Alma Mater Studiorum · Università di Bologna

Scuola di Scienze

Corso di laurea triennale in Informatica

Social network e watermarking: alcuni risultati preliminari

Presentata da:
Daniele Piergigli

Relatore:
Chiar.mo Prof.
Danilo Montesi

Sessione II

Anno Accademico 2015-2016

Indice

Elenco delle figure

Elenco delle tabelle

1 Introduzione

La diffusione dei social network ha permesso ad un elevato numero di persone di condividere i propri contenuti multimediali (testo, foto, video) con una larga platea di contatti [Figura 1 e 2]. Potenzialmente questi contenuti possono essere condivisi anche con persone non direttamente collegate al proprietario. Uno dei comportamenti più diffuso degli utenti dei SN è la condivisione di immagini, con il fine di mostrare una parte della loro vita privata solo con le persone a loro connesse senza trarne un profitto personale.

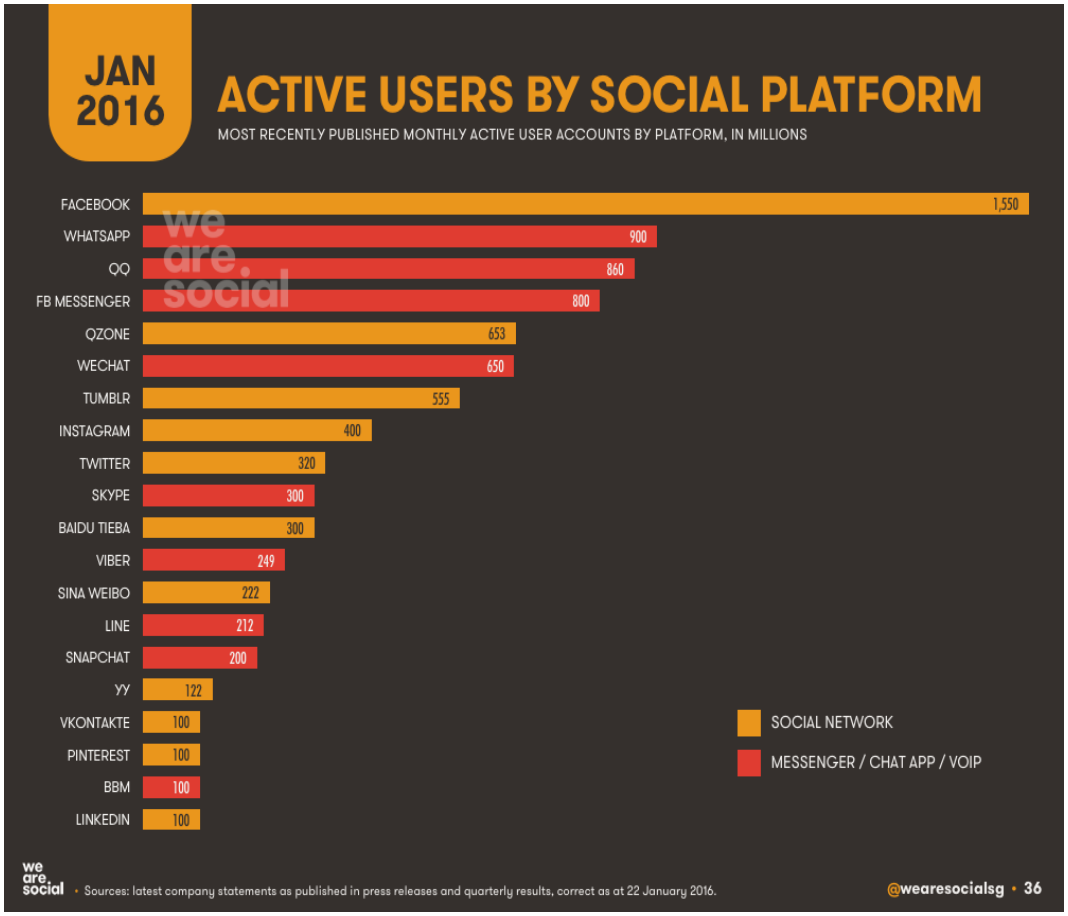

Figura 1: Numero di utenti attivi per ogni social network

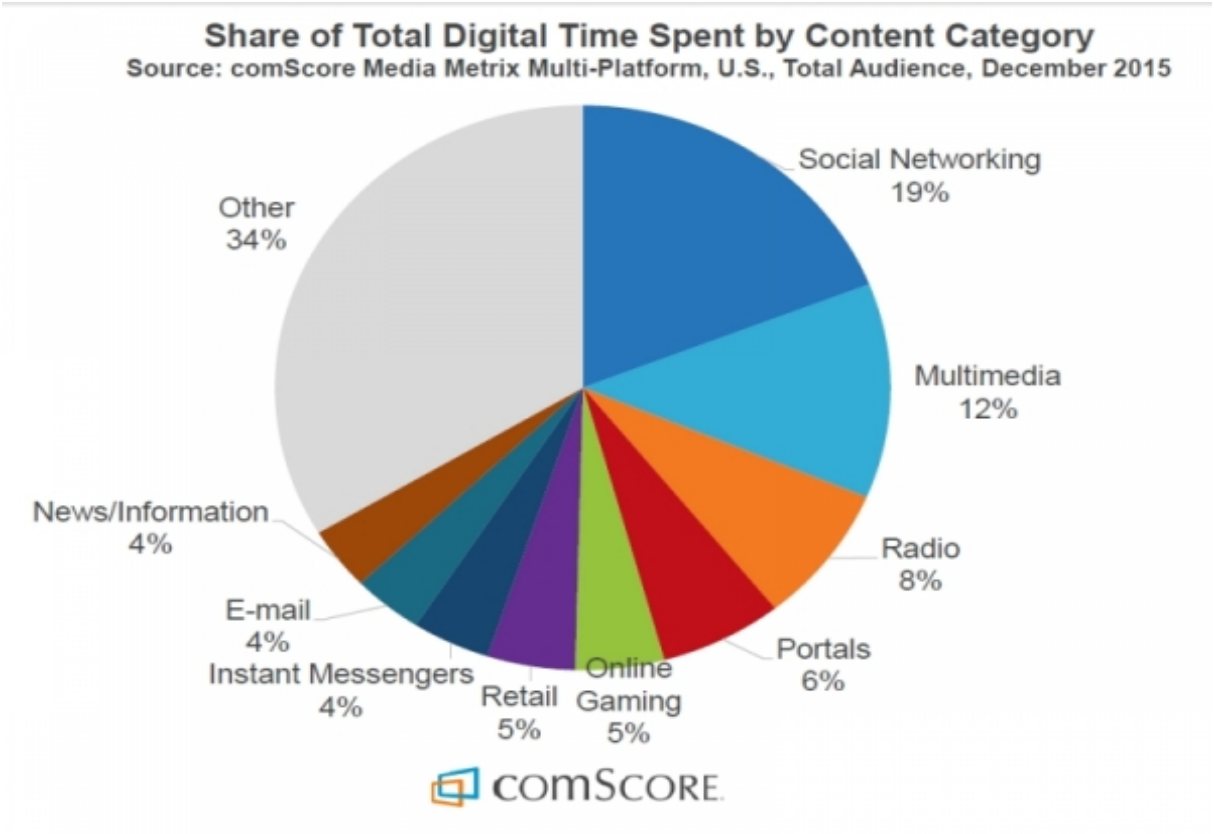

Figura 2: Tempo speso sui social network dagli utenti rapportato con il tempo totale di navigazione su internet annuale

In questo contesto diventa importante riconoscere e preservare la proprietà di un'immagine. La garanzia di riconoscere l'autore originario di un'immagine è utile in diverse situazioni, come per esempio in ambito forense, agevolando qualunque attività legale legata all'immagine.
L'utilizzo di un'immagine di proprietà altrui con fini illegali (lucro, pedopornografia) violerebbe diverse leggi tra cui la privacy dell'autore originale. Questo lavoro di tesi si è occupato quindi di valutare il diritto alla riservatezza che un utente deve avere quando condivide immagini sui social network.
Gli algoritmi di watermarking si occupano appunto di proteggere la nostra privacy da malintenzionati [25]. In pratica, alterando in maniera più o meno impercettibile all'utente, l'algoritmo di watermarking modifica l'immagine permettendo di inserire una firma (stringa o immagine) al suo interno. La sfida più difficile è portare a termine l'operazione preservando le caratteristiche iniziali dell'immagine.

1.1 Obiettivi

Lo studio effettuato in questo documento si prefigge lo scopo di analizzare e testare vari algoritmi di watermarking su immagini condivise. Andare a capire come i watermark vengano inseriti all'interno di una foto e come questo risponda alle alterazioni da parte dei social network permette di capire quali, tra questi algoritmi, si adattino di più alle esigenze di questo studio.
L'elaborato inoltre, si prefigge lo scopo di controllare se i social network inseriscano un qualche watermark all'interno dell'immagine caricata. Il test viene svolto per capire se i SN usino una qualche forma di firma digitale e se questa alteri l'immagine in qualche modo.
Lo studio non si prefigge in alcun modo lo scopo di migliorare questi algoritmi ed inoltre, sono stati presi in considerazione solo quegli algoritmi che, con le loro caratteristiche e proprietà, erano in grado di portare a un risultato nel contesto dei social network.
La differente quantità di algoritmi trovati per immagini in bianco e nero e quelle a colori è disomogenea, questo è dovuto al fatto che alcuni di quelli a colori fanno parte di software a pagamento e sono stati quindi trascurati volontariamente dai test, prediligendo algoritmi con sorgenti consultabili.
Capire come un'immagine venga modificata quando questa viene caricata sul web è parte principale dello studio, in quanto, capire se questa è stata soggetta ad un'alterazione delle proprie caratteristiche e proprietà al momento del download permette di ipotizzare il motivo per cui un algoritmo di watermarking fallisce.
Analizzare inoltre come la geolocalizzazione e i CDN (Content Delivery Networks) vengano usati all'interno dei social network ha permesso di approfondire gli aspetti riguardanti la condivisione e gestione delle immagini all'interno del sito stesso.

1.2 Argomenti trattati

Il documento è suddiviso in quattro capitoli:

- **Nel primo capitolo** l'elaborato spiega cosa significa "watermark", per cosa è utilizzato e quali sono le sue caratteristiche e proprietà. Successivamente vengono catalogati i vari tipi di algoritmi di watermarking e viene data una spiegazione del funzionamento di queste tecniche con eventuali pregi e difetti riscontrabili nel campo dei social network.

- **Il secondo capitolo** invece è incentrato su una analisi accurata di come i SN gestiscono le immagini, come queste vengono elaborate e quali campi dell'immagine vengono effettivamente alterati. I test in questo capitolo presentano una forma gerarchica, i test iniziano confrontando se le immagini caricate sono le stesse che vengono restituite dai social network, sino ad arrivare a una comparazione più generale che ne paragona il grado di alterazione. Viene infine controllato se l'utilizzo dei CDN e della geolocalizzazione da parte dei social network influiscono sull'alterazione di un'immagine.

- **Il terzo capitolo** è invece focalizzato interamente sulla prova degli algoritmi di watermarking. Qui viene analizzata la percentuale di successo nell'estrazione del watermark

nei differenti social network per ogni algoritmo, mostrando la capacità di uno algoritmo di adattarsi più a un particolare SN rispetto che ad un altro.

- **In fine nell'ultimo capitolo** vengono mostrate le conclusioni di questa tesi, con un approfondimento sui risultati e le future ricerche che potrebbero essere svolte in questo campo.

2 Algoritmi di watermarking

Il watermark digitale [13], è una sequenza di caratteri o codice inserito all'interno di un documento digitale, immagine, video o programma per identificare univocamente l'utente originale che lo ha creato [Figura 3]. L'obbiettivo principale degli algoritmi di watermarking è quello di proteggere gli interessi del creatore del contenuto contro l'uso e la distribuzione illegale del file multimediale.

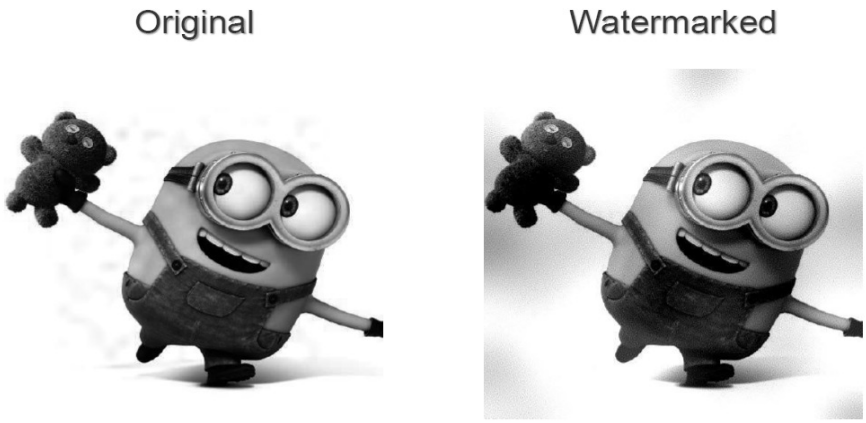

Figura 3: Watermark inserito all'interno di un'immagine in bianco e nero

Partendo dall'idea che è difficile prevenire questo tipo di attacchi (soprattutto nei social network), il watermark digitale assolve alla funzione di prova della proprietà del file per chi ne detiene il copyright, per identificarne le persone coinvolte e per allertare gli utenti quando stanno ricevendo documenti o programmi illegittimi. Lo studio si concentra solo nel campo del watermarking digitale applicato alle immagini.
La limitazione principale della tecnologia del watermark è la possibilità di avere un falso positivo nel quale, copie legali di un documento, immagine, video o programma vengono segnate come non accessibili da un utente autorizzato. Questo può accadere quando, ad esempio, i contenuti vengono corrotti intenzionalmente o vi è un deterioramento della firma digitale da parte di utenti o programmi malevoli.
Il processo di vita di un watermark, applicato ad un'immagine, si snoda attraverso quattro fasi [Figura 4]:

- **La creazione della firma digitale** viene eseguita da un primo algoritmo che genera un watermark da un'immagine o una stringa di testo.
- **La cifratura del watermark nell'immagine** effettuata da un algoritmo che prende la firma appena creata e la inserisce all'interno dell'immagine mediante uno dei tanti metodi di watermarking esistenti;

- **La decifratura del watermark dall'immagine**, utilizzando lo stesso metodo applicato al punto precedente ma in maniera inversa, restituisce la firma che l'immagine contiene.
- **Il confronto tra le due firme**, cioè tra il watermark creato dal primo algoritmo e quello estrapolato dall'immagine. Solo se queste coincidono si può avere la conferma che l'algoritmo di watermarking ha avuto successo.

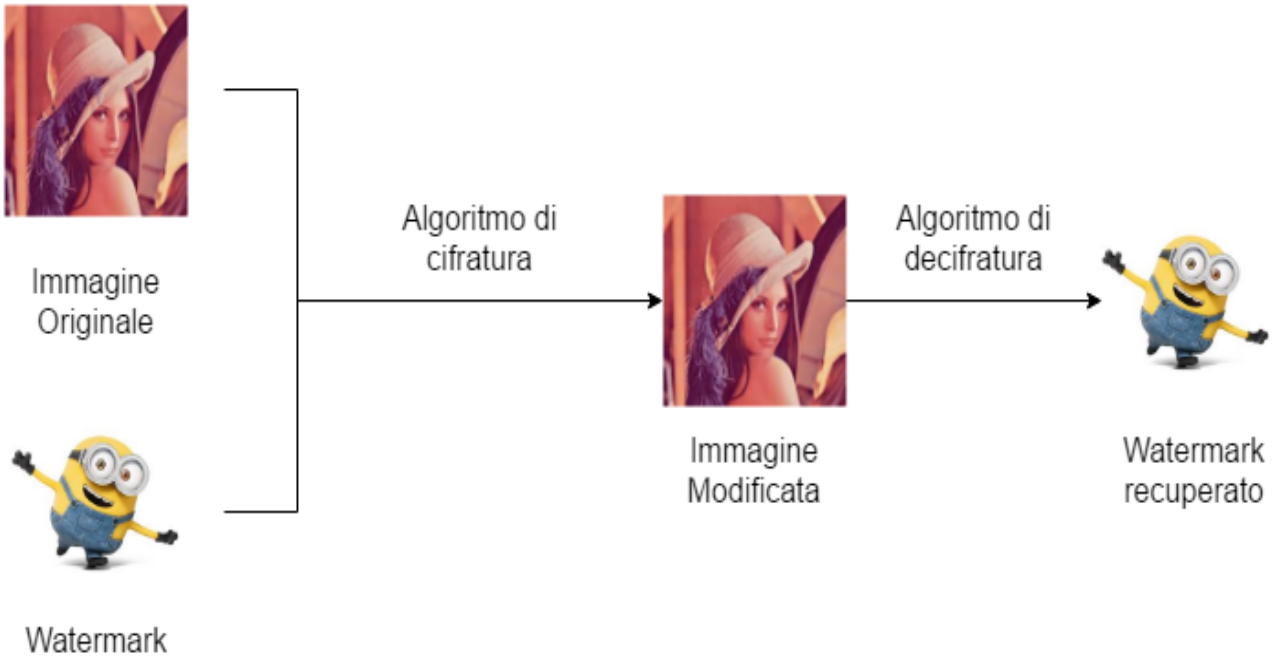

Figura 4: Processo di inserimento e estrapolazione di un watermark all'interno di un'immagine

Per prevenire attacchi, gli algoritmi devono soddisfare un certo numero di requisiti. L'integrità dell'immagine originale, ad esempio, non deve essere modificata dalla prospettiva dell'occhio umano, questo perché, se il watermark fosse visibile, un malintenzionato potrebbe estrarlo o distruggerlo.
Un altro problema tipico dei questo studio è stato riscontrato con la possibilità di una compressione o ottimizzazione dell'immagine da parte del social network. Questa potrebbe essere scalata, ruotata, ritagliata o filtrata andando a minare l'integrità del watermark. Per questo motivo gli algoritmi provati all'interno di questo elaborato hanno tutti la caratteristica di resistere, se non completamente quanto meno in parte, a questo tipo di alterazioni.

2.1 Classificazione

Gli algoritmi di watermarking, possono essere classificati in tre categorie a seconda della loro robustezza nel non perdere il watermark se soggetto a manipolazioni esterne e interne [Figura

5].

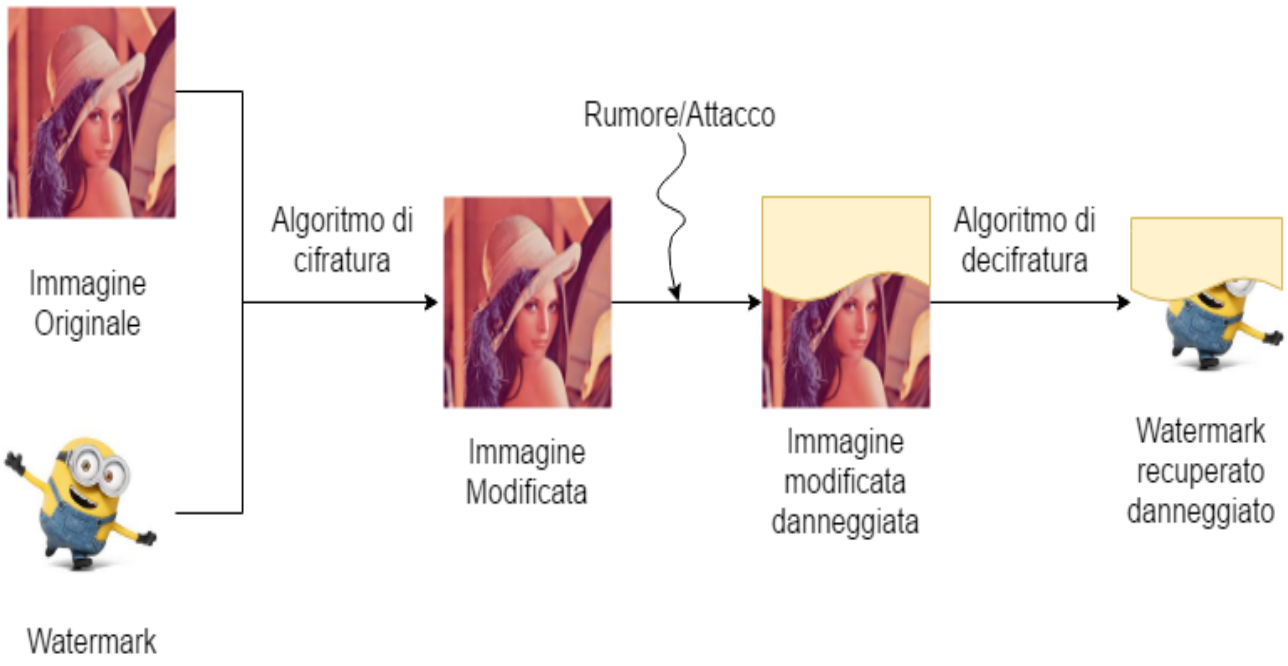

Figura 5: Esempio di attacco nel processo di creazione e estrapolazione di un watermark

2.1.1 Fragili

Questa categoria di watermark è generalmente indipendente dall'immagine data, così che un utente malevolo non può alterare il contenuto dell'immagine se non andando a manomettere la firma stessa. Il processo di autenticazione, consiste nel individuare la distorsione creata dal watermark di modo da individuare la regione dell'immagine che è stata modificata [Figura 6].
Il problema più grande dell'utilizzo di questo approccio risiede nella difficoltà di distinguere tra un attacco benevolo e uno malevolo. Per esempio, la maggior parte degli algoritmi fragili considerano un'immagine compressa come un'immagine che ha perso o alterato il proprio watermark anche se la semantica dell'immagine è rimasta inalterata.
Per questo motivo non è possibile utilizzare questa tecnica come punto di riferimento per i test di questa tesi, in quanto i social network applicano una compressione o ottimizzazione delle immagini.

Figura 6: Esempio di utilizzo di un algoritmo di watermarking fragile

2.1.2 Semifragili

Un algoritmo di watermarking appartenente a questa categoria è in grado di discriminare tra una manipolazione malevola dell'immagine, come ad esempio l'inserimento o la rimozione di una parte significativa della foto e una operazione globale che preserva la semantica del contenuto dell'immagine (compressione o ridimensionamento) [15].
Gli algoritmi semifragili sembrano quindi i migliori applicabili al campo dei social network, in quanto le immagini sono generalmente trasmesse e salvate in una forma compressa [Figura 7].

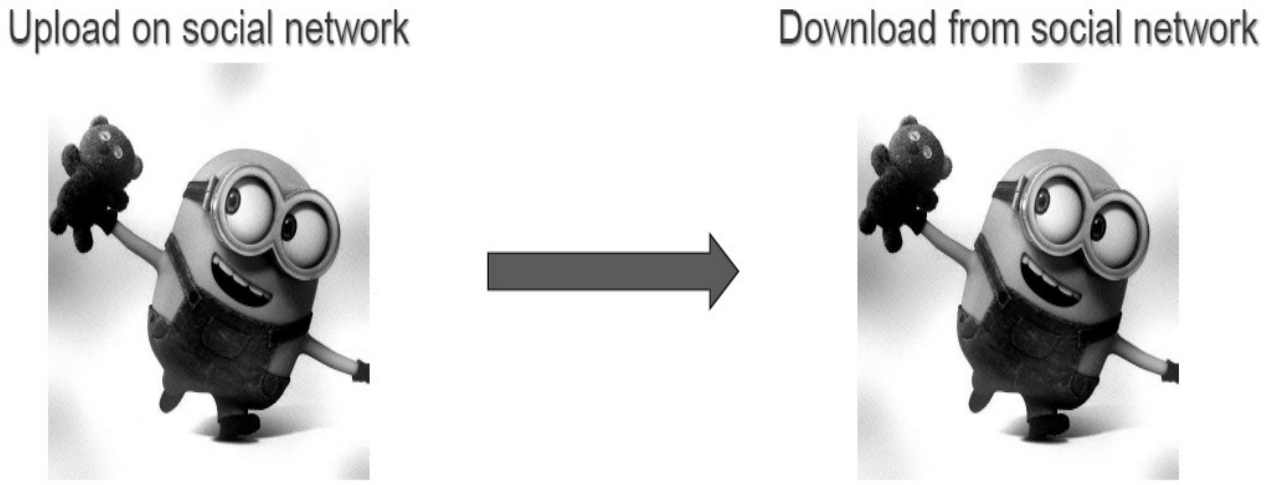

Figura 7: Esempio di utilizzo di un algoritmo di watermarking semi-fragile applicato ad un social network

2.1.3 Robusti

Questo tipo di watermark non può essere alterato facilmente in quanto resiste alla maggior parte degli attacchi di elaborazione del segnale. Questa tecnica permette di mantenere la firma digitale inserita all'interno del segnale in maniera permanente, in modo che, chiunque tenti di rimuovere o alterare la firma vada in contro al degrado della qualità del segnale stesso.
Questo watermark è utilizzato solitamente per assicurarsi la protezione del copyright del segnale, quindi non è adatto allo scopo di questa tesi, in quanto il segnale deve rimanere intatto o andrebbe a perdere di qualità nel momento in cui i social network andassero a effettuare una qualche alterazione su questo.

2.2 Proprietà

Il watermark può essere utilizzato per diversi scopi e quindi deve essere in grado di soddisfare determinate proprietà a seconda dell'utilizzo. Di seguito viene riportato un elenco dettagliato, andando a sottolineare i tratti presenti negli algoritmi di watermarking utilizzati qualora sia stato possibile effettuare una scelta sul tipo di proprietà applicabile.
Generalmente le proprietà comuni a tutti i watermark sono le seguenti:

- il legittimo proprietario o un'autorità di controllo deve poter estrarre facilmente le informazioni del watermark senza alterare le proprietà del file;
- il recupero della firma digitale deve provare l'identità del proprietario senza incorrere in ambiguità;

- la firma deve essere inserita all'interno del segnale da proteggere, per avere così una maggiore robustezza e quindi sicurezza contro le manomissioni.

2.2.1 Privati o pubblici

Il file può essere visibile a tutti gli utenti o solo a una determinata categoria di questi. In questo secondo caso si parla di watermark privato, cioè quando l'utente conosce a priori il contenuto del file a cui è stato applicato il watermark e ne possiede l'originale senza la firma digitale applicata.
Se invece non abbiamo bisogno di conoscere il contenuto dell'immagine per estrarre la firma digitale e non possediamo il file originale non marchiato, stiamo utilizzando un watermark pubblico. Questo ultimo caso è anche più soggetto all'identificazione e all'alterazione della firma. Il metodo risulta comunque utile quando il proprietario di un documento vuole rendere fruibile a chiunque la capacità di individuarlo.
In questo studio, gli algoritmi utilizzati, applicano watermark privati, in quanto, si vuole conoscere il contenuto del file marchiato ma non lo si vuole rendere fruibile a chiunque.

2.2.2 Ciechi o non ciechi

I watermark possono essere ciechi se per verificarne la presenza non si necessita del file originale, mentre sono detti non ciechi in caso contrario. Solitamente gli algoritmi che utilizzano watermark non ciechi sono più robusti, in quanto l'utente malevolo ha bisogno per forza dell'immagine originale per estrarre la firma digitale e inoltre solo il proprietario del file può dimostrare la presenza del marchio.
Per questa ricerca sono state adottate entrambe le proprietà, in quanto per alcuni degli algoritmi testati, non è stato necessario l'utilizzo dell'immagine originale. Questi estrapolano la firma digitale mediante dei punti chiave all'interno dell'immagine dove l'algoritmo è certo di averla inserita.

2.2.3 Invisibili

Anche se il watermark modifica solo una minuscola porzione dell'immagine, questa viene comunque degradata in quei punti. Questo degrado deve essere quindi il più piccolo possibile di modo da non alterare la percezione che l'occhio ha dell'immagine e quindi limitare il più possibile la capacità del malintenzionato di individuare il watermark.
Il grado di alterazione può essere deciso dal proprietario del file, il quale può sceglierlo solitamente passando da forti alterazioni, per aumentare la garanzia di robustezza della firma digitale proteggendola in maniera più efficace da eventuali attacchi, a deboli alterazioni, che non degradano l'immagine e in alcuni casi sono altrettanto efficaci nella sicurezza se il watermark rimane comunque invisibile all'utente [Figura 3].
Sono stati utilizzati nei test algoritmi che spaziano dalle piccole alle grandi alterazioni, in modo da dare risultati il più possibile completi su tutte le scelte possibilmente attuabili da un utente.

2.2.4 Codificati a chiave

Ad ogni firma digitale è associata una particolare sequenza di bit detta chiave, questa serve sia per produrre il segnale di watermark che per riconoscerlo all'interno di un documento. La chiave è privata e caratterizza univocamente il legittimo proprietario del documento, solo chi è in possesso della chiave è in grado di dimostrare la presenza della firma nel file.
Il numero possibile di queste deve essere tale da garantire una ottima robustezza dell'algoritmo, cioè il watermark può essere danneggiato solo se prima si riescono a decifrare tutte le chiavi.

2.2.5 Efficienti statisticamente

Un file firmato con un watermark deve essere facilmente riconoscibile se si è a conoscenza della relativa chiave. La probabilità che questa nella fase di riconoscimento venga rifiutata, pur essendo corretta, deve essere sufficientemente bassa o inesistente.
Gli algoritmi di watermarking testati hanno sempre permesso la corretta visualizzazione del watermark. Per questo motivo si può ipotizzare che è sempre possibile riconoscere la firma digitale anche se in parte degradata dalle alterazioni effettuate dai social network.

2.2.6 Invisibili statisticamente

Possedere diverse immagini firmate tutte con la stessa chiave deve rendere impossibile ad un utente malevolo l'accesso alla firma digitale impedendogli di rimuoverla. Per questo motivo gli algoritmi di watermarking devono essere in grado di generare segnali di watermark diversi per ogni immagine. Il riconoscimento della chiave all'interno dell'immagine, da parte di utenti non autorizzati, deve essere impossibile pur avendo tanti campioni su cui testare l'attacco.

2.2.7 Multipli

Deve essere possibile inserire un elevato numero di watermark all'interno dello stesso file, ognuna di queste firme digitali può essere riconosciuta mediante la corrispondente chiave. I questo studio, tutti gli algoritmi possono essere inseriti più volte nell'immagine, aumentandone la robustezza, ma diminuendone la possibilità di recuperare integro l'intero watermark dopo il caricamento su di un social network.

2.2.8 Robusti

Sulle immagini possono essere effettuate numerose operazioni per migliorare la loro qualità o per comprimere la loro dimensione. I watermark devono essere tali da non essere danneggiate da questo tipo di operazioni, né da operazioni mirate ad alterare o cancellare la firma digitale stessa.

2.2.9 Invertibili

Il legittimo proprietario del file deve poter rimuovere il watermark in qualsiasi momento. Questa è l'unica proprietà che può venire a mancare nel caso in cui si prediliga un algoritmo di watermarking atto a garantire la robustezza e la resistenza alle aggressioni. In questo elaborato, la firma digitale inserita in un'immagine, non può più essere rimossa.
In alcuni casi, per invertibilità, si intende la possibilità di generare un falso watermark e un falso file originale che sia uguale a quello ver. Dall'inserimento della falsa firma digitale in questo modo, si ottiene un documento che è perfettamente (invertibilità) o solo percettibilmente uguale (quasi invertibilità) a quello reale.
Se il watermark vuole essere una prova inconfutabile per l'applicazione del copyright, allora questo deve essere non invertibili o quasi non invertibili.

2.3 Principali algoritmi di watermarking

Esistono diversi tipi di algoritmi di watermarking per cifrare il watermark, questi solitamente si distinguono tra loro a seconda di come nascondono la firma digitale all'interno di un'immagine ma anche da come e quanto sfruttino le proprietà dei watermark. Infatti, alcuni algoritmi hanno ad esempio una invisibilità maggiore di altri, mentre altri prediligono la robustezza invece che la possibilità di estrarre la firma digitale dall'immagine.
Gli algoritmi più comuni di inserimento del watermark sono:

- **DWT** (Discrete Wavelet Transform);
- **DFT** (Discrete Fourier Transform);
- **SVD** (Singular Value Decomposition);
- **DCT** (Discrete Cosine Transform).

Le proprietà di questi algoritmi [1] sono solitamente combinate per aumentare la robustezza, di modo da avere un metodo sicuro di trasmissione dei dati. Combinando questi metodi le loro proprietà di resistenza agli attacchi, inoltre, vengono amplificate. Questo permette di avere non solo un watermark difficile da individuare ma anche difficile da rimuovere.
Ad esempio, le proprietà della DFT (scaling e l'invarianza allo spostamento di un'immagine), quelle del SVD (l'invarianza alla rotazione) e l'uso delle proprietà della DWT, possono essere combinate insieme per ottenere una resistenza maggiore a compressione e migliorare il passaggio di informazioni mediante un algoritmo di watermarking.
Di seguito, ogni algoritmo sarà analizzato più a fondo, per trovare il migliore da applicare a questo studio.

2.3.1 Discrete Wavelet Transform

Un wavelet (ondicella) è una rappresentazione, nello spazio e nel tempo, di un segnale mediante l'uso di una forma d'onda oscillante di lunghezza finita. Questa forma d'onda è scalata e traslata per adattarsi al segnale in ingresso. Per un'immagine viene posto un modello 2D della trasformata. Un singolo livello 2D-DWT decompone l'immagine in quattro differenti frequenze di sotto-banda (LL, LH, HL e HH). Ognuna di esse descrive modifiche locali alla luminosità e filtrano una posizione (high-pass o low-pass) dell'immagine originale in direzione sia verticale che orizzontale [Figura 8].
DWT ha molti vantaggi rispetto agli altri algoritmi grazie alla sua abilità di rappresentare le immagini in entrambi i domini (spaziali e di frequenza) simultaneamente e di separare le differenti componenti della frequenza di un'immagine.
La proprietà di separazione della frequenza di questo algoritmo di watermarking è utilizzata nelle immagini per inserire un watermark in differenti sotto-bande di una frequenza, ma in questa tesi è completamente inutile in quanto, mediante la compressione, la sotto-banda viene modificata perdendo la firma digitale.

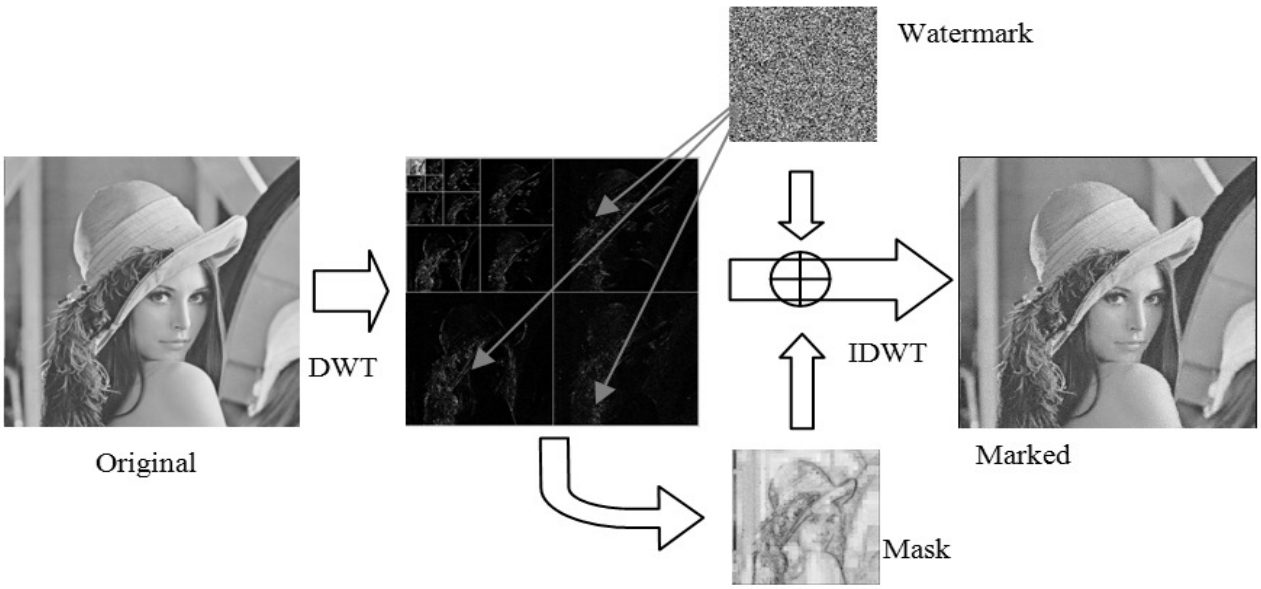

Figura 8: Esempio di applicazione di un algoritmo di watermarking DWT su di un'immagine[1]

2.3.2 Discrete Fourier Transform

DFT trasforma un'immagine dal dominio spaziale in una dominio di frequenza. La principale proprietà di questo metodo è quella di implementare un vettore binario di una certa lunghez-

[1]IDWT è utilizzato per definire l'algoritmo DWT inverso, utilizzato per decifrare il watermark

za che viene generato in maniera pseudocasuale (PRND) nello spettro dell'immagine. Questo viene inserito all'interno del coefficiente di grandezza della trasformata di Fourier, nascosto nell'immagine e disposto a cerchio, con un certo raggio, partendo dal centro di questa che viene poi applicato all'immagine [Figura 9].
Il coefficiente di grandezza è utilizzato per la rilevazione di rumori e potenza nella misurazione dello spettro di un'immagine, aumentare o diminuire questo valore permette di ridurre la varianza dello spettro.
Usando questo algoritmo il coefficiente ed il periodo di un'immagine nel dominio delle frequenze può essere separato. Per questo uno dei vantaggi principali della DFT è la resistenza agli attacchi geometrici, d'altro canto però, qualsiasi variazione all'interno dei bit di un'immagine che utilizza questo metodo danneggia il watermark e per questo motivo non è adatto agli esperimenti.

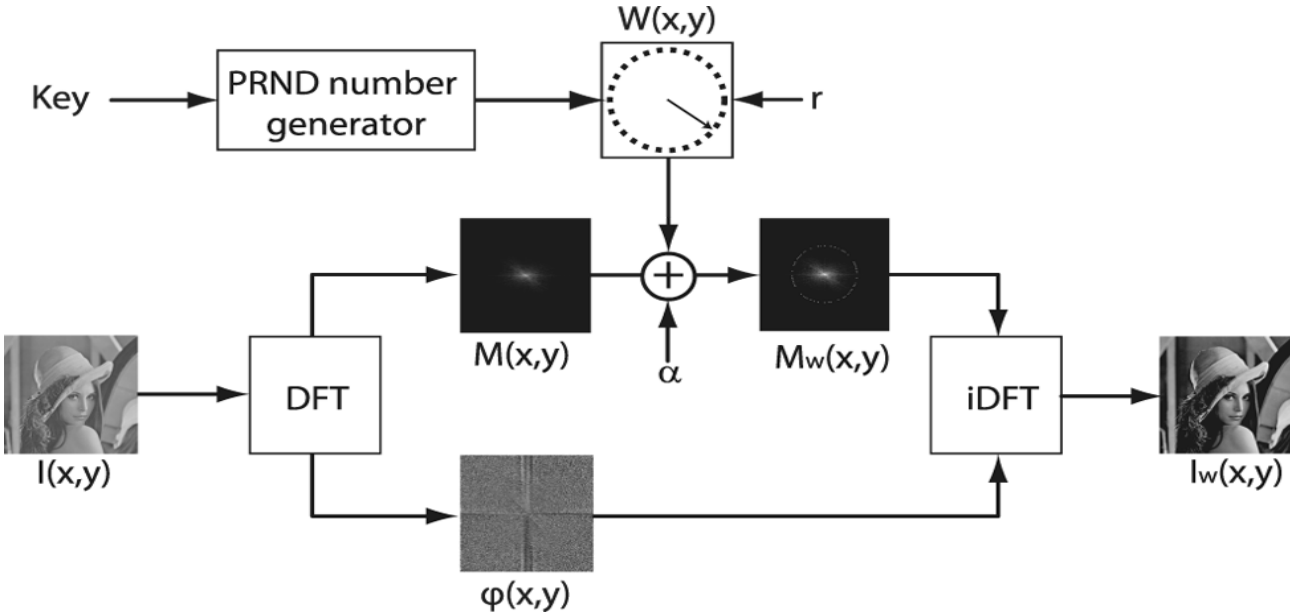

Figura 9: Esempio di applicazione di un algoritmo di watermarking DFT su di un'immagine[2]

2.3.3 Singular Value Decomposition

L'uso della SVD nell'ambito degli algoritmi di watermarking permette di cifrare il watermark all'interno di una matrice del dominio delle frequenze o del coefficiente del dominio spaziale di un'immagine invece che inserire la firma digitale direttamente all'interno dei coefficienti stessi di questa. [Figura 10].
Dato che la matrice non cambia a seconda dei differenti tipi di attacchi, il watermarking può

[2]IDFT è utilizzato per definire l'algoritmo DFT inverso, utilizzato per decifrare il watermark

essere estratto con una piccola perdita di informazione permettendo di comunicare in sicurezza e segretamente dei dati. Tuttavia, questo algoritmo da solo è completamente inutilizzabile e per questo motivo solitamente è applicato agli altri algoritmi di watermarking come un'ottimizzazione.
In questo caso di studio, la SVD è stata utilizzata negli algoritmi con lo scopo di irrobustire la difficoltà di attacco da parte di utenti malevoli.

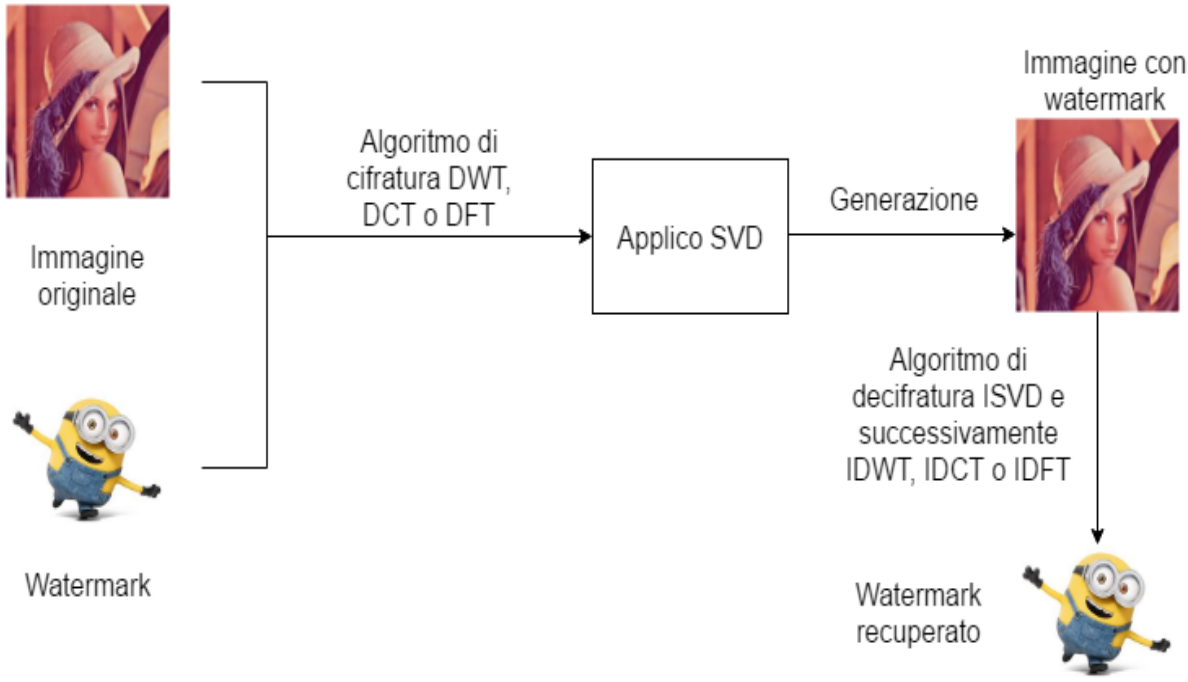

Figura 10: Esempio di applicazione di un algoritmo di watermarking SVD su di un'immagine ottimizzando l'algoritmo già presente[3]

2.3.4 Discrete Cosine Transform

La DCT [18] è una delle funzioni di trasformazione più popolari usate per la processazione di segnali, questa trasforma un segnale, dal dominio spaziale ad uno di frequenza. Un'immagine trasformata da un algoritmo di DCT è solitamente suddivisa in blocchi m x n non sovrapposti. Più in generale, questi blocchi sono formati da componenti 8 x 8 in modo da adattarsi alla composizione in byte di un'immagine.
Una delle migliori applicazioni del DCT utilizza una permutazione di lettura a zigzag basata sulla distribuzione dell'energia, sia dall'alto verso il basso sia dal basso verso l'alto. L'occhio umano è più sensibile al rumore nelle basse frequenze che in quelle alte, per questo l'energia

[3]ISVD è utilizzato per definire l'algoritmo SVD inverso, utilizzato per decifrare il watermark

dell'immagine originale è concentrata nel range delle basse frequenze.
Un watermark nascosto nella banda delle alte frequenze dovrebbe essere scartato dopo una compressione lossy, inoltre, dato che la firma digitale è sempre cifrata nel range della sotto-banda della bassa frequenza dell'immagine originale, permette di ottenere un watermark resistente a compressione.
Il problema maggiore di questo algoritmo è dato dal fatto che i coefficienti utilizzati sono inserti sempre con lo stesso metodo a zigzag, per cui il watermark, anche se invisibile all'occhio umano, può essere rintracciato nel codice dell'immagine. Motivo per cui, solitamente, per aumentare la robustezza agli attacchi di questo algoritmo, si utilizza anche un algoritmo di watermarking SVD [Figura 11].
L'elaborato si focalizza su questo tipo di algoritmo per i vari test, in quanto resiste alla compressione con piccole perdite e inoltre, se vi si aggiunge a questo l'algoritmo di cifratura SVD, riesce anche ad avere una certa robustezza agli attacchi malevoli degli utenti.

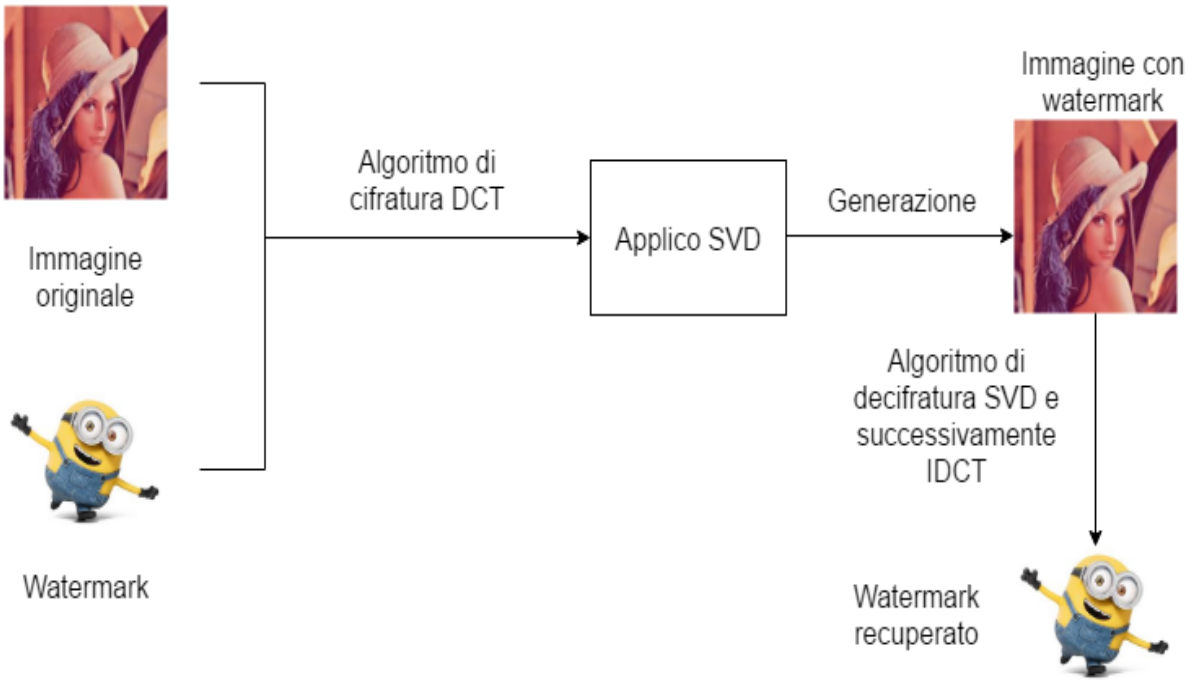

Figura 11: Esempio di applicazione di un algoritmo di watermarking DCT su di un'immagine ottimizzato tramite l'algoritmo di SVD[4]

[4]IDCT è utilizzato per definire l'algoritmo DCT inverso, utilizzato per decifrare il watermark

3 Social network e immagini condivise

In questo capitolo, lo studio, si focalizza sull'analisi dei social network. Vengono effettuati dei test di controllo sul contenuto delle immagini condivise al momento del caricamento e dello scaricamento dai SN e su come vengano gestite internamente da questi elaborando le nostre informazioni personali e geografiche.
Per prima cosa verranno analizzate le normative sulla privacy vigenti nei social network, analizzando le condizioni d'uso di due di questi, Facebook e Twitter. In questo modo è possibile verificare che lo studio di questo elaborato sia realmente necessario a proteggere gli utenti o se la privacy di questi venga già tutelata dai SN.
Dopo aver controllato quali tipi di algoritmi di watermarking sono più adatti per effettuare i test sulle immagini, è opportuno verificare anche come i social network le gestiscano, in modo da capire se le foto possono essere in qualche modo alterate, indebolendo la robustezza degli algoritmi algoritmi.
Per questo motivo lo studio si prefigge il compito di ad analizzare nel dettaglio in questo capitolo come vengano modificate le immagini una volta caricate sul social e se queste subiscono una alterazione tale da restituire un'immagine, diversa completamente o solo in parte, rispetto a quella originale.
Altro scopo della tesi è di verificare la possibilità che il social network stesso applichi una qualche forma di watermark. La verifica permetterebbe così di analizzare un algoritmo di watermarking già esistente.
Analizzarlo permetterebbe di comprendere più a fondo come un algoritmo si può adattare ai problemi legati alla compressione o allo scaling. Oltretutto sarebbe fonte di un'ulteriore indagine sulle modalità con cui viene applicato ad un'immagine (tempo, nome del profilo, ecc. . .).
Per controllare che non siano state apportate modifiche alle immagini, altri campi che devono essere controllati sono poi quelli legati alla geolocalizzazione e ai CDN (Content Delivery Networks), perché, viene logico pensare, che se un'immagine viene caricata su un social network e viene scaricata dall'altra parte del mondo, questa potrebbe contenere delle informazioni sulla sua posizione di caricamento e scaricamento.
Per effettuare i test, la ricerca è stata divisa in step che andranno a definire prima le regole su come eseguire gli esperimenti sui social, regolandone la risoluzione e il numero di foto su cui effettuare il test e poi sulle proprietà da analizzare [Figura 12]. Questo per capire se una certa immagine è stata in qualche modo alterata dai social network.
I social network che sono stati testati sono Facebook, Google+, Twitter, Telegram, Tumblr, Vk, WeChat, Linkedin, Whatsapp, Pinterest, Gmail, Wordpress.
Per altri, è necessario fare alcune considerazioni sul motivo per cui sono stati scartati dalle prove in quanto non idonei:

- **Instagram**: i test vengono comunque eseguiti anche se le immagini vengono scaricate con un tool online, in quanto, il sito non permette di caricare manualmente le foto se non da smartphone.

- **WeChat**: le immagini vengono trattate come file, motivo per cui i risultati rimangono sempre invariati rispetto all'immagine originale. I test vengono comunque effettuati su questo SN per verificare che, anche se trattati come file, le immagini non subiscano alterazioni.
- **Gmail**: funziona esattamente come WeChat e per lo stesso motivo non è stato preso in considerazione.
- **YouTube**: non è stato utilizzato negli esperimenti in quanto non permette di condividere delle foto in alcun modo.
- **Reddit**: non è stato incluso nelle prove in quanto l'unico modo di inserire una foto è tramite una condivisione da un altro social network e non direttamente.
- **Snapchat**: non viene considerato, in quanto, le immagini caricate tramite questo social network, non possono essere scaricate. L'unico modo per scaricarle, inoltre, è prima condividerle attraverso altri SN e poi scaricarle da questi. Eseguendo questa operazione, però, le immagini acquisiscono le proprietà e caratteristiche di una foto caricata direttamente su questi. Perciò i test saranno effettuati sui social network in cui queste sono state caricate.
- **Wordpress**: non è stato utilizzato perché l'immagine viene trattata come Wechat e Gmail.

Alcuni social network inoltre, offrono una scelta su come trattare l'immagine al momento del caricamento, se come file o come media, per cui nell'elaborato è stato sempre scelto come opzione di trattare una foto come media.

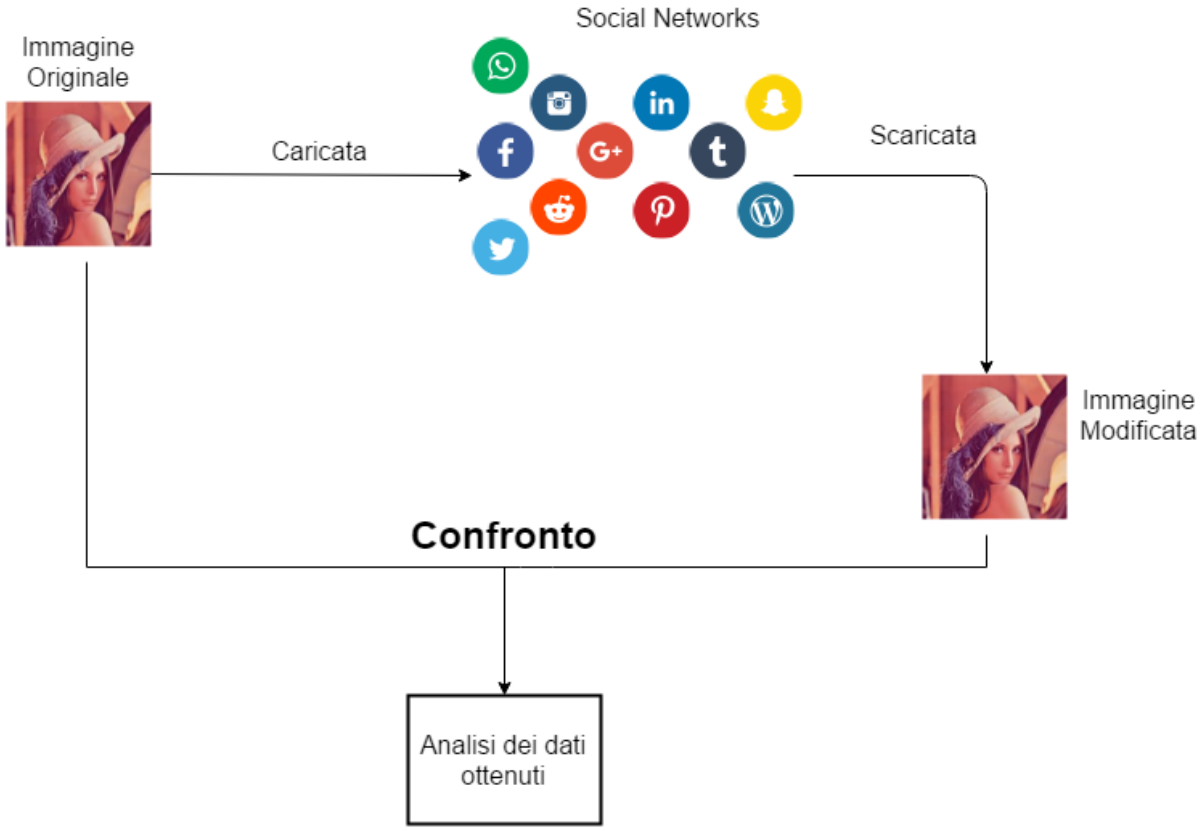

Figura 12: Metodo con cui sono stati effettuati i test sulle immagini

3.1 Considerazioni sul copyright per immagini

Prima di iniziare con i test, è necessario effettuare un controllo relativo alla necessità di utilizzare un watermark nei social network. Se questi tutelassero la privacy dei propri utenti, non sarebbe necessario alcun algoritmo di watermarking per proteggere una foto, in quanto, essi garantirebbero il copyright per le immagini caricate.
I SN sono autorizzati a disporre delle immagini, caricate da un utente, secondo la legge sui diritti d'autore e attraverso le condizioni di utilizzo che cambiano da social network a social network [17].
La legge italiana disciplina il diritto d'autore prevalentemente mediante la Legge 22 aprile 1941, n. 633 e successive modificazioni, e dall'artt. 2575 e seg. del Codice Civile. Secondo queste direttive, costituiscono oggetto del diritto d'autore "le opere dell'ingegno di carattere creativo, appartenenti al mondo della letteratura, della musica, del teatro e della cinematografia, delle arti figurative, dell'architettura, della scienza, sotto qualsiasi forma ed espressione". Inoltre l'autore ha il diritto esclusivo di pubblicare l'opera e di utilizzarla economicamente anche me-

diante la cessione dei diritti mantenendone comunque la paternità.
Nel caso specifico delle immagini, ad un fotografo, per venti anni dalla data di realizzazione della fotografia, spettano i diritti esclusivi di riproduzione, diffusione e spaccio ex art. 88 l. 633/41, salvo il caso che la fotografia sia stata commissionata nell'ambito di un contratto di lavoro (nel qual caso diviene titolare dei diritti il datore di lavoro).
Ogni esemplare di foto caricata sul web, deve contenere l'indicazione di chi detiene i diritti di utilizzazione economica (fotografo o datore di lavoro o committente) nonché la data dell'anno di produzione della fotografia. Qualora tali informazioni manchino, ai sensi dell'art. 90 comma 2 della l. 633/41, la riproduzione non è considerata abusiva (salvo che il fotografo provi la malafede del riproduttore).
Dunque, l'utilizzo su internet d'immagini fotografiche trovate in altri siti, contenenti le indicazioni sopra riportate, risulta possibile solo qualora si sia ottenuta l'autorizzazione del fotografo (ovvero del datore di lavoro o del committente nei casi tali soggetti siano i detentori dei diritti sulle immagini fotografiche). Invece, se le suddette indicazioni non sono presenti (caso abbastanza frequente) la riproduzione può avvenire liberamente, senza necessità di autorizzazione alcuna.
Detto questo, non è vietato da nessuna legge italiana scaricare una foto su di un profilo privato in un social network se questo è privo di un indicazione sul diritto d'autore. In questo caso le norme che regolano l'utilizzo delle immagini sono dettate dalle condizioni d'uso relative al singolo SN. Di seguito vengono analizzati nel dettaglio le condizioni d'uso di Facebook e Twitter.
L'utente è responsabile del proprio utilizzo dei servizi, dei contenuti postati sui servizi e di ogni conseguenza derivante da tali azioni. I social network non sono tenuti al monitoraggio o al controllo dei contenuti postati. Su Twitter si legge che qualunque utilizzo dei materiali postati tramite i servizi o ottenuti dall'utente tramite i servizi, si intendono a rischio esclusivo dell'utente. Non viene rilasciata approvazione o garanzia in ordine alla veridicità o affidabilità di quanto postato. Per quanto concerne i contenuti coperti da diritti di proprietà, l'utente concede a Facebook una licenza non esclusiva, trasferibile, che può essere concessa come sottolicenza, libera da royalty e valida in tutto il mondo, per l'utilizzo di qualsiasi contenuto pubblicato o in connessione con Facebook.
Questa licenza termina nel momento in cui si elimina l'account o il contenuto presente nell'account, salvo ulteriore condivisione con terzi. E se eliminare foto e video è semplice come vuotare il cestino del computer, è possibile che i contenuti rimossi, seppur non visibili, vengano conservati come copie di backup per un determinato periodo di tempo. Facebook chiede ai propri utenti di non pubblicare comunicazioni commerciali, di non caricare virus o altri codici dannosi, di non cercare di ottenere informazioni di accesso o accedere ad account di altri utenti, di non denigrare, intimidire o molestare altri utenti, di non pubblicare contenuti minatori, pornografici, con incitazioni all'odio o alla violenza, con immagini di nudo o di violenza forte o gratuita, di non sviluppare o utilizzare applicazioni di terzi con contenuti correlati all'alcol, a servizi di incontri o comunque rivolti a un pubblico adulto senza le dovute restrizioni di età, di

non usare Facebook per scopi illegali, ingannevoli, malevoli o discriminatori.
Tra gli oneri dell'utente di Facebook figurano inoltre l'impegno a non fornire informazioni personali false o creare un account per conto di un'altra persona senza autorizzazione. L'utente di Twitter manterrà i propri diritti sui contenuti che posterà e renderà disponibili sui servizi e con l'invio, la pubblicazione o visualizzazione concede a Twitter una licenza mondiale, non esclusiva e gratuita (con diritto di sublicenza) per l'utilizzo, copia, riproduzione, elaborazione, adattamento, modifica, pubblicazione, trasmissione, visualizzazione e distribuzione dei contenuti con qualsiasi supporto o metodo di distribuzione, sia attualmente disponibile sia sviluppato in seguito. L'utente accetta che i propri contenuti possano essere condivisi, distribuiti o pubblicati dai partner di Twitter. Gli eventuali utilizzi aggiuntivi da parte di costoro potranno avvenire senza il pagamento di un corrispettivo all'utente.
L'utente sarà responsabile del proprio utilizzo dei contenuti forniti e di ogni conseguenza che ne possa derivare, ivi incluso l'utilizzo da parte di altri utenti o partner terzi di Twitter. Vietato usare Facebook in caso di condanna per crimini sessuali. È vietato pubblicare o eseguire azioni che violino i diritti di terzi. Qualora tali condizioni non siano rispettate Facebook si riserva il diritto di rimuovere tutti i contenuti o le informazioni che gli utenti pubblicano. L'utente di Twitter si dichiara consapevole del fatto che l'utilizzo dei servizi potrà esporlo a contenuti offensivi, dannosi inadeguati o a post ingannevoli. Si riserva il diritto, ma non ha l'obbligo, di rimuovere o rifiutare la distribuzione di contenuti, di sospendere o chiudere utenze, di richiedere la restituzione di alcuni nomi utente; si riserva altresì il diritto di accedere, leggere, conservare e divulgare le informazioni che ritenga ragionevolmente necessarie per conformarsi a ogni legge, regolamento o procedimento, imporre l'osservanza delle condizioni, individuare, impedire o affrontare frodi o problematiche inerenti la sicurezza, proteggere i diritti, la proprietà o la sicurezza di Twitter, degli utenti e del pubblico.
Quando Facebook riceve una notifica di presunta violazione della proprietà intellettuale, procede alla rimozione o alla disattivazione dell'accesso ai contenuti oggetto della violazione. In caso di violazione reiterata dei diritti di proprietà intellettuale di terzi, l'account verrà disabiliato.
Di rilievo la circostanza che il social network declini ogni responsabilità per le eventuali condotte illecite sia online che offline posto che non controllano né guidano le azioni degli utenti. Twitter precisa che qualora il suo utente ritenga che i propri contenuti siano stati riprodotti in violazione di diritto d'autore, dovrà fornire una firma manuale o elettronica del titolare del diritto d'autore o del soggetto autorizzato ad agire per suo conto. Anche Twitter si riserva il diritto di rimuovere i contenuti che si presume costituiscano violazione senza alcun preavviso e laddove opportuno in caso di condotta reiterata, chiuderà l'account dell'utente.
Quindi, agendo opportunamente sul livello e sulle impostazioni del proprio profilo, è possibile limitare l'accesso e la diffusione dei propri contenuti, sia dal punto di vista soggettivo che da quello oggettivo [Figura 13]. È peraltro nota agli utenti di Facebook l'eventualità che altri possano in qualche modo individuare e riconoscere le tracce e le informazioni lasciate in un determinato momento sul sito, anche a prescindere dal loro consenso(tagging).

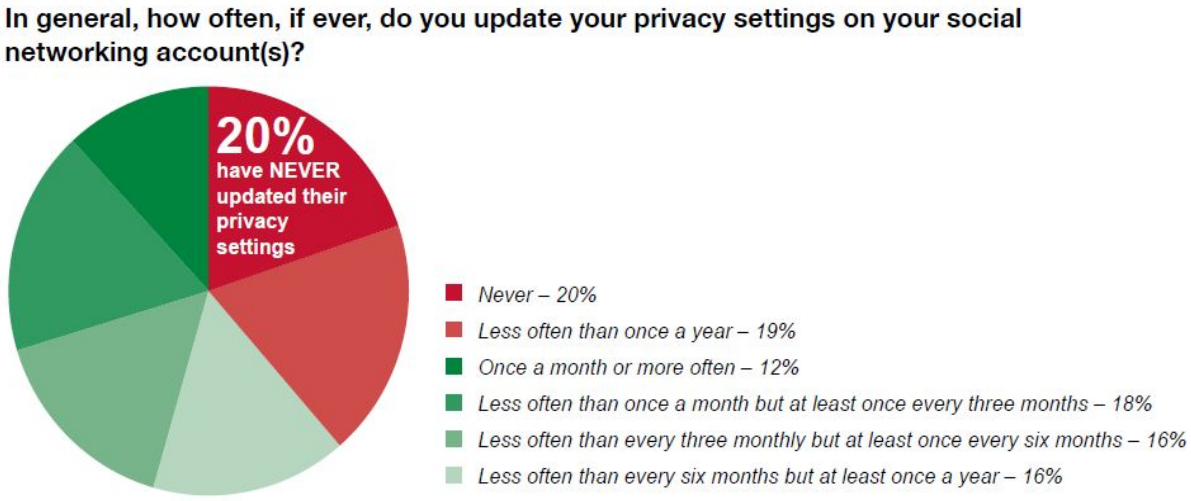

Figura 13: Protezione personale della privacy sui social network da parte degli utenti

Inoltre, i gestori del sito, pur reputandosi proprietari dei contenuti pubblicati, declinano ogni responsabilità civile e/o penale ad essi relativa.
In definitiva, coloro che decidono di diventare utenti di Facebook o Twitter sono ben consci non solo delle grandi possibilità relazionali offerte dal sito, ma anche delle potenziali minacce ai contenuti che vi inseriscono. I social network quindi non tutelano la privacy dei contenuti degli utenti in generale, motivo per cui lo studio di questo elaborato è da considerarsi adeguato a questo scopo.

3.2 Gestione della risoluzione

I social network gestiscono le immagini che carichiamo con specifiche risoluzioni scalando automaticamente le più grandi alla dimensione standard. Per questo motivo i test sono stati effettuati in base alla singola risoluzione standard richiesta dai SN utilizzando poi immagini sia più grandi che più piccole di questa. In questo modo si è potuto controllare, se le alterazioni subite dalle immagini, fossero applicate a una specifica o a tutte le risoluzioni.
Per questo motivo tutte le prove successive saranno svolte utilizzando i seguenti criteri:

- Per ogni social network sono state utilizzate 30 immagini;
- Le immagini sono 10 per tre diverse risoluzioni: "grandi" per quelle con risoluzione maggiore della standard, "standard" per quelle con la stessa risoluzione di quella richiesta dal social network e "piccole" per le più piccole di quelle richieste;
- Ogni profilo su cui è stato effettuato un test utilizza le stesse immagini;

- I confronti sono stati effettuati tra stessa immagine originale (quella data in input al social network al caricamento) e stessa immagine modificata (quella restituita dal sito al momento dello scaricamento) di ogni profilo e tra stesse immagini modificate di profili diversi per vedere se i confronti cambiavano per via di un diverso profilo utente;
- I test sono stati svolti mediante il codice in Appendice 1.

Nella Tabella 1 vi è un riepilogo delle risoluzioni usate per le immagini per ogni social network, da notare che a WeChat non è stata data una risoluzione, in quanto tratta le immagini come file, motivo per cui accetta una qualsiasi immagine con dimensione minore di 2 MB.
I test sono stati eseguiti utilizzando immagini più grandi della dimensione standard con una dimensione predefinita di 4096x2560 e con immagini più piccole della standard con dimensione di 200x200.

Social network	Standard
Facebook	2048x1280
Instagram	1080x1080
Google+	2048x1280
Twitter	564x376
Whatsapp	1600x1000
Pinterest	564x352
Telegram	1280x800
Linkedin	531x332
Tumblr	1280x800
Vk	1280x800
QQ	2048x1280
WeChat	*[5]

Tabella 1: Risoluzioni utilizzate per i test per ogni social network

3.3 Compressione al caricamento delle immagini

Quando un'immagine viene caricata su un social network, questa viene sottoposta ad una alterazione, che non è necessariamente dovuta alla dimensione superiore a quella standard dell'immagine, ma viene applicata, in alcuni social, anche alle foto che mantengono una risoluzione uguale o inferiore a quella imposta dal sito.
Per capire se questa alterazione era dovuta alla compressione che, per via di un algoritmo interno, i social network applicano alle immagini che non rispettano lo standard dimensionale, è stato eseguito un confronto sulla dimensione dei file tra stesse immagini caricate sul SN e quelle

[5] Le immagini trattate come file non hanno una risoluzione standard

poi scaricate.
I risultati sono stati confrontati utilizzando un programma che prendeva l'immagine di input e di output e restituiva, in percentuale, la differenza di dimensione tra i due file. Questo ha permesso di stilare la Tabella 2 nel quale viene proposta la media statistica della dimensione dei file per ogni social network e per ogni risoluzione delle immagini.
Quello che si nota è che stranamente, oltre ad effettuare una compressione, alcuni siti aumentano la grandezza dei file nel caso di immagini con risoluzione standard e piccole (campi della Tabella 2 con valore negativo). Questo non è dovuto alla compressione ma bensì a una ottimizzazione [16] che viene applicata dai social network sulle immagini e che permette di restituire dopo il caricamento una foto con qualità leggermente migliore dell'originale.
L'algoritmo utilizzato per migliorare le immagini è chiamato "Firefly" [8]. Questo utilizza una ottimizzazione dello sciame di particelle (PSO) applicato ad un'immagine in questo caso, così da perfezionare alcuni punti particolari di questa ultima.
Un primo risultato sulla compressione o l'ottimizzazione lascia già intendere che i social network applichino delle modifiche alle immagini, motivo per cui sicuramente gli algoritmi di watermarking analizzati debbono resistere a questo tipo di alterazioni.

Social network	Grandi	Standard	Piccole
Facebook	80.58%	7.06%	-3.83%
Instagram	93.27%	-0.58%	-0.13%
Google+	67.01%	0%	0%
Twitter	97.06%	4.71%	4.19%
Whatsapp	73.34%	-48.07%	-44.39%
Pinterest	97.70%	-0.30%	-0.28%
Telegram	87.37%	0%	0%
Linkedin	97.39%	-26.68%	-24.60%
Tumblr	79.86%	0%	0%
Vk	85.57%	0%	0%
QQ	78.24%	4.66%	-2.41%
WeChat	0%	0%	0%

Tabella 2: Statistica delle differenti grandezze delle immagini dopo che queste sono state scaricate dai social network

3.4 Correlazione fra nomi al download tra diversi profili

Successivamente verranno analizzati nel dettaglio i metadata che compongono un'immagine. Per il momento però, conviene soffermarsi sul nome di un'immagine, unico campo dei metadata visibile dall'utente senza utilizzare dei software di visualizzazione dei dati EXIF.
Controllare che il nome al caricamento e quello allo scaricamento dal social network sono dif-

ferenti è utile alla ricerca di un qualche watermark applicato da questi. Uno dei punti in cui potrebbe essere inserita una firma digitale potrebbe essere il nome che infatti, che anche se modificato in qualsiasi momento da un utente malevolo, lascia comunque un certo tipo di traccia all'interno del SN confermando la proprietà di un'immagine di un utente.
Il test è stato eseguito mediante un programma che confrontava i nomi tra stessa immagine scaricata e caricata sullo stesso social network, se questi erano diversi li restituiva come risultato del confronto.
I dati analizzati hanno permesso di trarre la conclusione che non tutti i social network mantengono il nome del file al momento del download, i siti che lo modificano non permettono di risalire al nome originale e il criterio con cui viene generato questo nuovo nome può essere solo supposto per alcuni di questi.
I metodi utilizzati per modificare il nome sono diversi, infatti alcuni social network lo modificano utilizzando il tempo o l'ordine cronologico di caricamento delle immagini sul sito (Tumblr, Telegram), altri dipendono dal profilo con cui si caricano le immagini (Facebook, Instagram) o per altri ancora invece il nome rimane lo stesso (Google+).
Per mostrare alcuni esempi, in Tabella 3, vengono mostrati come i nomi vengano alterati dal social network.

Social network	Input	Output
Facebook	Image1	13474955_294548354266650_2575934234214542233_o
Instagram	Image1	10012626_9837926311711183_1106302974_n
Google+	Image1	Image1
Twitter	Image1	Cs9N5fpXYAA7HD6
Whatsapp	Image1	0de056a6-552e-442e-b9e4-f87becb2e243
Pinterest	Image1	Image1
Telegram	Image1	photo601575848900929547
Linkedin	Image1	871f9134-651f-4a2e-a778-ee49bcf3c67e-large
Tumblr	Image1	tumblr_o890nnF7E61vo3v9ho1_1280
Vk	Image1	UWce9Sj-Q3U
QQ	Image1	3dF67G-854D
WeChat	Image1	1452919520

Tabella 3: Esempio di nomi applicati dai social network al download

Nella Tabella 4 vengono spiegati i risultati finali ottenuti dallo studio per questo tipo di esperimento. Con *"Diversi"* si intende che ogni immagine di un profilo è differente da quella con cui viene comparata e con *"Uguali"* si intende l'opposto. A volte nelle tabelle sarà indicato una specifica risoluzione con cui il test restituisce un certo risultato, in quel caso sarà indicato ad esempio con *"Immagini grandi diverse"*.

Social network	Grandi	Standard	Piccole
Facebook	Diversi	Diversi	Diversi
Instagram	Diversi	Diversi	Diversi
Google+	Uguali	Uguali	Uguali
Twitter	Diversi	Diversi	Diversi
Whatsapp	Diversi	Diversi	Diversi
Pinterest	Uguali	Uguali	Uguali
Telegram	Diversi	Diversi	Diversi
Linkedin	Diversi	Diversi	Diversi
Tumblr	Diversi	Diversi	Diversi
Vk	Diversi	Diversi	Diversi
QQ	Diversi	Diversi	Diversi
WeChat	Diversi	Diversi	Diversi

Tabella 4: Confronto dei nomi delle immagini di ogni social network per ogni dimensione di queste tra profili diversi

3.5 Confronto fra immagini tra diversi profili

Prima di definire i test sui social network eseguiti in questo paragrafo, è importante far notare come questi seguano una forma gerarchica, partendo dalla prova più rigida sino ad arrivare a quella più generica, in modo da controllare, il più dettagliatamente possibile, ogni possibile caratteristica di un'immagine.
Per questo è stata creata una scala dei confronti partendo appunto dalla definizione più stringente di file. I questo modo se qualcosa andasse a modificare anche solo un bit dell'immagine, ci si potrebbe accorgere facilmente a che livello della scala esso appare e se questi non è presente nel livello successivo.
Capire quindi in quale punto della scala l'immagine ha subito una alterazione risulta semplificata applicando questa organizzazione. La scala è strutturata nel seguente modo:

- Confronto SHA1;
- Confronto Metadata;
- Confronto Bit-a-Bit;
- Confronto statistico.

Il primo controlla sia i metadata che le differenze di bit tra immagini in un unico test, mentre gli altri controllano singolarmente una sola caratteristica [Figura 14].

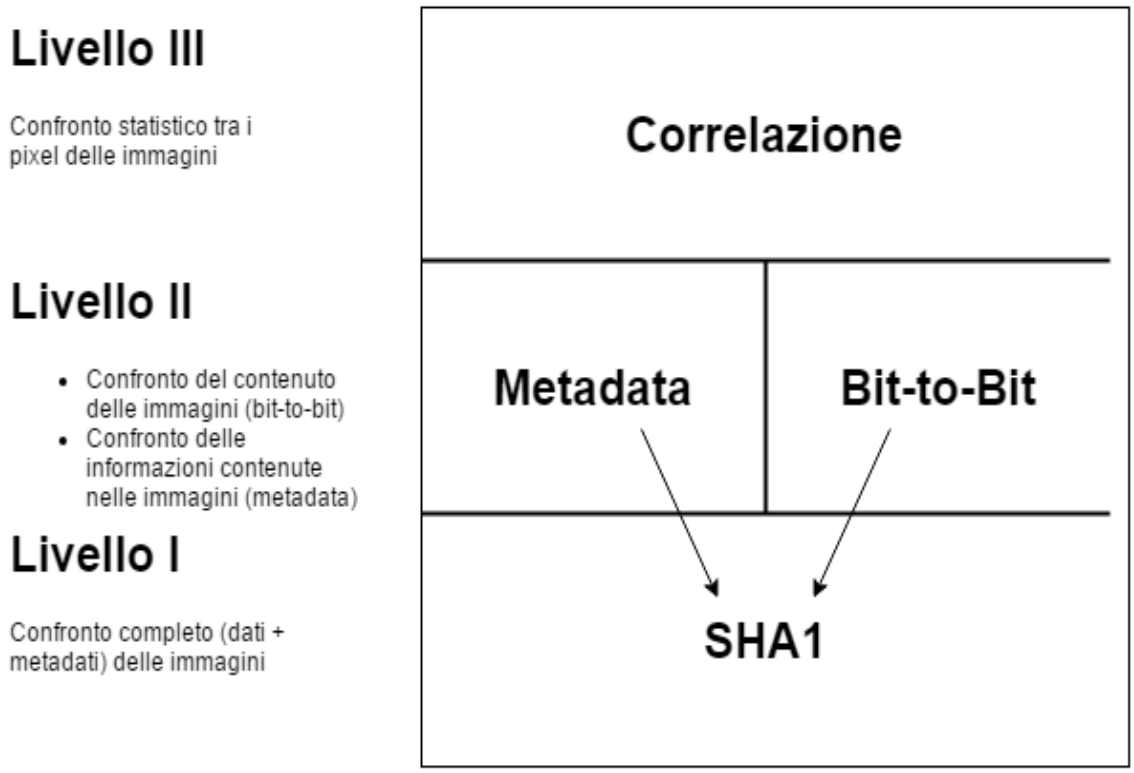

Figura 14: Scala dei confronti per i test effettuati

3.5.1 Confronto SHA1

Per questo test è stato utilizzato un algoritmo di hash per trovare delle differenze nei bit delle immagini tra le stesse scaricate e quelle caricate sul social network. Lo SHA1 (Secure Hash Algorithm) è un algoritmo crittografico con un message digest di 160bit che, anche se non più considerato sicuro è ottimo per gli esperimenti, in quanto, confrontarlo con un'immagine identica, permette di vedere se il file è stato corrotto in qualche modo.
Controllare lo SHA1 significa controllare che l'immagine sia integra e che quindi sia i bits di cui è composta l'immagine, sia i metadata che questi porta con se, non siano affette da alterazioni avvenute al momento del caricamento.
I test effettuati in Tabella 5 mostrano come le immagini abbiamo subito delle alterazioni evidenti dei bit e soprattutto si come alcune di queste siano modificate solo nella risoluzione più grande di quella standard, dando una conferma al precedente test di controllo sulla compressione di un'immagine. Da notare è anche la presenza di modifiche solo per Facebook e Instagram

nel confronto tra profili diversi di una stessa immagine. Questo lascia intendere che a parte per questo, l'algoritmo di alterazione utilizzato da ogni social network si applica allo stesso modo per immagini uguali e che quindi non dipende in alcun modo dal profilo utente.
Per quanto invece riguarda Facebook nei prossimi esperimenti saranno controllati più nel dettaglio i motivi di questi risultati, spiegando se questi dipendono dal profilo o da altro.
I test successivi sono una conseguenza di questo test, infatti, se lo SHA1 non restituisce delle prove di una avvenuta modifica di un'immagine, si può affermare, che nei prossimi confronti, non verrà trovata alcuna alterazione dei campi dove i risultati non hanno evidenziato differenze.

Social network	input vs output - stesso profilo SHA1	output profili diversi SHA1
Facebook	Diversi	Diversi
Instagram	Diversi	Diversi
Google+	Immagini grandi diverse	SHA1 uguali
Twitter	Diversi	SHA1 uguali
Whatsapp	Diversi	SHA1 uguali
Pinterest	Diversi	SHA1 uguali
Telegram	Immagini grandi diverse	SHA1 uguali
Linkedin	Diversi	SHA1 uguali
Tumblr	Immagini grandi diverse	SHA1 uguali
Vk	Immagini grandi diverse	SHA1 uguali
QQ	Immagini grandi diverse	SHA1 uguali
WeChat	SHA1 uguali	SHA1 uguali

Tabella 5: Risultati del test tragli SHA1 delle immagini caricate e scaricate dai social network

3.5.2 Confronto bit-a-bit

Per questo test abbiamo usato un confronto bit-a-bit del contenuto dell'immagine, questo compara due immagini binarie con l'intento di scovare il numero di bit che differiscono nella stessa posizione. A differenza dello SHA1, questo confronto permette di scovare degli errori escludendo la parte legata ai metadata, i quali potrebbero essere svuotati o modificati dal social network.
L'algoritmo utilizzato per il confronto si basa su una tecnica di differenza tra immagini che dopo aver controllato che non vi siano errori tra i bit di due immagini all'apparenza identiche, genera una nuova foto composta dalle sole differenze tra le due e ne calcola il riquadro di confine delle regioni non-zero. Se questo supera certi valori predefiniti, allora l'immagine è stata modificata dal social network [Figura 15].

Figura 15: Esempio di differenza tra il contenuto di due immagini simili ma non identiche

I risultati in Tabella 6 sono simili a quelli del precedente esperimento con lo SHA1, confermando l'ipotesi che le immagini subiscano una alterazione dovuta alla compressione o ottimizzazione al momento del caricamento sul sito. Per quanto riguarda Facebook e Instagram, invece, i risultati mostrano come la stessa immagine per profili diversi restituisca un risultato negativo, motivo per cui l'alterazione non viene fatta nel contenuto dell'immagine ma deve essere per forza nei metadata, essendo lo SHA1 la somma dei due test.

Social network	input vs output - stesso profilo bit-a-bit	output profili diversi bit-a-bit
Facebook	Diversi	Contenuti uguali
Instagram	Diversi	Contenuti uguali
Google+	Immagini grandi diverse	Contenuti uguali
Twitter	Immagini grandi diverse	Contenuti uguali
Whatsapp	Diversi	Contenuti uguali
Pinterest	Diversi	Contenuti uguali
Telegram	Immagini grandi diverse	Contenuti uguali
Linkedin	Diversi	Contenuti uguali
Tumblr	Immagini grandi diverse	Contenuti uguali
Vk	Immagini grandi diverse	Contenuti uguali
QQ	Immagini grandi diverse	Contenuti uguali
WeChat	Contenuti uguali	Contenuti uguali

Tabella 6: Risultati del test sul bit-a-bit tra le immagini caricate e scaricate dai social network

3.5.3 Confronto metadata

Il metadata è definito come un dato che fornisce informazione riguardo uno o più aspetti di questo, è utilizzato per riassumere informazioni di base su di un dato con il quale possiamo lavorare di modo da tracciarlo in maniera più specifica.
Nelle immagini digitali, questo include la creazione di un dato, il suo scopo, l'ora e la data di creazione, il creatore o l'autore del file, il luogo e la rete a con cui il computer ha creato il dato, l'uso comune, la grandezza del file e può anche includere matadata che descrivono la dimensione dell'immagine, la profondità di colore, la risoluzione e altri dati non significativi per i test [5].
In Tabella 7 vengono inseriti i nomi dei campi utilizzati dai social network per le immagini. Questi sono solo una parte dei campi che possono essere aggiunti ad una foto, con la possibilità per i più esperti di andare anche a modificare lo standard IPTC (informazioni sul copyright, i dettagli sull'immagine, la posizione, la categoria, ecc. . .).

Tag Name	Tipo
ExifTool Version Number	Double
File Name	String
Directory	String
File Size	String
File Modification Date/Time	String
File Access Date/Time	String
File Inode Change Date/Time	String
File Permission	String
File Type	String
File Type Extension	String
MIME Type	String
JFIF Version	Double
Resolution Unit	Double
X Resolution	Integer
Y Resolution	Integer
Current IPTC Digest	String
Original Trasmission Reference	String
Special Instructions	String
Profile CMM Type	String
Profile Version	Double
Profile Class	String
Color Space Data	String
Profile Connection Space	String
Profile Date Time	String
Profile File Signature	String

Primary Platform	String
CMM Flags	String
Device Manufacturer	String
Device Model	String
Device Attributes	String
Rendering Intent	String
Connection Space Illuminant	Float
Profile Creator	String
Profile ID	String
Profile Description	String
Blue Matrix Column	Float
Blue Tone Reproduction Curve	String
Device Model Desc	String
Green Matrix Column	Float
Green Tone Reproduction Curve	String
Luminance	Integer
Measurement Observer	String
Measurement Backing	Integer
Measurement Geometry	Integer
Measurement Flare	Integer
Measurement Illuminant	String
Media Black Point	Float
Red Matrix Column	Float
Red Tone Reproduction Curve	String
Technology	String
Viewing Cond Desc	String
Media White Point	Float
Profile Copyright	String
Chromatic Adaptation	Float
Image Width	Integer
Image Height	Integer
Encoding Process	String
Bits Per Sample	Integer
Color Components	Integer
Y Cb Cr Sub Sampling	String
Image Size	String
Megapixels	Double

Tabella 7: Metadata utilizzati dai social network nelle immagini condivise

Nei social network è importante l'utilizzo dei metadata in quanto, possiamo scoprire se l'immagine è stata alterata e se alcuni dei campi vengono rimpiazzati con dati nuovi generati dal sito, andando quindi a restituire un risultato positivo nel precedente test sul controllo dello SHA1.
In Tabella 8, 9 e 10 possiamo notare i risultati dei test sui metadata, i quali, come ipotizzato, sono stati alterati. Quanto meno ad essere modificati sono la risoluzione delle immagini più grandi della risoluzione standard, dovuta ad uno scaling dell'immagine alla risoluzione imposta dai social network e la dimensione del file, che per via della compressione JPEG o dell'ottimizzazione, sono state modificate.
I metadata contenenti campi riguardanti il tempo, sono diversi per ogni immagine e per ogni profilo ovviamente, questo perché è impossibile caricare nello stesso momento un'immagine su un social network.
Da notare in particolare è come, tranne per Facebook e Instagram, i risultati sui metadata sono gli stessi tra due profili diversi per lo stesso sito, questo significa che i metadata vengono sostituiti tutti nello stesso modo indipendentemente dal profilo in questione.
Qualsiasi modifica a questi campi viene sovrascritta nel caricamento successivo dell'immagine, motivo per cui i dati modificati non permangono e quindi, anche trovando qualcosa che può essere definito come univoco per una determinata immagine, non significa che su questa è stata applicata una qualche forma di watermark.

Social network	input vs output - stesso profilo metadata (grandi)	output profili diversi metadata (grandi)
Facebook	dimensione file, image width, image height, IPTC current digest, original transmission reference, DCT-progressive	IPTC current digest, original transmission reference
Instagram	dimensione file, image width, image height, IPTC current digest, original transmission reference, DCT-progressive	IPTC current digest
Google+	file size, image width, image height	Metadata uguali
Twitter	file size, image width, image height, DCT-progressive	Metadata uguali
Whatsapp	file size, image width, image height, DCT-progressive	Metadata uguali
Pinterest	file size, image width, image height, DCT-progressive	Metadata uguali
Telegram	file size, image width, image height	Metadata uguali
Linkedin	file size, image width, image height, DCT-progressive	Metadata uguali
Tumblr	file size, image width, image height	Metadata uguali
Vk	file size, image width, image height	Metadata uguali
QQ	file size, image width, image height	Metadata uguali
WeChat	Metadata uguali	Metadata uguali

Tabella 8: Risultati ottenuti sulle differenze nei metadata di immagini con risoluzione maggiore della standard

Social network	input vs output - stesso profilo metadata (standard)	output profili diversi metadata (standard)
Facebook	file size, IPTC current digest, original transmission reference, DCT-progressive	IPTC current digest, original transmission reference
Instagram	file size, IPTC current digest, original transmission reference, DCT-progressive	IPTC current digest
Google+	Metadata uguali	Metadata uguali
Twitter	file size, DCT-progressive	Metadata uguali
Whatsapp	file size, DCT-progressive	Metadata uguali
Pinterest	file size, DCT-progressive	Metadata uguali
Telegram	Metadata uguali	Metadata uguali
Linkedin	file size, DCT-progressive	Metadata uguali
Tumblr	Metadata uguali	Metadata uguali
Vk	Metadata uguali	Metadata uguali
QQ	Metadata uguali	Metadata uguali
WeChat	Metadata uguali	Metadata uguali

Tabella 9: Risultati ottenuti sulle differenze nei metadata di immagini con risoluzione standard

Social network	input vs output - stesso profilo metadata (piccole)	output profili diversi metadata (piccole)
Facebook	file size, IPTC current digest, original transmission reference, DCT-progressive	IPTC current digest, original transmission reference
Instagram	file size, IPTC current digest, original transmission reference, DCT-progressive	IPTC current digest
Google+	Metadata uguali	Metadata uguali
Twitter	file size, DCT-progressive	Metadata uguali
Whatsapp	file size, DCT-progressive	Metadata uguali
Pinterest	file size, DCT-progressive	Metadata uguali
Telegram	Metadata uguali	Metadata uguali
Linkedin	file size, DCT-progressivet	Metadata uguali
Tumblr	Metadata uguali	Metadata uguali
Vk	Metadata uguali	Metadata uguali
QQ	Metadata uguali	Metadata uguali
WeChat	Metadata uguali	Metadata uguali

Tabella 10: Risultati ottenuti sulle differenze nei metadata di immagini con risoluzione minore della standard

3.5.3.1 DCT-progressive

Uno dei risultati mostrati nelle precedenti tabelle è dato dalla modifica di un particolare campo dei metadata che mostra quale processo di codifica JPEG è stata apportata all'immagine. I

social network solitamente applicano il DCT-Progressive, il quale permette di mostrare un'anteprima di un'immagine, con grandezza minore rispetto all'originale (tumbnail), al momento del caricamento sul sito [Figura 16].
Questo campo non influenza in alcun modo l'immagine che quindi non viene alterata se non nei sui metadata. Al momento del caricamento questi presentano nel campo in questione la dicitura standard DCT-Baseline, motivo per cui, se l'immagine viene caricata una seconda volta, le immagini che avevano come modifica solo questo particolare campo diventano identiche alla foto del caricamento.
Questa alterazione quindi non è rilevante al in questo studio, ma viene comunque tenuta presente in quanto, chiunque effettuerà ulteriori test a riguardo, saprà già come impostare i test per eliminarla o inserirla nella propria ricerca.

Figura 16: Esempio di applicazione del DCT-Progressive a un'immagine

3.5.3.2 Considerazioni sui metadata di Facebook e Instagram

Per quanto riguarda Facebook e Instagram nei precedenti capitoli si era notato un cambiamento dello SHA1 e più precisamente dei metadata, con differenze anche tra profili diversi, nei campi contenenti l'IPTC Current Digest e l'Original Transmission Reference. Per analizzare il motivo di questa anomalia, lo studio si è concentrato sull'effettuazione di diversi test per capire

se questi parametri rimangano invariati per un'immagine specifica nel social network o vengano modificati in qualche modo se l'immagine viene postata su differenti profili o scaricata da un utente diverso.
I test sono stati possibili solo su Facebook, in quanto Instagram non permette la condivisone di una foto. Questi si dividono nelle seguenti categorie:

- Caricare l'immagine I_A sul profilo A, postare l'immagine I_A sul profilo B e scaricarla tramite il profilo A;
- Caricare l'immagine I_A sul profilo A, postare l'immagine I_A sul profilo B e scaricarla tramite il profilo B;
- Caricare l'immagine I_A sul profilo A e scaricare l'immagine I_A tramite il profilo B;
- Caricare l'immagine I_A sul profilo A e scaricare l'immagine I_A tramite il profilo A.
- Effettuare i test in momenti diversi in modo da controllare se il tempo è un fattore di modifica dei campi.

Per ogni di essi è stato svolto anche il viceversa in modo da avere una conferma ulteriore dei risultati ottenuti [Figura 17].

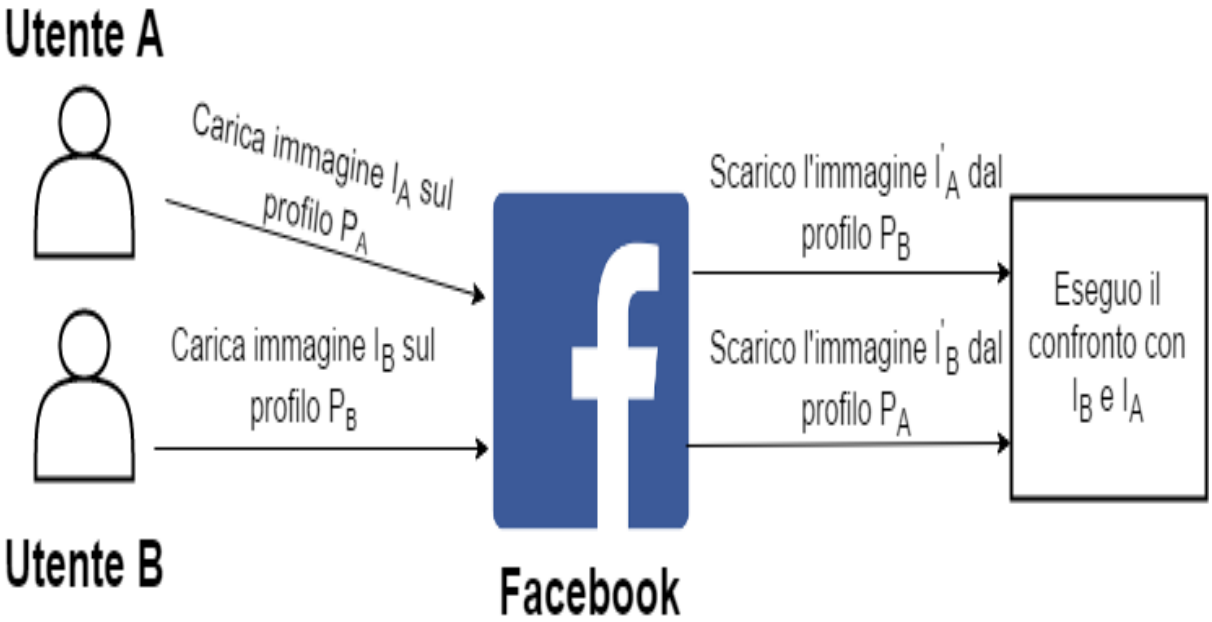

Figura 17: Metodo con cui sono stati effettuati i test sui metadata di Facebook

I test effettuati hanno evidenziato come questi campi siano gli stessi per ogni immagine caricata anche se questa viene spostata nel social network in altri profili o condivisa con altre persone. Quello che però resta impossibile da capire è in che modo questi campi vengano modificati. Non è possibile utilizzare alcun tipo di ingegneria inversa per sapere con quali elementi venga creato questo dato, le uniche supposizioni che si possono fare sono date dai test effettuati.
Questi mostrano come sicuramente i campi vengano modificati al momento della pubblicazione su Facebook e Instagram dell'immagine e che variano probabilmente in base al tempo, cioè al momento in cui l'immagine è caricata sul social e al profilo su cui l'immagine viene caricata originariamente.
Detto questo in Tabella 11 vi sono i risultati ottenuti per ogni test effettuato dimostrando che $I_A = I_B \Rightarrow I_A^{'} = I_A = I_B^{'} = I_B$.

Azione 1	Azione 2	Risultati ottenuti
Carico l'immagine I_A sul profilo A	La scarico con il profilo A	genero $I_A^{'}$
Carico l'immagine I_B sul profilo B	La scarico con il profilo B	genero $I_B^{'}$
Carico l'immagine I_A sul profilo A	La scarico con il profilo B	$I_A^{'} = I_A$
Carico l'immagine I_B sul profilo B	La scarico con il profilo A	$I_B^{'} = I_B$
Tramite il profilo A posto l'immagine I_A sul profilo B	La scarico con il profilo B	$I_A^{'} = I_A$
Tramite il profilo B posto l'immagine I_B sul profilo A	La scarico con il profilo A	$I_B^{'} = I_B$
Tramite il profilo A posto l'immagine I_A sul profilo B	La scarico con il profilo A	$I_A^{'} = I_A$
Tramite il profilo B posto l'immagine I_B sul profilo A	La scarico con il profilo B	$I_B^{'} = I_B$

Tabella 11: Risultati di ognuno dei test effettuati sui campi anomali di Facebook

3.5.4 Confronto statistico

Solitamente DIC (Digital Image Correlation) si basa sulla ricerca del massimo della matrice di correlazione definito come il sottoinsieme dell'array di intensità dei pixels su due o più immagini, il quale restituisce lo spostamento traslazionale tra queste.
È anche possibile stimare gli spostamenti per una risoluzione più piccola di quella del pixel, che viene spesso chiamata *"subpixel"*.
Per questo esperimento lo studio si è basato su l'applicazione di un software di terze parti, ImageJ, per la produzione dei risultati, utilizzando un plugin della GCSCA che genera un grafico sparso composto da i valori dei pixels delle due immagini confrontate con i rispettivi array di intensità.
I dati statistici restituiti dal programma sono:

- Il coefficiente di correlazione tra la due immagini confrontate;
- La pendenza della retta di regressione;

- La costante C che interseca l'asse y che da, in un range tra [-1,1], la correlazione tra due immagini. Se questa è 0 allora le foto sono *"Uguali"* altrimenti si considerano *"Diverse"* [Figura 18].

La correlazione può essere calcolata su di un singolo pixel o su di una area locale. Nel secondo caso, la grandezza è predefinita e viene calcolata una media per ogni area, il valore restituito verrà usato poi per calcolare il coefficiente di correlazione. Questo, che possiamo definire come C_v utilizza due immagini I_A e $I_A^{'}$, la prima è l'originale e la seconda è quella scaricata. Il risultato ottenuto, $C_v(I_A, I_B) = [-1, 1]$, indica di quanto le due immagini sono state traslate tra loro, con -1 la traslazione massima in senso antiorario e con 1 la massima in senso orario [Figura 19].

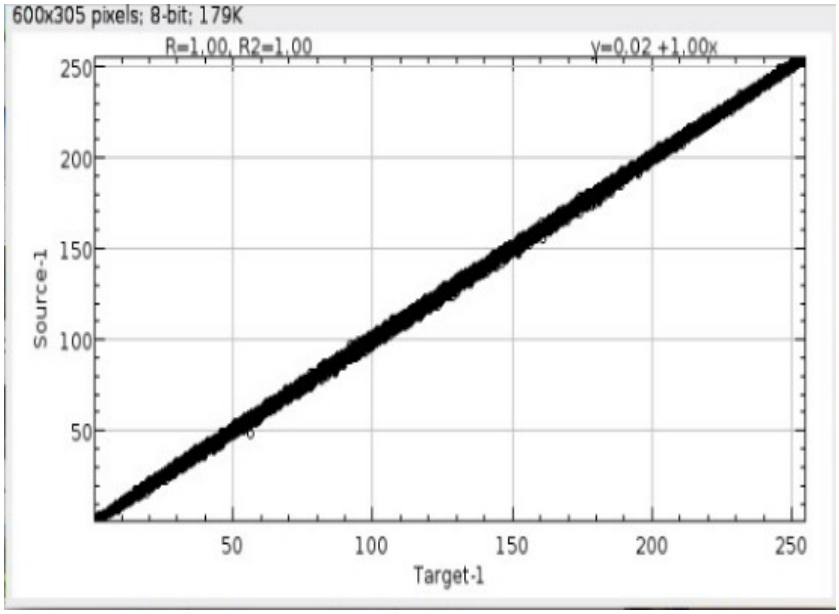

Figura 18: Grafico della correlazione tra due immagini

Figura 19: Coefficiente di correlazione di immagini traslate

Dato che i risultati, per ogni social network e per ogni risoluzione, generano valori di correlazione in un range ben definito ma molto vario, per calcolare una media di questi il più possibile precisa, è stato utilizzato un intervallo di confidenza (1).

Se U e V sono variabili random e la distribuzione di probabilità dipende da un qualche parametro θ, e

$$Pr(U < \theta < V) \geq \beta \ (dove\, \beta\, e\, un\, numero\, tra\, 0\, e\, 1) \tag{1}$$

allora il valore random *(U, V)* e un *"[(1-β)*100] % θ"* intervallo di confidenza.

I valori agli estremi dell'intervallo di confidenza sono chiamati *"limiti di confidenza"*. A questi è associata una probabilità cumulativa che caratterizza, indirettamente in termini di probabilità, la sua ampiezza, comparata con il valore massimo che può essere assunto misurando un valore random, cioè la probabilità che un evento casuale, descritto dalla variabile in questione, cada all'interno di questo range.
Sotto, nelle Tabelle 12, 13 i test hanno mostrato come ancora una volta la compressione o l'ottimizzazione generi delle alterazioni delle immagini.
Le prove sono state svolte applicando un intervallo di confidenza con $\beta = 0.95$ e con $\theta = 10$ cioè, la somma totale delle immagini testate per ogni cella della tabella. È stato utilizzato un intervallo di confidenza del 95% applicato alla somma dei valori di correlazione restituito dalle 10 immagini. I valore calcolato di ogni cella della tabella è, quindi, la media restituita da questo intervallo.
Le immagini con risoluzione superiore a quella standard non sono prese in esame in quanto, per

i test di correlazione, si deve per forza avere la stessa dimensione, cosa che per via dello scaling non si può ottenere. Provare anche solo a scalare l'originale, inoltre, genererebbe un falso positivo per questo test. Unico social network su cui poter eseguire l'esperimento è WeChat, che come descritto in precedenza nell'elaborato tratta le immagini come file, motivo per cui, è possibile avere la stessa grandezza in quanto lo scaling non è effettuato. Questo per il test sulla correlazione per immagini grandi restituisce risultato negativo per tutte le immagini.

Social network	input vs output - stesso profilo correlazione	output profili diversi correlazione
Facebook	0.18	0
Instagram	0.13	0
Google+	0	0
Twitter	0	0
Whatsapp	0.06	0
Pinterest	0.12	0
Telegram	0	0
Linkedin	0.17	0
Tumblr	0	0
Vk	0	0
QQ	0	0
WeChat	0	0

Tabella 12: Risultati del test su risoluzione piccola per la correlazione tra immagini

Social network	input vs output - stesso profilo correlazione	output profili diversi correlazione
Facebook	0.15	0
Instagram	0.12	0
Google+	0	0
Twitter	0	0
Whatsapp	0.09	0
Pinterest	0.10	0
Telegram	0	0
Linkedin	0.24	0
Tumblr	0	0
Vk	0	0
QQ	0	0
WeChat	0	0

Tabella 13: Risultati del test su risoluzione piccola per la correlazione tra immagini

3.6 Geolocalizzazione

La geolocalizzazione [21] permette agli utenti di ottenere alcuni servizi e funzionalità geografiche, come ad esempio la capacità di aggiungere una etichetta con la città dove si è scritto un post o si è scattata una foto.
Nel campo dei social network questo permette di collegare e coordinare utenti con persone vicine alla loro posizione o eventi nelle vicinanze tramite il loro indirizzo IP o mediante la trilaterazione dell'hotspot. Nel mondo mobile questi servizi si basano sul permesso di accedere alla propria posizione geografica mediante il GPS permettendo di ottenere dati in real time.
La geolocalizzazione permette quindi di ottenere informazioni su persone vicine e eventi, per questo motivo, social network come Facebook, Google+, VK, Twitter e Linkedin, utilizzano soprattutto questo strumento per scopi di marketing. Per questo motivo è importante controllare che le immagini caricate, che utilizzano la geolocalizzazione, non subiscano alterazioni da parte di questi.
Esistono diversi modi per utilizzare questo strumento di localizzazione e solitamente i social network ne utilizzano contemporaneamente più di uno:

- **IP base:** Questa forma di geolocalizzazione è la più comune e permette ai social network di trovare altre persone in base alla posizione comunicata dal indirizzo IP. Questo viene trasmesso al momento della connessione alla rete internet dal router dell'utente.

- **LAN base:** Più preciso della IP base, questo metodo è poco utilizzato da i social network. Solitamente viene utilizzato per avvertire di un accesso, da località non abituali di un utente, al sito. Inoltre, per localizzare un utente ad un determinato indirizzo, i social network utilizzano il metodo seguente.

- **LAT/LON base:** Il miglior tipo di geolocalizzazione. Permette di sapere esattamente, in coordinate, il punto geografico in cui un utente si trova utilizzando la latitudine e longitudine. Solitamente è utilizzato da pochi social network (Facebook, Google+, Twitter e Linkedin). Questi permettono di scrivere la latitudine e la longitudine nel loro motore di ricerca per travare utenti intorno ad queste coordinate. Con l'avvento dei GPS all'interno dei cellulari è possibile, per mezzo di questo metodo, avere informazioni molto più dettagliate sulle posizioni degli utenti.

3.6.1 Analisi mediante metadata

Per quanto riguarda le immagini, se vengono aggiunte informazioni geografiche al loro interno, si può supporre che siano contenute nei metadata. Il prossimo test di questo elaborato, allora, si è concentrato sul trovare anomalie nell'immagine caricata sui social network condividendo la propria posizione [Figura 20].
Il test utilizza sempre lo stesso schema dei precedenti esperimenti sui SN: le stesse trenta immagini originali per ogni sito, dieci per ognuna delle tre risoluzioni utilizzate (grandi, standard,

piccole). In questo modo si possono confrontare i vecchi risultati e vedere se vi sono anomalie di qualche tipo nei valori di ritorno.

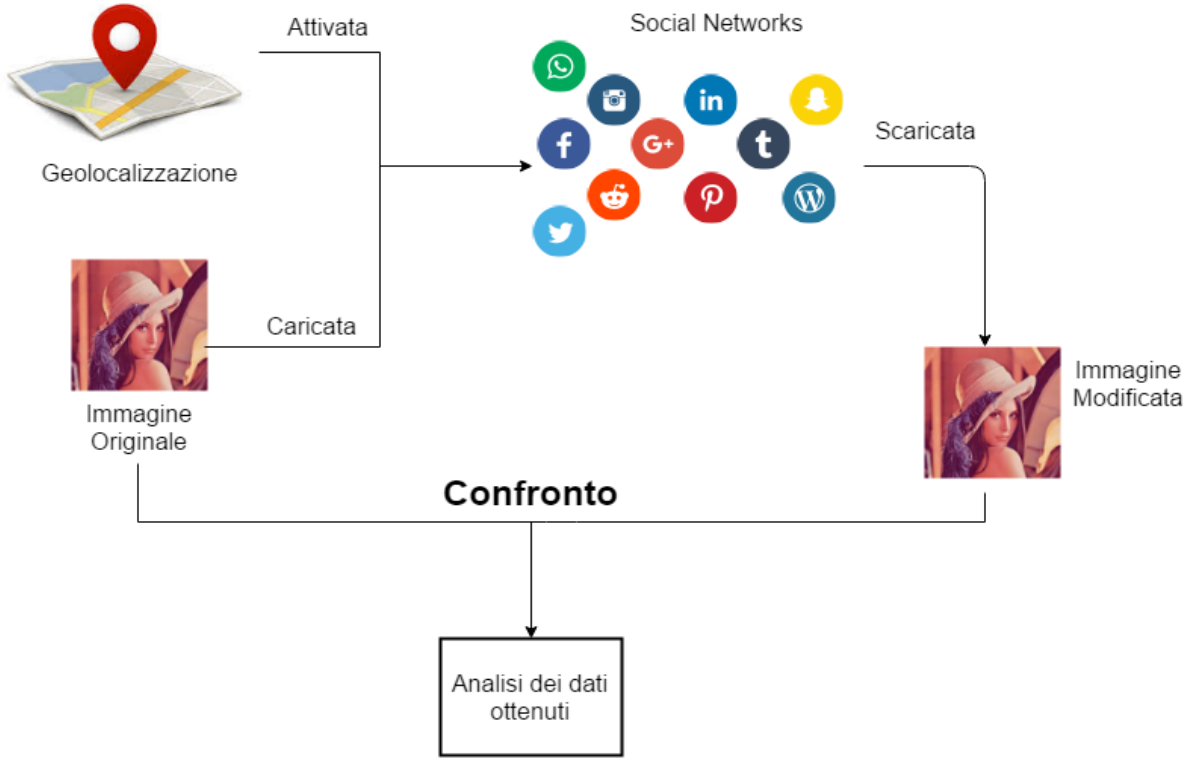

Figura 20: Esempio di funzionamento dei test su la geolocalizzazione

Dopo aver effettuato i test, i risultati non hanno mostrato alterazioni che non erano già presenti nei precedenti esperimenti sui metadata. Avere però un risultato negativo non significa non avere risultati, infatti da questo test emergono due evidenze.
La prima, è che i campi dei metadata che contengono coordinate GPS vengono rimossi e quindi la posizione non viene condivisa dal social network con l'utente, ma probabilmente viene salvata su di un server locale.
La seconda e anche la più interessante, mostra, che ripetendo gli esperimenti sui metadata di Facebook e Instagram con la localizzazione attiva e modificandola ogni volta che la si posta su un altro profilo, i valori anomali dei metadata non cambiano. Possiamo quindi escludere la posizione come uno dei parametri che alterano i campi anomali di questo social network.
Per quanto riguarda il controllo delle immagini scaricate da profili in zone diverse. Non è stato necessario un confronto come nei test precedenti per vedere se, cambiando il profilo, queste subiscono delle alterazioni. I campi non vengono modificati dalla posizione e quindi le immagini scaricate risultano tutte uguali nei risultati.

3.7 CDN

una CDN (Content Delivery Network) è una rete distribuita a livello mondiale di server proxy ripartito in più data center [14]. Il suo scopo è di garantire la disponibilità dei contenuti agli utenti finali con alte prestazioni, infatti una CDN offre gran parte dei contenuti che si trovano su internet oggi inclusi elementi che compongono il web (script, testi, ecc...), dati scaricabili (file, immagini, ecc...), applicazioni, file di streaming e social network.
I fornitori di contenuti pagano i CDN per usufruire della loro rete, in modo da rendere disponibile un determinato servizio o file su di un server il più vicino possibile ai propri utenti. A sua volta i CDN pagano i provider e gli operatori di rete per permettergli di ospitare i propri server nei loro data center.
I CDN sono formati da server nodi che sono sparsi in tutto il mondo e contengono gli stessi dati, in questo modo è più semplice raggiungere un informazione, sprecando meno banda e tempo di caricamento, questi sono agganciati a un server host che appartiene al fornitore di contenuti e che immagazzina successivamente i dati inviatigli dal CDN [Figura 21].
Oltre a migliorare le prestazioni e la disponibilità questi hanno anche il vantaggio di scaricare direttamente il traffico dei dati che arriverebbero dalle infrastrutture di origine del fornitore di contenuti, in questo modo questi ultimi ottengono dei risparmi sui costi e un grado di protezione dagli attacchi DoS utilizzando le loro grandi infrastrutture server per assorbire il flusso d'attacco.
Se all'inizio i contenuti venivano distribuiti utilizzando solo server gestiti e di proprietà dei CDN, oggi si iniziano ad avere soluzioni ibride applicando la tecnologia del P2P.

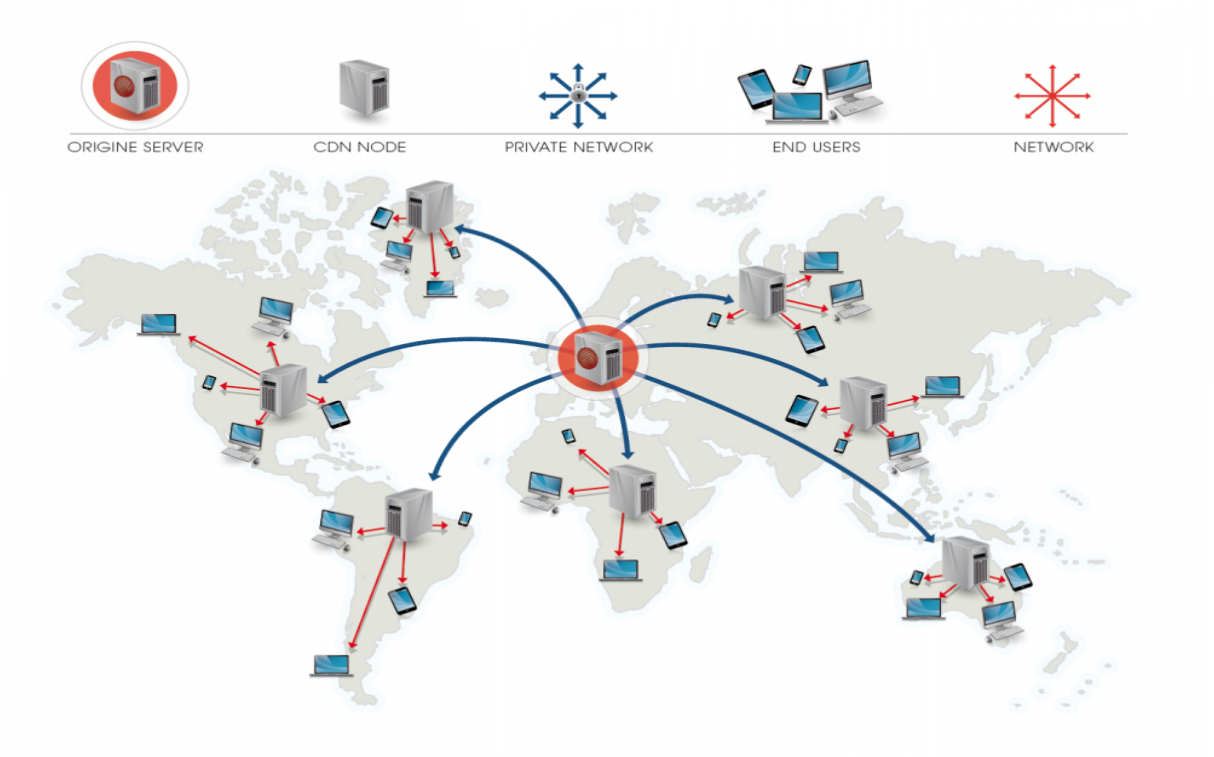

Figura 21: Esempio di funzionamento di una rete CDN

3.7.1 Gestione da parte dei social network

Come fornitori di contenuti i social network utilizzano i CDN per permettere ai propri utenti di caricare, scaricare e visualizzare contenuti in tutto il mondo nella maniera più veloce possibile. Per questo motivo in questo studio è importante analizzare se i CDN siano la causa delle alterazioni delle immagini o se comunque passino delle informazioni private degli utenti ai social network [19].
Inoltre i CDN sono sparsi in tutto il mondo, motivo per cui bisogna chiedersi se le alterazioni possano essere applicate solo in alcuni paesi specifici che effettuano maggiori controlli sulle identità e i movimenti sui social degli utenti.
Precedentemente la tesi ha mostrato come la geolocalizzazione non aggiunge nulla all'immagine. Possiamo quindi ipotizzare allora che i social network mantengano quell'informazione su di un server o non saperebbe possibile l'identificazione della posizione in cui è stata scattata un'immagine in un secondo momento.
Per questo nell'elaborato si è andati a controllare in quali paesi la popolazione è soggetta a più controlli sulla privacy [Figura 22]. Tra questi sono stati identificati tre paesi geograficamente distanti tra loro e su cui sono stati possibili eseguire dei test (Stati Uniti, Gran Bretagna, Russia e Giappone) e che, quindi, potrebbero aggiungere alcune informazioni alle immagini tramite i

CDN.
Per paesi interessanti dal punto di vista della privacy come Cina o Myanmar non è stato possibile eseguire i test, in quanto, questi, bloccano molti dei social network adottati per gli esperimenti.

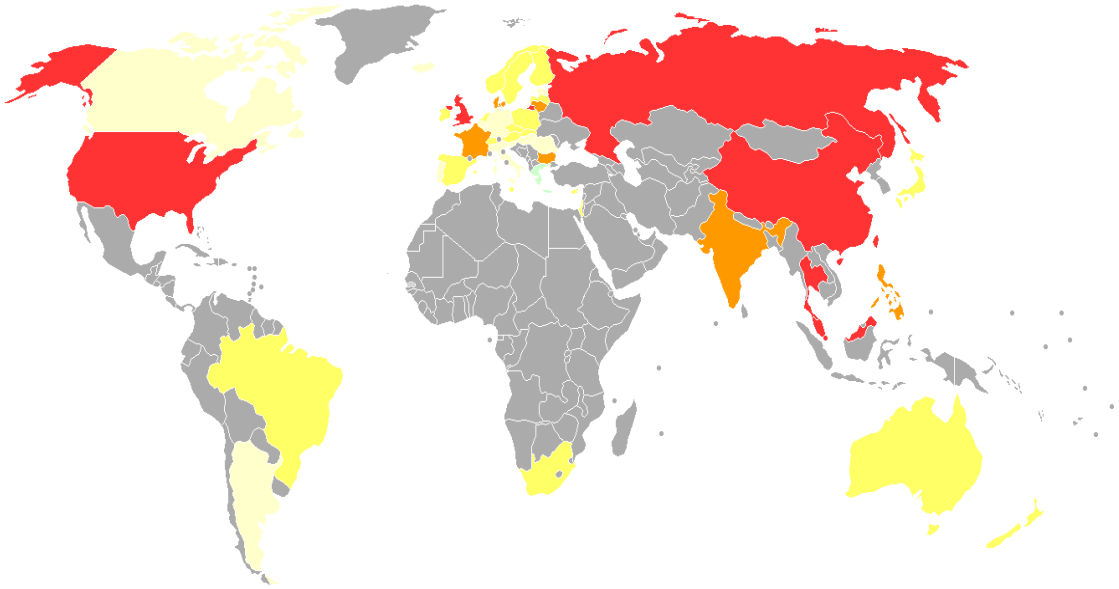

Figura 22: Mappa che mostra la privacy dei vari paesi del mondo[6]

Per quanto riguarda i CDN utilizzati dai social network abbiamo che Facebook, Instagram e Whatsapp utilizzano static.xx.fbcdn.net, Google+ ha il proprio CDN proprietario Google CDN, Twitter utilizza Amazon e Akamai, Pinterest e VK hanno un proprio CDN, Linkedin utilizza Google CDN e Amazon, WeChat utilizza Tencent, Tumblr utilizza Highwinds ed in fine per Telegram non è stato possibile capire il CDN utilizzato.

3.7.2 Test sui metadata

Per controllare le possibili alterazioni dovute ai CDN, lo studio, ha utilizzato una VPN per connettersi tramite un indirizzo IP mascherato di un altro stato di modo da ingannare il social network e figurare in un altro luogo geografico.
una VPN (Virtual Private Network) permette di collegarsi infatti ad un server come se fossimo fisicamente connessi a questo. Il server cambia a seconda dello stato in cui andiamo ad effettuare l'esperimento.
Dopo aver preso le solite trenta immagini per ogni social network e averle divise, dieci per ogni

[6]Nei paesi in cui la mappa mostra un colore più rosso, la popolazione è soggetta a più controlli (fonte "www.privacyinternational.org")

risoluzione (grandi, standard, piccole) come nei precedenti esperimenti, si è andati a mascherare l'indirizzo IP mediante il VPN e quindi a effettuare i test [Figura 23]. I paesi su cui sono stati condotti gli esperimenti, sono Italia, Stati Uniti, Gran Bretagna, Russia e Giappone tutti con un profilo creato da nuovo figurante in ognuno di questi stati per evitare problemi di geolocalizzazione.

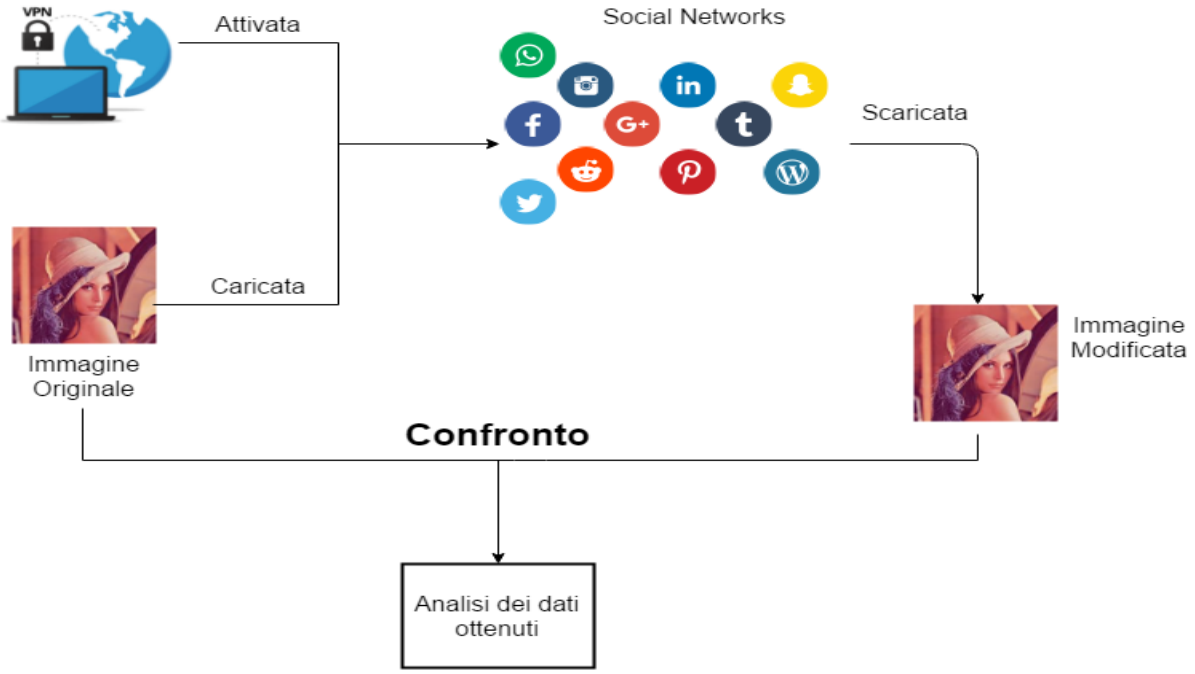

Figura 23: Esempio di impostazione dell'esperimento sui CDN

I risultati hanno sottolineato come i CDN non applichino alcun cambiamento alle immagini e come tutte le modifiche siano dovute al social network stesso, il quale le altera come già visto nei precedenti test. Per quanto riguarda i metadata, questi non vengono modificati, anche se, controllando le API di Akamei, è stata riscontrata la possibilità di effettuare delle ricerche di modo da conoscere la posizione in cui le foto sono state scattate e informazioni di provenienza sull'immagine (di chi è il profilo che ha scattato la foto, dove si trovava al momento della condivisione sul SN, ecc. . .).
Questo lascia supporre che il CDN non aggiunge nulla all'immagine, ma salva delle informazioni all'interno dei propri server sui dati caricati.
Un altro risultato è quello che deriva dai metadata di Facebook e Instagram, infatti anche cambiando il paese e effettuando i classici test sulle immagini, postandole e scaricandole da diversi profili di diversi stati, queste mantengono l'IPTC Current Digest e l'Original Transmission Reference originale, motivo per cui è possibile escludere i CDN come causa dell'alterazione di questi valori.

3.7.3 Considerazioni sugli indirizzi IP

Per quanto riguardagli indirizzi IP con cui sono stati effettuati i test, bisogna notare come il browser web utilizzato, nel momento in cui viene utilizzato per la prima volta, salva in automatico la posizione. Per questo motivo, per eseguire ogni esperimento, è stato necessario connettersi tramite VPN e disinstallare e reinstallare ogni volta il browser web, altrimenti, l'indirizzo IP salvato dal browser, veniva condiviso con il social network condividendo la nostra reale posizione e vanificando il test.
Costatare che una prima geolocalizzazione viene effettuata dal browser, mediante forse una CDN che comunica la nostra posizione tramite l'indirizzo IP, permette in qualsiasi paese ci si trovi, di avere tutte le informazioni, notizie e risultati, come se si fosse in quello di origine.
Esulando dalle motivazioni di questa ricerca, non sono stati effettuati test per validare questa ipotesi.

4 Algoritmi di watermarking applicati alle immagini condivise

Dopo aver controllato come i social network gestiscono le immagini e i vari tipi di algoritmi di watermarking che possono essere utilizzati in questo campo, sono stati eseguiti dei test concreti applicando gli algoritmi alle immagini da caricare sui SN.
Per prima cosa sono state necessarie ricerche per trovare questi algoritmi nel web, da notare che sono stati presi in visione solo quelli non a pagamento per lavorare con codice open source, di modo da verificare che le caratteristiche del codice permettessero di soddisfare i criteri di ricerca. Tutti gli algoritmi sono semifragili e basati su DCT, questi sono stati divisi in algoritmi in bianco e nero e algoritmi a colori a seconda del tipo di formato dell'immagine su cui sono applicabili.
Quelli in bianco e nero sono molti di più ma provengono da vecchi documenti quasi tutti pubblicati prima del nuovo millennio. Utilizzano quindi tecniche, magari superate, sufficienti comunque all'esecuzione dei test.
Per quanto riguarda invece gli algoritmi a colori, i software trovati sono quasi tutti a pagamento e quindi sono stati effettuati pochi test in questo campo. Le tecniche lasciano ben sperare per alcuni social network tuttavia, anche se ancora non permettono di recuperare almeno parte del watermark in caso di fallimento come quelli in bianco e nero.
Detto questo, la spiegazione dei test è per entrambi i casi pressoché simile. Questi sono stai effettuati nella seguente maniera [Figura 24]:

- Vengono prese in esame trenta immagini per ogni social network con tre differenti dimensioni (grandi, standard e piccole) come nei precedenti test;
- I watermark viene cifrato in ogni immagine;
- Le immagini vengono caricate e poi scaricate su tutti i social network;
- Il watermark viene decifrato e visionato per controllare che sia integro;
- l'esperimento si ripete per ogni algoritmo di watermarking utilizzato.

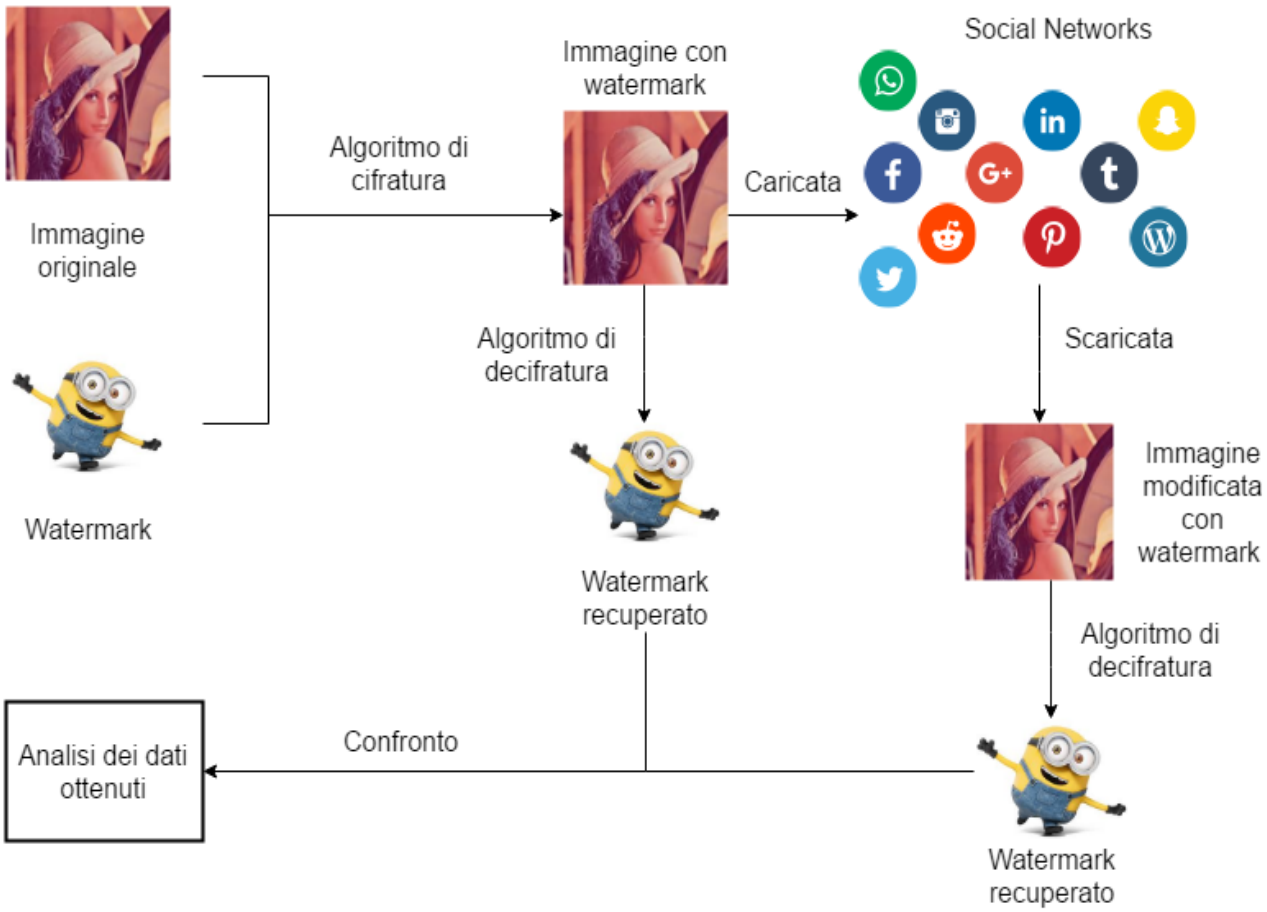

Figura 24: Esempio di impostazione dell'esperimento per il test degli algoritmi di watermarking

4.1 Bianco e nero

Gli algoritmi in bianco e nero per funzionare hanno bisogno di una dimensione quadrata dell'immagine, questo perché, per inserire il watermark, hanno bisogno di una risoluzione della foto che è potenza di due (i.e. $512x512 = 2^9x2^9$) per via dell'algoritmo di DCT utilizzato.
Per questo, tutte le immagini che sono state testate in questa ricerca, sono state impostate su dimensioni quadrate e trasformate in un formato .PGM in bianco e nero per permettere l'utilizzo delle librerie non a colori su cui si basano gli algoritmi.
Il primo test preso in esame è basato sul comportamento degli algoritmi su immagini completamenti bianche, completamenti nere o nere/bianche solo a metà. Questo esperimento è necessario per provare a capire come il watermark venga applicato ai vari pixels che compongono l'immagine.
L'esperimento è stato eseguito utilizzando dieci immagini per ognuna delle tre categorie citate precedentementeà Su queste è stata eseguita la cifratura e decifratura del watermark per ogni

algoritmo di watermarking in bianco e nero senza caricarlo e scaricarlo da alcun social network. I risultati in Tabella 14 mostrano come i vari algoritmi necessitino di bianchi e di neri per applicare la firma digitale. Si noti inoltre come per alcuni algoritmi, i due colori, devono anche essere distribuiti uniformemente o il watermark non viene applicato *("watermark non trovato")*.

Algoritmi	immagini nere	immagini mezze nere/bianche	immagini bianche
Koch [10]	Watermark non trovato	100%	Watermark non trovato
Cox [4]	Watermark non trovato	83%	Watermark non trovato
Dugad [6]	Watermark non trovato	Watermark non trovato	Watermark non trovato
Corvi [3]	Watermark non trovato	Watermark non trovato	Watermark non trovato
Bruyn [2]	Watermark non trovato	Watermark non trovato	Watermark non trovato
Kim [9]	Watermark non trovato	69%	Watermark non trovato
Frid [7]	Watermark non trovato	71%	Watermark non trovato
Wang [20]	Watermark non trovato	78%	Watermark non trovato
Xia [22]	Watermark non trovato	80%	Watermark non trovato
Xie [23]	Watermark non trovato	57%	Watermark non trovato
Xie2 [23]	Watermark non trovato	100%	Watermark non trovato
Zhu [24]	Watermark non trovato	Watermark non trovato	Watermark non trovato
Kund [11]	Watermark non trovato	52%	Watermark non trovato
Kund2 [12]	Watermark non trovato	12%	Watermark non trovato

Tabella 14: Risultati dell'esperimento su immagini solo bianche o solo nere o nere/bianche solo a metà

Per quanto riguarda le caratteristiche con cui sono composti gli algoritmi, questi devono essere divisi in due metodi, quelli basati su stringa e quelli basati su impronta.

4.1.1 Algoritmi basati su stringa

Questi algoritmi utilizzano un algoritmo basato sulla cifratura di una stringa di lunghezza minima X, con X il numero di bit minimi richiesti dall'algoritmo all'inserimento, che funge da watermark. Questo è il metodo più semplice e anche il più efficace da controllare. Se il valore di ritorno dopo la decifrazione è uguale alla firma iniziale, si può affermare con certezza che il test si è concluso con successo.
Gli algoritmi che utilizzano questa forma di watermark sono: Koch (32bit), Bruyn (32bit), Frid (96bit), Xie (32bit), Xie2 (32bit), Kund (32bit) e Kund2 (32bit), con tra parentesi la lunghezza minima X in bit.
Più si rispetta il numero minimo di bit, più si hanno dei risultati vicini alla perfezione, in quanto, all'aumentare dei caratteri formanti i bit richiesti dall'algoritmo, aumenta anche la possibilità di errore nella modifica dell'immagine da parte dei social network.

Social network	Koch	Bruyn	Frid	Xie	Xie2	Kund	Kund2
Originale	99%	67%	68%	59%	62%	59%	80%
Facebook	99%	85%	60%	69%	61%	1%	50%
Instagram	99%	85%	60%	69%	61%	1%	50%
Google+	99%	85%	62%	61%	61%	2%	38%
Twitter	94%	85%	62%	61%	61%	2%	38%
Whatsapp	94%	92%	62%	61%	69%	5%	60%
Pinterest	99%	98%	62%	66%	61%	2%	38%
Telegram	37%	85%	62%	61%	61%	2%	38%
Linkedin	99%	92%	62%	66%	69%	2%	35%
Tumblr	99%	85%	62%	61%	61%	2%	38%
VK	99%	85%	38%	61%	61%	2%	38%
QQ	94%	85%	38%	66%	61%	1%	42%
Wechat	99%	67%	68%	69%	62%	59%	80%

Tabella 15: Test degli algoritmi di watermarking basati su stringa per immagini in bianco e nero

In Tabella 15 vengono mostrati i risultati di questo esperimento. Come si può notare, alcuni algoritmi traggono vantaggio dall'ottimizzazione o dalla compressione dell'immagine, per migliorare la propria efficienza. L'originale, che sarebbe dato dalla cifratura e decifratura delle immagini senza caricarle o scaricarle dai social, infatti, genera risultati peggiori in alcuni casi. Non è stato possibile appurare questa teoria non conoscendo gli algoritmi utilizzati per la compressione e ottimizzazione delle immagini. Si può solo supporre che sia dovuto a una riduzione da parte della compressione della dimensione dell'immagine. Se il watermark riesce a resistere a questa alterazione, esso viene posizionato in un punto, più facilmente riconoscibile dall'algoritmo che si occupa della decifratura, della firma digitale.
Alcuni algoritmi resistono meglio degli altri alle alterazioni dei social network, infatti, sia l'algoritmo di Koch che quello di Bruyn ottengono ottimi risultati in tutti i test.
Da notare, come gli algoritmi siano stati testati su immagini con dimensioni differenti e quindi non rispecchiano omogeneamente i risultati. Per tutti gli algoritmi, infatti, è stato sempre possibile estrarre il watermark. Esistono dei casi in cui per uno stesso gruppo di immagini, alcune hanno dato risultati perfetti(100%) mentre altri, risultati molto negativi, generando quindi una diminuzione della media dei successi.
Avere come risultato un 68% di decifratura del watermark non significa che le immagini nei test non abbiano mai dato un risultato positivo (100%), ma che non lo è stato dato per tutte. Esistono algoritmi quindi che si adattano meglio a qualsiasi immagine e altri solo ad alcune.
Si può quindi affermare come, a seconda della dimensione, distribuzione della scala dei grigi e alterazione di un'immagine, questa può dare risultati perfetti o meno in tutti gli algoritmi testati e per ogni social network.

4.1.2 Algoritmi basati su impronta

Questi algoritmi utilizzano un metodo basato sulla cifratura di un'immagine come watermark. Questa deve essere ovviamente in bianco e nero e avere una dimensione che è potenza di due. I test hanno dimostrato che mantenere una grandezza di 512x512 permette di avere il miglior risultato possibile. Il metodo, anche se più complesso, è anche il più robusto e permette di mantenere, a differenza del precedente, una considerevole quantità in più di informazione all'interno dell'immagine.
Uno dei grandi problemi di questi tipi di algoritmi risiede nella ridotta capacità di nascondere il watermark all'interno dell'immagine. Di fatto, per aggiungere la firma digitale, questi creano un effetto *sporco* sulla foto, permettendo, ad un utente malevolo, di capire dove si trova parte del watermark [Figura 3]. Per questo motivo, oltre a controllare che la firma digitale sia integra, bisogna verificare, dopo la cifratura, il grado di visibilità di questa all'interno dell'immagine.
Gli algoritmi che supportano questo tipo di watermark sono: Cox, Dugad, Kim, Wang, Xia e Zhu. Per ognuno di loro è stata adottata la stessa immagine di watermark per eseguire gli esperimenti.
Mantenere una dimensione che si avvicina il più possibile a quella utilizzata dall'immagine usata come firma, permette di ottenere risultati migliori rispetto a dimensioni maggiori o minori. Il motivo risiede probabilmente nel metodo di applicazione della firma digitale.
Gli esperimenti applicati sono identici a quelli del precedente capitolo con la differenza che, invece di inserire una stringa, in questo caso l'algoritmo di watermarking richiede l'inserimento di un'immagine in bianco e nero.

Social network	Cox	Dugad	Corvi	Kim	Wang	Xia	Zhu
Originale	92%	86%	50%	97%	95%	99%	95%
Facebook	79%	100%	51%	61%	92%	95%	83%
Instagram	79%	100%	51%	61%	92%	95%	83%
Google+	79%	100%	53%	86%	94%	98%	87%
Twitter	70%	100%	53%	75%	94%	98%	87%
Whatsapp	70%	100%	53%	85%	94%	98%	86%
Pinterest	79%	100%	53%	85%	94%	98%	87%
Telegram	79%	14%	53%	79%	94%	98%	87%
Linkedin	79%	100%	53%	65%	93%	96%	85%
Tumblr	79%	100%	52%	86%	94%	98%	87%
VK	79%	100%	38%	79%	94%	98%	87%
QQ	70%	100%	38%	85%	94%	98%	86%
Wechat	92%	86%	68%	97%	95%	99%	95%

Tabella 16: Test degli algoritmi di watermarking basati su stringa per immagini in bianco e nero

In Tabella 16 vengono mostrati i risultati del test, i quali portano alle stesse conclusioni tratte dal precedente esperimento con il watermark basato su stringa. In questi casi l'ottimizzazione e la compressione non migliorano il test ma bensì lo deteriorano, in quanto, probabilmente, andare a modificare una firma digitale distribuita su tutta la foto, e non più pochi bit nascosti nell'immagine, crea delle difficoltà all'algoritmo di decifratura nel momento del ripristino del watermark.
In questo test il risultato migliore viene raggiunto dall'algoritmo di Dugad che, invece, restituisce dei risultati che rasentano il 100%. Esso permette di caricare e scaricare un'immagine da un social network mantenendo il proprio watermark, intatto, con qualsiasi foto in bianco e nero a disposizione, preservando le proprietà dell'immagine.

4.2 Colori

Per quanto riguardagli algoritmi di watermarking a colori, non è stato possibile, per via dei troppi software a pagamento, ottenerne un numero elevato per gli esperimenti. Per questo motivo, il numero dei test è limitato ai pochi disponibili, che restituiscono comunque risultati interessanti.
Per quanto riguarda il metodo con cui sono state sviluppate le prove, esso è identico a quello precedentemente effettuato tramite gli algoritmi in bianco e nero, con la sola differenza, che le immagini sono a colori e supportano ora più formati (.PNG, .JPG, ecc. . .). Per questo motivo la prima cosa da controllare sono stai i tipi di formati supportati da ogni social network.
Alcuni di loro non accettano alcuni tipi di formato, motivo per cui sono stati scelti solo formati che potessero essere utilizzati da tutti i social, cioè .PNG e .JPG.
Lo studio si è anche preoccupato di controllare lo standard con cui vengono salvate tutte le immagini che vengono scaricate dai social network. Esso viene trasformato da tutti in .JPG, ad accezione di Google+ e Tumblr che ne mantengono il formato originale intatto.
A differenza degli algoritmi di watermarking in bianco e nero, questi non hanno categorie in cui dividersi, sono basati tutti su stringa con chiave. Oltre ad inserire il watermark nell'immagine, questo viene anche protetto con una password d'accesso, di modo che, solo conoscendola, l'utente può accedere alla firma digitale. Questa viene inserita al momento della cifratura del watermark e rimane al suo interno sino alla decifratura.
Un'altra dote degli algoritmi a colori a differenza di quelli in bianco e nero è data dalla capacità di contenere al loro interno un file di testo e non una semplice sequenza di caratteri. L'utente maligno oltre a conoscere la password deve riuscire a estrarre un contenuto che non è di poche stringhe ma può anche essere lungo migliaia di caratteri.
Per questo motivo, i risultati non possono essere calcolati come per i precedenti esperimenti in percentuale di successo nel ricostruire l'intera stringa, ma bensì devono restituire, integra, l'intera sequenza di caratteri.

4.2.1 Rassegna degli algoritmi

Di seguito viene analizzato nel dettaglio ogni algoritmo di watermarking provato:

- **OpenPuff:** Il migliore dei tre software. È basato sulla stenografia e permette di lavorare con una vasta quantità di formati di immagine. Il programma permette di inserire fino a tre password per crittografare i file di testo, così da permettere un ottimo livello di sicurezza. L'unico problema di questo software è la sua incapacità di lavorare con immagini di piccole dimensioni, questo problema non ha permesso di lavorare con foto, standard o più piccole di una certa dimensione, con alcuni social network. Inoltre, sempre per via di questo problema, in alcuni casi non è stato possibile inserire la cifratura a tre chiavi o testi di dimensioni troppo grandi, in quanto, all'aumentare delle dimensioni di questi, deve necessariamente aumentare anche la dimensione dell'immagine per contenerli.

- **OpenStego:** Questo software di stenografia permette di inserire un file di testo cifrato da una password solo per il formato .PNG. Questo è l'unico limite di un software che, a differenza del precedente, supporta qualsiasi dimensione dell'immagine su cui inserire il watermark.

- **SecretBook:** Questo programma ha la capacità di inserire un watermark sotto forma di testo all'interno di un'immagine direttamente sul social network. Il software è una estensione di Google Chrome e ha come unico svantaggio di lavorare solo con foto caricate prima su Facebook. Per questo motivo, lo scopo dei test non è stato quello di caricare e scaricare l'immagine e vedere se il watermark fosse applicato correttamente, ma bensì di andare a controllare se, scaricando l'immagine con la firma digitale, questa mantenesse ad un secondo caricamento su social network il watermark. Inoltre dopo averla caricata e scaricata da un altro SN, se ricaricandola su Facebook, la firma rimanesse integra.

Gli algoritmi sono stati tutti testati allo stesso modo, eccetto per SecretBook, dove invece un'immagine veniva prima caricata su Facebook, e dopo averle applicato il watermark, veniva scaricata seguendo successivamente l'iter dei test come per tutti gli altri software.

4.2.2 Test su immagini a colori

In Tabella 17, 18 e 19 sono mostrati i risultati dell'esperimento sulle immagini a colori per ogni dimensione. I test restituiscono vari risultati a seconda del punto in cui il watermark o viene perso o riesce ad essere recuperato. Di seguito una rassegna dei vari tipi di errore che vengono mostrati nelle tabelle:

- **Ok:** se il watermark è stato estratto correttamente;

- **Errore di dimensione:** se dopo aver provato ad aggiungere l'immagine al programma di cifratura di OpenPuff si riceve un errore di *"format carriers error"* dovuto alla dimensione troppo piccola di quest'ultima;

- **Errore di formato:** se il formato non viene più letto dal software (ad esemoio dopo una conversione da PNG a JPG di alcuni social network);
- **Errore di recupero:** se al momento della decifratura non si è in grado di estrarre il watermark in quanto danneggiato dalle operazioni sul SN;
- **Errore di scaling:** se dopo uno scaling dell'immagine, non si può mantenere il watermark. Questo accade solo utilizzando Secretbook, in quanto la dimensione con cui viene scaricata un'immagine da Facebook, dopo aver inserito la firma digitale, diventa 960 x 592;

Per quanto riguarda Openpuff, il problema maggiore è dovuto alla incapacità di inserire il watermark per immagini con grandezza troppo piccola. Per alcuni social network come Whatsapp e Telegram non si è in grado di decifrare il watermark nascosto. Detto questo, il programma si comporta egregiamente in diverse situazioni, tranne con le immagini grandi, in quanto, il watermark non è in grado di resistere allo scaling.
Nel caso in cui si applichi il formato .PNG a un'immagine su Openpuff, la situazione invece cambia. I social network, trasformando il formato in .JPG, danneggiano il watermark non permettendo l'estrazione.
Lo stesso vale per OpenStego, che risulta resistente come il precedente software allo scaling utilizzando il formato .PNG. Nei SN che mantengono questa caratteristica senza cambiare il formato al caricamento i risultati sono perfetti.
Per quanto riguarda invece SecretBook, si ha che in tutti i casi in cui non viene eseguito uno scaling dell'immagine, il watermark rimane intatto. Sennonché, in alcuni casi, come Telegram o VK, anche se l'immagine ha una dimensione per cui lo scaling non viene eseguito, la firma digitale viene persa.
La capacità, sia di SecretBook e sopratutto di OpenPuff, di restituire dei risultati positivi in buona parte dei social network fa ben sperare nel campo degli algoritmi di watermarking a colori. Se si riuscisse a superare il problema dello scaling e della compressione/ottimizzazione e per OpenPuff della dimensione dei file, si avrebbero due software per foto a colori in grado di applicare un watermark a tutte le immagini.

Social network	Openpuff .JPG	Openpuff .PGN	Openstego	Secretbook
Facebook	Errore di recupero	Errore di formato	Errore di formato	Ok
Instagram	Errore di recupero	Errore di formato	Errore di formato	Errore di scaling
Google+	Errore di recupero	Ok	Ok	Ok
Twitter	Errore di recupero	Errore di formato	Errore di formato	Errore di scaling
Whatsapp	Errore di recupero	Errore di formato	Errore di formato	Ok
Pinterest	Errore di recupero	Errore di formato	Errore di formato	Errore di scaling
Telegram	Errore di recupero	Errore di formato	Errore di formato	Errore di scaling
Linkedin	Errore di recupero	Errore di formato	Errore di formato	Errore di scaling
Tumblr	Errore di recupero	Ok	Ok	Ok
VK	Errore recupero	Errore di formato	Errore di formato	Errore di scaling
QQ	Errore recupero	Errore di formato	Errore di formato	Errore di scaling
WeChat	Ok	Errore di formato	Errore di formato	Ok

Tabella 17: Applicazione del watermark sulle immagini a colori di dimensioni grandi

Social network	Openpuff .JPG	Openpuff .PGN	Openstego	Secretbook
Facebook	Errore di recupero	Errore di formato	Errore di formato	Ok
Instagram	Errore di recupero	Errore di formato	Errore di formato	Errore di scaling
Google+	Ok	Ok	Ok	Ok
Twitter	Errore di dimensione	Errore di formato	Errore di formato	Errore di scaling
Whatsapp	Errore di recupero	Errore di formato	Errore di formato	Ok
Pinterest	Errore di dimensione	Errore di formato	Errore di formato	Errore di scaling
Telegram	Errore di recupero	Errore di formato	Errore di formato	Errore di scaling
Linkedin	Errore di dimensione	Errore di formato	Errore di formato	Errore di scaling
Tumblr	Ok	Ok	Ok	Ok
VK	Ok	Errore di formato	Errore di formato	Errore di recupero
QQ	Errore di recupero	Errore di formato	Errore di formato	ok
WeChat	Ok	Errore di formato	Errore di formato	Ok

Tabella 18: Applicazione del watermark sulle immagini a colori di dimensioni standard

Social network	Openpuff .JPG	Openpuff .PGN	Openstego	Secretbook
Facebook	Errore di dimensione	Errore di formato	Errore di formato	Ok
Instagram	Errore di dimensione	Errore di formato	Errore di formato	Ok
Google+	Ok	Ok	Ok	Ok
Twitter	Errore di dimensione	Errore di formato	Errore di formato	Errore di recupero
Whatsapp	Errore di recupero	Errore di formato	Errore di formato	Ok
Pinterest	Errore di dimensione	Errore di formato	Errore di formato	Errore di recupero
Telegram	Errore di recupero	Errore di formato	Errore di formato	Errore di recupero
Linkedin	Errore di dimensione	Errore di formato	Errore di formato	Errore di recupero
Tumblr	Ok	Ok	Ok	Ok
VK	Ok	Errore di formato	Errore di formato	Errore di recupero
QQ	Errore di dimensione	Errore di formato	Errore di formato	Ok
WeChat	Ok	Errore di formato	Errore di formato	Ok

Tabella 19: Applicazione del watermark sulle immagini a colori di dimensioni piccole

5 Conclusioni

Una quantità sempre più grande di file multimediali vengono caricati sui social network e tra questi le immagini occupano una parte o la totalità (Pinterest) di questo flusso. Ciò spinge a trovare nuovi metodi per proteggere le immagini attraverso un copyright inattaccabile e di semplice utilizzo ed è per questo motivo che abbiamo preso la strada del watermark all'interno delle immagini, perché è ragionevole pensare che ciò che non può essere visto è anche più difficile da trovare.
Come prima cosa la tesi ha spiegato e analizzato in che modo i vari tipi di algoritmi di watermarking sono in grado di rimanere all'interno di un'immagine e di garantire, in certi casi, una difficile rimozione del watermark. Si è poi spostata sul verificare come le immagini vengano gestite all'interno di un social network, di modo da controllarne il comportamento nel mantenere intatta la firma digitale.
Dopo aver controllato la necessità di effettuare questo studio, controllando le normative sulla privacy dei social network, è stata quindi affrontata l'ipotesi di un watermark applicato all'immagine dai questi. Effettuando diversi test, soprattutto per quanto riguarda la geolocalizzazione e la gestione da parte dei CDN dei file multimediali, si è andati a controllare se, cambiando il paese in cui l'immagine viene condivisa, potessero essere alterati i valori interni di un'immagine quali SHA1, metadata e bit-to-bit.
In fine i test effettuati sui vari algoritmi di watermarking trovati in rete, hanno permesso di formulare un quadro di insieme di come i social network elaborino le immagini dopo il caricamento e quindi di come alterino la nostra firma digitale.
Andando a classificare i vari tipi di watermark che possono essere aggiunti ad un'immagine, sono stati analizzati diversi metodi applicabili ai SN, ma i più efficaci si sono rivelati quelli che utilizzano la tecnica DCT. Lavorare con i social network infatti, comporta la modifica della dimensione in altezza e larghezza di un'immagine, incorrendo in una perdita o aumento dei bits di questa dovuta a una compressione JPEG o ad una ottimizzazione della qualità. Con questo algoritmo è quindi possibile prevenire i problemi di compressione o quanto meno limitarli, mantenendo in tutti i casi parte della firma inalterata.
Investigare su come i social network elaborassero le immagini e le restituissero al momento del download, ha permesso di scoprire che nessun sito, a parte Facebook e Instagram, inseriscono una qualche firma nelle immagini. Per questi SN è infatti emersa la possibilità di una aggiunta, da parte del sito stesso, di un identificativo univoco all'interno dei metadata. Questo non significa che Facebook e Instagram inseriscano un proprio watermark nelle immagini, anche perché, quando l'immagine viene ricaricata sul social network, questa svuota i propri metadata e ne aggiunge di nuovi cambiando i valori dei campi, ma bensì, che Facebook e Instagram potrebbero essere in grado di monitorare un'immagine all'interno del proprio SN sapendo sempre a quale utente questa appartenga.
La geolocalizzazione mostra però che ad essere monitorate sono sole le informazioni del profilo utente, mentre, per quanto riguarda i CDN, non è stato possibile dire con certezza che le immagini non subiscano alcuna modifica, in quanto alcune API di siti che offrono questo servizio

permettono di estrarre informazioni sul luogo in cui è stata caricata l'immagine e sul profilo utente senza però inserire nulla nei metadata. I test effettuati al livello dei pixels di una foto hanno invece visto modifiche sono nei casi di ottimizzazione o ridimensionamento.
Nulla di permanente quindi viene inserito in un'immagine e gli unici cambiamenti sono dovuti alla compressione e all'ottimizzazione, che vengono effettuate dai social network al momento del caricamento dell'immagine.
Non avendo quindi trovato nulla che potesse intendere l'applicazione di un algoritmo di watermark all'interno delle immagini, sono stati eseguiti i test dei vari tipi di algoritmi di watermarking.
Avere una buona quantità di algoritmi in bianco e nero, ha permesso di effettuare una revisione abbastanza dettagliata di come questi si comportino con i social network e come, a seconda delle loro proprietà, siano in grado di resistere o meno alla compressione JPEG e all'ottimizzazione applicata da quest'ultimi. Si può inoltre affermare che queste due alterazioni delle immagini sono gli unici motivi per cui il watermark fallisce.
Alcuni degli algoritmi utilizzati hanno dato risultati perfetti con alcune immagini del campione utilizzato per i test e possono essere utilizzati pur non raggiungendo la completa ricostruzione della firma con tutti i campioni.
Analizzando le immagini a colori invece, non sono stati trovati algoritmi che eludono l'ostacolo della compressione JPEG, ma che si comportano egregiamente con file salvati in formato PNG. Lavorare con immagini che devono sottostare a determinate regole per poter inserire il watermark inoltre, limita l'utilizzo di questi algoritmi, esulando dallo scopo della tesi di avere algoritmi funzionanti in ogni situazione.
In conclusione, prendendo visione di tutto il lavoro svolto in questa ricerca, grandi passi devono essere ancora fatti nel campo del watermarking digitale. Gli algoritmi presi in considerazione erano troppo pochi per affermare che questo tipo di stenografia è un ottimo metodo di salvaguardia del copyright. Ciononostante, la tesi è in grado di dare delle indicazioni sui campi dove concentrare gli sforzi, infatti, gli algoritmi resistenti a compressione JPEG sarebbero in grado di fornire risultati eccellenti nel campo dei social network.
Maggiore attenzione sarebbe anche da dare ai CDN, dove la ricerca ha dato dei risultati meno completi per via dell'impossibilità di richiedere dati in maniera diretta senza passare per i social network. È logico pensare che le immagini siano salvate in questi server con le informazioni di ogni utente, viste le API che permettevano di richiedere questi dati, ma senza risultati certi, queste rimangono solo supposizioni.

6 Appendice

Di seguito sono riportati i codici utilizzati per le prove effettuate in ambiente Linux. Il primo permette di estrarre informazioni su come vengono trattate le immagini nei social network, mentre il secondo permette di inserire e estrarre i vari algoritmi di watermarking dalle immagini.

6.1 Codice esperimenti sui social network

```
# Per la corretta esecuzione del programma, si dovra':
# 1- creare una cartella che contenga il profilo con cui hai eseguito l'esperimento e
    aggiungere a quest'ultimo tre sottocartelle (grandi, standard, piccole);
# 2- per ognuna di queste sottocartelle aggiungi tante sottocartelle quanti sono i social
    network su cui andrai a testare;
# 3- le immagini in ogni cartella devono avere il nome "originale" + "un qualche numero"
    (originale1.jpg) per le immagini di input e "mod" + "un qualche numero" (mod1.jpg)
    per le immagini di output;
# 4- avvia il programma con python script.py "cartella_con_profilo"
    "file_per_i_risultati" "cartella_per_metadata" .
# Il programma confrontera':
# 1- SHA1;
# 2- Bit-To-Bit;
# 3- Metadata;
# Ogni test verra' eseguito tra immagini di input e output delle stesso profilo e
    immagini di output di differenti profili

# Libreries
import sys, os, subprocess, tempfile
import cv2
import numpy as np
from photohash import *
from os.path import basename
from scipy.misc import imread
from scipy.misc import imsave
from scipy.linalg import norm
from scipy import sum, average
from PIL import Image
from PIL import ImageChops
import imagehash
from datetime import datetime
```

```
# generate the sha1 of the image
def getsha1(image):
    sha1 = ""
    with tempfile.TemporaryFile() as tempf:
        proc = subprocess.Popen(['sha1sum', image], stdout=tempf)
        proc.wait()
        tempf.seek(0)
        sha1 = tempf.read()[:40]
    return str(sha1)

# compare result of different sha1
def sha1compare(x, y):
    hash1 = getsha1(x)
    hash2 = getsha1(y)
    if hash1 == hash2:
        return "equals"
    else:
        return "different"

# compare the difference between two images
def equaldifference(im1, im2):
    im1 = Image.open(im1)
    im2 = Image.open(im2)
    return ImageChops.difference(im1, im2).getbbox() is None

# get result for the difference
def resultdifference(im1, im2):
    if equaldifference(im1, im2):
        return "equals"
    else:
        return "different"

# write al the result in the file text
def writeResult(result, f, name1, name2):
    f.write("Confronto " + name1 + " - " + name2 + "\n")
    if result[0] == result[1] == "equals":
        f.write("    I confronti sono tutti uguali! \n")
    elif result[0] == result[1] == "different":
        f.write("    I confronti sono tutti diversi! \n")
    else:
```

```
            f.write("      Confronto sha1: " + result[0] + "\n")
            f.write("      Confronto bit-to-bit: " + result[1] + "\n")

# create exif file for next test of all images
def exifWriter(image1, image2, name1, name2, folder):
    path1 = image1.split("/")
    path2 = image2.split("/")
    if os.path.isfile(folder + name1[:-4] + ".txt") == False and path1[4][:3] == "mod":
        os.system("exiftool " + image1 + " > " + folder + name1[:-4] + ".txt")
    elif os.path.isfile(folder + name2[:-4] + ".txt") == False and path2[4][:3] ==
        "mod":
        os.system("exiftool " + image2 + " > " + folder + name2[:-4] + ".txt")

# combine all the result of comparison between two image
def checker(image1, image2, file_result, social_compare):
    result = []
    name1 = image1.split("/")
    name1 = ("_").join(name1[-4:])
    name2 = image2.split("/")
    name2 = ("_").join(name2[-4:])
    result.append(sha1compare(image1, image2))
    result.append(resultdifference(image1, image2))
    writeResult(result, file_result, name1, name2)
    exifWriter(image1, image2, name1, name2, sys.argv[3])

# list all file in all directory starting by current
def lister():
    for root, dirname, filenames in os.walk(sys.argv[1]):
        for filename in filenames:
            filepath.append(os.path.join(root, filename))

# return if the rule to compare image is satisfied
def rule(name1, name2):
    if (name1[1] == name2[1] and name1[2] == name2[2] and name1[3] == name2[3] and
        name1[4][-5:] == name2[4][-5:] and name1[4][:3] == "mod" and name2[4][:3] ==
        "ori") or (name1[1] != name2[1] and name1[2] == name2[2] and name1[3] ==
        name2[3] and name1[4] == name1[4] and name1[4][:3] == "mod" and name2[4][:3]
        == "mod" and name1[4] == name2[4]):
        return True
    else:
```

```
        return False

# use to know the status of calculation
def sizeChecker():
    i = 1
    for pathImage1 in filepath:
        for pathImage2 in filepath:
            name1 = pathImage1.split("/")
            name2 = pathImage2.split("/")
            if rule(name1, name2):
                i += 1
    return i

# make print of all calculation in a text file of results
def loopping (size):
    i = 1
    startTime = datetime.now()
    file_result = open(sys.argv[2], "a")
    for pathImage1 in filepath:
        for pathImage2 in filepath:
            name1 = pathImage1.split("/")
            name2 = pathImage2.split("/")
            percent = (float(i)*100)/float(size)
            if rule(name1, name2):
                sys.stdout.write("\033[F")
                print str(percent)[:4] + "%    " + str(datetime.now() - startTime)[:6]
                checker(pathImage1, pathImage2, file_result, name1[4])
                i += 1
    file_result.close()
    sys.stdout.write("\033[F")
    print "100.0%    " + str(datetime.now() - startTime)[:6]

# launch all the calculations
def main():
    print "0.00%    0:00:00"
    lister ()
    size = sizeChecker()
    loopping(size)

#START OF THE PROGRAM
```

```
# save all the path of file inside
filepath = []
if __name__ == "__main__":
    main()
```

6.2 Generazione degli algoritmi di watermarking in bianco e nero

```
# Per la corretta esecuzione del programma, si dovra':
# 1- creare una cartella che contenga le immagini in .pgm da utilizzare per inserire il
    watermark;
# 2- creare una cartella per ogni algoritmo di watermarking e inserire al suo interno i
    file per eseguire l'algoritmo;
# 3- avvia il programma con python script.py "cartella_con_immagini"
    "cartella_con_algoritmo" "carattere_di_embedding_o_de-embedding".
# Il programma funziona nel seguente modo:
# 1- Per prima cosa viene generata un appropriato file di firma da utilizzare
    successivamente nel algoritmo di inserimento del watermark tramite il comando
    "gen_coI_sig > cox.sig" (preso Cox come esempio di uno degli algoritmi di
    watermarking). Il file di output contiene dei parametri e una sequenza di numeri
    random di una distribuzione Gaussiana (la sequenza di watermark). Il programma
    genera solitamente una discreto livello di firma per un'immagine di 8-bit in gray-scale
    con dimensione 512 x 512.
# 2- L'inserimento del watermark nell'immagine avviene tramite il comando "wm_coI_e
    -s cox.sig -o coI_image.pgm image.pgm". La firma viene parsata per ottenere la
    particolare sequenza di watermark da adottare nell'immagine. L'immagine con il
    watermark viene inserita nel file coI_image.pgm ;
# 3- Per estrarre la firma dall'immagine viene eseguito il comando "wm_coI_d -s
    cox.sig -i image.pgm -o cox.wm coI_image.pgm". Siccome l'algoritmo di Cox preso in
    esempio non e' cieco, abbiamo bisogno dell'immagine originale come riferimento per
    estrarre la firma, altrimenti il programma scrivera' il comando "wm_coI_d -s bryun.sig
    -o bryun.wm bryun.pgm". La firma scaricata dall'immagine verra' inserita in cox.wm ;
# 4- La firma originale cox.sig verra' utilizzata per confrontare la firma scaricata
    cox.wm. Il risultato e' solitamente un fattore di correlazione tra 0 e 1, che indica
    la quantita' di firma trovata, con 1 lo stato di successo massimo. Il comando
    utilizzato e' il seguente "cmp_coI_sig -s cox.sig cox.wm".

import os, sys, subprocess

# Obtain the image path of all images
```

```
def imagePath(folder):
    for root, dirname, filenames in os.walk(folder):
        for filename in filenames:
            pathimage.append(os.path.join(root, filename))

# Create the new signature deembedding of the mark in the image
def dembedAlgorithm(name, image):
    nameImage = image.split("/")
    nameImage = nameImage[2]
    # if we don't need the original image
    if name == "bruyn" or name == "kund3" or name == "dugad" or name == "frid2"
        or name == "koch" or name == "kund2" or name == "xie" or name == "xie2":
        os.system("./wm_" + name + "_d -s " + nameImage[:-4] + ".sig -o " +
            nameImage[:-4] + ".wm " + name + "_" + nameImage)
    # if we need the original image
    else:
        os.system("./wm_" + name + "_d -s " + nameImage[:-4] + ".sig -i" + " " +
            image + " -o " + nameImage[:-4] + ".wm " + name + "_" + nameImage)
    proc = subprocess.Popen(["./cmp_" + name + "_sig -s " + nameImage[:-4] + ".sig "
        + nameImage[:-4] + ".wm"], stdout=subprocess.PIPE, shell=True)
    (out, err) = proc.communicate()
    f = open("result.txt", "a")
    f.write(nameImage[:-4] + " \n" + out + " \n")
    f.close()

# Create the signature and embed the watermark in the image
def embedAlgorithm(name, image):
    nameImage = image.split("/")
    nameImage = nameImage[2]
    os.system("./gen_" + name + "_sig > " + nameImage[:-4] + ".sig")
    os.system("./wm_" + name + "_e -s " + nameImage[:-4] + ".sig -o " + name +
        "_" + nameImage + " " + image)

def main():
    imagePath(sys.argv[1])
    # embed mark
    if checker == "e":
        for image in pathimage:
            embedAlgorithm(sys.argv[2], image)
    #de-embed mark
```

```
    else:
        for image in pathimage:
            dembedAlgorithm(sys.argv[2], image)

#START OF THE PROGRAM
pathimage = []
# If is "e" so embed altrought is "d" dembed
checker = ""
checker = sys.argv[3]
if __name__ == "__main__":
    main()
```

Riferimenti bibliografici

[1] Rahim Ansari, Mrutyunjaya M Devanalamath, K Manikantan, and S Ramachandran. Robust digital image watermarking algorithm in dwt-dft-svd domain for color images. In *Communication, Information & Computing Technology (ICCICT), 2012 International Conference on*, pages 1–6. IEEE, 2012.

[2] Olivier Bruyndonckx, Jean-Jacques Quisquater, and Benoit Macq. Spatial method for copyright labeling of digital images. *Proc. IEEE Nonlinear Signal and Image Processing*, pages 456–459, 1995.

[3] Marco Corvi and Gianluca Nicchiotti. Wavelet-based image watermarking for copyright protection. In *Scandinavian conference on image analysis*, pages 157–163, 1997.

[4] Ingemar J Cox, Joe Kilian, F Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing*, 6(12):1673–1687, 1997.

[5] CIPA DC-008-Translation-2012. Excangable image file format for digital still cameras. `http://www.cipa.jp/std/documents/e/DC-008-2012_E.pdf`, 2012.

[6] Rakesh Dugad, Krishna Ratakonda, and Narendra Ahuja. A new wavelet-based scheme for watermarking images. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 2, pages 419–423. IEEE, 1998.

[7] Jessica Fridrich. Combining low-frequency and spread-spectrum watermarking. In *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*, pages 2–12. International Society for Optics and Photonics, 1998.

[8] Ming-Huwi Horng and Ting-Wei Jiang. The codebook design of image vector quantization based on the firefly algorithm. In *International Conference on Computational Collective Intelligence*, pages 438–447. Springer, 2010.

[9] Jong Ryul Kim and Young Shik Moon. A robust wavelet-based digital watermarking using level-adaptive thresholding. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 2, pages 226–230. IEEE, 1999.

[10] Eckhard Koch and Jian Zhao. Towards robust and hidden image copyright labeling. In *IEEE Workshop on Nonlinear Signal and Image Processing*, pages 452–455. Neos Marmaras, Greece, 1995.

[11] Deepa Kundur and Dimitrios Hatzinakos. Digital watermarking using multiresolution wavelet decomposition. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 5, pages 2969–2972. IEEE, 1998.

[12] Deepa Kundur and Dimitrios Hatzinakos. Diversity and attack characterization for improved robust watermarking. *IEEE Transactions on Signal Processing*, 49(10):2383–2396, 2001.

[13] Martin Kutter, Sviatoslav V Voloshynovskiy, and Alexander Herrigel. Watermark copy attack. In *Electronic Imaging*, pages 371–380. International Society for Optics and Photonics, 2000.

[14] Chia-Feng Lin, Muh-Chyi Leu, Chih-Wei Chang, and Shyan-Ming Yuan. The study and methods for cloud based cdn. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011 International Conference on*, pages 469–475. IEEE, 2011.

[15] Ching-Yung Lin and Shih-Fu Chang. Semifragile watermarking for authenticating jpeg visual content. In *Electronic Imaging*, pages 140–151. International Society for Optics and Photonics, 2000.

[16] Shishir Nagaraja, Amir Houmansadr, Pratch Piyawongwisal, Vijit Singh, Pragya Agarwal, and Nikita Borisov. Stegobot: a covert social network botnet. In *International Workshop on Information Hiding*, pages 299–313. Springer, 2011.

[17] Ed Novak and Qun Li. A survey of security and privacy in online social networks. *College of William and Mary Computer Science Technical Report*, 2012.

[18] Christian Rey and J-L Dugelay. Blind detection of malicious alterations on still images using robust watermarks. In *Secure Images and Image Authentication (Ref. No. 2000/039), IEE Seminar on*, pages 7–1. IET, 2000.

[19] Salvatore Scellato, Cecilia Mascolo, Mirco Musolesi, and Jon Crowcroft. Track globally, deliver locally: improving content delivery networks by tracking geographic social cascades. In *Proceedings of the 20th international conference on World wide web*, pages 457–466. ACM, 2011.

[20] Houng-Jyh Wang, Po-Chyi Su, and C-C Jay Kuo. Wavelet-based digital image watermarking. *Optics Express*, 3(12):491–496, 1998.

[21] Yu Wu, Chuan Wu, Bo Li, Linquan Zhang, Zongpeng Li, and Francis Lau. Scaling social media applications into geo-distributed clouds. *IEEE/ACM Transactions on Networking (TON)*, 23(3):689–702, 2015.

[22] Xiang Gen Xia, Charles Boncelet, and Gonzalo Arce. Wavelet transform based watermark for digital images. *Optics Express*, 3(12):497–511, 1998.

[23] Liehua Xie and Gonzalo R Arce. Joint wavelet compression and authentication watermarking. In *ICIP (2)*, pages 427–431, 1998.

[24] Wenwu Zhu, Zixiang Xiong, and Ya-Qin Zhang. Multiresolution watermarking for images and video: a unified approach. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 465–468. IEEE, 1998.

[25] Athanasios Zigomitros, Achilleas Papageorgiou, and Constantinos Patsakis. Social network content management through watermarking. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 1381–1386. IEEE, 2012.