**UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI**

**FACULTATEA DE INFORMATICĂ**

****

LUCRARE DE LICENȚĂ

**Pixels editor**

**propusă de**

***Ana Maria-Daniela Condrea***

**Sesiunea:** *februarie, 2020*

**Coordonator științific**

Lect. Dr. Anca Ignat

**UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI**

**FACULTATEA DE INFORMATICĂ**

Pixels editor

*Ana Maria-Daniela Condrea*

**Sesiunea:** *februarie, 2020*

**Coordonator științific**

*Lect. Dr. Anca Ignat*

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Data \_\_\_\_\_\_\_\_\_\_\_\_ Semnătura \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DECLARAȚIE privind originalitatea conținutului lucrării de licență**

Subsemntatul(a) ………………………………………………………………………………………

domiciliul în …………………………………………………………………………………………………..

născut(ă) la data de ………………..…., identificat prin CNP ………….……………..………………..., absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de ………………………. specializarea …………………………………………………………, promoția …………………………., declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 si 5 referitoare la plagiat, că lucrarea de licență cu titlul: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_elaborată sub îndrumarea dl. / d-na \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice in vederea facilitării fasificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Dată azi, ………………………… Semnătură student …………………………

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Pixels editor*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de testetc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, *data*

Absolvent *Ana Maria-Daniela Condrea*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(semnătura în original)

# 1.Cuprins

[1.Cuprins 6](#_Toc31722514)

[2. Introducere 7](#_Toc31722515)

[3. Contribuții 11](#_Toc31722516)

[4. Implementare 12](#_Toc31722517)

[4.1. Crearea meniului 12](#_Toc31722518)

[4.1.1. Hide image 12](#_Toc31722519)

[4.1.2. Hide text 14](#_Toc31722520)

[4.1.2.1. Hide text invisible 14](#_Toc31722521)

[4.1.2.2. Hide text visible 15](#_Toc31722522)

[4.1.3. Hide audio 16](#_Toc31722523)

[4.1.3.1. Hide audio in image 16](#_Toc31722524)

[4.1.3.2. Hide image in audio 18](#_Toc31722525)

[4.1.3.3. Hide text in audio 18](#_Toc31722526)

[4.1.4. Hide video 19](#_Toc31722527)

[4.1.4.1. Hide image in video 19](#_Toc31722528)

[4.1.4.2. Hide text in video 20](#_Toc31722529)

[4.1.5. Watermark 20](#_Toc31722530)

[4.1.5.1. Add image logo to image 21](#_Toc31722531)

[4.1.5.2. Add text logo to image 22](#_Toc31722532)

[4.1.5.3. Add image logo to video 23](#_Toc31722533)

[4.1.5.4. Add text logo to video 24](#_Toc31722534)

[5. Testare 25](#_Toc31722535)

[6.Concluzii 28](#_Toc31722536)

[7. Bibliografie 29](#_Toc31722537)

# 2. Introducere

Întru-cât spațiul online nu este sigur și exista posibilitatea preluarii datelor care nu se doresc a fi descifrate s-au creat diferite modalitați de a ascunde un mesaj fără ca cei din jur sa observe acest lucru. Ascunderea unor informații prin criptografie presupune criptata acestora prin utilizarea unor funcții Hash și chei publice care transforma mesajul într-unul vizibila intregii lumi. Prin steganografie mesajul ascuns este plasat într-un alt obiect, doar expeditorul si receptorul stiu de existența sa.

Tema acestui proiect, acunderea de mesaje in imagini, audio sau video, a reieșit din dorinta de crea o aplicație care reușeste sa îmbine ascunderea de informatii sub diferite forme. De asemenea am ales acesta temă deoarece am vrut sa înteleg cum pot descompune obiectele in pixeli si cum pot crea noi obiecte pe baza unor algoritmi care modifică doar câțiva biți din obiectul inițial.

Consider că această aplicație ajută la o mai bună întelegere a conceputului de ascundere a unui mesaj. Reprezentarea grafică aduce un plus deoarece permite fiecarui utilizator sa testeze atât modul prin care se ascund mesaje cât si modul prin care se extrag mesajele.

Ideea mea pentru acest proiect de licență este dezvoltarea unei aplicații desktop ce se poate utiliza de către personele care doresc sa testeze modalitați diferite de a ascunde o informatie. Totodată ajută și la înțelegerea conceptului de steganografie întru-cât pe lângă algoritmii de baza aduce noi elemente care permit execuția acestora pe mai multe cazuri. Aplicația permite utilizatorului să încarce imagini, text, fișiere audio sau video alaturi de mesajul pe care dorește sa îl ascundă. După adaugarea tuturor parametrilor necesari se generează un nou obiect care va contine informația secretă. Receptorul fișierului rezultat anterior are posibilitatea de a prelua mesajul prin rularea comenzii de “unload”.

Steganografia(scriere ascunsă)  reprezintă arta și știința de a scrie mesaje ascunse de așa maniera încat, in afara de expeditor si receptor, nimeni să nu poata suspecta si nici descifra existența mesajului ascuns. Steganografia reprezinta un domeniu larg, cu multiple forme de implementare si aplicabilitați, insa in ce privește securitatea datelor, steganografia se incadreaza in categoria metodelor de *securizare prin obscuritate.*

**Steganografie si Criptografie**

Diferența majoră între steganografie și criptografie consta in faptul că steganografia protejează mesajul prin ascundrea lui față de ceilalți cititori, față de criptografie care afisează întrega informatie criptată, și care, oricât de idescifrabilă ar fi, atrage atenția și curiozitatea asupra ei.

Steganografia Criptografia

|  |  |
| --- | --- |
| * datele nu pot fi alterate, modificate   in timpul transferului. | * datele pot fi modificate |
| * suportă principiile de confidențialitatea   și autenticitate | * suportă atât principiile de confidențialitate si autenticitate, precum și integritatea datelor |
| * obiectivul principal este de a partaja un   mesaj secret | * principalul obiectiv este de a proteja datele partajate |
| * se poate implementa pe audio, video,   text sau imagini | * se poate implementa doar pe text |
| * mesajele sunt ascunse | * mesajele sunt codificate |

**Watermarking**

Watermarking-ul, sau marcarea transparentă, este operația de introducere a unei informații transparente, imperceptibile pentru sistemul vizual numita marcaj watermark in informatia gazdă, care poate fi text, semnal audio, imagine statica sau video. Steganografia reprezintă o formă de implementare a principiului de watermarking deoarece presupune ascunderea unor mesaje fară ca acestea să fie detectate.

Procesul de marcare transparentă pentru a fi cât mai corect și să realizeze protecția datelor trebuie sa îndeplinească două condiții:

1. introducerea marcajelor in datele gazdă înainte ca acestea sa fie partajate.
2. extragerea mesajelor din datele recepționate și compararea informației adaugate la emisie cu cea extrasă la recepție, pentru autentificare, în caz de atac asupra fișierului partajat.

O aplicație asemanatoare ar fi Image Steganography. Această aplicație este create in JavaScript și are ca și functionalitate ascunderea unei imagini într-o alta imagine cu un numar specificați de biți. De asemenea oferă și posibilitatea de a prelua mesajul ascuns din imaginea generate la pasul anterior. Totuși, există și câteva aspecte negative ale acestei apicații, Image Steganography nu permite utilizatorilor să apeleze aceste funcționalități pe mai multe domenii precum video, text sau audio și se rezumă doar la ascunderea de imagine in imagine. Acest lucru nu permite o vastă întelegere asupra conceptului de steganografie și limitează execuția programului doar la un singur element din cele patru medii de lucru pe care le permite această stiință.

O altă aplicație similară este Steganography(encode text into image). Aplicația are ca și funcționalitate ascunderea de text sau de imagini într-o alta imagine. Steganography este mult mai amplă față de aplicația anterioară dar prezintă anumite restricții:

* nu putem alege cați biți vrem sa modificăm
* nu permite utlizarea unor imagini mari, robuste
* nu poate fi folosit pentru ascunderea de informații in audio sau video
* nu prezintă modul clar de a utiliza aplicația

Diferența dintre proiectele existente in mediul online și proiectul meu, Pixels editor, îl reprezintă generalitatea acestuia pe toate cele patru medii de lucru ale steganorafiei. Pe langă fucționalitățile de ascundere si preluare de mesaje din imagini, audio sau video aplicația vine cu un plus, acesta fiind posibiliatetea de a adauga watermark in imagini sau video.

Modul de lucru al aplicației este simplu, întru-cât există meniu simplu cu submeniu clare care permite utilizatorului să acceseze funcționalitățile aplicației cu ușurință. Flow-ul aplicației este unul clar, ușor de înțeles și constă în preluarea unor date de intrare setate de utilizator si in funcție de necesitate acesta poate ascunde sau prelua un mesaj.

Aplicația relizează doua mari functionalități:

1. Ascunderea de informații: utilizatorul completează cu date valide câmpurile cerute după care urmează apăsarea butonului de Load.

algoritmi pentru ascunderea informației

1. Preluarea informației ascunse: utilizatorul completează câmpurile cu datele rezultate la pasul de încărcare după care apasă butonul de Unload.

algoritmi pentru preluarea mesajului ascuns

Atât pentru ascunderea de informații cât și pentru preluarea informației ascunse completarea câmpurilor este testată. Astfel fiecare componentă care trebuie completată este verificată in ideea corectitudinii contextului:

* căile introduse trebuie sa fie valide;
* fiecare componentă adaugată manual trebuie să fie setată pentru a fi vizibilă in program;
* in cazul prezenței unui radio button trebuie neaparat să alegem o valoare;

# 3. Contribuții

În cazul proiectului de licență, Pixels editor, contribuțiile studentului sunt unrmatoarele:

* documetarea studentului în legatură cu domneniul stenografiei (acsunderea de obiecte in alte obiecte pentru a putea fi transmise fară detecție);
* crearea unui plan și a unei strategii de realizare a proiectului ;
* crearea meniului principal al aplicației;
* divizarea meniului in submeniuri bine structurate;
* implementarea pseudo-codului necesar;
* crearea unor exemple clare pentru o mai bună intelegere a conceptului;
* căutarea și alegerea unei soluții optime;
* implementarea soluției;
* testarea soluției;

Pentru realizarea licenței am utilizat urmatoarele module:

* Python ca limbaj de programare cu interpretare de uz-general, interactiv, orientat pe obiecte și de nivel ȋnalt;
* OpenCV ca bibliotecă de prelucrare a imaginilor;
* Numpy pentru lucrul cu matrici de dimensiuni mari;
* Pillow pentru crearea de noi imagini;
* Tinkter pentru crearea meniului, este un modul standard pentru Python în care se poate realiza interfața grafică a aplicației;
* Soundfile pentru lucrul cu fișiere audio;

# 4. Implementare

## 4.1. Crearea meniului

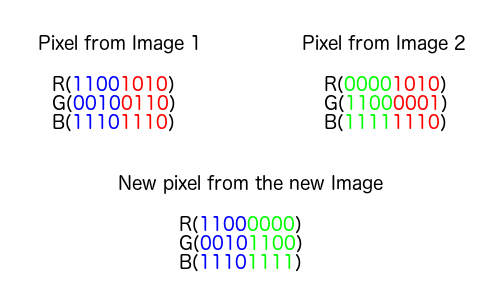
Meniul a fost creat cu ajutorul interfeței Tinkter din Python. Acesta este structurat in cinci mari module(Figure 1): ascunderea de imagini in imagini, ascunderea de text în imagini, ascunderea de imagini sau text în audio, ascunderea de imagini sau text in video si adăugarea de watermark in imagini si in video. Pagina principală constituie punctul de start in generearea sau preluarea obiectelor modificate în urma rulării comenzilor. Este realizat dintr-un frame care face legatura cu submeniurile fiecărui modul. *Figure 1-Meniul aplicației*

### 4.1.1. Hide image

Acest modul conține doua căi de acces una către imaginea care urmează a fi modificată, imaginea gazdă (cover image), primul input box din frame și una către imaginea care se dorește a fi ascunsă (image to hide), al doilea input box din frame.

Totodată acest submeniu oferă utilizatorului posibilitatea de a naviga în propriul computer și de a alege o imagine fară a scrie un path către aceasta, calea de acces setându-se automat în momnetul alegerii unei imagini. Pe lângă datele nesare pentru adaugarea unei imagini mai apare si un scale care lasă utilizatorului posibilitatea de a alege câți biți dintr-o imagine ar vrea sa transmit mai departe.

În partea de jos a submeniului apar trei butoane: Load, Unload și Back to Home. Primul realizează încărcarea imaginii care dorim a fi ascunsă în imaginea gazdă. O imagine este realizată la nivel de bază dint-o listă de pixeli care au o valoare cuprinsă intre 0 si 255. Pentru a putea realiza încarcarea datelor avem nevoie ca valoarea acestor pixeli să fie transformată în valoarea binară corespunzatoare. Acest proces este relativ simplu deoarece în imaginea coperta se modifica cei mai nesemnificativi n-biți cu cei mai semnificativi n-biți din imaginea care se doreste a fi transmisă fară a fi observată. Cazul de bază în acest proces îl reprezintă modificarea ultimilor 4 biți din imaginea copertă cu primii 4 biți din imaginea care trebuie ascunsă(Figure 2).



*Figure 2-Modificarea biților*

**

Al doilea buton este cel de Unload, este procesul invers, preluam imaginea care nu doream a fi detectată din imaginea rezultată la pasul de Load. Reconstruția imaginii transmise presupune extragerea celor mai semnificativi n-biți din imaginea modificată si adaugarea a 8-n biți pe care îi setam cu 0. Imaginea rezultată va suferi cateva modificari din cauza faptului cei 8-n biți setați cu 0 nu concid cu cei 8-n biți pe care îi avea inițial. Ca in cazul de încărcare a imaginii cazul de baza presupune modificarea a 4 biți(Figure 3).

Pixel from the new image Image reconstructed

**R(11000000) Unload R(11000000)**

**G(00111100) G(00110000)**

**B(11101111) B(11100000)**

*Figure 3-Reconstruirea imaginii*

** **

Al treilea buton este cel de Back to Home, acesta ne întoarce la pagina principala din meniu. Această operație se realizează prin apelul funcției ‘show\_page’ care face un salt (page.tkraise) la pagina la care am dori sa ajungem.

### 4.1.2. Hide text

Acest modul contine alte două submodule Hide text invisible și Hide text visble dar și un buton de Back to Home. Hide text se ocupa cu ascunderea unui text în pixelii imaginii cover care urmează a fi modificată prin apelul functiei de încărcare prezentă in fiecare submodule. In acest modul găsim totodată și funcționalitatea de extragere a textului din imaginea returnată anterior.

#### 4.1.2.1. Hide text invisible

Primul submodule de numește Hide text invisible și după cum sugerează titlul acestă componentă se ocupă cu ascunderea textului in imagine fara ca acesta sa poată fi detectat. Modulul de ascundere al textului invizibil contine un input box pentru adaugarea căii de acces a imaginii sau un buton de browse care permite alegerea unei imagini și automat completarea cu path-ul afferent acesteia. De asemenea contine un text box pentru inserarea textului care urmează a fi transmis mai departe prin intermediul imaginii, acest text vine insă cu o restricție, nu poate depași 256 de caractere. Principalele butoane din acest modul sunt Load, Unload dar și Back care ne întoarce la meniul anterior.

Butonul de Load executa modificarea imaginea gazdă astfel incât textul sa nu poate fi observant. Algoritmul se bazează pe modificarea valorii pixelilor din imagine în valoarea lor binară corespunzatoare. Totodată fiecare character din text este inlocuit cu codul sau ascii pentru ca ulterior acesta sa fie transformat tot într-o valoare binară pe 8 biți care va fi plasată într-o listă. Încărcarea în imagine se realizeză astfel: se parcurge imaginea pixel cu pixel si pe cel mai nesemnificativ bit din pixel se inlocuiește cu un bit din lista de biți(Figura 4).

* a = chr(97) = 01100001 în binar

Valoarea unui pixel Ultimii 3 biți din a Valoarea pixelului dupa Load

**R(11001100) 1 Load R(11001101)**

**G(00111010) 0 G(00111010)**

**B(11100110) 0 B(11100110)**

*Figure 4-Încarcarea textului in imagine*

Butonul de Unload parcurge imaginea nou creată pixel cu pixel și preia cel mai nedemnificativ bit. Lungimea textului care urmează a fi recreat se afla în primii trei pixeli pe ultimul bit. Dupa reconstruirea lungimii textului se va parcurge imaginea pană cand vom reuși să adunam toti biții necesari pentru a recrea fiecare caracter, ulterior textul pe care am dorit să îl ascundem.

#### 4.1.2.2. Hide text visible

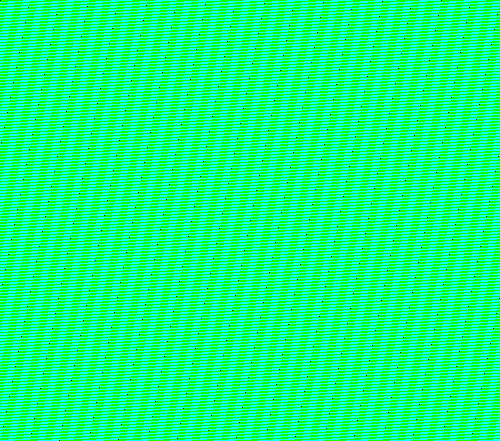
Al doilea submodule din modulul Hide text este Hide text visible care ascunde un text într-o imagine aleasă de utilizator. Diferenta majoră dintre acest submodule și submodulul anterior este ca apar cateva diferența minore între imaginea gazdă inițială și cea returnată după Load. Pentru apelul functiilor de Load si Unload este necesară setarea căii de acces către imagine și adaugarea textului care se dorește a fi ascuns. Cautarea imaginii se poate realiza și prin apasarea butonului de Browse a file care îi permit utilizatorului să aleaga o imagine din memoria caluclatorului. Pentru ascunderea textului se execută comanda de *load\_text\_random* prin apsarea butonului de Load a cărei funcșionalitate este:

* se parcurge textul dat și se transformă fiecare caracter in valoarea sa Ascii;
* se verifică daca lungimea textului este divizibilă cu 3 pentru parcurgerea imaginii
* se parcurge imaginea și se plasează la poziții random valorile Ascii ale caracterelor, poziții care sunt plasate într-un dictionar partajat de utilizatori.

Pentru extragerea textului se setează calea de acces către imaginea modificată anterior după care se apasă butonul de Unload. Acest proces constă în parcurgerea dicționarului de poziții și preluarea fiecarei valori din fiecare celulă de pixeli. După ce aceste valori au fost extrase se crează din nou textul prin transformarea fiecarei valori Ascii in caracter.

### 4.1.3. Hide audio

Hide audio se ocupă cu ascunderea unui fiseier audio într-o imagine, plasarea unei imagini sau a unui text într-un audio. Acest modul presupune crearea de noi fișiere wav care conțin informația care trebuie transferată. Este alcătuit din trei submodule:



Imaginea formată prin parcurgerea fișierului audio; dacă valoarea din audio este mai mică decât 0 atunci in imagine se plaseză ***RGB = 16385, abs(valoare) , lungimea reprezentării binare a valorii absolute***, în caz contrar se plasa in ***celula pixelului 0 pentru R iar pentru G și B similar cazului negativ***

*Figure 5-Creare imagine din audio*

#### 4.1.3.1. Hide audio in image

Primul submodule este cel de ascundere de audio într-o imagine nou creată. Acesta contine un Entry pentru adaugarea căii de acces catre fișierul audio sau un buton care permite alegerea unui fișier audio din cele existente și ulterior setarea path-ului. De asemenea mai conține două radioButton’s care ajută la crearea unei imagini gazdă unde va fi plasat fișierul wav. Pe langă aceste date mai apare și un Scale cu valori cuprinse între 1 și 14 care permit utilizatorului sa aleagă cați biți dorește sa ascundă. Ascunderea fișierului se realizeză prin apasarea butonului de Load după setarea tuturor variabilelor. Această operație presupune:

* crearea unei imagini gazdă care sa conțină valori ale pixelilor pe 16 biți (o imagine obisnuită are valori ale pixelilor pe 8 biți iar un fișier audio valori pe 16 biți, din această cauză s-a ales crearea unei noi imagini);
* crearea unei imagini din fișierul audio care presupune:
* extragerea valorilor din audio;
* setarea valorilor corespunzătoare pentru înaltime și lungime;
* crearea unei imagini cu fundal negru de înaltimea și lungimea setate precedent;
* parcurgerea acestei imagini si setarea fiecărui pixel cu valorile extrase din audio;
* dacă valoarea este negativă pe pixel-ul R(roșu) apare 16384 (valoarea maximă admisă pe 16 biți), pe pixel-ul G(verde) apare valoarea absolută, iar pe pixel-ul B(albastru) apare lungimea binarului creat din valoarea absolută;
* dacă valoarea este pozitivă pe pixel-ul R apare 0(valoarea minimă pe 16 biți), pe pixel-ul valoarea audio iar pe pixel-ul B apare lungimea binarului creat din valoarea audio;
* se plasează valorile din imaginea audio creată într-o listă pentu a fi transformate in valoarile binare corespunzătoare;
* în funcție de numarul de biți setați se modifică cei mai nesemnificativi biți din imaginea gazdă cu cei mai semnificativi biți din valorile audio;
* după plasarea tuturor valorilor audio în imagine se returnează noua imagine modificată;

Extragerea fișierului audio din imaginea create presupune setarea path-ului corespunzător de data aceasta către imagine nu către un audio apoi apsarea butonului de Unload care presupune parcurgerea imaginii gazdă doar între dimensiunile unde este plasat fișierul wav, preluarea ulimilor n biți si adugararea valorii 0 pană la valoarea indicate de pixel-ul G. Aceste valori binare sunt transformate în valorile int corespunzătoare, dacă prima valoare (a pixel-ului R) este 0 stim ca este pozitivă , în caz contrar este o valoare negative. Toate eceste valori extrase și modificate sunt plasate într-o lista care este dată ca parametru unei funcții pentru reconstrucția fișierului audio.

#### 4.1.3.2. Hide image in audio

Al doilea submodule este cel de ascundere a unei imagini într-un fișier audio. Pentru adugarea imaginii și a fișierului audio utilizatorul are două posibilităti, fie adaugă manual calea către acele fișiere după care trebuie sa apese si butonul de Set path pentru a le face vizibile în program, fie apasă butonul Browse a file care îi permite sa navigheze printre fișiere si alegerea unui obiect (setarea path-ului se face automat la legerea obiectului).

Cele doua funcționalităti prezente in acest submodule sunt cele de Load și Unload. Acunderea imaginii presupune preluarea fiecarei valori din celula unui pixel, crearea unei imagini din fișierul audio în aceeași manieră în care a fost creată în modulul *Hide audio in image.* Se parcurge imaginea audio creată și la poziții random se modifică valorile RGB cu valori din lista de valori din imagine. Din imaginea audio nou creată se extrag toate valorile și se plaseză într-o lista de valori pentru a reconstrui fișierul wav.

Pentru extragerea imaginii din audio se extrag pixelii de la pozițiile modificate anterior, se crează o imagine cu fundal negru de dimensiunile imaginii ascunse. Acestă imagine este parcursă și pentru fiecare celulă dintr-un pixel se setează o valoare din lista extrasa din audio.

#### 4.1.3.3. Hide text in audio

Al treilea submodul este cel de ascundere a unui text într-un fișier audio. Hide text in audio contine un Entry pentru setarea căii de acces către fișierul audio. Path-ul mai poate fi setat și prin apsarea butonlui de Browse a File care permite alegera unui fișier wav dorit. De asemenea conține și un câmp pentru adăugarea textului care se dorește a fi ascuns. Ascunderea și extragerea se realizeză prin apararea butoanelor de Load și Unload.

Butonul de Load presupune mai întai setarea căii de acces si adăugarea textului, după ce acestea au fost setate din fișierul audio se crează o imagine prin apelul funcției *make\_new\_image* din submodulul *Hide audio in image.* Ascunderea textului se realizeză prin apelul funcției *load\_text\_random* din submodulul *Hide text visible,* imaginea audio rezultată este parcursă pixel cu pixel iar fiecare valoare este plasată într-o lista de valori care vor reconstrui fișierul audio.

Butonul de Unload presupune setarea căii de acces către fișierul audio modificat anterior. Din acest fișier se extrag valorile textului de la pozițiile random pe care ambii utilizatori le dețin, valorile int extrase sunt de fapt valorile Ascii ale fiecărui caracter. După modificarea acestori valori în caracterul corespunzător se reconstruiește textul care se dorea a fi ascuns.

### 4.1.4. Hide video

Hide video se ocupă cu ascunderea unei imagini sau a unui text într-un videoclip și este alcatuit din alte două submodule Hide image in video și Hide text in video. Acest modul presupune crearea unui nou videoclip după ce se modifică un frame.

#### 4.1.4.1. Hide image in video

Hide image in video este primul submodule și se ocupă cu ascundrea unei imagini într-un fișier video. Acesta conține două input box-uri pentru adaugarea căilor de acces pentru video sau imagini, ulterior trebuie sa facem vizibil căile de acces în program prin apasarea butonului de Set Path. O metodă alternativă pentru setarea manuală a path-urilor este apăsarea butonului de Browse a file care permite alegerea unui fișier din computer. De asemenea mai există un scale care permite setarea numarului de biți care dorim a fi ascunși. Un videoclip este alcatuit din mai multe imagini care sunt derulate la un anumit timp. Ascunderea imaginii presupune setarea path-urilor și a numarului de biți, apoi apasarea butonului de Load care apelează funcția *final\_load().* Această funcție realizeză următoarele funcționalități:

* se calculează numarul de frame-uri din videoclip-ul dat ca parametru;
* se inițializeză valoarea parametrului *random\_frame* cu o valoare random între 0 și numarul de frame-uri;
* se extrage frame-ul de la poziția random setată anterior;
* se apelează funcția *load* din submodulul *Hide image in image* care are ca parametri frame-ul din video (este o imagine de la un anumit timp), imaginea setată și numarul de biți;
* rezultatul apelului este o nouă imagine care conține imaginea care se dorește a fi transmisă este plastă în video la poziția random;
* se afișeză videoclipul nou creat;

Extragerea imaginii din video presupune adaugarea parametrilor necesari pentru apelul funcției de *unload\_image\_from\_video().* Acestă funcție deschide videoclipul dat ca parametru (videoclipil creat la pasul anterior), extrage frame-ul de la poziția random setată la apelul Load apoi se apează funcția *unload\_show* din submodulul *Hide image in image.* Imaginea rezultată după acest apel este imaginea care s-a dorit a fi transmisă.

#### 4.1.4.2. Hide text in video

Hide text in video este al doilea submodule din modulul de ascundere a obiectelor in videoclip și realizeză ascunderea unui text într-un video. Acesta conține un Entry pentru adaugarea căii de acces către fișierul video, această cale mai poate fi adaugată prin apasarea butonului de Browse a file care permite alegerea unui fișier mp3 sau avi dintre cele existente in memorie. De asemenea mai apare un câmp pentru inserarea textului care se dorește a fi ascuns. Ascunderea textului se realizeză astefel: se setează parametrii necesari, se apasă butonul Load, se extrage un frame din videoclip de la o poziție random, se modifică acel frame prin apelul funcției *load\_text\_random* din submodulul *Hide image visible*. Imaginea modificată este plasată la poziția random in videoclip după care se salveză și afiseză noul videoclip.

Extragerea textului presupune setarea căii de acces către videoclipul modificat anterior, deschiderea acestuia și extagerea frame-ului în care se afla textul. Funcția Load returnează totodată și un dicționar care conține pozițiile din frame care au fost modificate. De la aceste poziții se preiau valorile pixelilor și se modifică valorile în caracterul corespunzător pentru reconstrucția textului.

### 4.1.5. Watermark

Modulul de Watermark este creat pentru adăugarea și eliminarea de logo în/din imagine sau video. Watermark conține alte 4 submodule: Add image logo to image, Add text logo to image, Add image logo to video și Add text logo în video.

#### 4.1.5.1. Add image logo to image

Add image logo to image este primul submodul și se ocupă cu adăugarea unui logo creat dintr-o imagine aleasă de utilizator la o imagine gazdă. Acest sumodul conține două input box-uri pentru adăugarea căilor de acces către imaginea cate urmează a fi modificată și către imaginea logo. De asemenea mai conține cinci RadioButton’s pentru setarea poziție logo-ului și un Scale pentru setarea parametrului alpha (gradul de vizibilitate a imaginii logo).

Adaugarea unui logo (Figure 6) se realizeză prin apsarea butonului de Load după setarea tuturor parametrilor. Acest buton apelează funcția *add\_watermark\_img()* care conține alte patru funcții:

* *create\_mask() ->* preia imaginea logo, crează o nouă imagine gri din aceasta apoi apelează funcția *threshold* din OpenCV care returnează o imagine în care partea alba reprezintă obiectul iar partea neagră reprezintă fundalul;
* *remove\_background() ->* preia imaginea logo și imaginea creată la pasul anterior, creată o noua imagine cu fundalul negru;
* *create\_invisible\_image ->* preia imaginea creată anterior și prin apelul funcțiilor *threshold, split, merge* din OpenCV se crează o nouă imagine fară background;
* *add\_logo() ->* se adaugă imaginea logo creată la pasul anterior peste imaginea gazdă prin apelul funcției OpenCV *addWeighted.*





*Figure 6 – Adaugare imagine logo*

Eliminarea unui logo (Figure 6) se realizează prin apăsarea butonului Unload după ce a fost adaugată calea de acces către imaginea modificată la pasul de Load. Mai întâi se crează o mască, o imagine cu fundal negru de aceeași dimensiune ca imaginea gazdă care la pozițiile setate la Load va avea masca creată la apelul funcției *create\_mask()*. Cu ajutorul funcției *inpaint* din OpenCV în locul unde se află imaginea logo se încearcă reconstrucția imaginii inițiale prin preluarea valorilor pixelilor din apropiere sau prin blurare.



*Figure 7- Eliminare imagine logo*

#### 4.1.5.2. Add text logo to image

Add text logo to image este al doilea submodul, acesta se ocupă cu adugarea unui text ca logo într-o imagine. Este alcătuit dintr-un input box pentru adăugarea path-ului către imaginea gazdă, un câmp pentru adăugarea textului, un câmp pentru setarea culorii textului, cinci RadioButton’s pentru setarea poziției, alte două RadioButton’s pentru a salva sau nu masca textului ( dacă salvăm masca se va crea o imagie cu fundal negru si cu textul logo alb, în caz contrat doar imaginea cu fundal negru de dimensiunile corespunzătoare) și un Scale pentru setarea nivelului de vizibilitate al textului, parametrul alpha.

Adăugarea textului logo presupune creare unei imagini, a unei măști, care va avea dimensiunile proporționale cu lungimea textului, va avea fundalul negru și prin apelul funcției OpenCV *putText* i se va adauga peste textul dat ca parametru cu culoarea specificată. Apoi se apelează funcțiile *create\_mask, remove\_background, create\_invisible\_image* din submodulul *Add image logo to image* pentru crearea măștii finale care prin apelul funcției *add\_logo\_text* va fi adaugată peste imaginea gazdă la poziția dată și cu nivelul de vizibiliatate setat.

Eliminarea textului logo presupune preluarea măștii creată prin apelul funcției *create\_txt\_image,* parcurgerea acesteia și extragerea fiecarei valori din celulele pixelilor într-o listă de valori apoi crearea unei imagini cu fundal negru de dimensiunile imaginii gazdă și plasarea la pozițiile setate la Load a valorilor din lista creată anterior. După ce această imagine a fost creată se apeleză funcția *inpaint* care va incerca eliminara logo-ului. Dacă la pasul de load am ales sa salvăm masca extragerea textului va fi mai puțin vizibilă întru-cât în acest caz se încearcă doar reconstrucția pixelilor unde se afla textul. Dacă am ales să nu salvam textul reconstrucția pixelilor se va face pe întrega poțiune unde se afla imaginea creată din textul dat.



**salvare mască**



**fără salvare de mască**



#### 4.1.5.3. Add image logo to video

Add image logo to video este al treilea submodul, acesta se ocupă cu adaugarea unei imagini logo la un videoclip. Submodulul conține două input box-uri, pentru adăugarea videoclipului și pentru adăugarea imaginii, Radio Button’s pentru selectarea poziției unde va fi plasat logo-ul si un Scale pentru setarea nivelului de vizibilitate a imaginii logo.

Inserarea imaginii logo presupune setarea tuturor paramereilor, apasarea butonului de Load care apeleză funcția *final\_load.* Aceasta funcție este împartită în doi pași importanți:

* apelul funcției *compute\_logo\_image* presupune apelul metodelor din submodulul *Add image logo to image* pentru a crea viitoarea imagine logo fară background;
* apelul procedurii *add\_logo\_video\_image* presupune parcurgerea videoclipului frame cu frame și crearea unei măști care va fi suprapusă peste fiecare frame.

Extragerea imaginii logo necesită setarea căii de acces către videoclipul creat la pasul de Load apoi apăsarea butonului de Unload. Procedura apelată la extragerea logo-ului este *remove\_logo\_video\_image* care cât timp videoclipul este deschis crează masca pentru extragerea logo-ului prin apelul metodei *compute\_mask* din submodulul *Add image logo to image,* apoi se incearca eliminarea logo-ului de la fiecare frame.

#### 4.1.5.4. Add text logo to video

Add text logo image este ultimul submodul și se ocupă cu adaugarea unui text logo la un videoclip. Acest submodule conține un input box pentru adăugarea căii de acces către videoclip, un câmp pentru adăugarea textului, un câmp pentru setarea culorii textului, Radio Button’s pentru setarea poziției și pentru salvarea măștii care va ajuta la extragerea logo-ului și un scale pentru setarea parametrului alpha.

Aăugarea textului logo în videoclip se relizeză similar cu adaugarea textului logo în imagine, singura diferentă este ca logo-ul se va adăuga la fiecare frame al videoclipului, un frame fiind o imagine care apare la un anumit timp din videoclip.

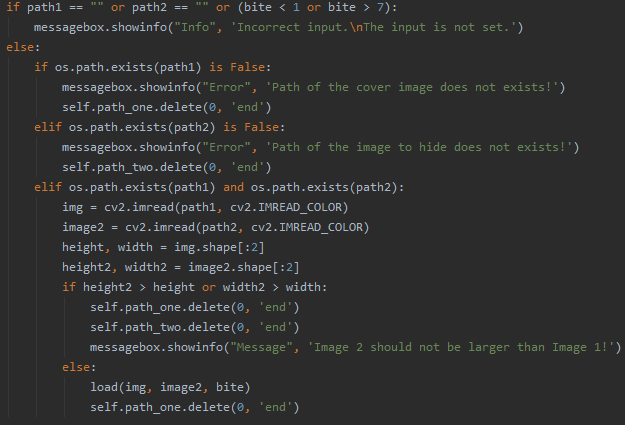
Extragerea textului logo din videoclip presupune adăugarea căii de acces către videoclipul creat anterior, parcurgerea videoclipului frame cu frame, crearea unei măști pentru extragerea logo-ului apoi apelarea funcțiilor OpenCV necesare. În funcție de alegerea facută la pasul de adăugare a logo-ului extragerea poate fi mai mult sau mai puțin vizibilă întru-cât dacă am ales să salvăm masca se vor modifica doar pixelii unde se afla scrisul, în caz contrat se va modifica întreaga regiune în care se afla imaginea creată din text.

# 5. Testare

Testarea presupune verificarea precondițiilor pentru Load și Unload pentru ca programul să ruleze corespunzător și sa furnizeze răspunsul corect.

Pentru primul modul în cazul apelului Load s-au folosit verificările următoare (cazul Unload conține doar verificarile de existentă și de setare al path-urilor):

se verifică dacă path-urile și numărul de biți sunt setate



se verifică existența fișierelor adăugate

se verifică dimensiunea imaginii de copertă în comparație cu imaginea ce urmează a fi adaugată peste aceasta

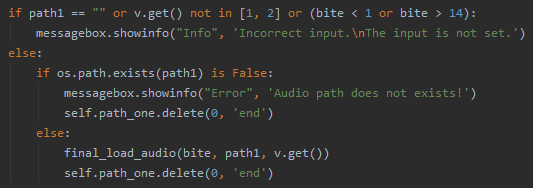
apel către funcția load cu parametri corespunzători

Modulul de ascundere a unui text în imagine presupune pe langă verificarea existenței și setării imaginii și următoarea verificare (cazul Unload presupune doar verificarea existenței și setării path-ului imaginii modificate):



se verifică dacă a fost adăugat text

Modulul Hide audio presupune pe langă setarea și verificarea căilor de acces către imagine/audio sau adăugarea textului urmatoarele verificări în cazul Hide audio in image:



se verifică numarul de biți

se verifică dacă a fost setat radio button-ul

Modulul Hide video presupune verificări similare cu cele din modulele Hide image și Hide text. Singura diferență este că în acest submodule se adaugă o cale de acces către un videoclip iar în cazul ascunderii de imagine se va verifica dimensiunea imaginii în comparație cu dimensiunea unui frame din videoclip.

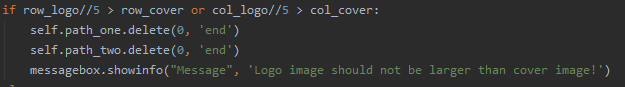
Modulul de Watermark presupune pe langă verificările existenței și setării path-urilor către imagine/ video și adăugarea textului următoarele examinări:

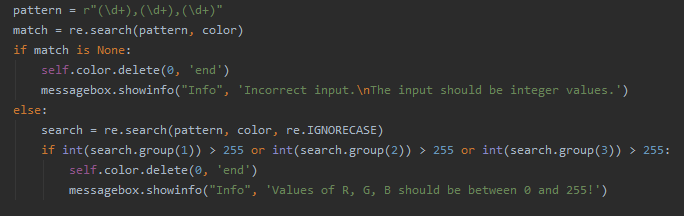
se verifică dacă parametrul alpha este corect

se verifică dacă a fost setată poziția de inserare a logo-ului



se verifică dacă imaginea logo este mai mică decât imaginea copertă





dacă se respectă pattern-ul dar numerele inserate sunt mai mari decât 255 atunci se cere reinserarea acestora

dacă nu a fost găsită o potrivire afisează un mesaj

se crează un pattern de adăugare a culorii textului logo apoi se verifică dacă pattern-ul a fost respectat

# 6.Concluzii

În concluzie sunt de părere că Pixels editor este o aplicație excelentă în ceea ce priveste ascunderea de imagini, text, audio în diferite medii vizuale. Posibilitatea de a avea o gamă largă din care să poți alege ce vrei sa ascunzi și unde să ascunzi este un atuu față de alte aplicații similare care nu oferă decât o funcționalitate pe rând.

Procesarea de imagini este importantă pentru ascunderea unui mesaj într-un alt obiect făra ca alți utilizatori sa observe existența acestuia, doar expeditorul si receptorul stiu de existența sa. Spre deosebire de steganografie ascunderea de imagini prin criptografie este mult mai vizibilă pentru celelalte persoane și automat mai predispusă să fie piratată.

Spre deosebire de alte aplicații asemănătoare, Pixels editor, oferă un nivel mult mai mare de libertate. În special, modul în care utilizatorul poate alege ce fisier să ascundă, câți pixeli să se modifice, ce tip de audio să adauge, la ce pozițe să adauge un logo, dacă să salveze sau nu masca unui logo, ce culoare să aibă logo-ul și nivelul de vizibilitate a logo-ului.

Ca îmbunătățiri și direcții viitoare, m-am gândit la crearea unui algoritm mai eficient de ascundere a unei imagini in imagini, să verific câți pixeli au fost modificați în funcție de modul de salvare a unei imagini png/jpg sau a unui video mp4/avi. De asemenea o altă îmbunătățire ar fi plasarea aplicației în mediul online, salvarea fișierelor într-o bază de date și citirea textului dintr-un fișier dat de către utilizator.

# 7. Bibliografie

* *“Digital Watermarking and Steganography”* by Ingemar J. Cox, Matthew L. Bloom;
* “*Information Hiding Techniques for Steganography and Digital Watermarking*” by Stefan Katzenbeisser and Fabien A. P. Petitcolas;
* “*Disappearing Cryptography*” by PeterWayner
* <https://towardsdatascience.com/steganography-hiding-an-image-inside-another-77ca66b2acb1>
* <https://null-byte.wonderhowto.com/how-to/steganography-hide-secret-data-inside-image-audio-file-seconds-0180936/>
* <https://stackoverflow.com/>
* <https://docs.python.org/3/library/tkinter.html>
* <https://pysoundfile.readthedocs.io/en/latest/>
* <https://pywavelets.readthedocs.io/en/latest/ref/2d-decompositions-overview.html?fbclid=IwAR3vansijHCq2Erz5uUsnmi9_2bovgq7NFXBOT2DeqFh0eML9RZp8rSlnlE>
* <https://www.learnopencv.com/read-write-and-display-a-video-using-opencv-cpp-python/?fbclid=IwAR3YaYxcRobGLSgV8HrKMCL_0U-lNU4_tF_3pOhJIzktv1Nq8_1e89Qgcxw>
* <https://searchsecurity.techtarget.com/definition/steganography?fbclid=IwAR1YRMAzfxY8ZtJSMrplA1aWrl-dHMoBznjnzmJhfEE4dlrO05J1Giab-VA>
* <https://github.com/intel-iot-devkit/Video-Analytics-OpenCV/tree/master/tutorials/opencv/Python/sample_09_DOG_video?fbclid=IwAR3vansijHCq2Erz5uUsnmi9_2bovgq7NFXBOT2DeqFh0eML9RZp8rSlnlE>
* <https://docs.opencv.org/master/d0/d86/tutorial_py_image_arithmetics.html?fbclid=IwAR0QPOY2TI0qZ7I8Ojg45DfSBPp8BLpU5hgd588Uq4m-XkfIf1dNdSMa-Rg>