

Batch Image Resizer and Converter

A desktop batch image resizer and converter: whole-folder resizing with width presets, JPG/PNG/WebP conversion, per-file savings report and before/after comparison — delivered built-to-order with full source, report, PPT and viva kit.

01. Title

Batch Image Resizer and Converter.

02. Introduction / Background

Resizing images one by one is drudgery — a student with 40 project photos or a seller with 200 product shots faces the same open-resize-save loop. Online batch tools demand uploading personal photos to a stranger's server and usually cap file counts or watermark output. This project builds a desktop batch resizer and converter that processes entire queues locally: drop in JPG, PNG or WebP files, pick a max-width preset, choose output format and quality, and process everything in one click with per-file before/after savings. The canvas-based pipeline runs identically as a desktop app or web page, fully offline.

03. Problem Statement

Bulk image resizing today means either tedious manual work, or trusting personal photos to free online services with limits and watermarks. Users need a private, unlimited, offline tool that resizes and converts whole batches while reporting exactly what was saved per file.

04. Objectives

- Queue any number of JPG/PNG/WebP files via drag-and-drop with thumbnail previews.
- Resize to width presets (1920/1280/800 px or custom) with aspect ratio always preserved and no upscaling.
- Convert output to JPG, PNG or WebP with a quality slider (40–100).
- Report per-file original vs resized dimensions, sizes and percentage saved.
- Provide before/after comparison and individual downloads with renamed files — fully offline.

05. Proposed Methodology

Files are read locally with the FileReader API and shown as thumbnails. On processing, each image is drawn to an offscreen canvas scaled so its width equals the target (scale capped at 1.0, never upscaling), then serialized with toDataURL in the chosen format and quality. A results table computes original vs resized dimensions and sizes; clicking a row renders the before/after comparison from the in-memory data URLs. Download buttons save each result with the configured filename prefix; originals are never modified.

06. System Architecture / Working

Three panels: the batch queue (drop zone + thumbnails), the settings panel (presets, custom width, format, quality, filename prefix, process button, progress bar), and the results area (savings table + before/after comparison). The pipeline is pure client-side canvas; Electron packaging turns the same code into a desktop app.

07. Hardware / Software Requirements

- Desktop/laptop running Windows, macOS or Linux; also runs as a plain web page.
- A modern browser engine (Chrome/Chromium 90+) or the Electron shell.
- No network connection required.
- 4 GB RAM recommended for batches of 50+ MP images.

08. Expected Outcomes

- A working batch tool processing multi-image queues with measurable size savings (typically 50–80% at quality 85).
- Per-file savings table and before/after views demonstrating the pipeline.
- Complete source, user manual, report, PPT and viva Q&A.;

09. Applications

- Students preparing project report and PPT images.
- Bloggers and sellers optimizing product photos for the web.
- Anyone converting PNG folders to WebP for faster websites.
- Classroom demonstration of the canvas image pipeline.

10. Future Scope

- Crop, rotate and watermark operations.
- EXIF-aware resizing with metadata preservation options.
- Folder watch mode: auto-process new files.