

Tehnici de verificare a autenticității imaginilor – ELA (Error Level Analysis)

Intro

Algoritm

Imagini -> Exemplu

Concluzii: de ce intr-o situatie e mai bine decat in alta, constrangeri etc

Intro

Evoluția rapidă a tehnologiilor de imagistică digitală din ultimii ani a condus la o utilizare foarte mare a imaginilor digitale pentru diverse scopuri. Acest lucru a condus la apariția a mai multor metode de manipulare și denaturare a conținutului imaginilor originale.

Modificarea imaginilor poate fi clasificată în trei tipuri:

- Îmbinare de imagini – este un proces de combinare a două sau mai multe imagini, rezultând o nouă imagine. Spre exemplu, o anumită regiune dintr-o imagine este copiată într-o altă imagine.



Fig. 1 Falsificarea unei imagini prin îmbinare

- Falsificare prin copiere-mutare – este un proces de copiere a unei regiuni dintr-o imagine în cadrul aceleiași imaginii. Regiunea copiată este de obicei modificată prin operații de rotație, scalare sau adăugare de zgomot.

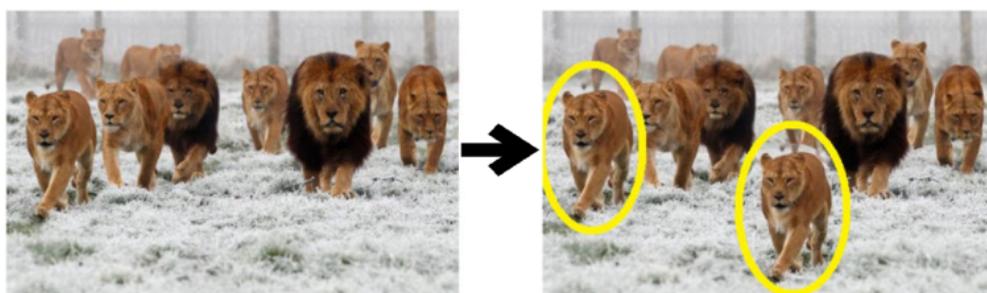


Fig. 2 Falsificarea unei imagini prin copiere-mutare de regiuni

- Retușarea imaginii – este un proces de retușare a pixelilor copiați astfel încât să se potrivească cu pixelii din jur. Acest lucru poate însemna îmbunătățirea sau reducerea unor caracteristici ale imaginii originale, fără a modifica sensul imaginii. Spre exemplu, această metodă poate fi întâlnită la imaginile de pe coperta anumitor reviste, pentru a retușa anumite părți ale corpului unei persoane pentru ca aceasta să pară mai atractivă.

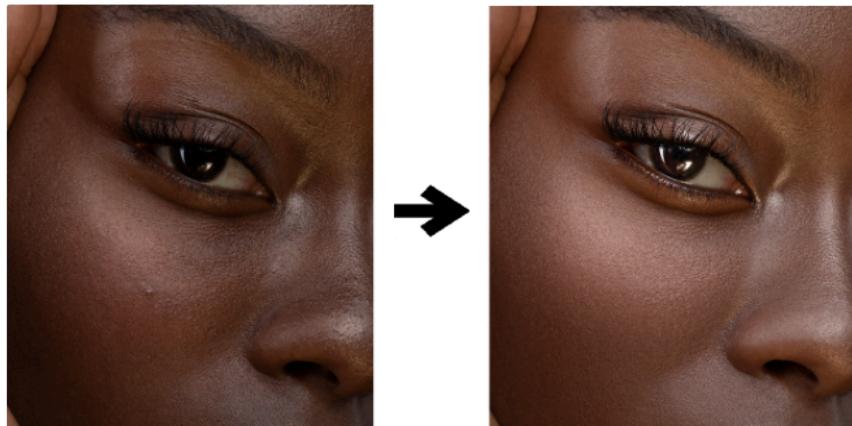


Fig. 3 Falsificarea unei imagini prin retușare (prelucrarea pixelilor)

Pentru a combate aceste acțiuni, s-a dezvoltat un nou domeniu, denumit expertiza criminalistică a imaginilor.

Expertiza criminalistică a imaginilor are ca obiect identificarea originii (identificarea aparatelor de luat vederi/de fotografiat/de filmat, determinarea condițiilor de folosire a acestora, etc.) și verificarea autenticității unei imagini. Studiile acestui domeniu sunt clasificate în două tipuri:

- Autentificare activă – necesită informații suplimentare despre imaginea originală (de exemplu extragerea unei caracteristici unice a imaginii, care să reprezinte semnătura imaginii)
- Autentificare pasivă – este o tehnică de detectare a falsurilor care nu necesită informații suplimentare despre imaginea originală. Această metodă se împarte la rândul ei în cinci categorii:
 - Detectarea falsificării imaginilor la nivel de pixeli – detectează anomalii statistice introduse la nivel de pixel într-o imagine digitală și se împarte în patru categorii: clonare (copiere-mutare), reeșantionare (redimensionare), îmbinare și statistic.
 - Detectarea falsificării imaginilor bazate pe format – utilizează corelațiile statistice introduse de o schemă de compresie cu pierderi specifică. Acest tip de metodă funcționează în principal pentru imaginile în format JPEG și se împarte, la rândul ei, în trei categorii: cuantizare JPEG, JPEG dublu, JPEG blocat
 - Detectarea falsificării imaginilor bazate pe cameră (aparatul de fotografiat) – exploatează artefactele introduse de obiectivul camerei, senzorul camerei sau de post-procesarea pe cip. Atunci când o imagine este captată de o cameră digitală, aceasta se deplasează de la senzorul camerei în memorie și trece printr-o serie de pași de procesare precum cuantizarea, corelarea culorilor, corecția gama, filtrare și comprimare (acești pași de procesare variază în funcție de modelul și artefactele fiecărei camere). Această metodă se împarte în patru categorii: aberație cromatică, matrice de filtre de culoare, răspunsul camerei și zgomotul senzorului

- o Detectarea falsificării imaginilor bazate pe mediul înconjurător fizic – modelează și detectează anomalii în interacțiunea 3D dintre obiectele fizice, lumină și aparatul de fotografiat. Această metodă se bazează în special pe mediul de iluminare în care este captat un obiect sau o imagine, deoarece iluminarea este foarte importantă pentru captarea unei imagini. Această metodă se poate axa pe trei direcții diferite, și anume: direcția luminii 2D, direcția luminii 3D și mediul de iluminare.
- o Detectarea falsificării imaginilor bazate pe geometrie – măsoară obiectele din lumea reală și poziția lor raportată la aparatul de fotografiat. Aceste metode pot fi ramificate în: metodă bazată pe punctul principal sau bazată pe măsurători metrice.

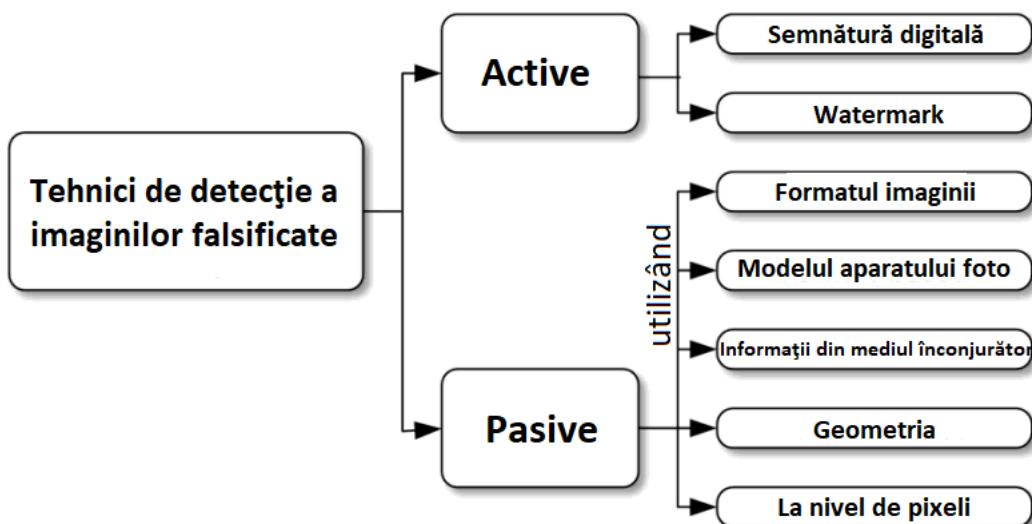


Fig. 4 Tehnici de detecție a imaginilor falsificate

Așa cum am expus mai sus, au fost dezvoltate multiple metode de detecție a falsurilor în imagini. Câteva exemple de astfel de metode sunt: analizarea erorii JPEG, filtrul median, Transformata Cosinus Discretă îmbunătățită (DTC), detectarea bazată pe DCT pe mai multe manipulații în lanț, estimarea pasului de cuantizare, etc.

Deoarece formatul JPEG (Joint Photographic Experts Group) este unul dintre cele mai utilizate formate, în special pentru site-urile web, mulți dintre acești algoritmi au fost gândiți în special pentru formatul JPEG.

Una dintre cele mai de succes tehnici de detectare a autenticității unei imagini este Analiza nivelului de eroare (ELA – Error Level Analysis).

Compresia și decompresia JPEG

JPEG este o tehnică de compresie cu pierderi, ce combină domeniul spațial cu domeniul frecvenței.

În procesul de compresie, imaginea inițială este împărțită în blocuri de dimensiune 8x8 și apoi îi este aplicată transformarea culorilor și subeșantionarea pentru a reduce datele de procesare. Apoi se trece în domeniul frecvență prin aplicarea transformatei DCT. Coeficientul DCT este ordonat în zig-zag și cuantificat utilizând tabelul de cuantizare. Codarea entropiei fără pierderi comprimă coeficienții de cuantizare pentru a forma un nou fișier JPEG.

Procesul invers are loc pentru decompresia JPEG. Mai întâi are loc decodificarea entropiei pentru a obține valorile cuantizate exacte ale coeficienților. Apoi, aceste valori sunt multiplicate conform tabelului, rezultând coeficientul decuantizat. În final se aplică transformata DCT inversă pentru a obține valorile pixelilor și se trece în spațiul de culoare necesar.

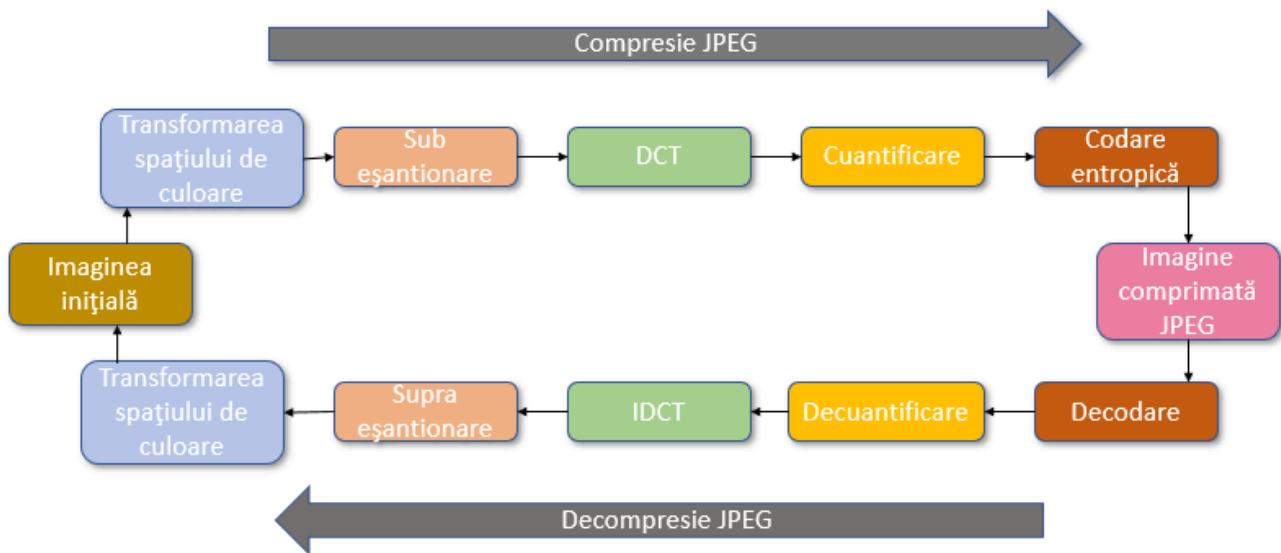


Fig. 5 Procesul de compresie și decompresie JPEG

Mărimea imaginii de compresie JPEG depinde de conținutul și calitatea imaginii. Imaginea omogenă cu puține obiecte necesită mai puțini biți în comparație cu o imagine de scenă complexă cu multe obiecte.

Majoritatea aplicațiilor de manipulare de imagine, precum Adobe Photoshop sau GIMP, vor resalva imaginea după modificare ca o nouă imagine JPEG.

ELA

ELA este un algoritm de compresie JPEG pentru detecția manipulării de imagini. Așa cum a fost prezentat în secțiunea Compresia JPEG, coeficientul frecvență al imaginii este cuantificat în funcție de tabelul de cuantizare, urmat de un proces de codare entropică. Neal Krawetz a adoptat acest proces în dezvoltarea unui algoritm care să aproximeze calitatea imaginilor JPEG. Calculul calității JPEG este definit ca fiind diferența dintre valoarea medie a tabelului de cuantizare a luminanței Y și cel a crominanței C_bC_r .

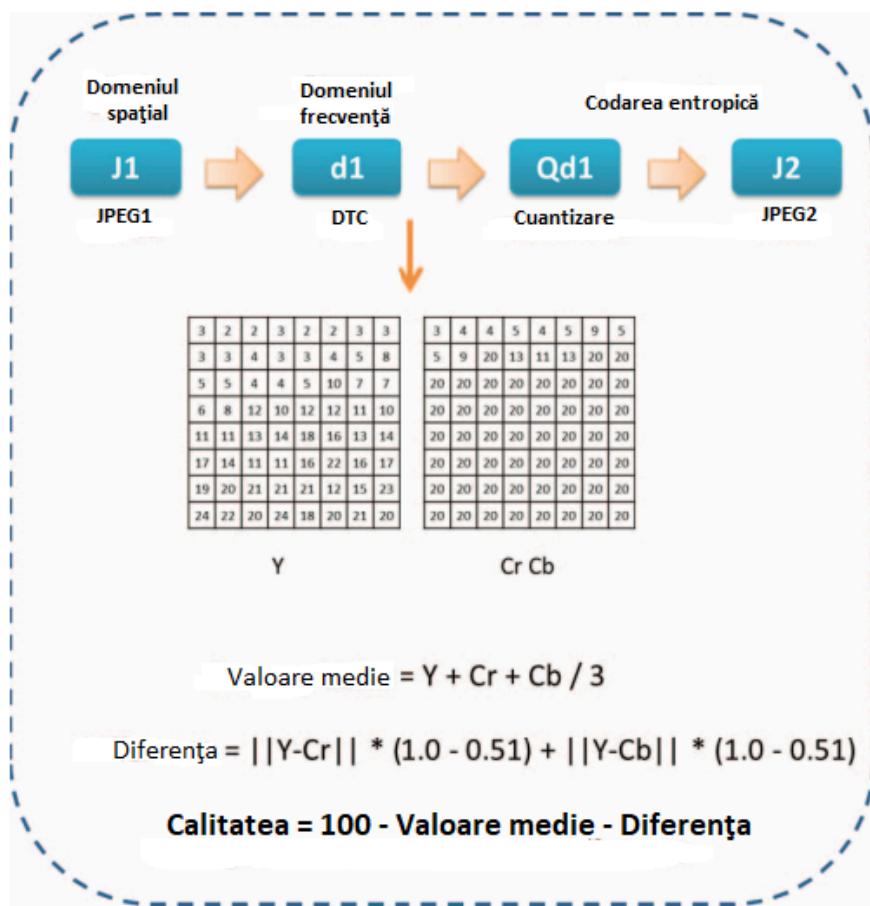
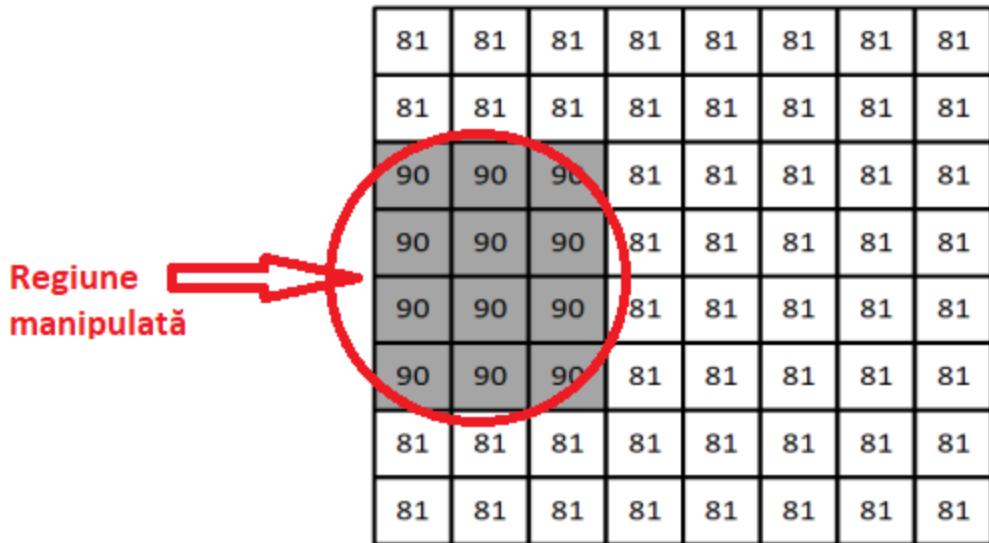


Fig. 6 Algoritmul de aproximare a calității JPEG

Utilizând algoritmul ELA, imaginea este împărțită în blocuri de dimensiune 8x8 pixeli și apoi este recomprimată independent cu o rată de eroare cunoscută (de exemplu 95%), pentru a face diferență între cele două (imaginea inițială și imaginea recomprimată). Dacă imaginea nu este modificată, fiecare bloc ar trebui să aibă aproximativ aceeași rată a calității. Dacă imaginea este modificată, regiunea din imagine manipulată va avea un nivel de eroare mai mare față de restul imaginii.



81	81	81	81	81	81	81	81
81	81	81	81	81	81	81	81
90	90	90	81	81	81	81	81
90	90	90	81	81	81	81	81
90	90	90	81	81	81	81	81
90	90	90	81	81	81	81	81
81	81	81	81	81	81	81	81
81	81	81	81	81	81	81	81

Fig. Valoarea calității obținută prin algoritmul ELA

Nivelul de eroare este dat de informația pierdută la salvarea imaginii în format JPEG. Gradul de eroare va crește pe măsură ce imaginea este salvată de mai multe ori. După un anumit număr de operații de salvare a imaginii, blocul poate atinge nivelul de eroare minim local, doar dacă nu există modificări în nivelul de calitate.

Când o imagine este captată de un aparat de fotografiat, aceasta este comprimată pentru prima oară în format JPEG și salvată în memoria acestuia. Dacă această imagine este ulterior deschisă într-un software precum Photoshop sau GIMP, editată și apoi salvată tot în format JPEG, ea va fi comprimată iarăși. Din acest proces reiese că imaginea originală a fost comprimată de două ori (o dată de aparatul de fotografiat și o dată de Photoshop/GIMP), însă regiunea editată a imaginii a fost comprimată o singură dată (de către Photoshop/GIMP).

Testare algoritm

Pentru testarea algoritmului ELA, am folosit 5 imagini în format JPEG din setul de date CASIA/Sp, preluate de pe <https://www.kaggle.com/sophatvathana/casia-dataset>.



Sp_D_CNN_A_ani0049_ani0084_0266



Sp_D_CNN_A_sec0041_pla0037_0098



Sp_D_CNN_R_art0044_art0018_0278



Sp_D_CRN_A_arc0082_ani0076_0378



Sp_D_CNN_A_nat0071_ani0024_0270

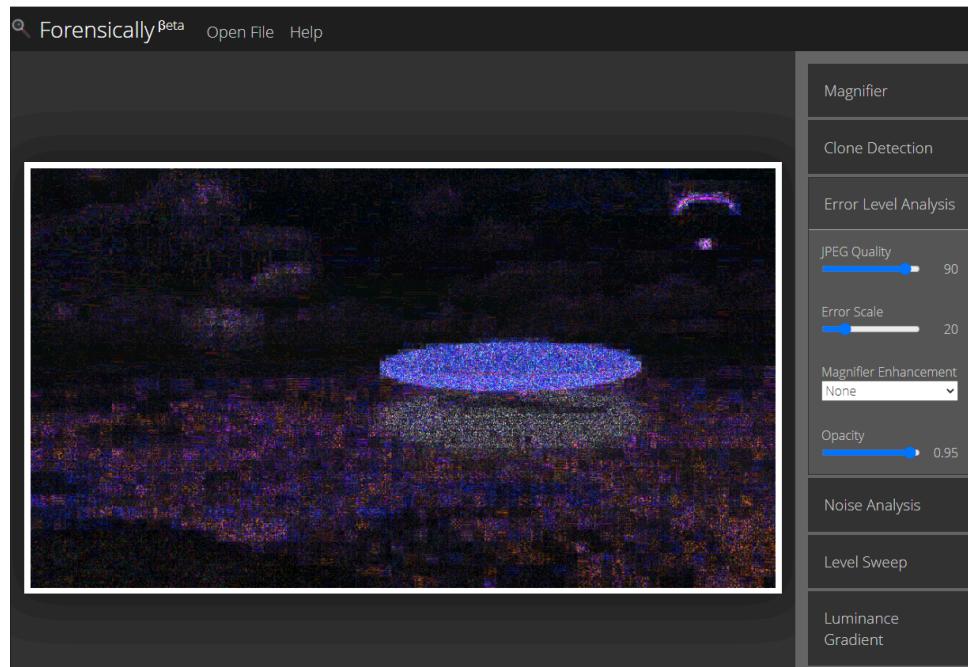
Fig. Imaginele din setul de date CASIA/Sp utilizate pentru testarea algoritmului ELA

Pentru a testa algoritmul ELA, am folosit trei platforme distincte:

- Aplicația 1: Site-ul <http://fotoforensics.com/> - permite introducerea unei imagini pentru analiză și returnează rezultatul obținut de algoritm (nu are alte opțiuni).



- Aplicația 2: Site-ul <https://29a.ch/photo-forensics/#error-level-analysis> – permite introducerea unei imagini și returnează rezultatul obținut; permite modificarea calității JPEG, rata erorii, opacitatea imaginii și adăugarea unei îmbunătățiri (egalizare de histogramă sau auto contrast).



- Aplicație 3: Aplicație rulată în Pycharm (deschisă în browser prin intermediul bibliotecii streamlit) – permite încărcarea unei imagini și returnează rezultatul algoritmului ELA; permite alegerea nivelului de recompresie și are opțiunea de a adăuga blur Gaussian de intensitate (radius) = 2 asupra imaginii.

Project File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - main.py

pythonProject > main.py

Project pythonProject C:\Users\Georgiana\PycharmProjects\pythonProject

main.py x

Package requirements 'streamlit==0.73.1', 'Pillow==8.1.0' are not satisfied

Install requirements Ignore requirements

```

1 import streamlit as st # streamlit 0.73.1
2 from PIL import Image, ImageChops, ImageEnhance, ImageFilter, ImageOps # pillow 8.1.0
3
4 st.title("ELA - Error Level Analysis")
5 markdown = """Aplicație care testează metoda Error Level Analysis de evidențiere a editărilor în
6 st.markdown(markdown)
7
8 upload_file = st.file_uploader("Încarcă o imagine", type=['jpg'])
9

```

Terminal: Local x + v

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/powershell>

PS C:\Users\Georgiana\PycharmProjects\pythonProject> streamlit run C:/Users/Georgiana/PycharmProjects/pythonProject/main.py

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>
Network URL: <http://192.168.2.6:8501>

ELA - Error Level Analysis

Aplicație care testează metoda Error Level Analysis de evidențiere a editărilor în imagini

Încarcă o imagine



Drag and drop file here

Limit 200MB per file • JPG

[Browse files](#)

Aplică blur

Alege nivelul de recompresie:



Experiment 1:

S-a testat imaginea Sp_D_CNN_R_art0044_art0018_0278 pe Aplicația 1, astfel: s-au introdus imaginea inițială. Imaginea în rezoluție 50%, imaginea în rezoluție 10% (modificarea rezoluției s-a efectuat utilizând aplicația GIMP):