

**SPECIALIZAREA INFORMATICĂ**

**Lucrare de licență**

**NotInSight – Aplicație de Steganografie**

**Absolvent**

**Vlad Cornoiu**

**Coordonator științific**

Conf. Dr. Ruxandra Olimid

**București**

**Iulie 2020**

**Abstract**

În prezent, securitatea și confidențialitatea datelor personale sau sensibile rămâne constant una dintre necesitățile prioritare în transferul informației pe internet. Din acest motiv, subiectul este într-o continuă dezvoltare, atât prin cercetările realizate de specialiștii în domeniu, cât și prin cele ale utilizatorilor canalelor de comunicare.

Scopul principal al acestei lucrări este acela de a furniza cunoștinte de bază cu privire la steganografia digitală dar și aprofundări cu privire la acest subiect. De asemenea, lucrarea propune o soluție pentru ascunderea informațiilor personale în transferurile de date pe internet. Mai specific, proiectul este constituit dintr-un serviciu pregătit să incorporeze date într-un fișier imagine. Acest proces este realizat printr-o serie de algoritmi aplicați asupra informației reprezentată în diferite forme: domeniul spațial și domeniul frecvențelor. Modulul conține de asemenea și soluția de decodificare a datelor secrete introduse, transformându-l într-un produs complet utilizabil pentru comunicarea pe internet.

Implementarea practicilor de steganografie a fost realizată în trei etape: transformarea datelor vizuale din domeniul spațial în cel al frecvențelor, depistarea zonelor potrivite pentru inserarea mesajului, introducerea mesajului în zonele identificate anterior și scrierea datelor sub forma compresată. Decodificarea mesajului secret este realizată prin reversul practicilor descrise anterior.

Pentru un plus de securitate, mesajului privat i-a fost aplicat un algoritm de criptare pentru o mai mare siguranța a datelor personale ce sunt transmise.

Lucrarea prezintă in final o serie de teste și comparatii ce au fost realizate în timpul perioadei de explorare, verificând încă o dată faptul că metodele selectate sunt un bun punct de plecare în steganografia digitală.

**Abstract**

In the present, security and privacy stand consistently as top priorities in personal use data-transfer all around the internet. For this reason, the subject holds an ever-improving state, not only by the research undertook by domain specialists, but also by daily-basis covering techniques trials of communication channels’ mere users.

The main purpose of this paper is to provide some basic knowledge of the practice of digital steganography and further insights within the aspects of this concept. Furthermore, this paper proposes a quick-to-handle solution for covering personal information throughout internet data transfer. More specific, this project introduces a microservice ready to embed a private data into an image file.

This process is materialized through a series of algorithms and procedures applied to data, both in spatial and frequency domain. This microservice is also ready to decode the secret message as well, making it feasible for the communication process on the internet.

The implementation of the steganography practices was achieved by bringing together the transformation from spatial to frequency domain, retrieval of areas which best fit the secret message, insertion of the secret message file in the aforementioned zones and writing the data in a compressed manor. Message retrieval was achieved by reversing the procedures described above.

The secret text was encrypted followed by the implementation of a shuffle algorithm process so that one may be more confident about the privacy of their transferred personal data.

The paper also presents a series of tests and comparisons that were made during the research period, proving once more that the selected methods holds a good starting point for future research in digital steganography.

Cuprins

[1. Introducere 5](#_Toc44652013)

[1.1. Motivatie 5](#_Toc44652014)

[1.2. Obiective 5](#_Toc44652015)

[1.3. Istoric 5](#_Toc44652016)

[1.4. Structura lucrarii 6](#_Toc44652017)

[1.5. Contributie personala 7](#_Toc44652018)

[2. Fundamente teoretice 9](#_Toc44652019)

[2.1. Imagine digitala si procesare de imagini 9](#_Toc44652020)

[2.2. Domeniu al frecventei (DCT) 10](#_Toc44652021)

[2.3. Steganografie 12](#_Toc44652022)

[2.4. Algoritmul LSB 13](#_Toc44652023)

[2.5. Algoritmul de compresie JPEG 15](#_Toc44652024)

[2.6. Sistemul de criptare AES 20](#_Toc44652025)

[3. Implementarea aplicatiei 21](#_Toc44652026)

[3.1. Tehnologii folosite si motivatia alegerii lor 21](#_Toc44652027)

[3.2. Modul de utilizare 22](#_Toc44652028)

[3.3. Detalii de implementare 24](#_Toc44652029)

[4. Experimente si performante 27](#_Toc44652030)

[4.1. Metrici de performanta 27](#_Toc44652031)

[5. Concluzii si directii viitoare de dezvoltare 30](#_Toc44652032)

[Bibliografie 31](#_Toc44652033)

# Introducere

## Motivație

Problema ce se situează la baza acestei lucrări este aceea a confidențialității datelor. Mai exact, este vorba despre dorința progresivă a persoanelor de a fi cât mai siguri în legătură cu păstrarea sau transferul informațiilor personale. Auzim deseori nevoia persoanelor din jur de a deține o metoda cât mai avansată de a-și păstra parolele, de a identifica un document sau o suită de date personale, de a ascunde informația față de receptori nedoriți sau de a comunica fără vreo grijă pe internet.

Desigur, pentru aceste situații, existența metodelor standardizate de criptare a datelor și informațiilor, precum și a protocoalelor și mecanismelor de comunicare ce nu permit terțelor să intervină poate construi un compromis destul de mic pentru majoritatea persoanelor. De asemenea, există posibilitatea nevoii de a transmite un mesaj a cărui însăși existență să fie păstrată secretă pe un canal de comunicare public. Pentru acestea, cercetarea și dezvoltarea unor noi metode de a spori siguranța utilizatorilor cu privire la datele transmise și utilizate sunt în continuă desfășurare.

## Obiective

Această lucrare are ca scop prezentarea unei metode de integrare a unor informații personale într-o structură de date ce poate face obiectul unei expuneri publice. Pentru a aduce un plus metodelor precizate anterior, prezenta lucrare propune prezentarea unei aplicații construite în jurul conceptului de steganografie și utilizarea acesteia pentru a ascunde datele fără a se pune problema unor eventuale suspiciuni.

Obiectivul este ca un mesaj confidențial să fie înglobat într-o imagine inofensivă într-un mod imperceptibil ochiul uman, printr-o serie de algoritmi de compresie și manipulare a datelor.

## Istoric

Exista un număr relativ restrâns de aplicații ce și-au propus dezvoltarea unor algoritmi de steganografie pentru a ascunde date.

Un prim exemplu este aplicația Xiao Steganography dezvoltată de echipa Nakasoft in Venezuela [1]. Ajunsă până la versiunea 2.6.1 în Noiembrie 2007 (ultimul release), Xiao Steganography este un utilitar cross-platform ce ii pune la dispoziție utilizatorului opțiunea de a alege metoda de criptare a datelor (ex. DES, Triple DES, MD5, SHA-1) înainte ca acestea să fie ascunse în fișierul inofensiv. Aplicația suportă doar formaturile de imagini digitale bmp și wav.

Un alt exemplu este aplicația OutGuess, dezvoltată în anul 1999 de către Niels Provos și o echipă germană [2]. În această aplicație, mesajul secret este preluat aleatoriu și introdus în zonele vizuale cele mai puțin perceptibile de către ochiul uman. De asemenea, este folosit și un principiu numit „deniable encryption” ce presupune imposibilitatea atacatorilor de a demonstra că există o informație reală în spatele unui text criptat și deci, a criptării în sine [3].

De menționat este și Steghide, aplicație ce își propune incorporarea unor date într-unul din formaturile jpeg, bmp, wav [4]. Acest util ajuns la versiunea 0.5.1. in 2003 dezvoltă un algoritm de interschimbare a valorilor pixelilor, astfel încât extracția unor anumite secvențe de informație (biți) din acești pixeli să fie cât mai apropiată de informația ce se vrea a fi introdusă (criptată cu algoritmul Rijndael – AES). Cu alte cuvinte, propune o abordare de tip „pattern-matching” pe structuri de date arborescente. Această abordare a fost aleasă pentru a rezista unor atacuri de tip statistic.

Se poate concluziona, astfel, că există un număr scăzut de aplicații ce au ca scop ascunderea datelor unui utilizator. Este observabil că acestea nu se rezumă doar la introducerea informațiilor de tip text în imagini digitale, ci oferă un spectru mult mai larg al formaturilor acceptate. Astfel, această lucrare nu are ca scop competiția cu celelalte proiecte menționate mai sus, ci găsirea și utilizarea unor metode și proceduri cu scopul introducerii datelor într-un mod cât mai imperceptibil.

## Structura lucrarii

Lucrarea este structurată în 5 capitole, primul capitol fiind constituit de introducerea curentă.

Capitolul 2 are ca scop prezentarea teoriei ce a stat la baza implementarii aplicației către cititorii acestei lucrări. Pornind de la aspecte teoretice de bază despre structura steganografiei și principiile procesării imaginilor digitale, capitolul continuă cu prezentarea unor prelucrări mai avansate ale imaginii în domeniul frecvențelor urmând ca ulterior să fie discutați algoritmi de steganografie și criptografie.

Capitolul 3 precizează detaliile de implementare ale algoritmului de steganografie și descrierea aplicației practice. În subcapitolele acestuia se vor regăsi detalii despre tehnologiile auxiliare utilizate în implementarea aplicației cât și o descriere a interfeței și modului de întrebuințare al acesteia.

Capitolul 4 își propune prezentarea unor metrici de performanță standardizate pentru măsurarea zgomotului introdus într-o imagine și compararea performanțelor algoritmului prezentat în lucrarea curentă cu cele ale altor algoritmi ce incorporeaza procesul de steganografie.

Lucrarea se încheie cu Capitolul 5 ce prezintă concluzii cu privire la soluția propusă, utilitatea acesteia cât și metode de îmbunătățire pentru viitor.

## Contribuție personală

Aplicabilitatea conceptelor enunțate în această lucrare au fost demonstrate printr-un serviciu pus la dispoziția utilizatorului, acesta parcurgând toți pașii de transformare și integrare a mesajului ce se dorește a fi ascuns.

O mare provocare a acestei lucrări a fost implementarea structurii pentru a construi o imagine digitala într-un format ce acceptă algoritmul de compresie JPEG. Întrucât experiența relevantă pentru acest algoritm a fost una minimă, a fost nevoie de multă documentație pentru a înțelege modul în care acesta funcționează și cum poate fi manipulat.

Deși implementarea tuturor algoritmilor este realizată cu succes de către aplicație, nivelul de robustețe nu este la nivel de producție, astfel încât rămâne pentru moment la stadiul de dovadă a conceptelor teoretice descrise în lucrarea propriu-zisă.

# Fundamente teoretice

## Imagine digitală și procesare de imagini

Definitia unei imagini poate fi formalizata prin prisma unei functii bidimensionale de forma unde x si y sunt coordonate spatiale, iar valoarea functiei pentru parametri (x,y) este reprezentata de intensitatea imaginii la coordonatele propriu-zise. Discretizand valorile , si prin procesele de esantionare (restrangerea domeniului de definitie) si cuantizare (restrangerea codomeniului), imaginea devine una digitala, constituita dintr-o serie finita de elemente cu pozitie bine stabilita ce poarta numele de pixeli [5].

Procesarea imaginilor digitale (DIP – Digital Image Processing) se defineste printr-un set de algoritmi ce presupun utilizarea unui calculator pentru a modifica aspectele unei imagini. Aceste modificari sunt realizate pentru a imbunatati calitatea interpretarii umane asupra imaginii sau pentru o mai buna adecvare in perceptia sistemelor computerizate [6]. DIP este o sub-categorie a procesarii semnalului digital (DSP – Digital Signal Processing), mai precis o restrangere a acestuia la spatiul bidimensional.

Spre deosebire de domeniul spatial unde atat analiza cat si modificarile sunt realizate direct la nivelul pixelilor, in functie de valorile lor in imaginea propriu-zisa, in domeniul de frecventa sunt reprezentate valori ce privesc rata de schimbare a valorilor pixelilor in domeniul spatial. Desi prezinta un nivel de abstractizare crescut, domeniul de frecventa ofera posibilitatea de a optimiza mult mai facil o imagine din punctul de vedere al calitatii, spatiului de memorie ocupat, etc. folosindu-se de conceptele procesarii semnalului digital.

In practica exista multe metode pentru procesarea datelor unei imagini digitale atat in domeniul spatial cat si in cel al frecventelor. Printre aceste practici se numara KLT (Karhunen-Loeve Transform), DFT (Discrete Fourier Transform) [7], WHT(Walsh-Hadamard Transform) si DCT (Discrete Cosine Transform).

De interes pentru aceasta lucrarea este discretizarea transformatei cosinus, DCT, datorita urmatoarelor avantaje:

* prezinta o capacitate crescuta pentru “compactarea energiei”, avantaj descris in detaliu in articolul “Discrete Cosine Transform”, publicat in Ianuarie 1974 de catre Nasir Ahmed, T. Natarahan si K.R. Rao [8]
* aproximeaza asimptotic eficienta compresiei realizata de transformata Karhunen-Loeve (ineficienta punct de vedere computational) [44]
* spre deosebire de DFT, coeficientii rezultati sunt decorelati, imbunatatind atat perfomantele computationale cat si pe cele ale modelarilor matematice la nivel de probabilitati [43].

## Domeniu al frecventei (DCT)

DCT este reprezentata de o functie liniara, inversabila ce are ca scop transformarea informatiilor din domeniul spatial in cel al frecventelor printr-o suma de sinusoide de diferite amplitudini si frecvente [9].

DCT-II a fost propus initial pentru compresia imaginilor digitale [10]. si se defineste prin ecuatiile prezentate in Formula **2.1** [45]:

(2.1)

unde:

* i, j – variabile discrete pentru domeniul spatial
* p(x, y) este valoarea pixelului la coordonatele x si y
* N reprezinta lungimea blocului pe care se realizeaza DCT
* M reprezinta latimea blocului pe care se realizeaza DCT

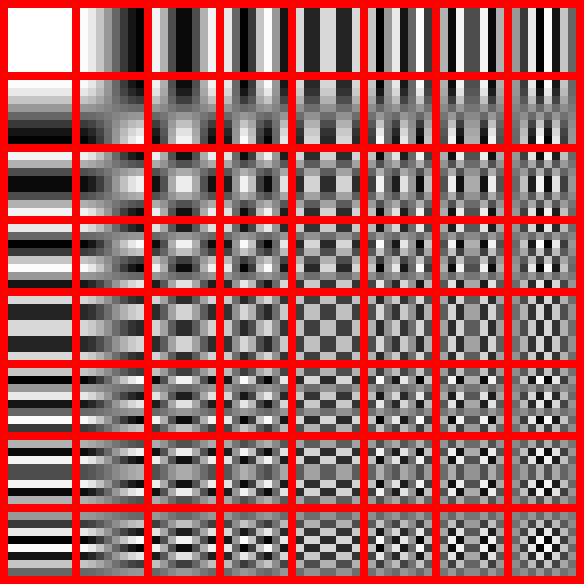
Implicit, inversa transformatei, IDCT, este reprezentata prin ecuatiile din Formula **2.2** [45]:

(2.2)

Intuitiv, transformarea se traduce printr-o translatie a intensitatii pixelilor din domeniul spatial (valorea unui pixel) catre amplitudine in spatiul frecventelor. Datorita acestei transformari putem elimina frecventele inalte, acestea fiind greu perceptibile de catre ochiul uman.

Cu alte cuvinte, reprezentativ pentru cazul prezent de analiza asupra procesarii imaginilor digitale, ne vom folosi de DCT definit pe o structura de 8 x 8. Blocul de pixeli de 8 x 8 va putea fi definit ca o combinatie liniara a functiilor cosinus ce se definesc pe aceste dimensiuni. Figura **2.1** constituie o reprezentare grafica a acestor functii:

Figura 2.1 - Reprezentare grafica a functiilor cosinus din DCT pentru o structura de 8x8 – [11]



## Steganografie

Steganografia este o metodologie de securitate cibernetica definita prin capacitatea de a incorpora informatie privata intr-o structura ce contine date publice. Tinta steganografiei este aceea de a transmite intr-un mod cat mai sigur date cu caracter personal sau confidentiale astfel incat insasi existenta lor sa nu fie cunoscuta decat de catre emitator si receptor.

Etimologia cuvantului “steganografie” dateaza inca din secolul al XVI-lea si provine dintr-o alaturare a cuvintelor grecesti “steganos” (ascuns, tainuit) si “graphe” (scriitura) [12].

In era moderna, unul din obiectivele steganografiei este acela de a trimite informatie personala (fisiere text, imagini, fisiere audio, etc.) prin intermediul transferurilor de date astfel incat sansele de a detecta prezenta acesteia de catre metodele de steganaliza existente sa tinda catre 0. Alte obiective sunt constituite de cresterea robustetii datelor trimise (rezistenta la diferite operatii de modificare a fisierului: rescalare, compresare) si majorarea capacitatii informatiei pe care acesta o poate ingloba pastrand proprietatile mai sus descrise [13].

Un exemplu actual de aplicare a steganografiei este conceptul de “digital watermarking” (procedeu concentrat in principal pe robustetea datelor transmise).[14] Notiunea presupune introducerea intr-o modalitate ascunsa si imperceptibila a unui marcator specific ce poate atesta autenticitatea informatiilor sau chiar identitatea proprietarului informatiei.

O structura ce contine atat parametrii necesari cat si procedeul de baza al steganografiei se regaseste in **Figura 2.2**:

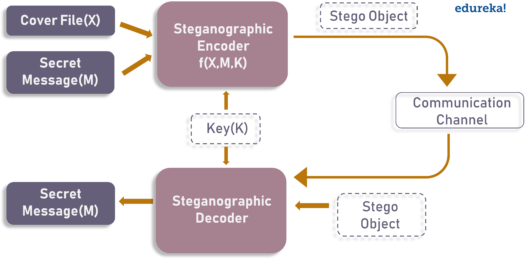


Figura 2.2 - Structura de baza a Steganografiei – [15]

Observam ca, procedural, Steganografia digitala are nevoie de o structura de date ce contine date publice, un mesaj secret, un algoritm de codificare a mesajului privat in structura de date cat si inversul acestui algoritm pentru a permite recuperarea mesajului secret de catre receptor.

Una din structurile de date in care este posibila inserarea mesajului secret este fisierul imagine, in acest caz vorbind despre steganografie in imagini digitale.

## Algoritmul LSB

In ceea ce priveste reprezentarea in domeniul spatial al informatiilor unei imagini digitale, cel mai nesemnificativ bit (Least Significant Bit - LSB), asa cum ii spune si numele, este bit-ul ce cauzeaza cea mai mica schimbare a datelor la nivel perceptual. Cu cat magnitudinea datelor creste, cu atat importanta sa scade. Spre exemplu, in prelucrarea unei imagini digitale reprezentata in format Red-Green-Blue (RGB), modificarea LSB-ului intensitatii culorii rosu dintr-un singur pixel va produce diferenta prezentata in **Figura 2.2**:

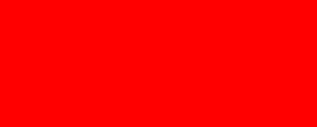
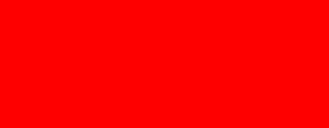


Figura 2.3 - Diferenta perceptuala RGB dupa 1 LSB modificat

(255, 0, 0) (254, 0, 0)

Mai mult, in **Figura** **2.3** se poate observa diferenta atunci cand in reprezentarea unui pixel in RGB sunt modificati ultimii biti pentru fiecare intensitate de culoare a acestuia:

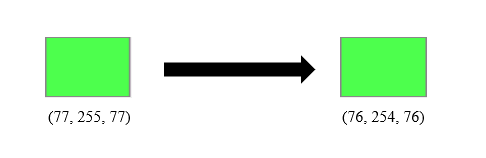


Figura 2.4- Diferenta perceptuala RGB dupa 3 LSB modificati

Algoritmul LSB este printre cei mai simpli algoritmi de steganografie, atat ca si concept, cat si din punct de vedere al implementarii. Acesta se foloseste de avantajul reprezentarii informatiei digitale in unul formaturile cunoscute (RGB, YcbCr, CMYK, etc.) si presupune manipularea bitilor cel mai putin semnificativi. Tinta acestor modificari este inglobarea informatiei secrete astfel incat pierderile efective ale calitatii imaginii sa nu afecteze perceptia potentialelor persoane ce intervin in procesul de comunicare si sa nu creeze suspiciuni.

Algoritmul continua pana la integrarea tuturor caracterelor in imaginea initiala. Asupra imaginii din **Figura 2.4** am aplicat un algoritm LSB naiv. A fost integrat un mesaj raspandit in toti LSB ai coeficientilor de culoare a pixelilor. Rezultatul se afla in **Figura 2.5**:



Figura 2.5 - Female (NTSC test image) - [16]

Figura 2.6 - Fisier ce contine informatie secreta



Figura 2.7 - Rezultat inserare date secrete

Se poate observa ca din punct de vedere perceptual, imaginile sunt identice. Plecand de la idea ca in fiecare pixel al imaginii initiale se pot ascunde 3 biti din mesajul secret, intr-o imagine de rezolutie 256x256 (aproximativ 66000 pixeli ~ 200kb) se pot ascunde 24 kb de date (echivalentul a 12 pagini dintr-o carte) fara ca acestea sa fie observate de catre ochiul uman. In plus, aproximativ jumatate din informatia secreta nu a produs modificari la nivelul datelor Figurii **2.5** intrucat LSB ai pixelilor coincideau cu bitii ce se doreau a fi introdusi.

Cu toate acestea, algoritmul naiv nu rezista atacurilor statistice (analiza zgomotului introdus) si nici si celor structurale (compresie, robustete).

Pentru a-i imbunatati performantele, algoritmului LSB i-au fost adaugate urmatoarele modificari:

* Criptarea mesajului secret pentru un plus de siguranta si un coeficient statistic superior
* Dezvoltarea unei metode de inserare bazata pe o parcurgere matriceala in spirala
* Introducerea algoritmului in procesul de compresie JPEG astfel incat modificarile nu sunt realizate direct in domeniul spatial ci in cel al frecventelor unde modificarile nu mai sunt la fel de usor de identificat.

## Algoritmul de compresie JPEG

Prin definitie, compresia unei imagini este reprezentata de executarea unei suite de algoritmi cu scopul de a reduce costul pentru stocarea sau transferul de date. Acesti algoritmi se pot folosi de avantaje ce tin de perceptia vizuala cat si de proprietati statistice pentru a-si optimiza rezultatele.

Compresiile se impart in doua mari categorii [17]:

* Compresii ireversibile cu pierderea unor date considerate redundante pentru perceptia umana (“lossy”), utilizate frecvent datorita compromisului realizat intre calitatea imaginii si spatiul de memorie utilizat de imaginea compresata.
* Compresii reversibile fara pierdere de date (“lossless”) folosite in principal pentru situatiile inflexibile (ex. Imagistica medicala)

Dupa cum spune si numele, tehnicile de compresie de tip “lossy” a datelor sunt focusate asupra economisirii spatiului ocupat de date ulterior si nu pe fidelitatea si acuratetea datelor in procent de 100%. In mod ideal, datele pierdute sunt minime sau imperceptibile de catre om.

Spre deosebire de algoritmii de compresie “lossless” a caror tinta este dezvoltarea unui proces reversibil pentru reprezentarea datelor prin exploatarea redundantei statistice, algoritmii de compresie “lossy” sunt constituiti pe teoria “rate-distortion” [18]. Aceasta teorie se bazeaza pe determinarea numarului minim de biti dintr-un mesaj (input) ce pot fi transferati printr-un canal de comunicare, astfel incat datele sa poata fi reconstruite la destinatie fara a depasi o anumita valoare de distorsionare.

In ceea ce priveste categoria de tip “lossy”, printre cei mai cunoscuti algoritmi de compresie a imaginilor digitale se numara [19]:

* Algoritm de compresie fractala (algoritm cu o rata de utilizare scazuta din cauza performantelor mediocre, folosit pentru comprimarea texturilor)
* Algoritmul de compresie JPEG (algoritm folosit la majoritatea imaginilor vizualizate online, find un compromis optimizat intre spatiul de memorie ocupat si calitatea imaginii redate) bazat pe Discrete Cosine Transform (DCT)

De interes pentru aceasta lucrare este algoritmul de compresie JPEG ce se materializeaza prin secventializarea urmatorilor pasi:

1. **Transformarea reprezentarii culorilor din format RGB (Red, Green, Blue) in format YCBCR (Y – componenta de luminescenta), CB – componenta de crominanta in albastru, CR – componenta de crominanta in rosu)**

Acest pas se realizeaza, de obicei, printr-o inmultire de matrici 3x3. O scurta reprezentare grafica a transformarii se regaseste in **Figura 2.8**:

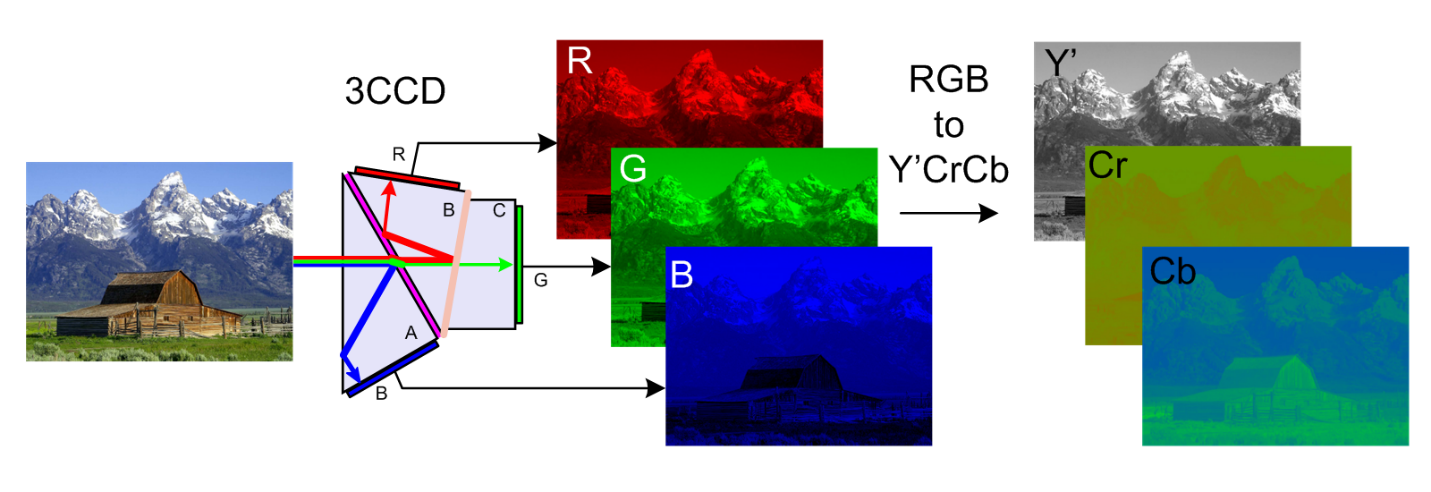


Figura 2.8 - Transformare RGB -> Y’CBCR, [20]

Algoritmul de calcul al valorilor Y, Cb si Cr este urmatorul [21]:

(2.3)

(2.4)

(2.5)

1. **Reducerea valorilor componentelor de crominanta printr-un factor de 2**

Procesul poarta numele de esantionare a crominantei (chroma subsampling) si se bazeaza pe sensitivitatea mai mare a ochiului uman pentru luminozitate si nu pentru culoarea informatiei. Cea mai intalnita rata de esationare in compresia JPEG este 4:2:0 si arata ca in **Figura 2.9**:

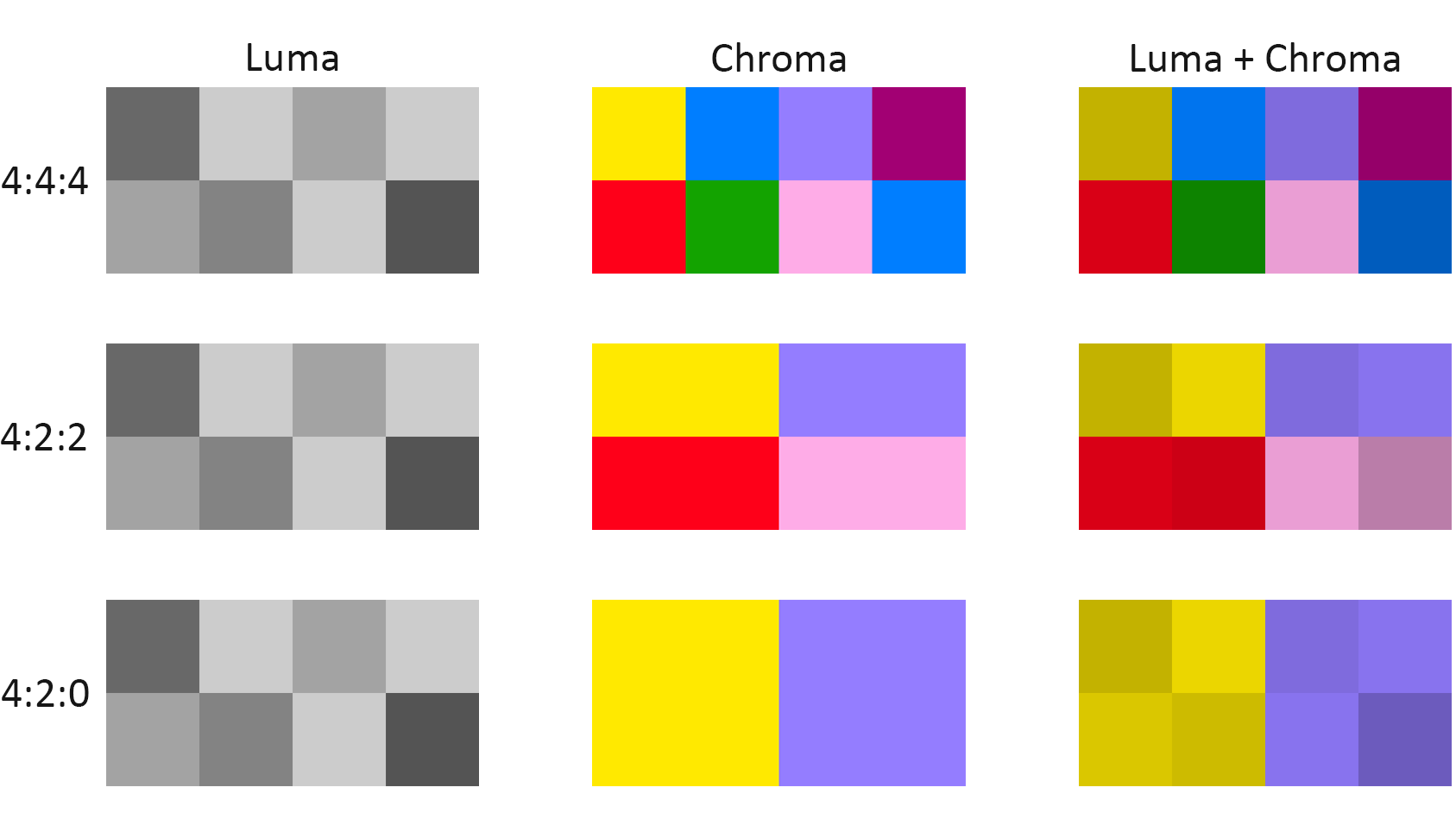


Figura 2.9 – Chroma Subsample Example: 4:4:4 – 4:2:2 – 4:2:0 – Sursa [22]

1. **Aplicarea DCT pe bloc-uri adiacente de pixeli 8x8**

Valorile matricilor 8x8 se incadreaza in intervalul [0,255]. Urmand a fi folosita functia cosinus ce este centrata in 0 (i.e., codomeniul este [-1,1]) vom centra in 0 si valorile matricii printr-o scadere a valorii 128.

Plecand de la formula descrisa in Subcapitolul 1.3. aceasta devine [23]:

(2.6)

Pentru a optimiza, termenii cosinus pot fi precalculati si stocati in tabele [24].

Mai este de precizat terminologia pentru coeficientii rezultati in urma transformarii. D[0][0] este numit Direct Current Term (DC) si este coeficientul de frecventa zero, obtinut ca o medie a tuturor culorilor pixelilor din blocul de 8x8 transformat [40]. Restul coeficientilor poarta numele de Alternating Current Terms (AC) si reprezinta componente cu frecventa crescatoare spre coltul din dreapta jos al matricei (D[8][8]).

1. **Cuantizarea amplitudinilor furnizate de DCT**

Componentele cu o frecventa inalta sunt inlaturate complet, urmand a pastra doar micile variatii in culoare si luminozitate pe suprafete mai mari. Pentru aceasta, coeficientii intorsi de DST sunt divizati prin valoarea corespondenta in matricea de cuantizare Q si apoi rotunjiti la intregi. Formal si particularizat pentru cazul nostru [24]:

(2.7)

unde:

* Q – matricea de valori cuantificate a coeficientilor bloc-ului dupa DCT
* D – matricea de valori a coeficientilor bloc-ului dupa DCT
* Qmat – matrice de cuantizare cu valori specifice in functie de procentul de detaliu al imaginii necesar dupa compresie

1. **Codarea rezultatelor**

Aceasta este realizata printr-o metoda de parcurgere a matricii in Zig-Zag si o restrangere a datelor in functie de lungimea sirului de numere consecutive care le preced (Rule Length Encoding). Mai precis, pentru matricea din **Figura 2.10**:

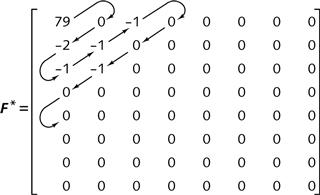


Figura 2.10 - Tehnica de alaturare a termenilor cuantizati - Sursa [25]

codarea va arata astfel:

[(0,7), 79], [(1,2), -2], [(0,1),-1], [(0,1),-1], [(0,1),-1], [(2,1),-1], [(0,0)]

unde structurile de tipul [(a,b),c] se definesc astfel:

* a – numarul de 0-uri necontorizati aflati inaintea numarului current
* b – numarul de biti necesari pentru a reprezenta in memorie numarul current
* c – numarul current

Aceasta codare este trecuta printr-o compresie Huffman (compresie de tip “lossless”) ce ofera un plus de compactare pentru economisirea spatiului pe disc [26].

Decodarea ce se realizeaza asupra unei imagini digitale codate cu ajutorul algoritmului de compresie JPEG este gestionata printr-o identificare a detaliilor pentru etapele de compresie si cuantizare a datelor initiale (detalii ce se regasesc in metadatele fisierului imagine) si parcurgerea in ordine inversa a etapelor descrise mai sus.

Performanta metodei de compresie JPEG este masurata de eroarea dintre imaginea originala si imaginea rezultata in urma aplicarii algoritmului. Statistic, metode precum MSE (Mean Square Error) sau PSNR (Peak-Signal Noise Ration) nu sunt corelate cu performanta subiectiva. Cuantizarea introduce partea semnificativa a erorii de reconstructie, celelalte erori fiind introduse de rotunjiri ale coeficientilor DCT in perioada de codare si decodare. De aceea, in practica, matricea de cuantizare este aleasa in functie de proprietati ale perceptiei umane.

## Sistemul de criptare AES

Rijndael o familie de algoritmi de criptare iterationali de tip cifru bloc (cipher block) cu o lungime variabila a dimensiunii blocurilor si cheii de criptare [27]. AES este o specificatie ce standardizeaza algoritmul de criptare Rijndael pentru urmatorii parametrii:

* Lungimea blocului de 128 biti
* Lungimea cheii de criptare: 128, 192 sau 256 biti

Selectand lungimea cheii de criptare la 256 biti (securitatea cea mai mare a acestui standard), algoritmul preia un input de 128 biti (16 bytes) si ii trece prin 14 runde de transformari succesive. Fiecare runda este constituita din 4 operatii ce sunt aplicate sub forma unor operatii pe matrice [28]:

* Add Round Key (o cheie specifica fiecarei runde este construita din cheia secreta initiala, fiecare depinzand de cea creata anterior) : intre input-ul rundei si cheie se realizeaza o operatie de XOR
* Sub-Bytes : transformarea fiecarui byte pe baza unui tabel de substitutie
* Shift-Rows: interschimbare valori pe fiecare rand din matrice
* Mix Columns: fiecare coloana este inmultita cu matricea

Algoritmul Rijndael a fost ales pentru deveni AES intrucat chiar dupa cateva runde, continutul input-ului este complet modificat si difuzat pe intreaga structura. De asemenea, computational, acesta nu necesita foarte multe resurse pentru operatiile de criptare si decriptare.

In ceea ce priveste steganografia si studiul curent, acest algoritm a fost ales pentru a cripta datele secrete datorita securitatii pe care il ofera informatiei (utilizat in prezent pentru datele secrete ale NSA – Agentia de Securitate Nationala a Statelor Unite ale Americii) cat si datorita pachetelor standard ale limbajelor de programare ce ofera o implementare a acestuia (*javax.crypto*)

# Implementarea aplicatiei

## Tehnologii folosite si motivatia alegerii lor

Aplicabilitatea integrarii unei metode de steganografie in procesul de compresie a datelor cat si posibilitatea de a recupera aceste date a fost demonstrata printr-o aplicatie web ce urmeaza a fi detaliata in acest capitol.

Pentru a crea o aplicatie web care sa sustina interactiunea cu un utilizator prin intermediul unui browser, a fost aleasa extensia Spring Boot a framework-ului de baza Spring [29]. Pentru dezvoltatorii software, Spring Boot este folosit pentru a pune la dispozitie un mediu prin care acestia pot sa creeze un microserviciu de sine statator. Cateva din avantajele pe care acest framework le prezinta si au fost decisive in selectarea lui pentru dezvoltarea aplicatiei web sunt usoara gestionare a procesarii cererilor intre utilizator si servicii prin intermediul sistemului de adnotari pe care acesta le defineste, incorporarea unui server in dependintele framework-ului astfel incat deploy-ul este realizat automat, minimele configurari necesare pentru a pregati aplicatia si integrarea acestuia cu limbajul de programare Java.

Desi Spring este pregatit pentru a fi integrat cu o suita de limbaje de programare precum Java, Kotlin, Scala, Groovy, pentru acest proiect a fost ales limbajul Java fara prea multe dificultati. Comunitatea developerilor este una dintre cele mai dezvoltate iar documentatia disponibila este mult mai detaliata decat pentru celelalte. Acest limbaj ofera o utilizare facila a diferitelor servicii gestionate cu ajutorul tool-ului maven (tool de automatizare a descarcarii si mobilizarii resurselor software exterioare celor standard) intrucat si acestea au fost scrise tot in Java. Printre acestea se numara:

* *jersey-media-json-jackson*,pentru transformarea datelor din obiecte POJO in Java in format JSON, acesta fiind, alaturi de XML, cel mai utilizat format de date in comunicarea pe web
* *ejml*, ce defineste structuri de tipul *SimpleMatrix* ce usureaza operatiile de adunare, scadere, inmultire, inversare, etc. pe matrici. Acest pachet a fost folosit in principal pentru teste

De asemenea, in uneltele standard java au fost regasite urmatoarele pachete ce au oferit in plus in decizia de a lucre cu Java.

* *javax.crypto,* utilizat pentru operatiile de criptare si decriptare cu ajutorul *AES*
* *java.io* si *java.nio* ce au facilitat operatiile cu stream-uri de date in jurul fisierelor

Pentru interfata utilizatorului a fost folosit framework-ul open-source Vaadin. Acesta ofera capacitatea de a implementa interfete grafice direct in limbajul Java, iar integrarea cu Spring Boot este una facila. Intrucat atentia acestui proiect s-a indreptat mai mult spre realizarea algoritmului de steganografie, Vaadin a fost o alegere usoara pentru a crea rapid un mediu usor accesibil si prietenos pentru utilizator.

VBinDiff este un alt produs software ce a fost utilizat pe parcursul dezvoltarii acestei aplicatii [30]. Acesta este un tool open-source ajuns la ultima versiune in 2017, folosit pentru compararea in forma hexazecimala a datelor fisierelor de dimensiuni de pana la dimensiuni de 4GB. Acesta a fost vital in nevoia de a examina datele secrete introduse pana in momentul in care a fost dezvoltat si algoritmul de decodificare.

Un alt util folosit pentru cresterea productivitatii a fost Postman [31]. Cu ajutorul acestuia am reusit sa creez request-uri de tip REST catre serviciile de procesare a imaginililor fara a fi nevoie de a porni de fiecare data interfata grafica.

## Modul de utilizare

In aceasta sectiune va fi prezentat modul de utilizare al aplicatiei. Interfata grafica realizata este una simplista dar imbogatita de o serie de verificari astfel incat utilizatorului nu i se ofera ocazia de a periclita buna functionare a algoritmilor din spate.

Primul ecran al aplicatiei, prezentat in figura **3.1** este creat pentru a-i oferi utilizatorului posibilitatea de a decide daca isi doreste sa inceapa procesul de codare sau pe cel de decodificare.

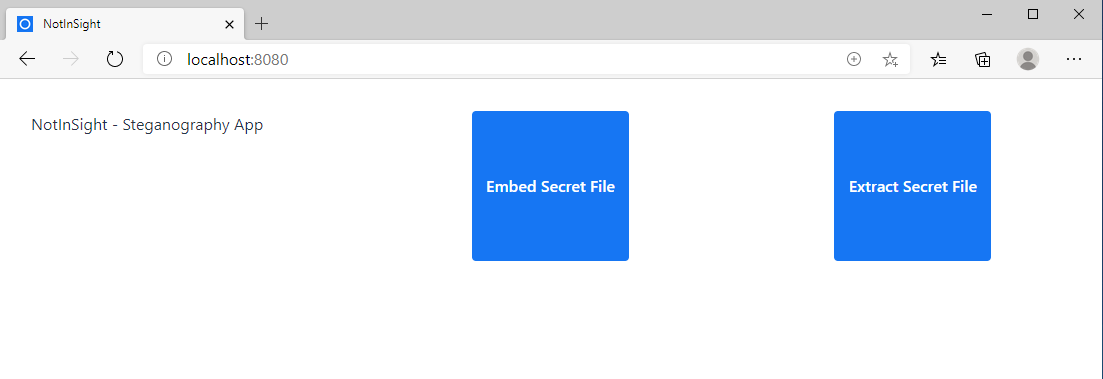


Figura 1.1 - ecran initial aplicatie NotInSight

Alegand optiunea de “Embed”, utilizatorul se gaseste pe o pagina unde isi poatea incarca un fisier ce contine datele secrete cat si pe cel in care vor fi integrate. Acesta are posibilitea de a incarca fisierele prin intermediul butoanelor “Upload Cover File” si “Upload Secret File” sau mai simplu, prin *drag & drop* in casuta specifica, dupa cum se poate observa in Figura **3.2**:

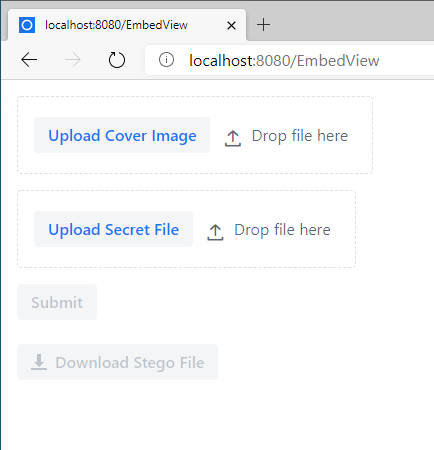


Figura 3.2 - Ecran Embed aplicatie NotInSight

Atat in partea de frontend cat si in partea de backend exista o serie de verificari ce nu vor permite utilizitorului sa incarce un format de date neacceptat de serviciile de procesare a imaginii. De asemenea, pentru a pastra o fidelitate cat mai mare a fisierului stego in raport cu imaginea initiala, utilizatorului nu ii este permis a incarca un fisier secret a carui dimensiune sa depaseasca ~ 10-15% din dimensiunea fisierului cover. Se poate observa in Figura **3.3** o notificare ce va avertiza utilizatorul despre nevoia de a schimba datele de intrare:

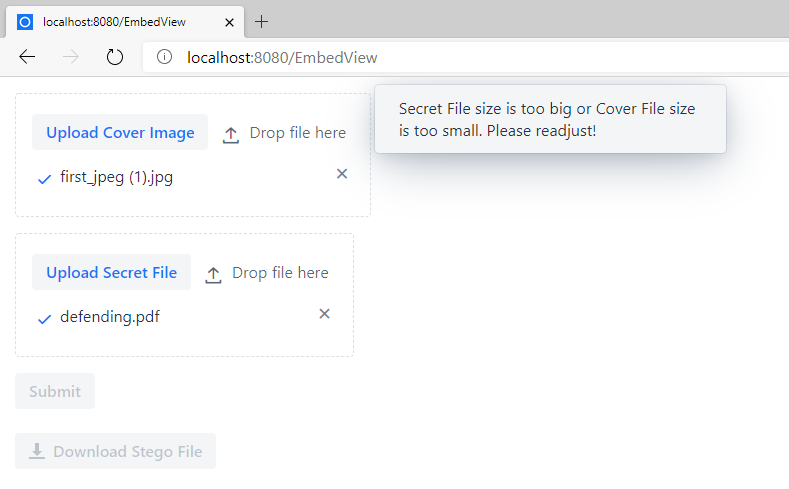


Figura 3.3 - Avertisment dimensiune fisiere

Dupa ce au fost incarcate cele doua fisiere, butonul de “Submit” devine disponibil iar utilizatorul poate sa trimita datele pentru a fi procesate. In momentul in care fisierul stego a fost creat, este afisatao notificare (Figura **3.3**) , butonul de descarcare devine disponibil, iar utilizatorul poate descarca fisierul.



Figura 3.4 - Notificare encodare reusita

In ecranul “Extract”, acelasi gen de ecran il intampina pe utilizator si ii ofera posibilitatea de a incarca un fisier ce contine date secrete. Dupa ce acestea au fost extrase, utilizatorul este notificat de terminarea procesului si are posibilitatea de a descarca fisierul.

## Detalii de implementare

Pentru inceput, informatia grafica a fisierului imagine a fost incarcata folosind libraria ImageIO (*javax.imageio.ImageIO*). Cu ajutorul acesteia, aplicatia accepta majoritatea formatelor de fisier imagine, aceasta fiind transformata la final in format JFIF (JPEG File Interchange Format), format ce accepta algoritmul de compresie JPEG.

Informatia fisierului secret a fost stocata ca sir de bytes. Asupra acestui sir de bytes a fost aplicat algoritmul de criptare AES, utilizand libraria *javax.crypto* din pachetul *java.base*. Dupa cateva experimente cu libraria, am folosit modul Cypher-Block Chaining (CBC). Un generator de numere aleatoare (Secure Random) a fost folosit pentru a crea vectorul de initializare necesar criptarii primului bloc de 16 bytes de text in clar. Textul criptat rezultat in urma aplicarii AES este folosit ca vector de initializare pentru cel de-al doilea bloc de criptat, al doilea text criptat este folosit pentru ce de-al treilea bloc s.a.m.d. Desi neparalelizabila (timpul de executie creste), din punct de vedere al securitatii datelor, aceasta metoda este mult mai sigura decat altele precum Electronic Codebook (ECB) ce genereaza un output identic pentru un acelasi input [46].

Dupa criptarea mesajului secret incepe procesarea imaginii. Acesteia ii este aplicata o transformare din spatial de culori RGB in YCbCr folosing formulele **2.3**, **2.4**, **2.5**, urmata apoi de procesul de downsample. Acesta a fost realizat pentru componentele Cb si Cr printr-o reducere succesiva a coeficientilor pe verticala si orizontala, rezultatul fiind media aritmetica a acestora.

Etapa de scriere a segmentelor si markerilor specifici formaturilor unei imagini JPEG a fost una anevoioasa. Segmentele de date contin informatiile necesare decompresarii imaginii pentru a o putea vizualiza (tabelele de cuantizare a luminescentei si crominantei, tabelele Huffman, etc.). Intrucat majoritatea detaliilor despre informatiile continute de aceste segmente se gasesc in cod hexazecimal, am folosit programul VBinDiff pentru a putea identifica rapid informatiile scrise intr-un fisier in limbaj hexazecimal. De asemenea, site-ul [47] a fost foarte util pentru a putea testa coerenta acestor date din header-ul imaginii.

Pentru DCT, initial am implemetat formula clasica ce a dus la complexitate O(n4) per bloc de 8x8. Imaginile de calitate inalta ajung la dimensiuni foarte mari: la o imagine de 1920x1080 vorbim despre 32400 blocuri de 8x8 numai pentru una din componentele Y, Cb sau Cr -> aproximativ 132 milioane de operatii per componenta doar pentru transformarea DCT. A trebuit astfel cautata o metoda alternativa pentru a diminua aceasta complexitate. Am gasit, astfel, formula AAN propusa de Arai, Agui si Nakajama [38]. Aceasta se bazeaza pe proprietatea de periodicitate a functiei cosinus si pe posibilitatea de a precalcula termenii ce folosesc aceasta functie. Urmarind informatiile disponibile pe site-ul [48] (grup ce gestioneaza libraria open-source pentru compresia JPEG si cei ce au realizat implementarea standardizata a acesteia) am reusit verific ca si acestia folosesc o versiune a acestui algoritm.

Mesajul secret a fost inserat folosind o tehnica mai putin conventionala pentru dispersarea acestuia in intreg pachetul de coeficienti DCT rezultati in urma transformarii. Primele informatii introduse au fost cele ce precizeaza lungimea fisierului secret, dimensiunea (in bytes) a extensiei acestuia si extensia fisierului in sine. Coeficientii au fost asezati in serii de matrice patratice astfel incat asezarea lor sa nu fie una liniara si predictibila. Pentru a reduce distorsiunea produsa imaginii au fost evitati coeficientii de tip DC cat si cei egali cu 0.

Compresia Huffman a fost realizata cu ajutorul unor tabele standard. Inainte de a descoperi existenta markerilor si segmentelor de date aferente in structura unei imagini digitale JPEG am incercat implementarea unor arbori Huffman. Desi implementarea a fost cu succes, nu am reusit sa integrez si o metoda prin care sa transpun si codificarea in detaliile imaginii, astfel incat sistemul de operare sa le poata citi si astfel sa decompreseze fisierul. Aici, de ajutor a fost aplicatia JPEGSnoop ce ofera informatii detaliate si foarte bine structurate cu privire la tabelele de compresie, cuantizare s.a. [39]. Am preluat astfel o serie de imagini digitale intr-un format ce foloseste compresie JPEG atat din surse proprii cat si din arhive de pe internet pentru a compara tabelele pe care acestea le folosesc. Am ales astfel o serie de tabele standard, folosite de cele mai multe camere digitale si produse software pentru editare de imagini.

In urma acestor procedee, fisierul Stego este creat si livrat utilizatorului. Acesta are posibilitatea de a-l descarca si trimite pe mediul de comunicare pe care si-l doreste. Imaginea digitala este in format JPEG File Interchange Format(JFIF), unul dintre cele mai folosite de pe internet [41].

Ajunsa la destinatar, acesta poate incarca imaginea in aplicatie pentru a fi trimisa catre serviciile ce rezolva decodarea mesajului secret, aplicand pasii descrisi anterior in ordine inversa: citirea markerelor si segmentelor de date ce contin informatia necesara codarii (tabele cuantizare, tabele compresie, detalii despre dimensiuni, etc.), decodarea Huffman pentru a obtine coeficientii DCT si extragerea mesajului criptat din acestia. Decriptarea este realizata cu ajutorul AES iar mesajul secret este recuperat.

# Experimente si performante

## Metrici de performanta

Metricile de performanta ce privesc imaginile digitale reprezinta un subiect destul de controversat in analiza calitatii unui fisier imagine intrucat foarte multe dintre acestea nu sunt considerate a fiind corelate cu subiectivismul. Procesul de determinare a acuratetei unei imagini se numeste IQA (Image Quality Assessment) si presupune atat o analiza obiectiva cat si una subiectiva.

PSNR (Peak Signal Noise Reduction) este una dintre cele mai cunoscute metode de aproximare a distorsiei, fiind un raport intre puterea maxima a unui semnal si puterea zgomotului ce perturba reprezentarea acestuia [32].

Pentru a testa fidelitatea fisierelor Stego, am selectat 3 imagini ce fac obiectul studiilor si analizelor asupra imaginilor digitale si am comparat rezultatele aplicarii procesului de inserarea a mesajului secret cu cele ale altor algoritmi ce ofera posibilitatea de a integra informatie secreta intr-un fisier imagine.

Pentru a ne asigura ca toti algoritmii opereaza la capacitate maxima a fost selectat un text de aprox. 3KB pentru teste

Imaginile de referinta sunt urmatoarele:



Figura 4.3 - Peppers - [37]

Figura 4.2 - Airplane (F-16) - [35]

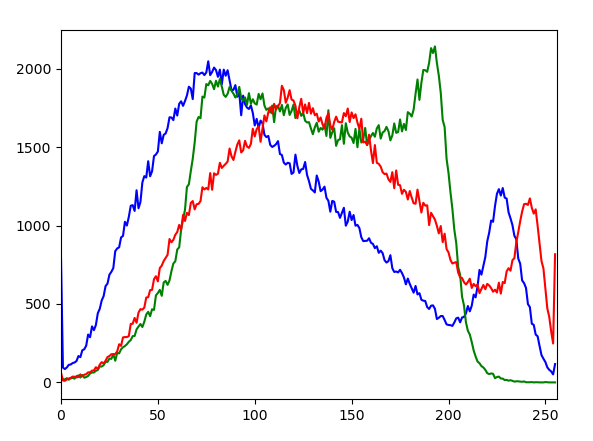
Figura 4.1 - Baboon - [34]

In comparatie cu alti algoritmi ce realizeaza steganografie, rezultatele obtinute pentru aceste imagini au fost prezentate in Tabelul **4.1**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SecretFile = 3KB | | MSE | PSNR | SSIM[42] | BRISQUE[43] |
| Xiao Steganography | Baboon | 0.047 | 66.168 | 1.000 | 18.600 |
| Peppers | 0.047 | 66.163 | 1.000 | 18.501 |
| Airplane | 0.047 | 66.204 | 1.000 | 8.149 |
| HideNSend | Baboon | 52.321 | 36.264 | 0.986 | 16.490 |
| Peppers | 17.425 | 40.619 | 0.978 | 10.299 |
| Airplane | 19.519 | 40.105 | 0.982 | 20.515 |
| NotInSight | Baboon | 48.875 | 36.470 | 0.985 | 13.111 |
| Peppers | 21.138 | 39.817 | 0.976 | 12.104 |
| Airplane | 18.646 | 40.284 | 0.985 | 22.667 |
| LSB | Baboon | 95.647 | 34.630 | 0.982 | 12.328 |
| Peppers | 95.462 | 35.231 | 0.928 | 18.202 |
| Airplane | 102.396 | 34.945 | 0.960 | 24.194 |

Tabel 4.1 - Comparare algoritmi steganografie

Este bine de tinut cont de faptul ca imaginile prelucrate cu majoritatea produselor software de steganografie nu sunt rezistente la tipul de compresie JPEG atunci cand sunt transmise pe internet, acesta constituid un factor important in ceea ce priveste fiabilitatea produsului rezultat.

 In Figurile **4.4** – **4.9** se pot observa histogramele realizate asupra imaginilor initiale si asupra celor prelucrate cu algoritmul NotInSight. Acestea demonstreaza o deviatie minima in ceea ce priveste procentul statistic realizat asupra coeficientilor pixelilor.

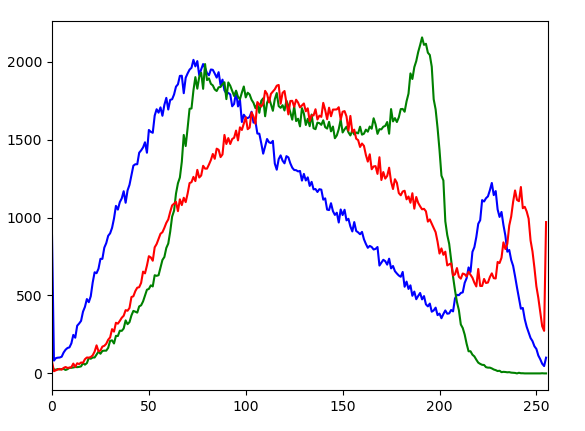


Figura 4.5 - Histograma Stego Baboon

Figura 4.4 - Histograma Baboon

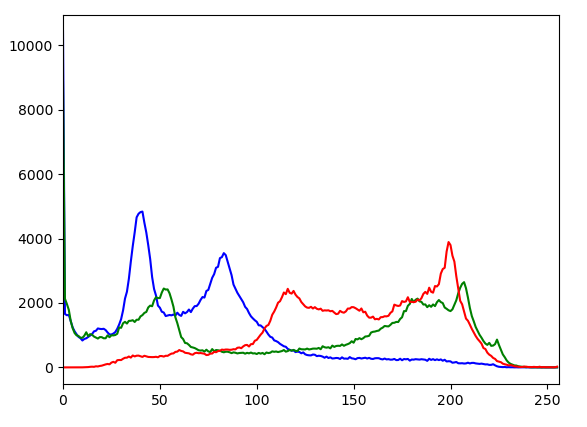
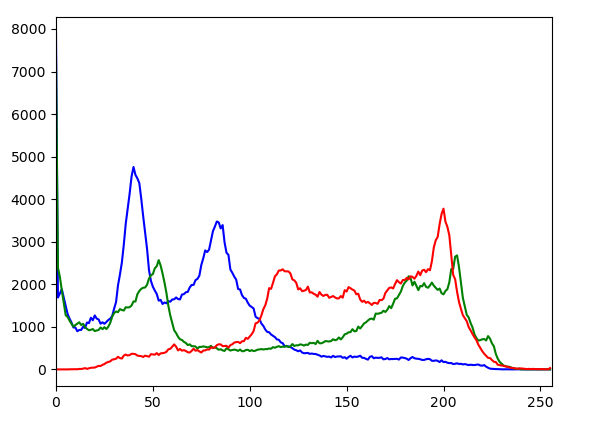


Figura 3.6 - Histograma Peppers

Figura 4.7 - Histograma Stego Peppers

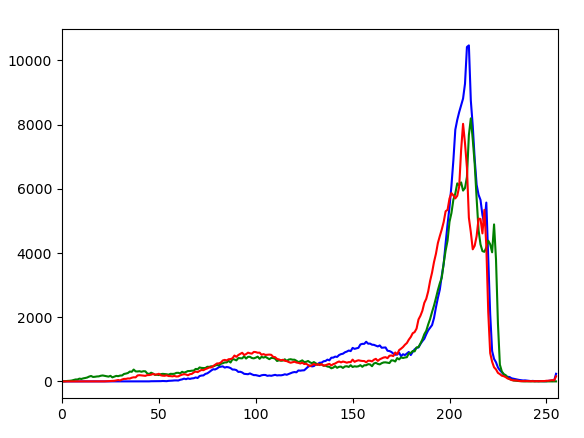
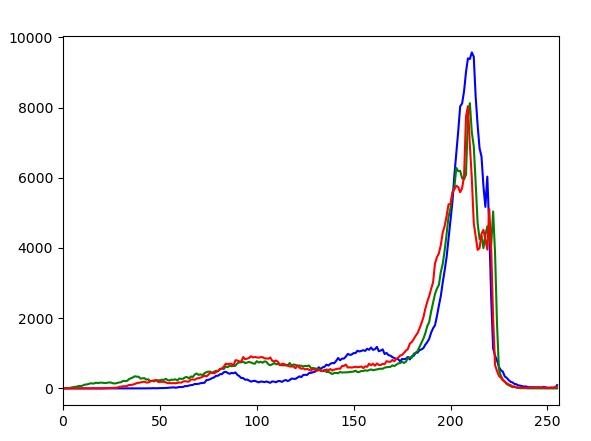


Figura 4.9 - Histograma Stego Airplane

Figura 4.8 - Histograma Airplane

# Concluzii si directii viitoare de dezvoltare

Steganografia digitala este un domeniu amplu al ascunderii informatiilor pe internet. Acesta este intr-o continua dezvoltare, fiind limitat doar de creativitatea persoanelor ce implementeaza aceste practici, dupa cum a fost verificat si in lucrarea curenta.

Am dezvoltat, astfel, un produs ce este capabil de a extrage informatia grafica dintr-o imagine intr-un format oarecare urmata de preluarea un fisier cu date confidentiale de la utilizator. Informatia grafica a fost trecuta printr-un process de transformare din domeniul spatial in cel al frecventelor iar datele confidentiale au fost ascunse in coeficientii obtinuti sub acest nivel. In final, a fost aplicat un algoritm de compresie ce corespunde standardului JPEG astfel incat imaginea rezultata este pregatita de a fi transferata pe orice mediu de comunicare. Decodificarea mesajului secret este posibila prin executarea pasilor in ordine inversa, implementati de asemenea in cadrul aplicatiei.

Desi executia algoritmilor descrisi in aceasta lucrare ating scopurile propuse in debutul lucrarii, aplicatia prezinta si o serie de imbunatatiri ce fac obiectul unei dezvoltari ulterioare.

Desi arhitectura actuala a aplicatiei implica executia frontend-ului si a backend-ului pe acelasi sistem de gestionare a datelor, fisierele incarcate de utilizator (in special cel secret) ar putea fi criptate inainte de a se realiza transferul catre partea de servicii. Acest lucru ofera o securitate suplimentara in cazul unui atac de tip Man in the Middle [42].

De asemenea, asupra interfetei grafice se pot aduce o serie imbunatiri ce ar putea creste calitatea experientei utilizatorului atunci cand foloseste aplicatia.

In concluzie, lucrarea constituie un punct de plecare in directia modalitatilor de ascundere a informatiei intr-o imagine digitala, oferind explicatia detaliata a pasilor parcursi pentru realizarea acesteia.

# Bibliografie

[1] List of freeware, 35 Best Free Steganography Software For Windows, Accesibil online: <https://listoffreeware.com/list-of-best-free-steganography-software-for-windows/>, Data accesarii 27.06.2020

[2] Niels Provos, OutGuess, Accesibil online: <https://web.archive.org/web/20150831083519/http://outguess.org/>, Data accesarii 27.06.2020

[3] Canetti, R., Dwork, C., Naor, M., & Ostrovsky, R., Deniable encryption. In Annual International Cryptology Conference (pp. 90-104). Springer, Berlin, Heidelberg. (1997, August).

[4] Sourceforge, Steghide - manual, http://steghide.sourceforge.net/documentation/manpage.php, 10 Oct 2003

[5] Gonzalez, Rafael C. "Richard E. Woods Digital Image Processing, Pearson." (2018).

[6] McAndrew, Alasdair. *A computational introduction to digital image processing*. Chapman and Hall/CRC, 2015.

[7] Rao, Kamisetty Ramamohan, Do Nyeon Kim, and Jae Jeong Hwang. *Fast Fourier transform-algorithms and applications*. Springer Science & Business Media, 2011.

[8] Ahmed, Nasir, T\_ Natarajan, and Kamisetty R. Rao. "Discrete cosine transform." *IEEE transactions on Computers* 100.1 (1974): 90-93.

[9] Rao, K. Ramamohan, and Ping Yip. Discrete cosine transform: algorithms, advantages, applications. Academic press, 2014.

[10] Ahmed, Nasir. "How I came up with the discrete cosine transform." Digital Signal Processing 1.1 (1991): 4-5.

[11] <https://en.wikipedia.org/wiki/File:Dctjpeg.png>, Data accesarii, 29.06.2020

[12] Merriam-Webster, “Steganography.”, Accesibil online: <https://www.merriam-webster.com/dictionary/steganography> , Data accesarii: 29.06.2020

[13] Cheddad, Abbas, Joan Condell, Kevin Curran and Paul Mc Kevitt,. "Digital image steganography: Survey and analysis of current methods." Signal processing 90.3 (2010): 727-752.

[14] Cox, I., Miller, M., Bloom, J., Fridrich, J., & Kalker, T., “Digital watermarking and steganography”, Morgan kaufmann, 2007.

[15] Choudary, A. *Steganography Tutorial | A Complete Guide for Beginners | Edureka.*, 2019 Accesibil online: <https://www.edureka.co/blog/steganography-tutorial>, Data accesarii: 29.06.2020

[16] <http://sipi.usc.edu/database/database.php?volume=misc&image=1#top>, Data accesarii: 26.06.2020

[17] Singh, Manjari, Sushil Kumar, Siddharth Singh, Manish Shrivastava "Various image compression techniques: Lossy and lossless." International Journal of Computer Applications 142.6 (2016): 23-26.

[18] Blau, Yochai, and Tomer Michaeli. "Rethinking Lossy Compression: The Rate-Distortion-Perception Tradeoff." *arXiv preprint arXiv:1901.07821* (2019).

[19] Wikipedia, Lossy Compression, Accesibil online: <https://en.wikipedia.org/wiki/Lossy_compression>, Data accesarii: 29.06.2020

[20] <https://en.wikipedia.org/wiki/YCbCr#/media/File:CCD.png>, Data accesarii: 29.06.2020

[21] Eric Hamilton, JPEG File Interchange Format, 1992, Accesibil online: <https://www.w3.org/Graphics/JPEG/jfif3.pdf> , Data accesarii: 29.06.2020

[22] Jack, Keith. *Video demystified: a handbook for the digital engineer*. Elsevier, 2011

[23] William Buchanan, DCT (Discrete Cosine Transform), Accesibil online: <https://asecuritysite.com/comms/dct2>, Data accesarii 29.06.2020

[24] Cabeen, Ken, and Peter Gent. "Image Compression and Discrete Cosine Transform”, College of Redwoods, 2008.

[25] Poynton, Charles. *Digital video and HD: Algorithms and Interfaces*. Elsevier, 2012.

[26] Huffman, David A. "A method for the construction of minimum-redundancy codes." *Proceedings of the IRE* 40.9 (1952): 1098-1101.

[27] Daemen, Joan, and Vincent Rijmen. "AES proposal: Rijndael." (1999).

[28] FIPS, PUB. "197." Advanced encryption standard (AES) 26 (2001).

[29] Spring Team, Spring Boot, Accesibil online: <https://spring.io/projects/spring-boot>, Data accesarii: 27.06.2020

[30] Christopher J. Madsen, Vbindiff, Accesibil online: <https://www.cjmweb.net/vbindiff/>, Data accesarii: 27.06.2020

[31] Postman, Inc., Postman, Accesibil Online: <https://www.postman.com/>, Data accesarii: 29.06.2020

[32] Wikipedia, Peak signal-to-noise ratio, Accesibil online: <https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio>, Data accesarii: 28.06.2020

[33] <http://sipi.usc.edu/database/database.php?volume=misc&image=5#top>, Data accesarii: 28.06.2020

[34] <http://sipi.usc.edu/database/database.php?volume=misc&image=10#top>, Data accesarii: 28.06.2020

[35] <http://sipi.usc.edu/database/database.php?volume=misc&image=11#top>, Data accesarii, 28.06.2020

[36] <http://sipi.usc.edu/database/database.php?volume=misc&image=12#top>, Data accesarii, 28.06.2020

[37] <http://sipi.usc.edu/database/database.php?volume=misc&image=13#top>, Data accesarii, 28.06.2020

[38] Arai, Yukihiro, Takeshi Agui, and Masayuki Nakajima. "A fast DCT-SQ scheme for images." *IEICE TRANSACTIONS (1976-1990)* 71.11 (1988): 1095-1097.

[39] Calvin Haas, JPEGsnoop 1.8.0 - JPEG File Decoding Utility, Accesibil online: <https://www.impulseadventure.com/photo/jpeg-snoop.html>, Accesat la data 01.07.2020

[40] CSE 228 Week 3 Part 1, UC San Diego, Jacobs School of Engineering, Accesibil online: <https://cseweb.ucsd.edu/classes/sp03/cse228/Lecture_5.html>, Data accesarii: 01.07.2020

[41] JPEG JFIF, W3, Accesibil online: <https://www.w3.org/Graphics/JPEG/>, Data accesarii: 01.07.2020

[42] Wikipedia, the free encyclopedia, Man-in-the-middle attack, Disponibil online: <https://en.wikipedia.org/wiki/Man-in-the-middle_attack>, Data accesarii 02.07.2020

[43] Nuno Vasconcelos, Discrete Cosine Transform, UCSD, Disponibil online: <http://www.svcl.ucsd.edu/courses/ece161c/handouts/DCT.pdf>, Data accesarii: 02.07.2020

[44] Cranley, Nicola. Handbook of research on wireless multimedia: quality of service and solutions. Ed. Liam Murphy. Information Science Reference, 2009.

[45] Stanford Edu, The Discrete Cosine Transform (DCT), Accesibil online: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/jpeg/dct.htm>, Data accesarii: 02.07.2020

[46] Block cipher mode of operation, Wikipedia, the free encyclopedia, Accesibil online: <https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation>, Data accesarii: 02.07.2020

[47] AE27FF, JPDump, Accesibil online: <https://cyber.meme.tips/jpdump/#>, Data accesarii: 02.07.2020

[48] Independent JPEG Group, Accesibil online: <http://www.ijg.org/>, Data accesarii: 02.07.2020