

**SPECIALIZAREA INFORMATICĂ**

**Lucrare de licență**

**Securizarea Informațiilor folosind Tehnici Steganografice**

**Absolvent**

**DLarisa**

**Coordonator ştiinţific**

Conf. Dr. Boriga Radu

**București**

**Iulie 2021**

**Abstract**

Prin natura lor, oamenii sunt ființe curioase. În era comunicării digitale, noi, ca populație globală, am devenit obișnuiți cu a cere informații gratuite, care să fie livrate în timp real. Cu toate acestea, nimic în viață nu este obținut fără a plăti un preț, iar acest flux constant de trafic de date vine cu un cost; un cost pe care mulți dintre cei care folosesc internetul nu îl înțeleg sau îi minimizează importanța. Accesul la această cantitate masivă de date este extrem de valoros, iar în schimbul lui ni se cere, ca utilizatori, să partajăm informații sensibile, care ne pot pune în pericol confidențialitatea online. Din acest motiv, subiectul securizării informațiilor trimise în mediul virtual este de interes atât pentru specialiști, cât și pentru utilizatori, și se află într-o continuă dezvoltare.

Lucrarea de față își propune să introducă cititorul în domeniul steganografiei, o ramură importantă a securității digitale, prin prezentarea noțiunilor de bază, prin studiul și compararea unor algoritmi existenți ce fac uz de aceste concepte și prin propunerea unor îmbunătățiri ale metodelor selectate, creând astfel o bună bază de plecare, teoretică, în acest domeniu.

**Abstract**

Humans are curious beings by nature. In the age of digital communication, we, as a global population, have become accustomed to asking for free information to be delivered in real time. However, nothing in life is achieved without paying a price and this constant flow of data traffic comes at a cost; a cost that many internet users do not undestand or simply downplay. Access to this massive amount of data is extremely valuable and, in exchange for it, we, as users, are asked to share sensitive informations that may jeopardize our online privacy. For this reason, the subject of securing the information sent online is of interest for both specialists and normal people and is constantly evolving.

This paper’s goal is to present to the reader the field of steganography, an important branch of digital security, by introducing the basic notions, by studying and comparing existing algorithm that make use of these concepts and by proposing improvements to the chosen strategies, hence making a great theoretical starting point in this field.

Cuprins

[**1. Introducere** 5](#_Toc74228067)

[1.1. Motivație 5](#_Toc74228068)

[1.2. Obiective 6](#_Toc74228069)

[1.3. Contribuția personală 6](#_Toc74228070)

[1.4. Structura lucrării 6](#_Toc74228071)

[**2. Fundamente teoretice** 8](#_Toc74228072)

[2.1. Steganografia 8](#_Toc74228073)

[2.2. Imagini digitale 11](#_Toc74228074)

[2.3. Un studiu al tehnicilor steganografice 13](#_Toc74228075)

[*2.3.1. Algoritmul LSB simplu* 13](#_Toc74228076)

[*2.3.2. Algoritmul LSB îmbunătățit cu filtrarea pixelilor după MSB* 16](#_Toc74228077)

[*2.3.3. Algoritmul LSB folosind conceptul de liste înlănțuite* 17](#_Toc74228078)

[*2.3.4. Algoritmul LSB folosind K-means clustering* 18](#_Toc74228079)

[*2.3.5. Algoritmul LSB propus* 20](#_Toc74228080)

[*2.3.6. Algoritmul DCT* 25](#_Toc74228081)

[**3. Analiza performanței** 33](#_Toc74228082)

[3.1. Tehnologii utilizate și justificarea selectării acestora 33](#_Toc74228083)

[3.2. Măsurarea performanței 34](#_Toc74228084)

[**4. Concluzii** 43](#_Toc74228085)

[**5. Bibliografie** 44](#_Toc74228086)

# **1. Introducere**

## 1.1. Motivație

Viața în era contemporană este marcată de accesul și utilizarea mediului online, care îmbunătățește, dar și ușurează considerabil traiul utilizatorilor prin beneficii evidente. Însă, odată cu digitalizarea unui număr considerabil de aspecte ale existenței cotidiene se produce și o creștere semnificativă a riscului, în materie de securitate, la care suntem expuși în sfera virtuală. Prin urmare, nu este de mirare că există o cerere din ce în ce mai ridicată pentru metode noi și îmbunătățite de protecție.

Problema care stă la baza acestei lucrări este aceea de confidențialitate a informațiilor. O cantitate impresionantă de date este transmisă zilnic, pe internet, iar dintre acestea, multe au caracter sensibil, fie că ne referim la banalele poze de familie, postate pe rețelele de socializare, care au o încărcătură afectivă, până la conturi și parole bancare, de care depinde bunăstarea traiului nostru.

Bineînțeles, pentru a ascunde informațiile importante față de receptorii nedoriți există metode standardizate de criptare, precum și protocoale sau certificate securizate de comunicare, care vin în ajutorul utilizatorului, nepermițând accesul neautorizat terțelor malicioase, ce ar putea interveni și le-ar putea compromite. Criptografia este domeniul de interes pentru aceste funcționalități, dar are un vădit dezavantaj prin faptul că datele criptate sunt ușor identificabile, printr-o simplă observare, atrăgând astfel atenția asupra caracterului potențial valoros al acestora și făcându-le vulnerabile în fața atacatorilor care le urmăresc. Cu aceste idei în minte, putem deduce că un bun punct de plecare în securizarea informațiilor sensibile ar fi mascarea acestora, astfel încât să treacă neobservate, drept material inofensiv și banal, prin traficul de date și să nu devină ținta agresorilor.

Steganografia are drept obiectiv idea prezentată mai sus, fiind o alternativă, mai sigură, pentru transmiterea datelor cu caracter personal în mediul online și devine, astfel, un subiect de interes pentru rezolvarea problemei cu care se confruntă societatea contemporană.

## 1.2. Obiective

Obiectivul lucrării de față este acela de a studia conceptele definitorii, tehnicile și metodele steganografice care să vină în ajutorul procedeelor tradiționale de securizare digitală a informaților și să o ofere un strat suplimentar de protecție împotriva terțelor primejdioase.

Scopul este de a crea un algoritm sigur, cu capacitatea de a transmite un mesaj confidențial într-o formă modificată, imperceptibilă ochiului uman, care să nu ridice eventuale suspiciuni și să păstreze într-o manieră exactă conținutul original.

## 1.3. Contribuția personală

Pentru a arăta relevanța conceptelor descrise în cadrul acestei lucrări, dar și pentru a demonstra că acestea pot fi puse în practică, a fost conceput un serviciu, care să vină în ajutorul utilizatorului și să parcurgă toți pașii necesari codificării unui mesaj secret într-o fotografie, punându-se accent pe transformarea acestuia într-un canal inofensiv, nedetectabil de ochiul uman. De asemenea, se propun îmbunătățiri la unii dintre cei mai bine cunoscuți algoritmi din domeniul steganografiei și se combină cu elemente din domeniul criptografiei, tocmai pentru a mări gradul de securitate și pentru a atinge obiectivul propus de această lucrare.

O mare provocare a reprezentat înțelegerea noțiunilor prezentate și formularea unor concluzii pertinente pornind de la literatura de specialitate. Datorită cunoștințelor limitate din acest domeniu, a fost necesară multă documentație pentru a pricepe, interpreta și aplica conceptele teoretice studiate.

## 1.4. Structura lucrării

Lucrarea este împărțită în cinci capitole:

Primul capitol este cel actual, care a constat în prezentarea introducerii, a motivației studiului acestui domeniu, dar și a obiectivelor vizate de lucrarea prezentă.

Al doilea capitol este cel mai complex. Acesta își propune prezentarea noțiunilor introductive din domeniul steganografiei, al procesării imaginilor digitale și discutarea câtorva algoritmi steganografici bine-cunoscuți. De asemenea, vom propune o nouă metodă de încorporare a mesajului secret într-o fotografie, folosind și o tehnică bine-cunoscută de criptare a datelor, AES, reușind astfel să facem legătura dintre domeniul criptografiei și cel al steganografiei și să obținem un nivel de securitate mai ridicat.

Capitolul al treilea prezintă tehnici standardizate de măsurare a performanței algoritmilor vizați și compararea rezultatelor obținute prin intermediul acestora.

În capitolul patru este oferită o concluzie în legătură cu domeniul studiat.

Lucrarea conchide cu capitolul cinci, care constă în lista referințelor bibliografice.

# **2. Fundamente teoretice**

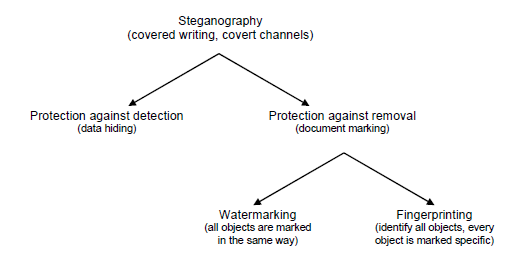
## 2.1. Steganografia

Mai întâi să înțelegem care este domeniul de activitate și cu ce se ocupă steganografia. Pe scurt, steganografia este arta și știința comunicării într-un mod prin care existența unui mesaj secret să nu poată fi detectată. Din punct de vedere etimologic, termenul provine din alăturarea cuvintelor grecești *steganos* (gr. original: *στεγανός*), care înseamnă *ascuns/tăinuit*, și *graphia* (gr. original: *γραφία*), care înseamnă *scriere* și a fost folosit încă de la sfârșitul secolului al XVI-lea [1].

Istoric vorbind, steganografia apare în forme rudimentare încă din Antichitate. Cel mai vechi caz documentat de steganografie datează din 500 î.Hr. când istoricul Herodotus scrie povestea lui Histaiaeus. Acesta, prizonier fiind, dorea să trimită un mesaj lui Aristagoras din Miletus ca să se revolte împotriva regelui persan. Pentru ca mesajul să nu fie interceptat a ras capul unui sclav doritor, a tatuat mesajul pe scalpul acestuia și a așteptat până când părul a crescut pentru a-l trimite mesager [2].

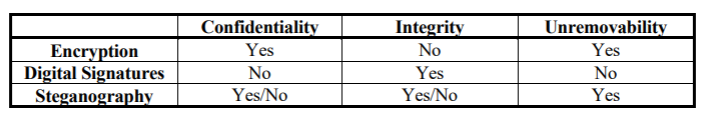
În perioada modernă, pe parcursul celui de-al II-lea Război Mondial, germanii au inventat micro-punctele: acestea erau documente complete, imagini și planuri reduse ca dimensiune la aceea a unui punct și atașat ca semn de punctuație [3], descrise de J. Edgar Hoover, directorul FBI din acea perioadă, ca fiind „capodopera inamicului în materie de spionaj” [4].

Și exemplele pot continua, dar steganografia s-a dezvoltat foarte mult, în special în ultimii ani, datorită evoluției rapide și la scară largă a tehnologiei. În era contemporană, steganografia digitală se bazează pe codificarea biților dintr-un mediu inofensiv, cum ar fi fișiere text, poze, fișiere audio sau video, cu date secrete, astfel încât probabilitatea de a detecta existența mesajului secret de către terțele malițioase, folosind tehnici de steganaliză existente, să se apropie cât mai mult de zero. Rezistența în fața procedeelor distincte care au ca scop alterarea fișierelor, fără pierderea informației (exemplu: compresare sau rescalare), dar și îmbunătățirea problemei capacității de stocare a mesajelor secrete în mediile de conversie, cu precădere poze și fișiere text, care prezintă limitări datorită dimensiunii reduse, în comparație cu celelalte medii multimedia existente, sunt alte obiective importante ale steganografiei, doar ca să numim câteva.

Unul dintre cele mai bine cunoscute exemple de steganografie modernă este conceptul de *watermarking*[[1]](#footnote-1). Metodele steganografice nu sunt, de obicei, robuste, împotriva modificării datelor sau prezintă doar o robustețe limitată. Pe de altă parte, watermarking-ul are funcționalitatea adițională de rezistență împotriva încercărilor de a elimina datele ascunse. Astfel, watermarking-ul, în detrimentul steganografiei, este utilizat ori de câte ori mediul care acoperă informațiile sensibile este distribuit la scară largă (spre exemplu bancnotele) și poate ajunge la părți care cunosc existența informației securizate, încercând să o altereze [5]. Printre aplicațiile watermarking-ului enumerăm protecția drepturilor de autor, protecție împotriva copierii (de interes este subiectul pirateriei digitale sau falsificarea bancnotelor) și autentificarea conținutului.

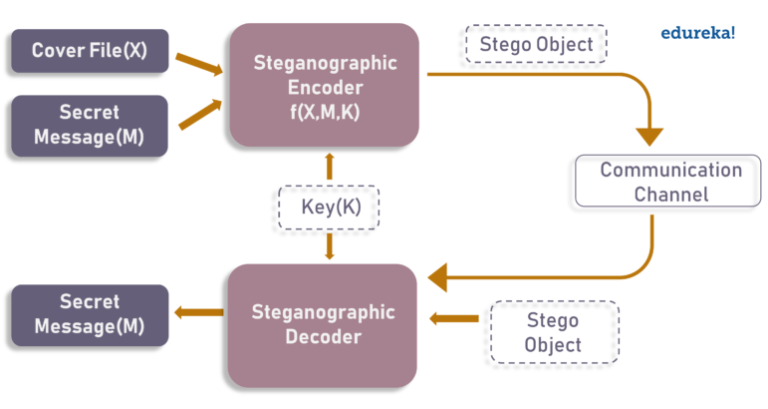
Figură 2.1. 1 - Legătura dintre Steganografie și Watermarking [6]

Dar de ce preferăm steganografia în detrimentul criptografiei? În esență, ambele au același scop, fiind utilizate pentru a asigura confidențialitatea datelor, dar folosesc mecanisme total diferite pentru a-l atinge. Astfel, avantajul clar al domeniului prezentat constă în faptul că mesajul secret transmis nu atrage atenția asupra sa. Mesajele criptate, ușor identificabile cu ochiul liber, oricât de impenetrabile ar fi, trezesc interes și pot fi în sine incriminatoare în țările în care criptarea este ilegală. Cu alte cuvinte, steganografia este mai discretă decât criptografia atunci când dorim să trimitem informații cu caracter sensibil. Pe de altă parte, mesajul ascuns prin steganografie este mai ușor de extras, dacă este descoperită prezența acestuia. Tocmai de aceea, pentru un plus de protecție și rezultate vizibil îmbunătățite, un prim pas logic este acela de a îmbina cele 2 ramuri ale securității. Pentru o înțelegere mai clară a diferențelor dintre cele două domenii este atașată următoarea figură:



Figură 2.1. 2 - Comparație între criptare, semnături digitale și steganografie [6]

Diagrama de mai jos prezintă o schemă de bază a unui model steganografic:



Figură 2.1. 3 - Schemă a unui model steganografic de bază [7]

După cum arată imaginea, atât mediul de acoperire (X), cât și mesajul secret (M) sunt introduse în codificatorul steganografic. Funcția steganografică, notată cu f(X, M, K), încorporerază mesajul secret în fișierul de acoperire dat ca input. Se obține un obiect steganografic care este foarte similar, din punct de vedere al aspectului, cu mediul original. Acesta este procesul de codificare. Pentru operația inversă, obiectul steganografic este introdus în decodificator și se obține astfel mesajul secret.

În funcție de natura *obiectului de acoperire*[[2]](#footnote-2), steganografia poate fi împărțită în mai multe tipuri, dintre care menționăm (cele mai comune):

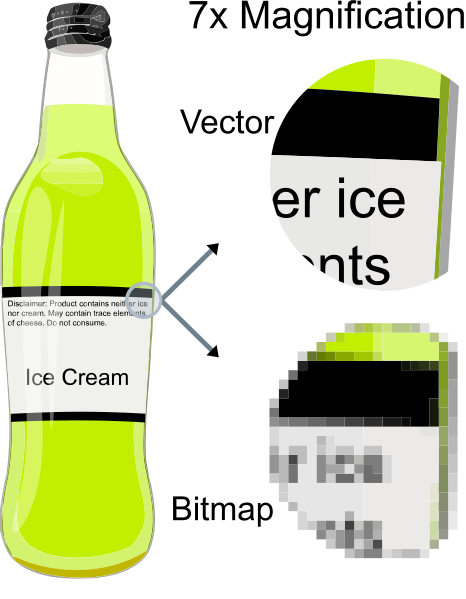
1. Steganografia de tip Text
2. Steganografia de tip Imagine
3. Steganografia Video
4. Steganografia Audio

În cele ce urmează, ne vom concentra pe al doilea tip de steganografie. În lumea digitală, imaginile sunt utilizate pe scară largă, iar pentru domeniul steganografiei ele reprezintă un mediu de acoperire de interes nu numai datorită răspândirii acestora în sfera virtuală, dar și datorită numărului ridicat de biți prezenți în reprezentarea digitală a unui astfel de fișier multimedia, care permit ascunderea unei cantități ridicate de informație.

## 2.2. Imagini digitale

Acum că avem o bună cunoaștere asupra fundamentelor steganografice este timpul să discutăm câteva concepte legate de procesarea imaginilor, pentru a putea înțelege rolul și modul de utilizare al acestora în domeniul steganografiei.

O imagine poate fi definită ca o funcție bidimensională , în care și caracterizează coordonatele spațiale ale planului, iar valoarea funcției în orice pereche de parametrii este dată de intensitatea (sau *gray level* – eng. original) imaginii în acel punct. Când , și sunt toate cantități finite, discrete, ele definesc o imagine digitală. Orice imagine este compusă dintr-o secvență finită de elemente, fiecare cu o valoare și localizare spațială unică. Aceste elemente poartă numele de *pixeli* [8]. Deci, ne putem gândi la o imagine ca la o matrice de pixeli, având un număr fix de coloane și rânduri. În sine, termenul de *imagine digitală* se referă, de obicei, la imagini *bitmap* (formatul .bmp) sau la imagini de tip *raster* (formatul .png, .jpeg etc...) și nu la cele e tip *vectorial* (formatul .svg, .pdf etc...).



Figură 2.2. 1 - Exemplu de imagine vectorială vs grafică de tip raster [63]

Pe scurt, la nivel conceptual, cele 3 tipuri de imagini prezentate se definesc astfel:

1. Imaginea Vectorială – sunt definite de puncte într-un plan cartezian, care sunt conectate prin linii și curbe pentru a forma poligoane și alte forme. Punctele determină direcția vectorului, fiecare putând avea diferite proprietăți precum culoarea, forma, curbura, grosimea sau umplerea. [9]
2. Imagine de tip raster – este o matrice de puncte, care reprezintă o rețea dreptunghiulară de pixeli. Astfel, acest tip de imagine este caracterizat de lățimea și înălțimea în pixeli, dar și de numărul de biți per pixel. [10]
3. Bitmap – se mai numește și matrice de biți (traducere din eng. original).

După cum am menționat mai sus, pixelul este unitatea de bază a unei imagini digitale. În sistemele cromatice, o culoare este, de obicei, reprezentată fie de 3 canale, fie de 4[[3]](#footnote-3). În cele ce urmează, ne vom axa pe modelul RGB (eng. original: *Red* – roșu; *Green* – verde; *Blue* - albastru). Deoarece acestea trei sunt elemente primare aditive, fiecare culoare poate fi reprezentată ca sumă ponderată a unei componente roșii, verzi și albastre, explicându-se, astfel, de ce acest model este cel mai des întâlnit și utilizat în afișarea imaginilor. Fiecare pixel din imagine este compus din 3 valori (roșu, verde, albastru) care sunt reprezentări pe 8 biți (sau 1 byte) a numerelor din intervalul 0 – 255. Faptul că putem reprezenta în mod binar imaginea digitală ne va fi util în secțiunea următoare, când prezentăm câțiva algoritmi steganografici.

## 2.3. Un studiu al tehnicilor steganografice

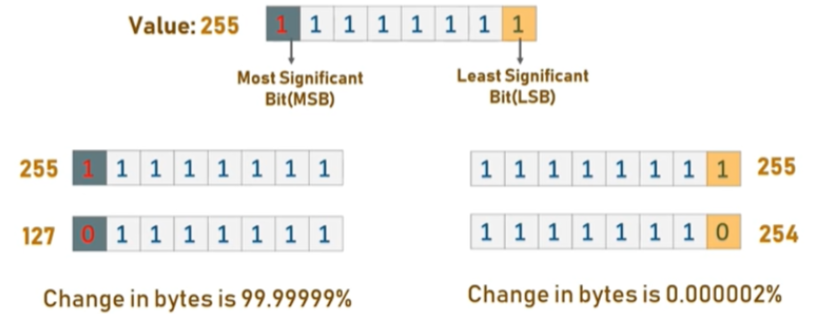
Variate metode steganografice au fost propuse în ultimii ani, cele mai multe dintre acestea bazându-se pe teoria sistemelor de substituție. O astfel de metodă încearcă să înlocuiască părțile redundante ale obiectului de acoperire cu un mesaj secret. Principalul dezavantaj pentru o astfel de abordare este slăbiciunea relativă la modificările produse mediului de acoperire. Recent, dezvoltarea de noi tehnici robuste de watermarking a contribuit la progrese în construcția sistemelor steganografice robuste și sigure.

În cele ce urmează, vom trece în revistă o clasificare a celor mai des întâlnite tehnici steganografice, în funcție de modificările aduse fișierului de acoperire, în cadrul procesului de încorporare a mesajului, așa cum sunt prezentate și în [5]:

* *Sisteme de Substituție*: înlocuirea părților redundante ale obiectului de acoperire cu mesajul secret;
* *Tehnici de Transformare a Domeniului*: încorporează mesajul într-un spațiu de transformare al mediului de acoperire (spre exemplu, în domeniul de frecvență[[4]](#footnote-4));
* *Metode Statistice*: codificarea informațiilor se realizează prin modificarea mai multor proprietăți statistice ale mediului de acoperire și fac uz de testarea ipotezelor pentru procesul de extragere;
* *Tehnici de Distorsiune*: încorporează mesajul secret folosindu-se de distorsiunea semnalului și măsoară abaterea de la obiectul de acoperire original în etapa de decodare.

### *2.3.1. Algoritmul LSB simplu*

Dacă aș oferi unei unei persoane 000.000 $ și i-aș spune că poate să modifice o singură cifră în 1 înainte să îi predau suma aleasă, probabil că ar schimba cifra din stânga, astfel încât să obțină 100.000 $, nu pe cea din dreapta care ar rezulta într-un singur dolar. Chiar dacă în ambele cazuri înlocuim numai o singură cifră, cea din dreapta este cea mai puțin semnificativă, deoarece are cel mai mic efect asupra totalului monater. Acesta este conceptul folosit și în cadrul algoritmului LSB: mesajul secret va fi încorporat în bitul cel mai nesemnificativ din fiecare canal al culorii unui pixel.



Figură 2.3.1. 1 - Diferența dintre LSB și MSB [7]

Această metodă steganografică este, probabil, una dintre cele mai simple pentru a ascunde informații în cadrul unei imagini și, totuși, este surprinzător de eficientă, deoarece ochiul uman nu poate distinge cu ușurință diferențe insignifiante la nivelul culorilor.



Figură 2.3.1. 2 - Diferențe când 1 bit este modificat sau 3 biți sunt modificați [imagine autor]

Această tehnică poate fi extinsă la 2, 4 sau chiar 8 biți, dar acest lucru poate provoca distorsiuni și zgomot, rezultând într-o imagine cu pierderi din punct de vedere al calității [11].

Deci, dacă dorim să ascundem un mesaj secret într-o fotografie, mai întâi transformăm textul în cod ASCII, apoi în text binar și, parcurgând secvențial fiecare pixel al mediului de acoperire, înlocuim fiecare LSB al fiecărui canal de culoare cu bitul corespunzător din textul secret. Acesta este cel mai simplu tip de algoritm LSB. Datorită problemelor de securitate, o primă recomandare pentru a îmbunătăți performanța acestui algoritm este ca utilizatorul să nu utilizeze o imagine din domeniul public, deoarece ar putea fi ușor comparată cu cea originală și atacatorul ar remarca, destul de rapid, diferențele dintre cele două. De asemenea, deoarece ordinea contează (atunci când încercăm să codăm mesajul) și fiecare pixel conține un șir lung de biți ( biți), imaginea digitală nu poate fi comprimată, decupată, redimensionată, îmbunătățită sau să fie aplicată asupra ei orice altă procedură care ar schimba pixelii, așa că acest algoritm funcționează fie cu formatul .bmp, fie cu .png, dar nu .jpeg, spre exemplu. Evident, acest lucru este destul de incomod și obținem astfel o primă slăbiciune importantă a acestui procedeu steganografic [[5]](#footnote-5).

*Figura 2.3.1.4* a fost obținută din *2.3.1.3*, asupra căreia am aplicat algoritmul LSB.

Figură 2.3.1. 3 - Lenna.png [12]

Figură 2.3.1. 4 - LennaStego.png [imagine autor]

Se poate remarca, cu ușurintă, faptul că, din punct de vedere al percepției vizuale, imaginile sunt identice. După cum am discutat anterior, fiecare pixel are 3 canale de culoare, deci poate ascunde 3 biți de informație secretă. Astfel, într-o imagine de rezoluție 255x255 (ex: Lenna.png; aproximativ 65.000 pixeli ~ 200 kb) putem încorpora, cu aproximație, 25 kB de date cu caracter senzitiv, ce reprezintă echivalentul a 12.5 pagini dintr-o carte. În plus, LSB are și beneficiul că un număr semnificativ de biți din mesajul secret corespundeau deja cu cei din imaginea de acoperire. Statistic vorbind, doar jumătate din biții dintr-o imagine vor trebui să fie modificați, în medie, pentru ascunderea mesajului [13].

După cum am menționat și mai sus, algoritmul original nu rezistă atacurilor, fie ele de tip statistic (analiza zgomotului introdus), sau de tip structural (robustețe, compresie etc...). De aceea, trebuie aduse îmbunătățiri pentru a obține o mai bună performanță.

În 2017, Mohidul Islam et al. aduc o optimizare interesantă. Autorii explică faptul că albastrul este cea mai puțin vizibilă culoare pentru ochiul uman, prin urmare conclud că cel mai bine utilizat pentru încorporarea datelor este canalul albastru din formatul RGB [14]. Idea este coroborată și de descoperirile lui Sahu U. și Mitra S. în domeniul steganografiei video (aceștia folosesc un algoritm LSB bazat pe selectarea aleatoare a cadrelor fișierului video și a poziției pixelilor) [15].

G.R. Manjula și A. Danti propun un algoritm LSB bazat pe funcții hash, de tipul 2-3-3, ceea ce înseamnă că doar 8 biți sunt stocați într-un pixel, doi în canalul roșu, 3 în cel verde și 3 în albastru. A fost folosită funcția hash , fiind poziția unui bit într-un pixel, este poziția de bit pentru cei 8 biți secreți care urmează să fie încorporați și este numărul de LSB utilizați pentru procesul de ascundere [16]. O astfel de tehnică a oferit rezultate clar superioare față de algoritmul original.

În cele ce urmează, vom explora câțiva algoritmi îmbunătățiți de LSB, care au fost propuși recent.

### *2.3.2. Algoritmul LSB îmbunătățit cu filtrarea pixelilor după MSB*

Principala slăbiciune a algoritmului LSB constă în faptul că încorporarea mesajului secret se realizează în pixeli consecutivi, fiind foarte ușor detectat de un potențial atacator. Metoda aleasă aduce un plus securității deoarece filtrează ce pixeli urmează a fi folosiți, pentru a face dificilă identificarea prezenței unui mesaj secret.

Autorul introduce noțiunea de pixel *întunecat* (eng. original: *dark*) și pixel deschis (eng. original: *light*). Un pixel este întunecat dacă, dintre cele 3 canale de culoare, minim 2 au MSB (eng. original: *Most Significant Bit*) egal cu 0; dacă, în schimb, minim 2 canale au MSB egal cu 1, atunci vorbim de un pixel deschis [17]. Utilizatorul alege, la începutul programului, în care tip de pixeli dorește să ascundă mesajul.

Deși această idee rezolvă problema alegerii aleatoare a pixelilor care urmează să fie codificați, eficiența algoritmului va depinde de calitatea imaginii. În cadrul mediului de acoperire, pixelii întunecați și deschiși sunt răspândiți în mod inegal, deci nu ne putem folosi de toată capacitatea fotografiei, astfel că dimensiunea posibilelor mesaje secrete care se doresc a fi încorporate scade dramatic. Deci nu este o alegere neapărat viabilă.

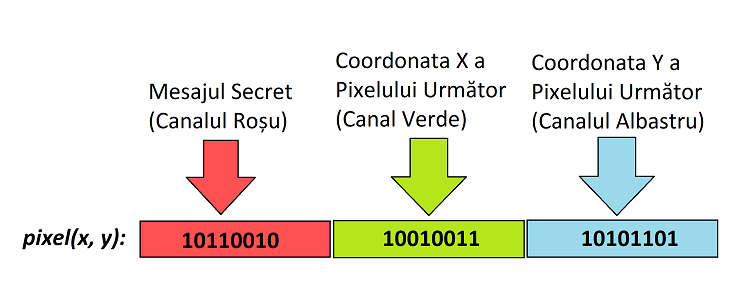
### *2.3.3. Algoritmul LSB folosind conceptul de liste înlănțuite*

Această metodă aduce câteva îmbunătățiri importante algoritmului de bază. În primul rând, mesajul secret nu mai este încorporat în pixeli consecutivi, ci se folosește conceptul de *listă înlănțuită* [18].

***Lista înlănțuită*** este o structură de date asemănătoare din punct de vedere conceptual cu vectorii, dar cea mai importantă deosebire dintre aceste două structuri de date constă în plasarea lor în memorie. Vectorii sunt poziționați în ordine secvențială în memorie, spre deosebire de listele înlănțuite în care fiecare element (numit în literatura de specialitate „nod”) stochează, pe lângă datele propuse, și adresa următorului nod din memorie [19].

Folosind această idee, mesajul secret va fi încorporat în pixeli random din imaginea de acoperire. Astfel, se ating două avantaje importante:

* În primul rând, neutilizarea secvențialității îngreunează detectarea textului ascuns și crește nivelul de securitate.
* În al doilea rând, adresa primului octet al mesajului secret devine cheie steganografică. După cum știm din lucrul cu listele înlănțuite, adresa primului nod este întotdeauna stocată într-un pointer pentru accesarea viitoare a datelor din structură. Pierderea acestei adrese duce, implicit, la pierderea datelor stocate. Deci metoda oferă un plus de securitate și prin prezența unei chei, cunoscute doar de utilizator.



Figură 2.3.3. 1 - LSB folosind Conceptul de Liste Înlănțuite [Imagine Autor]

Un alt beneficiu interesant este faptul că mai multe mesaje secrete pot fi încorporate în fișierul de acoperire. Deducem, astfel, că putem avea mai multe chei steganografice. Această caracteristică ar putea fi foarte utilă atunci când există mai mulți receptori și fiecare dintre aceștia trebuie să primească propriul mesaj secret [18]. Suplimentar, textul transmis ar putea fi fragmentat și codificat pe bucăți. Aceste lucru este foarte important deoarece se asigură că un potențial atacator are nevoie de toate cheile steganografice pentru a putea decodifica mesajul integral și, este posibil, ca terța malicioasă să nu conștientizeze dimensiunea textului transmis, deci să nu reușeașcă spargerea totală a acestuia.

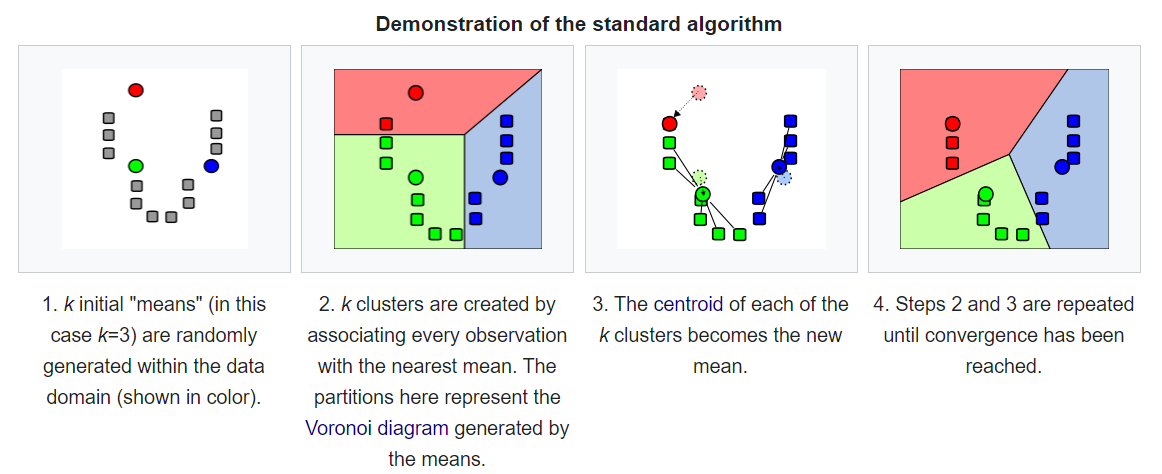
Pe de altă parte, algoritmul se confruntă cu aceeași problemă ca metoda prezentată anterior. Deși, din punct de vedere al capacității fotografiei de acoperire, spațiul în care un mesaj secret poate fi încorporat este mai mare decât în tehnica anterioară (în cele mai multe cazuri; depinde foarte mult de fotografia selectată și de tipul de pixeli folosiți de utilizator), tot este redus de cel puțin trei ori, deoarece folosim doar canalul roșu al unui pixel pentru a încorpora mesajul propriu-zis.

### *2.3.4. Algoritmul LSB folosind K-means clustering*

Aceast algoritm este cel mai complex dintre toți cei analizați mai sus. Această abordare a fost descrisă în [20], [21], [22] și [23], ca să numim câteva lucrări. Pentru a putea înțelege cum funcționează, trebuie să luăm cunoștință de câteva noțiuni preliminarii.

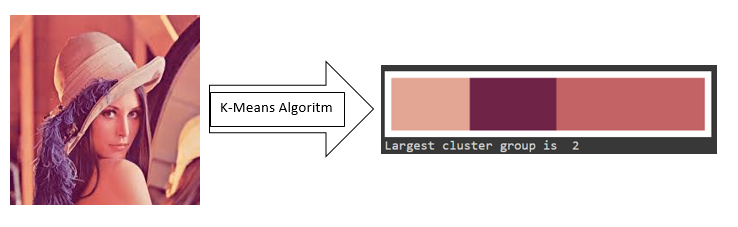
***Clusterele*** sunt grupuri de obiecte similare (în cazul nostru, pixeli). Datele din interiorul unui *cluster* sunt similare, fie că este vorba de culoare, dimensiune, formă etc..., dar se deosebesc de datele din interiorul unui alt *cluster*. Procesul de creare a *clusterelor* poartă denumirea de *clustering*.

Metoda de grupare ***K-Means*** face parte din categoria de algoritmi de învățare nesupravegheați, fiind una dintre cele mai populare metode utilizate pentru a aplica procedeul de *clustering*. Această tehnică clasifică un set de date în clustere disjuncte. În prima etapă, se calculează centre geometrice și în a doua etapă se face clasificarea datelor în funcție de distanța dintre acestea și centrele geometrice, fiecare fiind asignată unui cluster. Cea mai utilizată metodă pentru calculul distanței este distanța Euclidiană. Odată ce gruparea este realizată, se recalculează noul centru geometric al fiecărui cluster, iar datele sunt regrupate. Fiecare cluster din partiție este definit de datele membre și de centrul său. Centrul geometric pentru fiecare cluster este punctul în care suma distanțelor față de toate obiectele membre este redusă la minim. [24]



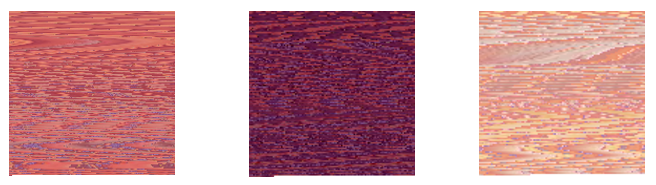
Figură 2.3.4. 1 - Demonstrație Algoritm Standard K-Means [24]

Având acum baza teoretică, vom prezenta algoritmul LSB îmbunătățit din [22]: Asupra imaginii de acoperire se aplică algoritmul de K-Means și se formează 3 clustere.



Figură 2.3.4. 2 - Demonstrație formarea celor 3 clustere folosind Lenna.png [Imagine Autor]

Pixelii sunt repartizați în cele trei clustere, care sunt convertite, mai departe, în fotografii de sine-stătătoare:



Figură 2.3.4. 3 - Imaginile formate din cele 3 Clustere [Imagine Autor]

În următorul pas, mesajul secret este criptat și împărțit în trei fragmente de lungime egală; iar fiecărui fragment îi este repartizat o fotografie și este încodat în aceasta folosind tehnica LSB.

Această tehnică întărește securitatea deoarece mesajul secret este împărțit în mai multe fragmente, fiind mult mai dificil de spart (mai ales că în lucrarea originală, asupra acestuia a fost aplicat un algoritm de criptare) și, folosind clustering-ul, obținem o imagine cu zgomot, care este mai rezistentă la atacurile potențialilor terțe malicioase, aceștia nemaiputând să distingă între poza originală și pixelii care au fost modificați.

Pe de altă parte, culorile nu sunt distribuite egal și ne așteptăm ca unele clustere să fie mai mari decât altele. Astfel, dacă avem un mesaj de o dimensiune mai ridicată și facem împărțirea acestuia în trei bucăți egale, așa cum este descris în algoritmul original, este posibil să avem pierdem din mesajul transmis, de aceea, o posibilă îmbunătățire ar fi să împărțim mesajul în trei bucăți direct proporționale cu dimensiunea celor trei clustere obținute; sau, alternativ, să avem mai multe clustere.

### *2.3.5. Algoritmul LSB propus*

În cele ce urmează, vom propune un algoritm de steganografie bazat pe tehnica LSB. Motivul alegerii acestei metodologii se datorează faptului că, în literatura de specialitate, este bine-cunoscut că un algoritm steganografic folosind LSB este mai vulnerabil decât alte tehnici precum DCT (eng. original: *Discrete Cosine Transform*) [[6]](#footnote-6) sau DWT (eng. original: *Discrete Wavelet Transform*), doar pentru a numi câteva [5]. Prin urmare, acestă idee contribuie la steganografia LSB, într-o încercare de a dezvolta un algoritm mai sigur, solid și robust, menținându-și în același timp reputația ca o implementare relativ simplă, în comparație cu alte metode.

Un prim pas logic constă în criptarea mesajului. În acest fel, se face legătura dintre criptografie și steganografie, iar securitatea algoritmului este îmbunătățită substanțial. Pasul de criptare protejează împotriva posibilității extragerii neautorizate a datelor din fișierul de acoperire, prin atacuri de steganaliză [[7]](#footnote-7). Ne putem gândi că acest aspect servește drept ultimă soluție în cazul în care un atac de steganaliză are succes. Pentru criptarea mesajului secret vom folosi algoritmul AES, un algoritm simetric, rapid și care se realizează fără pierderea anumitor date din memorie.

#### **2.3.5.1. AES**

AES (eng. original: *Advanced Encryption Standard*) este un subset al familiei de algoritmi Rijndael [25], algoritmi este de tip bloc (eng. original: *block cipher*). AES prezintă următoarele caracteristici:

* Blocul are lungimea de 128 de biți;
* Cheia de criptare are lungimea variată și poate lua valori din mulțimea (reprezentare pe biți).

AES a fost selectat de guvernul SUA pentru a proteja informațiile clasificate și, în zilele noastre, este folosit pe tot mapamondul cu scopul de a cripta date cu caracter sensibil, fiind standardul pentru securitatea cibernetică. Institutul Național de Standarde și Tehnologii (NIST din eng. original) a început cercetările pentru AES în anul 1997, când, în urma examinărilor și a testelor, a fost observat că DES (eng. original: *Data Encryption Standard*) a început să devină vulnerabil la atacurile cu forță brută [26].

Pentru o securitate optimă, vom folosi cheia de 256 de biți (cea mai mare oferită de standard). AES funcționează, mai degrabă, pe bytes decât pe biți. Algoritmul are lungimea de bloc de 128 de biți, adică 16 bytes, care sunt stocați într-o matrice de , astfel:

Dimensiunea cheii utilizate specifică numărul de runde de transformări succesive care convertesc textul original, în mesajul criptat. Pentru o cheie de dimensiune 256, ca cea aleasă, sunt necesare 14 runde. Fiecare rundă constă în patru operații (*SubBytes, ShiftRows, MixColumns, AddRoundKey*), iar algoritmul este prezentat așa cum este el descris în [27]:

1. *KeyExpansion*: cheile de rundă (eng. original: *round key*) sunt derivate din cheia originală de criptare, fiecare depinzând de cea creată anterior, folosind algoritmul de „AES Key Schedule”.
2. *AddRoundKey*: între starea inițială (mesajul clar) și prima cheie de rundă se realizează operația de XOR-are.
3. **13 Runde a câte patru Operații**:
   1. *SubBytes*: fiecare octet este înlocuit de un altul conform tabelei de substituție S-box;
   2. *ShiftRows*: este un pas de transpoziție în care ultimele trei rânduri ale matricei sunt shift-ate cu un anumit număr de pași (dacă începem numărătoarea rândurilor de la zero, atunci, valorile primului rând sunt shift-ate la stânga cu o coloană; pe al doilea rând cu două coloane și pe al treilea cu trei, iar rândul 0 rămâne neschimbat):

* 1. *MixColumns*: fiecare coloană a matricei stare este înmulțită cu , obținându-se o nouă matrice;
  2. *AddRoundKey*: XOR-are între cheia de rundă corespunzătoare și matrice.

1. **A 14-a Rundă (Runda Finală)**:
   1. *SubBytes*
   2. *ShiftRows*
   3. *AddRoundKey*

Cifrările de tip bloc au două proprietăți cheie care îi fac imuni în fața atacatorilor și anume confuzia și difuzia. Formal, confuzia înseamnă că inputul și outputul nu au nicio conexiune între ele și difuzia se referă la faptul că modificări mici ale inputului au un impact gigantic asupra outputului [28]. În AES, confuzia este adăugată la pasul *SubBytes*, iar difuzia se produce la pașii *ShiftRows* și *MixColumns*. De asemenea, AES este eficient și din punct de vedere computațional deoarece criptarea, respectiv decriptarea, nu fac uz de multe resurse.

În continuare, pentru a îmbunătăți securitatea algoritmului LSB, vom lua în considerare faptul că este slab împotriva atacurilor de steganaliză din princina pixelilor modificați care sunt așezați unul după altul. De aceea, propunem să îi amestecăm, pentru a putea folosi întreaga capacitate a fotografiei de acoperire și a crește gradul de siguranță.

#### **2.3.5.2. Algoritmul modern de amestecare Fisher-Yates (versiunea lui Durstenfeld)**

După cum am menționat și anterior, vom genera o secvență unică, aleatoare de pixeli a unei imagini. Secvența poate fi creată manual și de utilizator. Aceasta va acționa drept cheie steganografică, iar fără ea nu vom putea recupera mesajul secret. Adăugăm astfel un nou strat de securitate algoritmului.

Pentru a obține acest lucru, vom folosi versiunea modernă a algoritmului de amestecare Fisher-Yates, introdusă de Richard Durstenfeld în 1964 [29] și popularizată de Donald E. Knuth, în 1969, în cartea sa *The Art of Computer Programming* [30], deoarece aceasta are o complexitate de , față de a implementării naive. Algoritmul de permutare a pixelilor este:

**Date de Intrare**: n = numărul total de pixeli ai imaginii de acoperire;

m = numărul total de bytes ai mesajului secret \* 3; (pentru că avem nevoie de 3 pixeli pentru LSB)

**Date de Ieșire**: vectorul P de pixeli amestecați;

w = 0;

**while** w n-1 do

aux[w] = w; w++;

**end**;

w = 0;

**while** w m-1 do

j = valoare random din intervalul [0; m-w-1];

swap(aux[w], aux[j]);

**end**;

P[m]; w = 0;

**while** w m-1 do

P[w] = aux[n-1-w];

**end**;

În esență, folosim acest algoritm pe post de PRNG (eng. original: *Pseudo-Random Number Generator*). Întrucât secvența numerelor generate aleatoriu depinde, în întregime, de input, acest tip de PRNG nu poate produce mai multe permutări distincte decât numărul maxim de stări posibile distincte. Chiar și atunci când numărul stărilor posibile depășește numărul permutărilor, natura neregulată a generării produce ca unele permutări să apară mai des decât altele. Astfel, pentru a minimiza bias-ul, numărul total posibil de stări ar trebui să depășească numărul de posibile PRNG-uri cu mai multe ordine [31]. În cazul nostru, ar trebui să ne asigurăm că este suficient de mare, iar, de cele mai multe ori, acest obiectiv este îndeplinit.

Ca să putem folosi pixelii trebuie să cunoaștem locația lor în poză, deci să știm rândul și coloana. Aceste informații se obțin ușor. Dacă considerăm *P* ca fiind numărul pixelului din secvența generată aleatoriu și *N, M* reprezintă numărul de linii, respectiv de coloane ale pozei, atunci:

După ce vom folosi secvența pentru a încorpora mesajul secret în pixelii corespunzători, aceasta va fi codificată folosind AES și va fi transmisă împreună cu poza de acoperire pentru procesul de decodificare.

#### **2.3.5.3. Criptarea/Decriptarea**

Pe scurt, algoritmul propus se prezintă astfel:

**Date de Intrare**: I = imaginea; M = mesajul secret; P1 = parolă pentru mesajul secret; P2 = parolă pentru secvența de Pixeli

**Date de Ieșire**: S = imaginea modificată; Pix = secvența criptată de pixeli

***Pas 1***: Se criptează mesajul secret (M) cu AES folosind parola secretă, P1, rezultând N.

***Pas 2***: Se generează secvența aleatoare de pixeli, P, pornind de la lungimea mesajului secret criptat și de la mărimei imaginii de acoperire, I.

***Pas 3***: Se încorporează N în pixelii generați aleatoriu în P, folosind LSB și obținem S.

***Pas 4***: Se criptează P cu AES, folosind P2 și obținem Pix.

***Pas 5***: Returnăm S și Pix.

Pentru decriptare, se parcurg pașii în ordine inversă, astfel:

**Date de Intrare**: S = imaginea modificată; Pix = secvența criptată de pixeli; P1 = parolă pentru mesajul secret; P2 = parolă pentru secvența de Pixeli

**Date de Ieșire**: M = mesajul secret

***Pas 1***: Se decriptează Pix folosind P2 (AES) și obținem P.

***Pas 2***: Din S extragem biții din pixelii marcați, din P, și obținem N.

***Pas 3***: Se decriptează N folosind P1 (AES) și obținem M.

***Pas 4***: Se returnează M.

Este important ca cele două parole, pentru mesajul secret și pentru secvența aleatoare de pixeli, să fie transmise într-un mod sigur pe canalul de comunicații, altfel algoritmul devine inutil. Ca recomandare, acestea ar trebui să fie diferite ca să nu faciliteze munca unei terțe malițioase. Există mai multe protocoale sigure din care utilizatorul poate alege. Dintre acestea amintim Diffie-Hellman cu semnături digitale sau RSA [32]. Nu vom intra în detalii deoarece nu constituie tema lucrării de față.

Metoda propusă nu este foarte eficientă din punct de vedere al memoriei deoarce trebuie să trimitem secvența criptată de pixeli, alături de imaginea rezultată prin tehnica steganografică, dar facem acest compromis pentru a obține o mai bună securitate.

### *2.3.6. Algoritmul DCT*

Tehnicile steganografice pot fi clasificate în metode aplicate în domeniul spațial și metode aplicate în domeniul de frecvență [33].

Până acum au fost prezentate doar metode care făceau uz de domeniul spațial și anume algoritmii de tip LSB, deoarece procesarea se aplică direct asupra valorilor pixelilor din imaginea de acoperire. Aceste metode sunt utilizate pe scară largă datorită simplității lor. Cu toate acestea, sunt extrem de vulnerabile la modificări, chiar și mici, care au fost aplicate fișierului de acoperire. Un atacator poate executa pur și simplu tehnici de procesare a semnalului pentru a distruge în întregime informațiile secrete. În multe cazuri, chiar și micile modificări rezultate din sistemele de compresie cu pierderi (eng. original: *lossy compression*) produc pierderea mesajului secret.

Conceptul de *compresie a imaginilor digitale* se referă la reducerea dimensiunii fișierului, fără a scădea calitatea la un nivel inacceptabil. Există două tipuri de compresie a imaginilor digitale [34]:

1. Compresia reversibilă fără pierderi de date (eng. original: *lossless*): este redusă dimensiunea unui fișier, menținând, în același timp, calitatea ca înainte de comprimare. Deci calitatea datelor nu este compromisă, iar fișierul poate fi restaurat în forma sa originală. Câțiva algoritmi de tip „lossless” sunt codificarea Huffman, codificare pe lungimea de rulare (eng. original: *run-length encoding*) etc...
2. Compresia ireversibilă cu pierderi de date (eng. original: *lossy*): în această tehnică sunt eliminate datele care sunt considerate redundante pentru percepția umană, pentru a micșora spațiul de memorie ocupat. Fișierul comprimat nu poate fi restaurat în forma originală exactă. În acest tip de compresie, calitatea datelor este periclitată și dimensiunea datelor se modifică, algoritmii fiind construiți pe teoria rată-distorsiune [35] (eng. original: *rate-distortion*). Aceasta implică stabilirea celei mai mici cantități de biți, din fișierul de intrare, ce poate fi transmisă, prin canalul de comunicație, pentru ca, la destinație, să poată fi receptate cu o valoare a distorsiunii minimale. Dintre cele mai utilizate tehnici *lossy* enumerăm: algoritmii de compresie fractală, DCT (eng. original: *Discrete Cosine Transform*) etc...

În metodele care sunt aplicate pe domeniul de frecvență, cum ar fi DCT (eng. original: *Discrete Cosine Transform*) [36], DWT (eng. original: *Discrete Wavelet Transform*) [37] sau DFT (eng. original: *Discrete Fourier Transform* [38]) [21], imaginea de acoperire este transformată mai întâi într-un domeniu diferit. Apoi, coeficienții transformați sunt prelucrați pentru a ascunde informațiile secrete. Noii coeficienți sunt transformați înapoi în domeniu spațial pentru a obține imaginea steganografică [5]. Acest tip de tehnici ascund mesajele în zone semnificative ale pozei, ceea ce le face mai robuste la atacuri, cum ar fi compresia, decuparea etc..., față de abordarea LSB. Cu toate acestea, ele rămân imperceptibile pentru sistemul senzorial uman, dar sunt mai complexe din perspectiva calculului și prezintă o capacitate de încorporare mai mică. Multe metode de transformare a domeniului sunt independente de formatul imaginii și pot supraviețui conversiei între formatele *lossy* și cele *lossless*.

Algoritmul Jsteg [39] este o versiune modificată a algoritmului de codificare JPEG standardizat și a fost inventat de Derek Upham. Ideea principală din spatele acestui algoritm este de a manipula porțiunea (DCT) a procesului standard de codificare JPEG pentru a încorpora datele secrete. JSteg codifică datele secrete în bitul cel mai puțin semnificativ (LSB) al coeficienților[[8]](#footnote-8) care nu sunt egali cu -1, 0 sau +1 [40]. Practic, algoritmul JSteg este copia exactă a metodei de încorporare LSB, dar în domeniul spațial.

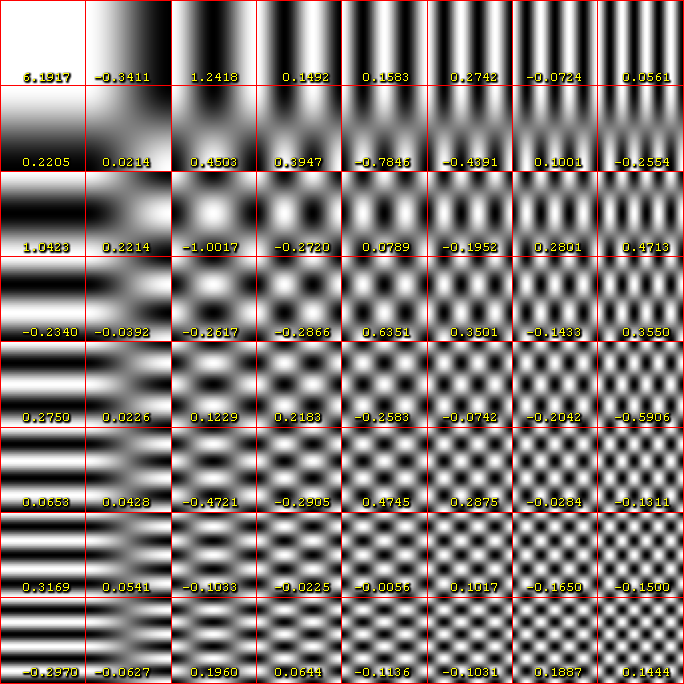
DCT este reprezentat de o funcție liniară, inversabilă care exprimă o secvență finită de date într-o sumă de funcții ale cosinusului oscilând la diferite frecvențe. [41]

Există multe versiuni de DCT, dar, de interes, este **DCT** de tip **II** (eng. original: *orthonormal Forward Discrete Cosine Transform*) și **DCT** de tip **III**, care este inversa DCT-II și se mai notează și **IDCT** (eng. original: *Inverse Discrete Cosine Transform*). Un coeficient **DCT**, , este exprimat prin următoarea ecuație [42]:

unde este intensitatea (sau *gray level* – eng. original) pixelului în punctele , și ***N*** este dimensiunea blocului de pixeli.

Fiind inversa lui DCT-II, **IDCT** are următoarea formulă [42]:

unde, .

Înainte ca orice transformare să poată fi implementată, imaginea de acoperire trebuie împărțită în blocuri de 8x8, care nu se suprapun. Aceste blocuri sunt introduse secvențial în DCT, începând din colțul din stânga sus al imaginii, avansând spre dreapta, până în colțul din dreapta jos.

Figură 2.3.6. 1 - Reprezentare Grafică a DCT și Coeficienților Corespunzători, aplicați pe o Imagine 8x8 [67]

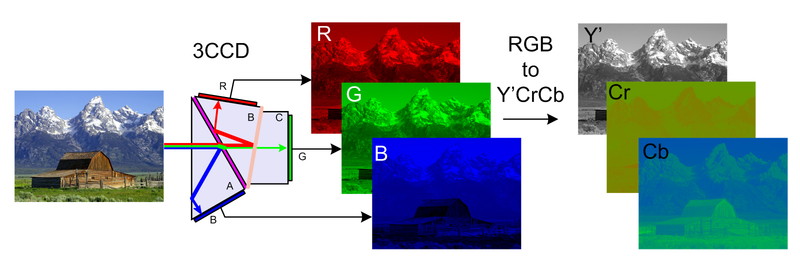
Deoarece algoritmul folosește blocuri de pixeli de 8x8, atunci ***Formula 2.3.6.1.*** [13] devine:

unde, .

După cum am precizat, DCT se aplică unui bloc de 8x8 pixeli, obținând astfel **64** coeficienți, dintre care **1 DC** (eng. original: *Direct Current Term*), care este reprezentat de D[0][0], egal, ca valoare, cu media tuturor pixelilor din blocul selectat de dimensiune 8x8 și **63 AC** (eng. original: *Alternating Current Terms*), reprezentând schimbarea culorii în bloc, crescând frecvența cu cât ne apropiem de colțul din dreapta jos (D[7][7]) [43].

Pașii algoritmului steganografic JSteg sunt următorii, așa cum au fost ei prezentați de autorul lui [44]:

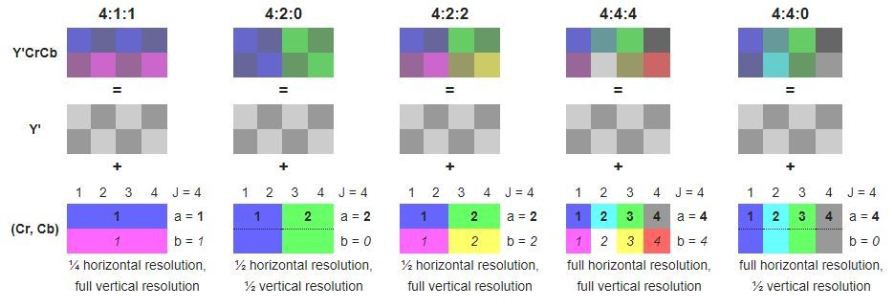
* Mai întâi, se citește imaginea de acoperire și se face verficarea dacă lungimea și lățimea acesteia sunt multiplu de 8. În cazul în care nu respectă condiția, imaginea este redimensionată astfel încât să poată fi împărțită uniform în blocuri de 8x8 pixeli.
* După cum am menționat anterior, codificarea JPEG este de o codificare cu pierderi, ceea ce înseamnă că apar modificări pentru a fi economisit spațiul de memorie. Imaginea este convertită din formatul *RGB*, în formatul *YUV* (sau mai este notat *YCbCr*).



Figură 2.3.6. 2 - Transformare RGB -> YcrCb [45]

Canalul **Y** stochează gradul de luminozitate al unui pixel, iar **U** și **V** sunt canalele cromatice [46]. D.L. Currie și C.E. Irvine ilustrează în [47] că ochiul uman este mult mai sensibil la schimbările de luminozitate ale pixelilor, decât la modificările de culoare. Formulele de conversie din formatul RGB în YCbCr sunt [5]:

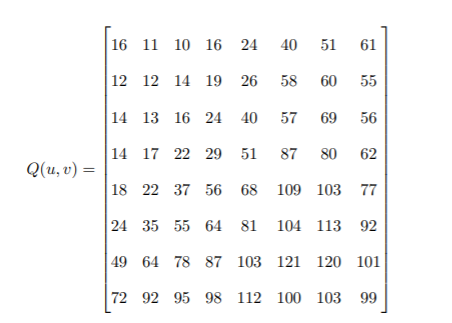
* Canalele U și V sunt, în mod normal, reduse la 50% din lățimea și înălțimea inițiale, deoarece ochiul uman are mai multă sensibilitate la variațiile de luminozitate decât la culoare. Acest procedeu poartă denumirea de eșantionare a culorilor (eng. original: *Chroma Subsampling*), iar, pentru compresia JPEG, cea mai întâlnită rată este 4:2:0 [46].



Figură 2.3.6. 3 – Tipuri Chroma Subsampling [48]

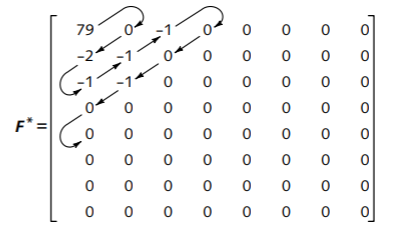
* Imaginea de acoperire este împărțită în blocuri de pixeli de 8x8. Asupra acestora este aplicat DCT și se obține matricea celor 64 de coeficienți.
* Următoarea etapă este cunatificarea coeficienților furnizați în matricea DCT. O slăbiciune biologică a ochiului uman este exploatată și anume: ochiul uman este destul de bun pentru a distinge diferențele de luminozitate în frecvențe joase, dar nu în frecvențele înalte [49]. Pentru a cuantifica numerele reale din matricea DCT (notată cu *D*), vom împărții coeficienții la valorile corespunzătoare din matricea standard *Q* și îi vom rotunji la cea mai apropiată valoare întreagă, obținând astfel o nouă matrice (*Matrix*), în care coeficienții cu magnitudine mică sunt cei mai afectați de cuantificare și vor avea valoarea zero, iar coeficienții semnificativi vor rămâne aproape de valoarea lor inițială [50]. Aceasta este etapa în care datele sunt pierdute. Pentru steganografie, natura cu pierderi a cuantificării este una de dorit, deoarece ajută la obturarea modificărilor realizate de încorporarea mesajului secret. Formal, acest pas poate fi descris matematic, astfel:

Matricea standard de cuantificare este:



Figură 2.3.6. 4 - Matrice Standard de Cuantificare JPEG [51]

* Coeficienții sunt sortați printr-o metodă de parcurgere a matriciei în Zigzag. De-a lungul acestui zigzag, se utilizează o formă de codificare a lungimii de rulare (eng. original: *Run Length Encoding* - **RLE**), care comprimă coeficienții cu valoarea 0 (lucru foarte benefic, deoarece multe dintre valorile așezate spre colțul din dreapta-jos al matricei sunt zero). Ulterior, se realizează o codificare Huffman.



Figură 2.3.6. 5 - Scanare ZigZag [49]

Folosind exemplul de mai sus, codarea RLE va arăta: [(0, 7), 79], [(1, 2), -2], [(0, 1), -1], [(0, 1), -1], [(0, 1), -1], [(2, 1), -1], [(0, 0)], unde, pentru o configurație **[(l, m), n]**, termenii din componența acesteia au următoarele semnificații:

* + **l**: este definit ca numărul de zeroruri care se găsesc înaintea valorii de la poziția curentă;
  + **m**: numărul de biți care sunt suficienți pentru a construi, în memorie, valoarea de la poziția curentă;
  + **n**: valoarea de la poziția curentă.
* Deoarece coeficientul DC (*D[0][0]*) este media tuturor pixelilor din bloc, modificările aduse asupra acestuia determină cea mai mare distorsiunea a imaginii rezultate. Ca atare, algoritmul JSteg evită acești coeficienți și coeficienții AC care au valorile în mulțimea , pentru procesul de încorporare a mesajului secret. Ca ***îmbunătățire***, propunem să luăm această restricție și să o extindem în sensul în care vom limita selecția de coeficienți AC doar la cei . Propunerea vine în urma observației din [46] care menționează că AC cu valoare negativă sunt mai susceptibili la corupția datelor și, de asemenea, provoacă mai multe distorsiuni în imagine.
* Blocurile de 8x8 sunt alăturate și obținem imaginea steganografică.

# **3. Analiza performanței**

Pentru analiza performanțelor, au fost implementați algoritmii descriși în secțiunile 2.3.1., 2.3.3., 2.3.4., 2.3.5., 2.3.6. și anume algoritmii LSB Simplu, LSB folosind conceptul de Liste Înlănțuite, LSB folosind K-Means Clustering, algoritmul propus de LSB și DCT. Deoarece primii 4 algoritmi primesc ca input și pot funcționa doar utilizând imagini de tip .png (sau alt format de tip *lossless*), algoritmul pentru DCT folosește o versiune modificată de Jsteg, unde primește ca input tot fișiere cu extensia .png; de aceea pașii de RLE și codificare Huffman nu au mai fost necesari în implementarea algoritmului, deoarece aceia sunt specifici compresiei de tip JPEG.

## 3.1. Tehnologii utilizate și justificarea selectării acestora

Algoritmii au fost implementați folosind limbajul Python3 [52], iar ca mediu de dezvoltare, a fost utilizată platforma Google Colab [53].

Python a fost ales deoarece facilitează lucrul cu fișierele, fie ele text sau imagine (librăriile *CV2* [54] sau *Pillow* [55], în funcție de operațiile care sunt necesare asupra fotografiilor; *CV2* este mult mai rapidă și complexă, în cadrul căreia există funcții pentru calculul coeficienților DCT, funcție pentru transformarea din RGB în YcbCr și invers etc..., spre deosebire de biblioteca *Pillow* care este folosită pentru operații de tipul tăiere și redimensionare de fotografii). Algoritmul AES poate fi implementat prin intermediul librăriei *Pycrypto* [56], iar biblioteca *hashlib,* care vine preinstalată cu Python, este folosită pentru funcțiile hash, în acest fel verificând validitatea unei parole. *Bitstring* [57] este un modul Python conceput pentru a ajuta la crearea și analiza datelor binare într-un mod cât mai simplu și natural cu putință.

De asemenea, Python este bine cunoscut ca fiind folosit pentru tehnici de învățare automată, cum este algoritmul K-Means care a fost implementat folosind librăria *sklearn* [58].

Google Colab, sau, pe scurt, Colab, a fost utilizat ca mediu de dezvoltare deoarece este o unealtă open-source, dezvoltată de Google Corporation, care oferă acces la spațiu Cloud, cu resurse GPU gratuite și puternice, dar și o memorie de disc destul de mare de aproximativ 70 GB.

## 3.2. Măsurarea performanței

Pentru testele de performanță au fost selectate următoarele fotografii, des utilizate în studiul asupra imaginilor digitale:

Figură 3.2. 1 - Lenna.png [12] Figură 3.2. 2 – Pepper.png [59] Figură 3.2. 3 – Baboon.png [59]

Prima fotografie are dimensiunea 225x225, a doua are dimensiunea 320x320 și ultima 512x512. Prima fotografie a fost selectată special cu dimensiuni nedivizibile la 8 pentru a vedea cum se compară imaginea rezultată din algoritmul DCT modificat cu restul.

Mesajul secret a fost ales să aibă dimensiunea de aproximativ 2.5 KB, pentru a observa cum se descurcă algoritmii în fața unui mesaj secret de dimensiuni ridicate.

Din punct de vedere vizual, obținem următoarele rezultate:

* + LSB Simplu:



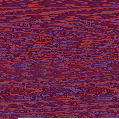
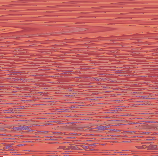
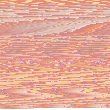
*Figură 3.2. 4 – Stego LSB Simplu*

* + LSB Liste Înlănțuite:



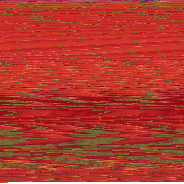
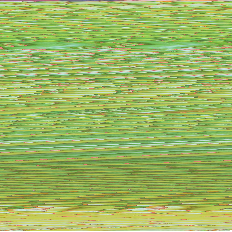
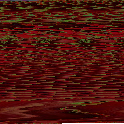
*Figură 3.2. 5 – Stego LSB Liste*

* + LSB K-Means Clustering (după cum am menționat, vom obține 3 fotografii pentru fiecare imagine introdusă):
    - Pentru Lenna.png:

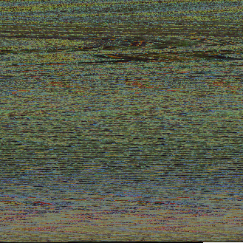
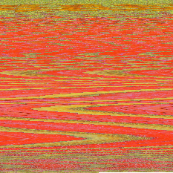
*Figură 3.2. 6 – LSB K-Means Lenna.png*

* + - Pentru Pepper.png:

*Figură 3.2. 6 – LSB K-Means Pepper.png*

* + - Pentru Baboon.png:

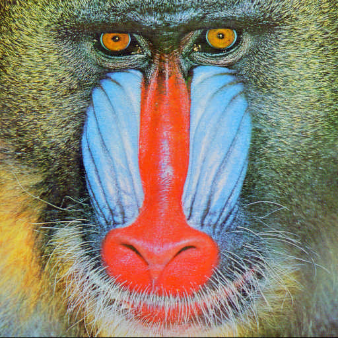
*Figură 3.2. 7 – LSB K-Means Fruits.png*

* + LSB Algoritm Propus:



*Figură 3.2. 8 – Stego LSB Original*

* + DCT:

*Figură 3.2. 9 – Stego DCT*

După cum se poate observa, cu excepția LSB-ului prin K-Means care returnează pozele clusterelor și DCT-ului atunci când este dat ca input o fotografie cu dimensiuni nedivizibile cu 8, caz în care algoritmul va redimensiona fișierul de acoperire, adăugând padding, ceea ce rezultă într-o poză mai neclară, ceilalți algoritmi întorc imagini identice cu cele oferite ca input, deci, obiectivul steganografic de a rămâne vizual nedetectabil, este îndeplinit.

Dacă luăm în calcul capacitatea de încorporare a mesajului, algoritmii LSB Simplu, LSB cu Liste Înlănțuite și algoritmul LSB propus, au trecut testul cu brio: aceștia au înglobat întregul mesaj secret în imaginea de acoperire, fără pierderi da date. În contrast, LSB folosind K-Means Clustering nu a reușit să codeze tot fișierul text în poza Pepper.png, deoarece ultimul cluster era prea mic. Dacă, în schimb de 3 clustere, folosim 5, reușim să rezolvăm această problemă. În mod similar, dacă împărțim mesajul în funcție de dimensiunile celor trei clustere evităm problema de capacitate. DCT are aceași problemă: pentru Lenna.png nu reușește că codeze tot mesajul secret.

Ca timp de execuție, algoritmii au următoarele performanțe:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Algoritmi** | **Imagini** | **Codare** | **MedieC** | **Decodare** | **MedieD** |
| **LSB Simplu** | *Lenna.png* | 0.04 | 0.066 | 0.011 | 0.013 |
| *Pepper.png* | 0.07 | 0.014 |
| *Baboon.png* | 0.09 | 0.015 |
| **LSB Liste** | *Lenna.png* | 0.15 | 0.28 | 0.047 | 0.148 |
| *Pepper.png* | 0.24 | 0.048 |
| *Baboon.png* | 0.45 | 0.053 |
| **LSB K-Means** | *Lenna.png* | 0.58 | 1.61 | 0.014 | 0.018 |
| *Pepper.png* | 1.35 | 0.019 |
| *Baboon.png* | 2.92 | 0.022 |
| **LSB Propus** | *Lenna.png* | 0.29 | 0.49 | 0.172 | 0.223 |
| *Pepper.png* | 0.41 | 0.213 |
| *Baboon.png* | 0.78 | 0.285 |
| **DCT** | *Lenna.png* | 1.57 | 2.87 | 0.687 | 4.698 |
| *Pepper.png* | 2.96 | 1.588 |
| *Baboon.png* | 4.09 | 2.423 |

*Figură 3.2. 10 – Tabel Timpi Execuție*

Se poate observa că metoda propusă este destul de rapidă în faza de codare, ocupând mai mult timp în etapa de decodare. Datorită operațiilor de criptare și permutare a pozițiilor pixelilor ne așteptam la o creștere a timpului de execuție, dar remarcăm că se compară destul de mine cu timpii celorlați algoritmi.

**PSNR** (eng. original: *Peak Signal to Noise Ratio*) este metrica primară utilizată pentru a măsura cât de perceptibil a fost zgomotul rezultat, cauzat de algoritm în rezultatele colectate [60]. PSNR este o metrică de eroare și este considerat a fi raportul dintre forța maximă a unui semnal digital și cantitatea de *noise* nedorit, care îl afectează. O valoare mai mare de PSNR implică o distorsiune mai mică. Formula de calcul este:

unde, este numărul maxim de pixeli al imaginii de acoperire.

**MSE** (eng. original: *Mean Square Error*) măsoară media pătratului „erorii”; eroarea fiind suma cu care estimatorul diferă față de cantitatea de estimat [61]. Cu cât MES are o valoare mai mică, cu atât eroarea este mai mică. Dată o imagine *I*, fără zgomot, de dimensiune *m*x*n* și aproximarea ei zgomotoasă *K*, MSE este calculat folosind:

**SSIM** (eng. original: *Structural Similarity*) este folosit pentru a măsura asemănarea dintre două fotografii, din punct de vedere al percepției. Diferența dintre această metodă și tehnicile menționate anterior este aceea că PSNR și MSE calculează erori absolute, în schimb, SSIM este un model bazat pe percepție, care consideră degradarea imaginii ca o schimbare percepută în informațiile structurale, încorporând totodată fenomene perceptive importante, printre care enumerăm atât termeni de mascare a luminozității, cât și termeni de mascare a contrastului. Informația structurală este ideea că pixelii au interdependențe puternice, mai ales atunci când sunt apropiați spațial. Aceste dependențe poartă informații importante despre structura obiectelor din scena vizuală. SSIM poate lua valori între -1 și 1: un scor egal cu 1 înseamnă că cele două fotografii sunt foarte asemănătoare și -1 reprezintă contrariul. Indicele SSIM este calculat pe diferite blocuri ale unei imagini. Măsura dintre două astfel de blocuri *x* și *y*, cu dimensiunea comună de *N*x*N*, este [62]:

unde, este media lui *x*, a lui *y*, și varianța lui *x*, respectiv *y*, este covariația lui *x* și *y*, și sunt două variabile pentru stabilizarea diviziunii cu numitor slab, iar *L* reprezintă intervalul dinamic al valorilor pixelilor.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Algoritmi** | **Fotografii** | | **MSE** | **Medie** | **PSNR** | **Medie** | **SSIM** | **Medie** |
| **LSB Simplu** | *Lenna.png* | | 1.760 | 0.603 | 45.673 | 58.206 | 0.994 | 0.997 |
| *Pepper.png* | | 0.037 | 62.419 | 0.999 |
| *Baboon.png* | | 0.014 | 66.527 | 0.999 |
| **LSB Liste** | *Lenna.png* | | 1.886 | 0.674 | 45.371 | 55.238 | 0.993 | 0.997 |
| *Pepper.png* | | 0.100 | 58.122 | 0.999 |
| *Baboon.png* | | 0.038 | 62.222 | 0.999 |
| **LSB K-Means** | *Lenna.png* | Cluster1 | 0.050 | 0.047 | 61.083 | 62.521 | 0.999 | 0.998 |
| Cluster2 | 0.091 | 58.516 | 0.999 |
| Cluster3 | 0.105 | 57.906 | 0.998 |
| *Pepper.png* | Cluster1 | 0.072 | 59.514 | 0.997 |
| Cluster2 | 0.023 | 64.458 | 0.998 |
| Cluster3 | 0.038 | 62.257 | 0.999 |
| *Baboon.png* | Cluster1 | 0.012 | 67.098 | 0.998 |
| Cluster2 | 0.011 | 67.443 | 0.999 |
| Cluster3 | 0.023 | 64.430 | 0.998 |
| **LSB Propus** | *Lenna.png* | | 1.835 | 0.646 | 45.493 | 56.111 | 0.994 | 0.997 |
| *Pepper.png* | | 0.074 | 59.397 | 0.999 |
| *Baboon.png* | | 0.029 | 63.444 | 0.999 |
| **DCT** | *Lenna.png* | | - | 41.634 | - | 32.537 | - | 0.877 |
| *Pepper.png* | | 21.162 | 34.875 | 0.924 |
| *Baboon.png* | | 62.106 | 30.199 | 0.830 |

*Figură 3.2. 11 – Tabel Performanțe Execuție*

La DCT, nu am putut verifica metricile de performanță pe Lenna.png deoarece algoritmul redimensionează orice fotografie care nu are dimensiunile divizibile cu opt și nu puteam să comparăm două imagini cu proporții diferite. În general, algoritmul DCT oferă performanțele cele mai slabe. Era și de așteptat în condițiile în care acesta realizează o compresie a imaginilor date ca input.

Algoritmul propus are performanțe bune în comparație cu celelalte metode prezentate. Remarcăm că performanțele acestuia sunt direct proporționale cu dimensiunea fotografiei și a mesajului secret introdus.

În final, analizăm histogramele:

|  |  |  |  |
| --- | --- | --- | --- |
| **Algoritm** | **Fotografii** | **Poză Originală** | **Poză Stego** |
| **LSB Simplu** | *Lenna* | C:\Users\Larisa\Desktop\pin1.png | C:\Users\Larisa\Desktop\pout1.png |
| *Pepper* | C:\Users\Larisa\Desktop\pin2.png | C:\Users\Larisa\Desktop\pout2.png |
| *Baboon* | C:\Users\Larisa\Desktop\pin3.png | C:\Users\Larisa\Desktop\pout3.png |
| **LSB Liste** | *Lenna* | C:\Users\Larisa\Desktop\pin1.png | C:\Users\Larisa\Desktop\pout1.png |
| *Pepper* | C:\Users\Larisa\Desktop\pin2.png | C:\Users\Larisa\Desktop\pout2.png |
| *Baboon* | C:\Users\Larisa\Desktop\pin3.png | C:\Users\Larisa\Desktop\pout3.png |
| **LSB K-Means** | *Cluster1 – Pepper* | C:\Users\Larisa\Desktop\pcluster1_img.png | C:\Users\Larisa\Desktop\pa.png |
| *Cluster2 – Pepper* | C:\Users\Larisa\Desktop\pcluster2_img.png | C:\Users\Larisa\Desktop\pb.png |
| *Cluster3 – Pepper* | C:\Users\Larisa\Desktop\pcluster3_img.png | C:\Users\Larisa\Desktop\pc.png |
| **LSB Propus** | *Lenna* | C:\Users\Larisa\Desktop\pin1.png | C:\Users\Larisa\Desktop\pout1.png |
| *Pepper* | C:\Users\Larisa\Desktop\pin2.png | C:\Users\Larisa\Desktop\pout2.png |
| *Baboon* | C:\Users\Larisa\Desktop\pin3.png | C:\Users\Larisa\Desktop\pout3.png |
| **DCT** | *Lenna* | C:\Users\Larisa\Desktop\pin1.png | C:\Users\Larisa\Desktop\pout1.png |
| *Pepper* | C:\Users\Larisa\Desktop\pin2.png | C:\Users\Larisa\Desktop\pout2.png |
| *Baboon* | C:\Users\Larisa\Desktop\pin3.png | C:\Users\Larisa\Desktop\pout3.png |

*Figură 3.2. 12 – Tabel Histograme*

După cum putem remarca, histogramele prezintă o deviere minimală din perspectiva modificărilor aduse pixelilor în urma aplicării algoritmilor. Algoritmul propus are performanțe bune alături de celelalte metode prezentate. Remarcăm că performanțele acestuia sunt direct proporționale cu dimensiunea fotografiei și a mesajului secret introdus.

# **4. Concluzii**

Nicio comunicare nu poate fi considerată privată pe internet. Încălcările de confidențialitate se întâmplă în fiecare an, inclusiv când vine vorba de guverne care exploatează legile, printre alte incidente de invazie a confidențialității expuse de whistleblowers precum Edward Snowden, care a spus „Nu vreau să trăiesc într-o lume în care tot ceea ce spun, tot ceea ce fac, orice expresie de creativitate sau iubire sau prietenie este înregistrată, mai ales fără acordul meu.” Securitatea contra acestor probleme este furnizată sub forma criptografiei: având datele de pe internet criptate, doar pentru a fi decriptate de către cei care sunt autorizați să le vizualizeze. Cu toate acestea, criptografia nu ascunde faptul că există un canal de comunicare între două sau mai multe părți. Aici intervine steganografia. Cu steganografia digitală, comunicarea sub acoperire poate fi stabilită între două sau mai multe terțe, utilizând suporturi digitale de acoperire inocente, care păstrează în secret datele încorporate în interiorul acestora. Aceste date pot fi apoi extrase numai de cei care se așteaptă ca ele să fie ascunse în interiorul fișierului de acoperire. Iar printre cele mai des utilizate fișiere multimedia transmise prin intermediul internetului se numără fotografiile.

În lucrarea de față am însușit noțiunile principale care stau la baza steganografiei și a procesării imaginilor digitale, am revizuit câțiva algoritmi importanți din domeniul steganografiei, le-am comparat performanțele și am propus o metodă nouă de ascundere a datelor, care s-a dovedit a fi destul de eficientă, cu rezultate promițătoare, trecândând cu brio testele la care a fost supus și reușind să țină pasul cu alți algoritmi din lucrări de specialitate. Din păcate, tehnica propusă are și limitări deoarece nu este foarte eficientă din punct de vedere al memoriei, dar trebuie să realizăm acest compromis pentru securitate suplimentară.

În încheiere, lucrarea de față reprezintă un punct de start în studiul tehnicilor de mascare a datelor cu caracter confidențial într-un mediu multimedia, în cazul de față fiind aleasă imaginea digitală, și sunt oferite informații și instrucțiuni amănunțite a procedeelor necesare pentru înfăptuirea acesteia.

# **5. Bibliografie**

|  |  |
| --- | --- |
| [1] | Wiktionary, „steganography,” 14 Octombrie 2019. [Interactiv]. Available: https://en.wiktionary.org/wiki/steganography. [Accesat 01 Iunie 2021]. |
| [2] | E. Cole, Hiding in Plain Sight: Steganography and the Art of Covert Communication, Statele Unite ale Americii: Bob Ipsen, 2003, p. 6. |
| [3] | A. Siper, R. Farley și C. Lombardo, „The Rise of Steganography,” în *Proceedings of Student/Faculty Research Day, CSIS, Pace University*, New York, 2005. |
| [4] | Wikipedia, „Microdot,” 22 Februarie 2021. [Interactiv]. Available: https://en.wikipedia.org/wiki/Microdot. [Accesat 01 Iunie 2021]. |
| [5] | S. Katzenbeisser și F. A. P. Peticolas, Information Hiding Techniques for Steganography and Digital Watermarking, Statele Unite ale Americii: Artech House INC., 2000. |
| [6] | R. Popa, 1998. [Interactiv]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.9413&rep=rep1&type=pdf. [Accesat 01 Iunie 2021]. |
| [7] | A. Choudary, „edureka!,” 25 Noiembrie 2020. [Interactiv]. Available: https://www.edureka.co/blog/steganography-tutorial. [Accesat 01 Iunie 2021]. |
| [8] | R. C. Gonzalez și R. E. Woods, Digital Image Processing - Third Edition, New Jersey: Prentice Hall, 2007. |
| [9] | Wikipedia, „Vector graphics,” 15 Mai 2021. [Interactiv]. Available: https://en.wikipedia.org/wiki/Vector\_graphics. [Accesat 01 Iunie 2021]. |
| [10] | Wikipedia, „Raster graphics,” 21 Mai 2021. [Interactiv]. Available: https://en.wikipedia.org/wiki/Raster\_graphics. [Accesat 01 Iunie 2021]. |
| [11] | N. Santos, „Steganography and Watermarking - Part II.C. Techniques and Tools: Forensic Data Analysis,” în *CSF: Forensics Cyber-Security (Tecnico Lisboa)*, Lisabona, 2015. |
| [12] | Noiembrie 1972. [Interactiv]. Available: https://en.wikipedia.org/wiki/File:Lenna\_(test\_image).png. [Accesat 02 Iunie 2021]. |
| [13] | J. Krenn, Ianuarie 2004. [Interactiv]. Available: https://www.krenn.nl/univ/cry/steg/article.pdf. [Accesat 03 Iunie 2021]. |
| [14] | M. S. Islam, A. Hossin, R. K. Shah și P. K. Bipin, „Bit Adjusting Image Steganography in Blue Channel using AES and Secured Hash Function,” *International Journal of Computer Science and Mobile Computing,* vol. 6, nr. 11, pp. 25-30, 2017. |
| [15] | U. Sahu și S. Mitra, „Secure Data Hiding Technique Using Video Steganography,” *International Journal of Computer Science & Communication Networks,* vol. 5, nr. 5, pp. 348-357, 2015. |
| [16] | G. Manjula și A. Danti, „A Novel Hash Based Least Significant Bit (2-3-3) Image Steganography in Spatial Domain,” *International Journal of Security, Privacy and Trust Management (IJSPTM),* vol. 4, nr. 1, 2015. |
| [17] | A. Jamsheed și B. Janet, „Improved LSB image steganography using MSB filtering of pixels,” National Institute of Technology, Tiruchirappalli, India, 2019. |
| [18] | M. Nosrati, R. Karimi, H. Nosrati și A. Nosrati, „Embedding stego-text in cover images using linked list concepts and LSB technique,” *Journal of American Science,* vol. 7, nr. 6, pp. 97-100, 2011. |
| [19] | T. H. Cormen, C. E. Leiserson, R. L. Rivest și C. Stein, Introduction to Algorithms - Third Edition, Statele Unite ale Americii: Massachusetts Institute of Technology, 2009. |
| [20] | S. S. Shetty, K. Athmaranjan, S. D. Rai și S. R. Shetty, „Image Steganography Using K-Means and DES Algorithm,” *International Journal of Research in Engineering, Science and Management,* vol. 3, nr. 6, pp. 197-201, 2020. |
| [21] | S. Xu și S. Lai, „An Optimal Least Significant Bit Based Image Steganography Algorithm,” în *ICIMCS '14: Proceedings of International Conference on Internet Multimedia Computing and Service*, New York, 2014. |
| [22] | B. Pillai, M. Mounika, P. J. Rao și P. Sriram, „Image Steganography Method Using K-Means Clustering and Encryption Techniques,” în *Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Jaipur, 2016. |
| [23] | I. Kich, E. B. Ameur și A. Souhar, „New Image Steganography Method Based on K-means Clustering,” în *Proceedings of the 2nd International Conference on Big Data, Cloud and Applications*, Tetouan, 2017. |
| [24] | Wikipedia, „k-means clustering,” 06 Mai 2021. [Interactiv]. Available: https://en.wikipedia.org/wiki/K-means\_clustering. [Accesat 04 Iunie 2021]. |
| [25] | V. Rijmen și J. Daemen, „AES Proposal: Rijndael,” 1999. |
| [26] | H. B. Westlund, „NIST reports measurable success of Advanced Encryption Standard,” *Journal of Research of the National Institute of Standards and Technology,* 2002. |
| [27] | United States National Institute of Standards and Technology (NIST), „Announcing the ADVANCED ENCRYPTION STANDARD (AES),” Federal Information Processing Standards Publication 197 , 2001. |
| [28] | K. Sakiyama, Y. Sasaki și Y. Li, Security of Block Ciphers: From Algorithm Design to Hardware Implementation, Wiley Publishing, 2016. |
| [29] | R. Durstenfeld, „Algorithm 235: Random permutation,” *Communications of the ACM,* vol. 7, nr. 7, pp. 420-421, 1964. |
| [30] | D. E. Knuth, The Art of Computer Programming, Addison-Wesley, 1969. |
| [31] | A. Jörg, „Generating Random Permutations - Teză de Doctorat,” Australian National University, Australia, 2010. |
| [32] | S. Singh, The Code Book - The Science of Secrecy from Ancient Egypt to Quantum Cryptography, Statele Unite ale Americii: Anchor Books, 1999. |
| [33] | A. Cheddad, J. Condell, K. Curran și P. Kevitt, „Digital image steganography: Survey and analysis of current methods,” *Signal Processing,* vol. 90, nr. 3, pp. 727-752, 2010. |
| [34] | M. Singh, S. Kumar, S. Singh și M. Shrivastava, „Various Image Compression Techniques: Lossy and Lossless,” *International Journal of Computer Applications,* vol. 142, nr. 6, pp. 23-26, 2016. |
| [35] | Y. Blau și T. Michaeli, „Rethinking Lossy Compression: The Rate-Distortion-Perception Tradeoff,” în *Proceedings of the 36th International Conference on Machine Learning*, California, 2019. |
| [36] | A. Nasir, T. Natarajan și K. Rao, „Discrete Cosine Transform,” *IEEE Transactions on Computers,* Vol. %1 din %2C-23, nr. 1, pp. 90-93, 1974. |
| [37] | A. Haar, „Zur Theorie der orthogonalen Funktionensysteme,” *Mathematische Annalen,* vol. 69, nr. 3, pp. 331-371, 1910. |
| [38] | G. Strang, „Wavelets,” *American Scientist,* vol. 82, nr. 3, pp. 250-255, 1994. |
| [39] | J. Kodovsky și J. Fridrich, „Quantitative structural steganalysis of jsteg,” *Information Forensics and Security, IEEE,* vol. 5, nr. 4, pp. 681-693, 2010. |
| [40] | M. Kumar, „STEGANOGRAPHY AND STEGANALYSIS OF JOINT PICTURE EXPERT GROUP (JPEG) IMAGES - DISSERTATION,” The University Of Florida, 2011. |
| [41] | K. Rao și P. Yip, Discrete cosine transform: algorithms, advantages, applications, Statele Unite ale Americii: Academic Press Professional, Inc., 1990. |
| [42] | A. Khamrui și J. Mandal, „A Genetic Algorithm based Steganography using Discrete Cosine Transformation (GASDCT),” în *International Conference on Computational Intelligence: Modeling Techniques and Applications*, 2013. |
| [43] | UCSD Edu, „CSE 228 Week 3 Part 1 - JPEG,” [Interactiv]. Available: https://cseweb.ucsd.edu/classes/sp03/cse228/Lecture\_5.html. [Accesat 07 Iunie 2021]. |
| [44] | D. Upham, „JSteg source,” [Interactiv]. Available: https://zooid.org/~paul/crypto/jsteg/. [Accesat 07 Iunie 2021]. |
| [45] | Wikimedia, „File:CCD.png,” 04 Martie 2021. [Interactiv]. Available: https://commons.wikimedia.org/wiki/File:CCD.png. [Accesat 07 Iunie 2021]. |
| [46] | D. Fu, Y. Q. Shi, D. Zou și G. Xuan, „JPEG Steganalysis Using Empirical Transition Matrix in Block DCT Domain,” în *IEEE Workshop on Multimedia Signal Processing*, 2006. |
| [47] | D. L. Currie și C. E. Irvine, „Surmounting the Effects of Lossy Compression on Steganography,” în *19th National Information System Security Conference*, 1996. |
| [48] | A. Babcock, „Chroma Subsampling” 04 Martie 2019. [Interactiv]. Available: https://www.tings.com/tv/learn/chroma-subsampling. [Accesat 07 Iunie 2021]. |
| [49] | C. Poynton, Digital Video and HD - Second Edition, Statele Unite ale Americii: Elsevier INC., 2012. |
| [50] | S. Johansson și E. Lenngren, „Steganographic embedding and steganalysis evaluation - Bachelor’s Thesis,” [Interactiv]. Available: http://www.diva-portal.org/smash/get/diva2:770253/FULLTEXT01.pdf. [Accesat 07 Iunie 2021]. |
| [51] | M. Amiruzzaman, „STEGANOGRAPHIC COVERT COMMUNICATION CHANNELS AND THEIR DETECTION - Thesis,” [Interactiv]. Available: https://etd.ohiolink.edu/apexprod/rws\_etd/send\_file/send?accession=kent1310505218&disposition=inline. [Accesat 07 Iunie 2021]. |
| [52] | Python Software Foundation, „Python 3.9.5 Documentation,” 07 Iunie 2021. [Interactiv]. Available: https://docs.python.org/3/. [Accesat 07 Iunie 2021]. |
| [53] | Google Corporation, „Welcome To Colaboratory,” 2021. [Interactiv]. Available: https://colab.research.google.com/notebooks/intro.ipynb?utm\_source=scs-index#scrollTo=5fCEDCU\_qrC0. [Accesat 07 Iunie 2021]. |
| [54] | Pypi.org, „opencv-python 4.5.2.54,” 07 Iunie 2021. [Interactiv]. Available: https://pypi.org/project/opencv-python/. [Accesat 08 Iunie 2021]. |
| [55] | Pypi.org, „Pillow 8.2.0,” 01 Aprilie 2021. [Interactiv]. Available: https://pypi.org/project/Pillow/. [Accesat 08 Iunie 2021]. |
| [56] | Pypi.org, „pycrypto 2.6.1,” 17 Octombrie 2013. [Interactiv]. Available: https://pypi.org/project/pycrypto/. [Accesat 08 Iunie 2021]. |
| [57] | Pypi.org, „bitstring 3.1.7,” 05 Mai 2020. [Interactiv]. Available: https://pypi.org/project/bitstring/. [Accesat 08 Iunie 2021]. |
| [58] | scikit-learn.org, „sklearn.cluster.KMeans,” [Interactiv]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. [Accesat 08 Iunie 2021]. |
| [59] | 16 Ianuarie 1998. [Interactiv]. Available: https://johnloomis.org/ece563/notes/basics/depth/color\_depth.html. [Accesat 07 Iunie 2021]. |
| [60] | M. Kharrazi, H. Sencar și N. Memon, „Cover selection for steganographic embedding,” în *IEEE International Conference*, 2006. |
| [61] | K. Scicluna, „An Alternative LSB Based Video Steganography - Bachelor's Thesis,” MCAST Institute of Information and Communication Technology, 2018. |
| [62] | Z. Wang, A. Bovik, H. Sheikh și E. Simoncelli, „Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing,* vol. 13, nr. 4, pp. 600-612, 2004. |
| [63] | Wikipedia, 02 Iunie 2017. [Interactiv]. Available: https://en.wikipedia.org/wiki/File:VectorBitmapExample.svg. [Accesat 01 Iunie 2021]. |
| [64] | G. Rafferty, „towardsdatascience,” 18 Iunie 2018. [Interactiv]. Available: https://towardsdatascience.com/steganography-how-spies-rickroll-each-other-6a831d7df39e. [Accesat 02 Iunie 2021]. |
| [65] | Wikipedia, „Steganalysis,” 12 Mai 2020. [Interactiv]. Available: https://en.wikipedia.org/wiki/Steganalysis. [Accesat 05 Iunie 2021]. |
| [66] | E. Roberts, „The Discrete Cosine Transform (DCT),” [Interactiv]. Available: https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/jpeg/dct.htm. [Accesat 05 Iunie 2021]. |
| [67] | Wikimedia, „File:Dct-table.png,” 19 Septembrie 2020. [Interactiv]. Available: https://commons.wikimedia.org/wiki/File:Dct-table.png. [Accesat 07 Iunie 2021]. |

1. Watermarking-ul este un subiect larg, care are profunde legături cu domeniul steganografiei, însă nu reprezintă obiectivul lucrării de față, așa că a fost rezumat în câteva rânduri. O cercetare mai aprofundată a acestei ramuri este lăsată la latitudinea cititorului. [↑](#footnote-ref-1)
2. Numim obiect/mediu/fișier de acoperire orice tip de fișier multimedia care acționează drept *paravan* în steganografie și în care va fi integrat mesajul secret, rezultând astfel în obiectul steganografic. [↑](#footnote-ref-2)
3. Cele 4 canale sunt cyan, magenta, galben și negru. [↑](#footnote-ref-3)
4. În opoziție cu domeniul de frecvență este cel spațial. În cadrul celui din urmă, modificările sunt aplicate direct pe pixeli, așa cum se întâmplă în sistemele de substituție, spre exemplu. În schimb, domeniul de frecvență este caracterizat de un nivel de abstractizare ridicat, deoarece sunt procesate valorile intensităților pixelilor și a semnalului digital. Acesta constituie o arie de interes (cu precădere în procesarea imaginilor digitale) deoarece oferă posibilitatea de optimizare mult mai eficientă a unei imagini din punct de vedere al calității acesteia, per total, sau al spațiului de memorie ocupat. [8] [↑](#footnote-ref-4)
5. Este de remarcat faptul că această slăbiciune este prezentă doar în cazul în care dorim să ascundem un text într-o imagine. Aceeași regulă nu se aplică dacă am dori să codificăm o poză secreta în mediul de acoperire, caz în care orice modificare a imaginii de acoperire nu va corupe fișierul secret. [64] [↑](#footnote-ref-5)
6. DCT este cel mai folosit algoritm de compresie lossy și este utilizat pentru fotografiile de tip .JPEG. [↑](#footnote-ref-6)
7. Steganaliza este studiul detectării mesajelor ascunse folosind tehnici de steganografie și este analogul criptanalizei pentru criptografie. [65] [↑](#footnote-ref-7)
8. Este vorba de coeficienții AC. Detalii suplimentare în pagina următoare. [↑](#footnote-ref-8)