of the strongest modules in the whole product.

Include

- EXIF-based orientation correction
- remove stale EXIF orientation after physical rotation
- safe mode conversion between image modes
- controlled alpha flattening for JPG
- broken/partial metadata sanitation
- optional ICC/profile normalization

Safe mode conversions

Examples:

- **P** -> **RGB**
- **RGBA** -> **RGB** with chosen background
- **LA** -> **L** or **RGB**
- **CMYK** -> **RGB** or target profile
- grayscale palette images -> **L** or **RGB**

Alpha handling for JPG

Provide explicit background options:

- white

- black
- custom color
- checker preview only

6. Color profile normalization

This is extremely valuable.

Important design rule

sRGB normalization and **CMYK normalization** must be separate target modes.

Do not attempt to auto-decide between them silently.

Mode A — Normalize to sRGB

Primary default for general use.

Use when target output is:

- JPG for web/general delivery
- PNG
- screen assets
- mixed archives needing standardization

Mode B — Normalize to CMYK

Separate print-oriented mode.

Use only when target output format and workflow make sense for CMYK.

Recommended practical outputs for CMYK mode:

- TIFF
- JPG (when specifically requested)

Why this matters

PNG-centered workflows should remain **sRGB-first**.

CMYK is a separate production/export branch, not the default normalization target.

Proposed profile modes

- preserve embedded ICC profile
- convert to standard sRGB ICC profile
- convert to selected CMYK ICC profile
- strip ICC profile after conversion if explicitly requested

Recommended defaults

- default target color space: **sRGB**
- CMYK conversion: **advanced / print mode only**

7. Grayscale conversion

Add a dedicated batch mode:

Convert all input images to grayscale

Variants

- grayscale with preserved original dimensions
- grayscale + JPG export
- grayscale + PNG export
- grayscale + DPI set
- grayscale + screen-fit

Naming suggestion

- `_gray`

8. Border trim

Very useful for scans, screenshots, exports, and document captures.

Purpose

Automatically trim uniform borders.

Border types

- white border
- black border
- near-white threshold
- near-black threshold
- custom background sample

Modes

- trim all sides
- trim only top/bottom
- trim only left/right
- trim and then fit to aspect/output size

9. Contact sheet

Very strong practical feature.

Purpose

Generate preview sheets from folders/batches.

Options

- rows / columns
- cell size
- filename under thumbnail

- page DPI
- output as JPG or PNG
- sort order by filename/date
- optional frame number for multi-frame extraction

Useful outputs

- selection sheet
- proof sheet
- archive inventory sheet

10. Watermark / text overlay

Modes

- image watermark
- text watermark
- corner placement
- centered placement
- tiled watermark

Options

- opacity
- margin
- scale relative to image
- font size
- background box optional

11. Animated / multi-frame handling

This must be explicit and predictable.

A. GIF / animated WebP input policy

Modes

- first frame only
- all frames processed and re-exported
- extract frames to image sequence

Sequence naming

- 001
- 002
- 003
- ...
- 999

B. Multi-frame TIFF handling

Requested feature

Convert multi-frame TIFF into separate PNG files with sequence numbers.

Recommended behavior

- create one subfolder per source file
- export frames as:
 - 001_filename.png
 - 002_filename.png
 - 003_filename.png
- preserve frame order exactly

Optional extras

- keep source DPI metadata where meaningful
 - apply grayscale/normalization per frame if selected
-

12. SVG / SVGZ rasterization

Requested feature

Convert SVG / SVGZ with target DPI such as 300 or 600.

Recommended interpretation

Since SVG is vector, DPI should be used as a **rasterization/export parameter**, not as ordinary raster metadata only.

Modes

- SVG -> PNG at 300 DPI
- SVG -> PNG at 600 DPI
- SVG -> JPG at 300 DPI
- SVG -> JPG at 600 DPI
- SVG -> custom pixel size

Important note

For SVG, DPI should influence the rendered raster result.

13. GIF / WebP DPI handling

Requested idea

GIF/WebP/SVG — simple conversion to 300 or 600 DPI.

Recommended implementation

Split this into two behaviors:

For SVG

- rasterize using requested DPI

For GIF/WebP raster files

- allow DPI metadata assignment when exporting to suitable output

- or convert frames/images to JPG/PNG/TIFF with requested DPI metadata

Recommended UX wording

Instead of saying “change GIF/WebP to 300 DPI”, present it as:

- **export with DPI metadata = 300 / 600**
- **rasterize SVG at 300 / 600 DPI**

This will reduce user confusion and keep behavior honest.

High-value batch presets

These presets will make the product feel immediately useful.

Preset 1 — Web JPG 75

- normalize EXIF
- normalize to sRGB
- flatten alpha on white
- export JPG quality 75
- strip unnecessary metadata

Preset 2 — Web JPG 90

- normalize EXIF
- normalize to sRGB
- flatten alpha on white
- export JPG quality 90
- preserve ICC or convert to sRGB ICC

Preset 3 — Lossless PNG Standard

- normalize EXIF
- normalize to sRGB
- preserve alpha
- export PNG balanced compression

Preset 4 — Print Metadata 300 DPI

- preserve pixels
- set DPI = 300
- preserve ICC or convert to selected profile

Preset 5 — Print Metadata 600 DPI

- preserve pixels
- set DPI = 600
- preserve ICC or convert to selected profile

Preset 6 — 1080p Cover

- normalize EXIF
- crop/cover to 1920x1080
- export JPG or PNG

Preset 7 — 2160p Cover

- normalize EXIF
- crop/cover to 3840x2160
- export JPG or PNG

Preset 8 — A4 Ratio Crop

- crop to A4 ratio
- keep resolution or optionally resize

Preset 9 — Grayscale Archive

- normalize EXIF
- convert to grayscale
- export PNG or JPG

Preset 10 — TIFF Frames to PNG Sequence

- split multi-frame TIFF
- export numbered PNG sequence

Metadata policy

This should be global and user-selectable.

Modes

- preserve all metadata that is safely transferable
- preserve only DPI and color profile
- strip all metadata

Recommended default

- preserve only **DPI + ICC/color info**

This gives predictable outputs without dragging messy EXIF/XMP baggage into every export.

Output naming policy

Consistent naming matters a lot.

Recommended suffixes

- `_jpg60`
- `_jpg75`
- `_jpg90`
- `_png`
- `_dpi300`
- `_dpi600`
- `_1080p`
- `_1440p`
- `_2160p`
- `_16x9`
- `_a4`
- `_a3`
- `_gray`
- `_srgb`

- `_cmyk`
- `_trim`

Frame extraction naming

Use fixed 3-digit numbering:

- `001_...`
- `002_...`
- `003_...`

Recommended folder workflow

Input

- source folder or mixed picked files

Managed workspace

- `input_staged/`
- `work/`
- `output/`
- `logs/`

Output organization

- by operation preset
- optionally one final consolidated output folder
- optional ZIP packaging

Strong MVP scope

If we keep v1 focused, the strongest MVP is:

Core MVP modules

1. Convert to JPG

2. Convert to PNG
3. Set DPI only
4. Fit to 1080p / 1440p / 2160p
5. Fit to 16:9 / A4 / A3
6. Normalize / Fix Image
7. Normalize to sRGB
8. Grayscale conversion
9. TIFF frames -> numbered PNG sequence
10. Contact sheet

Extended v1.1 modules

1. HEIC / HEIF adapter
2. SVG / SVGZ rasterization
3. RAW adapter via rawpy
4. Watermark / text overlay
5. Border trim
6. CMYK conversion branch

Important caution points

1. CMYK should not be default

For a JPG/PNG-centered tool, **sRGB must be the default target**.

2. PNG should stay honest

PNG is lossless. Treat its "quality" as compression preset, not visual degradation level.

3. DPI should not be confused with resolution

Changing DPI alone does not increase detail.

4. Legacy formats need graceful failure

PSD/WMF/IFF/XIF should produce clear "best-effort / unsupported feature" messages, not silent corruption.

5. Multi-frame inputs need explicit policy

Never guess silently whether to use first frame or all frames.

Ideal first product menu

Convert

- Convert to JPG
- Convert to PNG
- Convert to Grayscale JPG/PNG

Normalize

- Fix EXIF Orientation
- Normalize to sRGB
- Convert to CMYK
- Flatten Alpha for JPG
- Sanitize Metadata

Resolution / layout

- Set DPI Only
- Fit to 1080p / 1440p / 2160p
- Fit to 16:9
- Fit to A4 / A3
- Border Trim

Sequence / preview

- TIFF Frames to PNG Sequence
- GIF/WebP Frames to Sequence
- Contact Sheet

Markup

- Watermark
- Text Overlay

Recommended product positioning

Audion Image Tools should feel like:

- a practical batch image normalizer
- a delivery/export utility
- a repair tool for chaotic mixed image archives
- a bridge between screen assets, scans, prints, and portable archives

Not an “editor clone”, but a **high-trust transformation engine**.

Best next step

After this spec, the next correct document is:

Audion Image Tools — Project Structure & Module Layout

That file should define:

- folder tree
 - module names
 - CLI commands
 - launcher names
 - config schema
 - portable environment contents
 - wheel/licensing list
 - future GUI structure
-

Technical notes grounded in current docs

Pillow documents `ImageOps.exif_transpose()` for applying EXIF orientation and removing the orientation tag afterward, and it also provides `contain()`, `cover()`, `fit()`, and `pad()` helpers for geometry workflows.

□cite□turn713825view4□turn713825view5□turn713825view6□turn713825view7□

Pillow's `ImageCms` module uses the LittleCMS2 engine and supports reusable color transforms, which makes batch normalization to a chosen target profile practical for repeated conversions.

□cite□turn549885view2□turn549885view0□turn549885view1□

For JPEG output, Pillow states that quality values above 95 should generally be avoided, and that `100` mostly disables parts of JPEG compression while greatly increasing file size with little visual gain. Pillow also notes that JPEG supports a `dpi` option on save. □cite□turn713825view1□turn229522view0□

For PNG, Pillow documents `compress_level` from 0 to 9 and notes that PNG can embed ICC profile data, which is why PNG presets in this project should be treated as compression presets rather than visual quality tiers. □cite□turn713825view2□

Pillow supports multi-frame TIFF access via `seek()`, `tell()`, and `n_frames`, which directly supports the TIFF-to-numbered-PNG-sequence feature. □cite□turn229522view2□

HEIC/HEIF support can be added to Pillow through `pillow-heif`, CairoSVG officially converts SVG to PNG and other raster outputs with DPI-aware export options, and rawpy exposes RAW decoding through `rawpy.imread()` for CR2/CR3/DNG workflows.

□cite□turn625178view1□turn625178view2□turn625178view3□

Pillow's own format documentation also confirms some of the intentional limitations behind the support tiers above: PSD support is limited to older Photoshop-written PSD files, and WMF/EMF loading on Windows defaults to 72 DPI unless another DPI is explicitly requested.

□cite□turn229522view3□turn229522view4□