

Operatori Laplace in detectia muchiilor/conturului

**Membrii echipei**

**Dezvoltator**: Iryna Kinzerska*(e-mail: iryna.kinzerskaa@gmail.com)*

**Manager**: Vanca Rafael Marian *(e-mail: vanca.rafael.marian@gmail.com)*

**Nume îndrumător**: Camelia Costina Florea

**Data predării proiectului (zi/lună/an)**:

Sinteza Lucrarii

Tema proiectului nostru o reprezinta Operatori Laplace in detectia muchiilor/conturului (generarea automata a mastii de convolutie de diverse dimensiuni). Operatorul Laplace este un operator diferential de ordinal n al doilea in spatiul euclian n-dimensional. Acestea au o viteza mare de procesare astfel tema aleasa este optima pentru folosirea practica.

Programul folosit pentru implementarea proiectului nostru este google colab, ne am folosit de documentele gasite in mediul online pentru a incepe implementarea.

Datele de intrare sunt reprezentate de o imagine pe nivele de gri, dimensiunea operatorului folosit fiind de (3x3,5x5,7x7 etc) respectiv parametrii generare automata operator.

Datele de iesire/afisare rezultat reprezinta imafinea originala + histograma, imaginea gradient, imaginea continad muchiile(binar).

Ca si rezultate ale proiectului nostru avem prezentarea imaginii dupa aplicarea tuturol pasilor descrisi mai jos.

Introducere

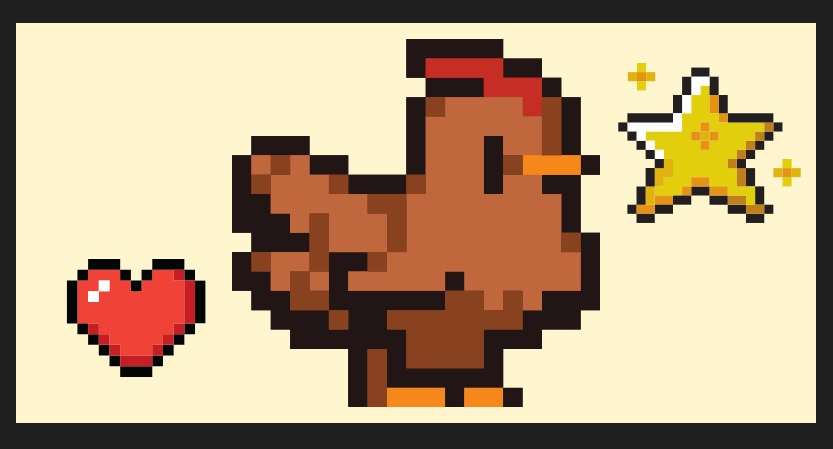
Detectarea trecerii prin zero a derivatei de ordinul doi permite detectarea precisa a muchiei indiferent de faptul cat de brusca este schimbarea luminozitatii. Astfel medile care utilizeaza procesare de imagine, computer vision, machine vision s.a.m.d vor folosi acest algoritm de detective. Prin algoritmul implementat am reusit sa parcurgem pasii propusi, astfel am citit imaginea, o convertim in Gray, bluram imaginea cu gaussian pentru a scapa de zgomot, aplicam Laplacean si in final afisam.

Edge detection este folosit pentru segmentarea imaginilor si extragerea de date folosite in zone ca de exemplu procesare de imagine, computer vision sau machine vision.

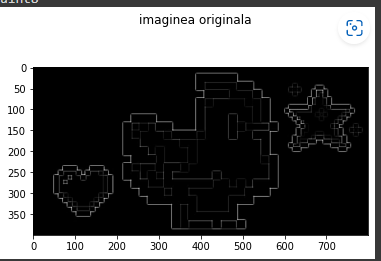
Noi am implementat algoritmul de detectia a muchiilor folosind Operatorii Laplacean, utilizand in prima parte libarii gata facute pentru fiecare pas, deschis mai jos in document.

Apoi am inceput prin implementarea algoritmul prin parti. Astfel am urmarit sa vizualizam imaginea dupa aplicarea fiecarui pas.

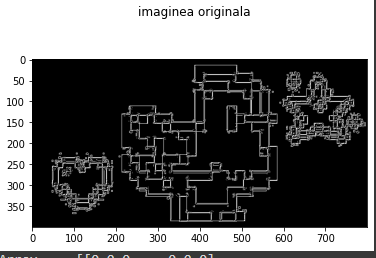
Imaginea originala



Imaginea obtinuta dupa aplicarea convolutiei:



Imaginea obtinuta dupa determinarea zero\_crossing:



Teoretic

Zero-Crossing

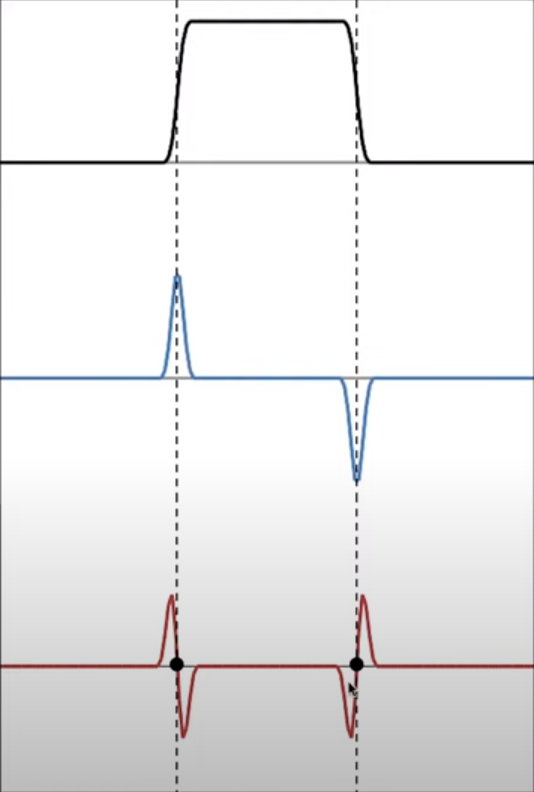
Aplicam Laplacian

Aplicam Gaussian

Convertim in Grey

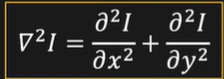
Citim imaginea

In primul rand avem nevoie de prima derivata respectiv de derivata a doua care ne va da trecerile prin zero ce indica edge urile.

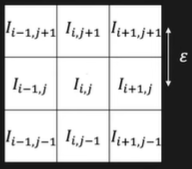


Un fact foarte interesant despre derivata de ordinul doi il reprezinta faptul ca la crossing nu avem peaks, defapt avem zero, treceri prin zero foarte puternice. Apoi daca putem detecta trecerile prin zero putem detecta si edge urile.

Apoi trecem la operatorul Laplacean care reprezinta suma derivatelor de ordinul doi, derivata de oridinul doi al imaginii in functie de x si derivata de ordinul doi in functie de y.

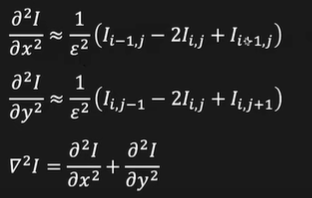


Pentru implementare avem nevoie de asa zisele masti de convolutie. Acestea sunt de cel putin 3 pixeli astfel vom folosi un operator de cel putin 3x3.

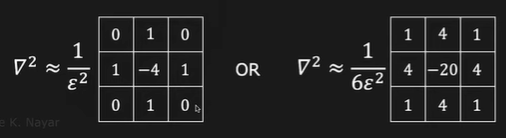


Sa zicem ca vrem sa aflam Laplace pentru centrul imaginii noastre. Ce vom face va fi sa luam diferenta dintre pixelul din extrema dreapta minus pixel din centru, vom diviza vu epsilon minus diferenta dintre pixelul din centru si pixelul din stanga apoi din nou toata procedura divizata de epsilon si se ajunge la calculele din urmatoarea imagine. (Acelas lucru si pentru componenta pe y).

Derivata de ordinul 2 a imaginii pentru x si pentru y:

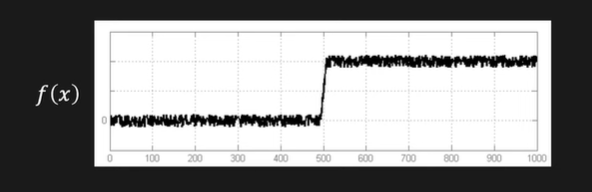


Exemple de masti de convolutie:



Avem totusi o problema, am determinat derivata secunda pentru x si pentru y dar totusi edge urile pot sa fie si pe colturi, de exemplu la 45%. Astfel se poate utiliza un filtru de mai sus.

Efectul zgomotului:



Din cauza lui edge urile apar ca nu find perfecte, astfel pentru a scapa de acest zgomot vom folosi Gaussian

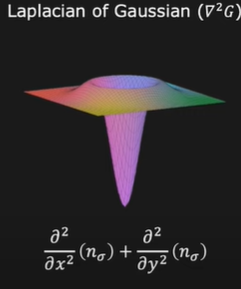
Derivata de ordinul 1 aplicata unei functii f si a unui gaussian.



Putem aplica un operator laplacian unui gaussian si apoi imagini.



Aplicarea gradientului:



Implementarea solutiei adoptate:

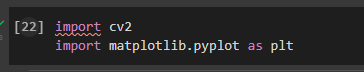
In primul pas noi am implementat acest algorit utilizand librarii gata facute, astefel folosind platforma google colab pentru versionarea priectului nostru am folosit apoi librariile cv2 respectiv matplotlib.pyplot as plt

Librarii utilizate in proiectarea cu librarii gata facute:

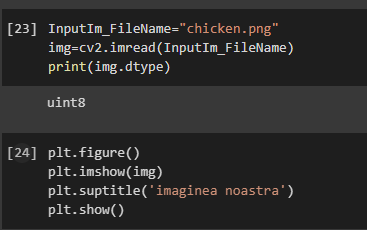


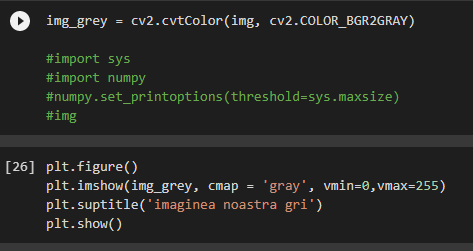
In cea dea a doua parte am inceput implementarea proiectului nostru fara a folosi aceste librarii gata facute.

Astfel in prima parte am adaugat librarile cv2, , matplotlib.pylot as plt.

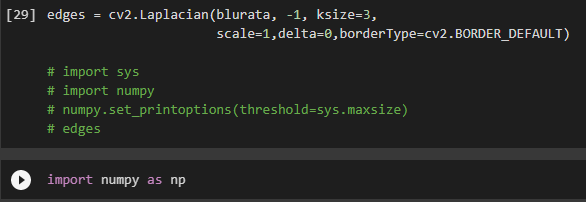


In urmatoarea parte am urmat schita prezentata la partea teoretica, si anume am inceput prin a adauga imaginea noastra si a o transforma in grey.

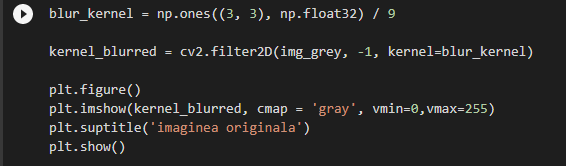




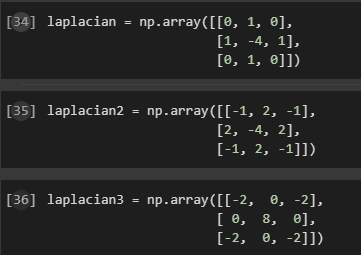
Am aplicat Laplaciand de ordinul 2

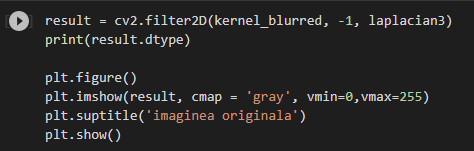


Asa cum am prezentat in partea teoretica, avem nevoie de aplicarea Gaussian ului pentru eliminarea zgomotului din functia noastra. Astfel pentru aplicarea blurului cu gaussian am avut nevoie de a genera kernel si apoi sa aplicam convolutia impreuna cu acel kernel.

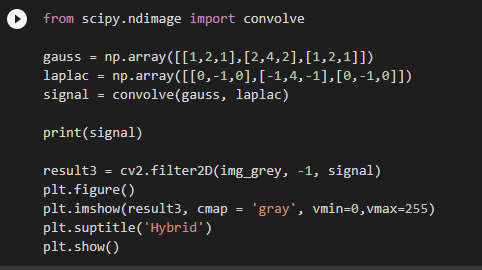


In urmatorul pas am trecut la creearea Laplacian ului nostru, si anume la generarea kernel ului si aplicarea acestuia (convolutia). Tot in partea teoretica am definit mastile pe care le folosim.

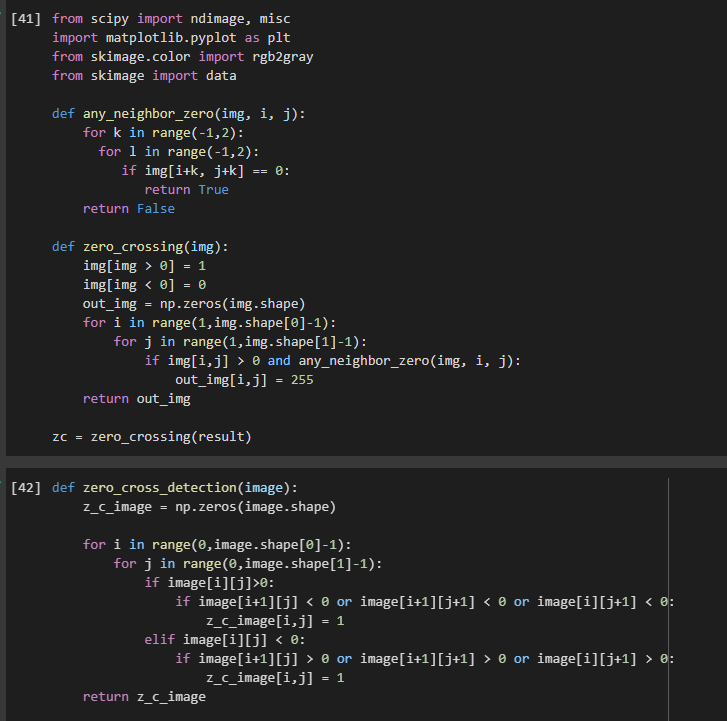




Sa facut implementarea si impreuna cu o masca hibrid.



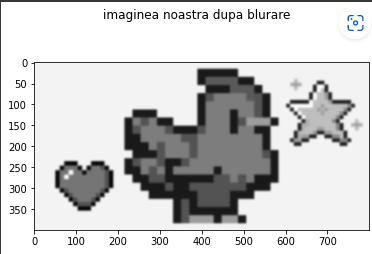
In ultima parte am cautat zero-crossings, acesta fiind ultimul pas din proiectul nostru.



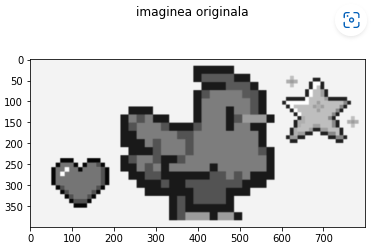
Rezultate experimentale:

In urma aplicarii codului prezentat in rubrica anterioara, am ajuns la urmatoarele rezultate.

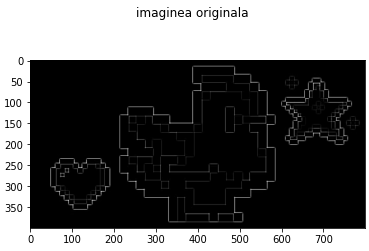
Imaginea blurata:



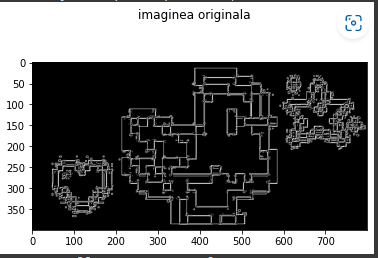
Imaginea dupa aplicarea gaussian:



Imaginea dupa aplicara Laplacian, masca:



Zero-Crossing:

1. 

Concluzii:

Pe scurt tema noastra a fost implementata cu succes, aceasta poate sa fie imbunatatita prin aprofundare si munca suplimentara. Ca si grad de dificultate aceasta putem spune ca este de nivel mediu, dar fiind o echipa buna am reusit impreuna sa ducem la bun sfarsit, sa lucram si sa ne corelam munca intr un mediu armonios.

Ca si rezultate finale, aceastea mai trebuie lucrate, imaginea finala se afiseaza la un nivel prea inchis.

Bibliografie:

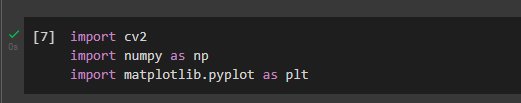
[(153) Edge Detection Using Laplacian | Edge Detection - YouTube](https://www.youtube.com/watch?v=uNP6ZwQ3r6A&t=319s)

[Laplace operators in differential geometry - Wikipedia](https://en.wikipedia.org/wiki/Laplace_operators_in_differential_geometry)

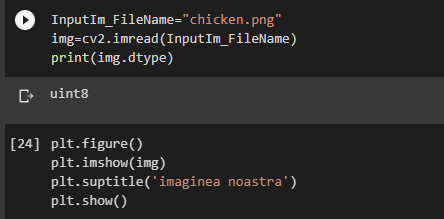
https://utm.md/meridian/2010/MI\_2\_2010/4\_Calmicov\_I\_Analiza\_algoritmilor.pdf

Anexe:

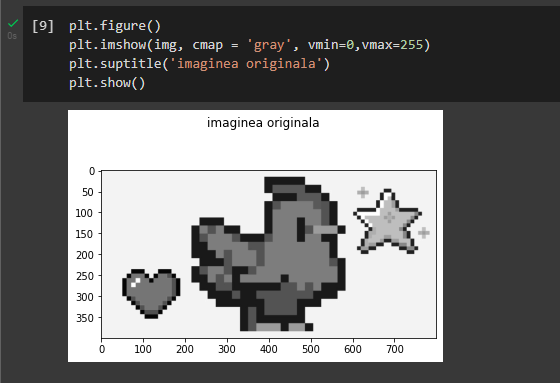
Librariile importate:

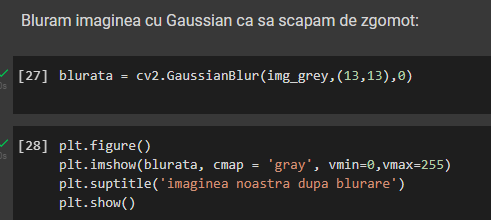


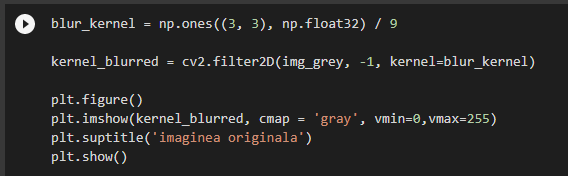
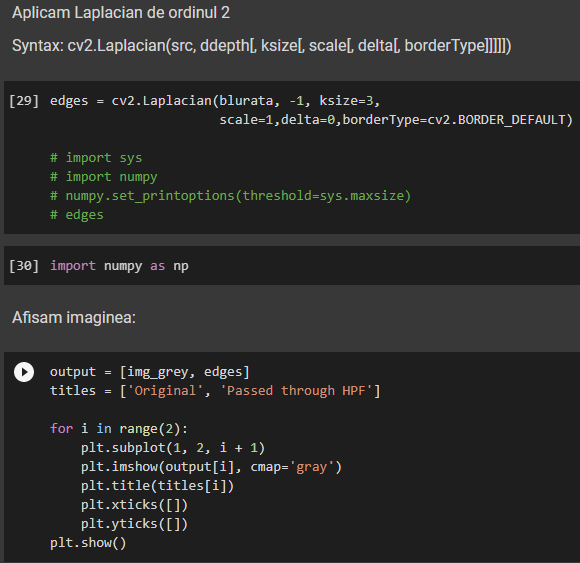
Imcarcam Imaginea:



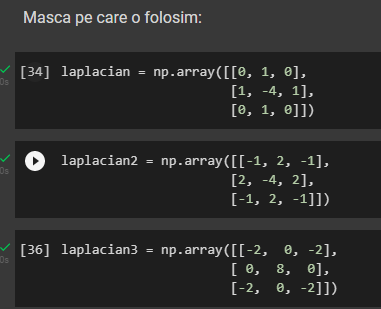
Imaginea afisata in format gry:



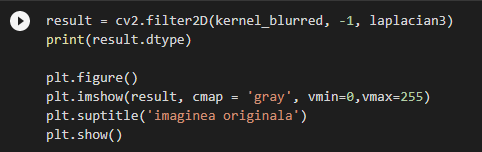


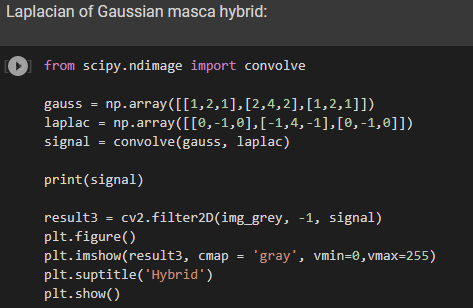


Masca pe care o folosim:



Convolutia:





Imaginea finala:

