

Come comprimere le immagini per migliorare SEO e user experience

Su Your Inspiration Web si è già parlato di [come gestire immagini responsive nel modo più efficiente](#), sia su dispositivi desktop sia su mobile, affrontando la questione anche dal punto di vista del “peso” delle immagini. Proprio su quest’ultimo argomento vorrei porre l’accento con questo articolo, poiché incide su diversi aspetti del lavoro di noi professionisti del web, primi tra tutti la user experience e la SEO.

Già, perché **più il caricamento di un contenuto è rapido, meno l'utente è infastidito, con conseguente aumento delle conversioni**. È un concetto così importante che tra gli autori di tool volti a supportarlo troviamo Mozilla, Google e Yahoo!. Usando [GTmetrix](#) avremo contemporaneamente a disposizione i tool degli ultimi due, nonché un clone del pannello “Net” di Firebug. Generando un report per un sito non ancora ottimizzato ci imbattemmo subito nella necessità di “comprimere le immagini”, nonostante siano già tutte “Salvate per web” con tanto di casellina “Ottimizzata” selezionata (i [suoi effetti](#) purtroppo non vengono considerati dai tool appena citati). Abbiamo quindi bisogno di altre soluzioni.

Comprimere le immagini vettoriali

Saprete già tutti che i web font e le immagini in formato SVG sono vettoriali, e che rappresentano **la scelta ottimale per realizzare immagini semplici** come potrebbero esserlo delle icone, poiché sono leggere e ridimensionabili senza perdita di qualità. La loro adozione è già di per sé un modo di alleggerire le nostre pagine, ma ricordate di comprimerle lato server, ad esempio tramite il [mod deflate di Apache](#). Sempre riguardo alle immagini SVG, è possibile convertire le scritte in linee grafiche anziché incorporare i font, riducendo così ulteriormente il loro peso. Esistono dei tool che consentono di testare visivamente in tempo reale varie impostazioni per eliminare *parti in eccesso* dei file SVG, come [questo editor](#).

Comprimere le immagini raster

Sono due i tipi di compressione per le immagini raster: **lossy** e **lossless**. La prima riduce maggiormente il peso del file a scapito di una degradazione impostabile in modo da essere accettabile, ma proprio a causa di questo calo di qualità, dopo la compressione l’immagine dev’essere guardata e valutata. La seconda invece mantiene la qualità originale e l’immagine compressa non necessita di revisione, ma **le dimensioni del file risultano maggiori della sua versione lossy**. In particolare vi segnalo [Mozjpeg](#), un algoritmo di compressione lossless per immagini Jpeg molto efficace sviluppato da Mozilla e [testabile online](#).

Dunque, **quale tipo di compressione scegliere?** Il mio consiglio è che se le vostre priorità sono

la produttività o la SEO, e vi basta superare la validazione dei tool, è preferibile la compressione lossless. Se invece sono la performance o la user experience (ambiti che richiedono un'analisi ragionata della situazione) meglio optare per la compressione lossy (che offre comunque vantaggi in ambito SEO).

Comprimere più immagini alla volta

Avere la possibilità di manipolare molte immagini alla volta velocizza notevolmente lo sviluppo. Ecco perché ho testato la compressione lossless di diversi tool su una dozzina di immagini a media risoluzione trovate cercando "Fashion wallpaper" su Google Immagini, selezionando foto ritoccate e non, immagini scontornate, con colori brillanti o opachi, orientate in verticale, ecc. Il miglior risultato su Windows lo ha raggiunto [FileOptimizer](#), mentre su Mac ha avuto la meglio [ImageOptim](#). In fase di sviluppo è anche possibile usare strumenti incredibilmente utili come [Yeoman](#) o soltanto il suo componente **Grunt**.

Comprimere le immagini a seconda del browser utilizzato

Questa strategia comporta un aumento dei tempi di sviluppo, ma anche dei risultati ottimizzati all'estremo sulla base dell'audience. Considerando infatti oltre ai classici GIF, PNG e JPEG anche con i formati [WebP](#) e [JPEG XR](#) (che supportano trasparenza e animazioni) otteniamo due nuovi concorrenti nella gara per l'ottimizzazione (e altri ne stanno arrivando, come [JPEG 2000](#)). Attenzione però: **non potete essere sicuri che tutte le immagini codificate in questi nuovi formati siano più "leggere"**. Sta a voi determinare il vincitore della gara, immagine per immagine... oppure fare alcune conversioni di prova a fini statistici e, in base a esse, comprimere tutte le immagini. Una volta compresse le immagini, potrete scegliere il vostro metodo per il browser-detect preferito, valutandone i pro e i contro: molto diffusi sono [Modernizr](#) o [imgsupport](#) e la [lettura lato server](#) o [client dell'userAgent](#). Dopodiché, dovrete fare in modo che certe versioni di una stessa immagine vengano mostrate soltanto su certi browser, organizzandole ad esempio in cartelle "per browser". L'implementazione a mio avviso migliore e che precorre i tempi è quella di usare l'elemento "picture" (grazie a [picturefill](#)) e lasciare che sia il browser a scartare i formati non supportati.

Un plugin per WordPress

Ebbene sì, esiste un plugin anche per questo! :) [EWWW Image Optimizer](#) esegue la compressione lossless non appena viene fatto l'upload di un'immagine, oltre ad avere altre utili funzioni. Il risultato però non è garantito: in alcuni casi GTmetrix non rileva la compressione fino a che non si trova una configurazione ottimale del plugin. Inoltre, in caso di hosting non compatibile si è costretti a usare la compressione cloud che richiede un pagamento mensile.

Conclusioni

Ottimizzare le immagini è diventato al giorno d'oggi strategicamente importante per progetti di medio-alto livello. Gli strumenti di sviluppo non mancano e le possibilità di implementazione stanno aumentando grazie all'integrazione di nuove tecnologie da parte dei browser.