**Documentatie**

Structuri

* pixel contine cele trei canale de culori R G B
* imagine contine un vector de pixeli si dimensiunea imaginii
* detectie contine scorul, coordonatele coltului stanga sus a ferestrei, dimensiunea ferestrei si culoarea cu care trebuie
* vect\_detectii contine un vector de detectii si lungimea sirului

Headere

Pentru criptare/decriptare:

criptare.h

* criptare care liniarizeaza imaginea primita in fisierul imagine\_init, modifica vectorul de pixeli obtinut si rescrie imaginea modificata (criptata) in fisierul imagine\_fin, iar in fisierul cheie returneaza seed-ul si sv-ul.
* rescrie construieste in memoria externa o imagine primita ca parametru in forma liniarizata. Header-ul se copiaza din fisierul inital fisier\_init transmis ca parametru.
* liniarizare primeste ca parametru un fisier .bmp si transforma imaginea intr-un vector de pixeli. Returneaza adresa la care a fost memorat.
* XORSHIFT32 genereaza un numar aleator in functie de seed-ul transmis ca parametru.
* permutare creeaza o permutare de n elemente cu ajutorul sirului pseudo-aleator r.
* permutare\_imagine aplica o permutare asupra pixelilor unei imagini.
* criptare\_cu\_XOR primeste o imagine cu pixelii permutati si aplica criptarea cu XOR. Este folosita a doua jumatate a sirului pseudo-aleator r.
* generare\_sir\_nr\_pseudo\_aleatoare creeaza un sir pseudo-aleator de lungime n folosind algoritmul XORSHIFT32 pornind de la valoarea initiala seed.

decriptare.h

* permutare\_inversa generaza permutarea initiala si o cu ajutorul ei permuta invers imaginea.
* decriptare\_cu\_XOR primeste o imagine criptata si aplica decriptarea cu XOR. Este folosita a doua jumatate a sirului pseudo-aleator r.
* decriptare decripteaza o imagine primita ca parametru folosind o cheie secreta. Apeleaza functiile permutare\_inversa si decriptare\_cu\_XOR.

test\_chi\_patrat.h

* frecvente\_pe\_canale creeaza trei vectori de frecventa pentru fiecare canal de culoare.
* test\_chi\_patrat\_pe\_un\_canal calculeaza valoarea χ2 pentru un canal de culoare.
* test\_chi\_patrat afiseaza valoarea testului chi patrat pentru toate canalele de culoare.

Pentru Pattern-matching

grayscale.h

* grayscale\_image transforma o imagine color intr-una gri.

cross\_correlation.h

* calculare\_scor calculeaza intensitatea\_medie si deviatie\_standard pentru un sablon si o fereastra date si le foloseste in determinarea scorului de corelatie.
* template\_matching construieste un vector de detectii pentru un sablon si o fereastra date.

colorare.h

* colorez foloseste apelurile utile ale functiei color\_fereastra pentru a colora toate detectiile din vector
* aleg\_culoare decide culoarea corespunzatoare fiecarui sablon

eliminare\_non-maxime.h

* elimin\_non\_maxime parcurge vectorul sortat de detectii si elimina detectiile non maxime marcandu-le cu un scor negativ, mai mic ca -1

identificare\_patternuri.h

* identificare\_patternuri ruleaza functia template\_matching pentru fiecare sablon, reuneste detectiile intr-un vector final, il sorteaza, elimina non-maximele si apeleaza colorez.