

Tema 1 – Redimensionarea imaginilor cu pastrarea continutului

(a) Rezultatele obtinute la punctul (1.1) – micșorarea imaginii pe latime – cu 50 pixeli pentru imaginea 'castel.jpg':



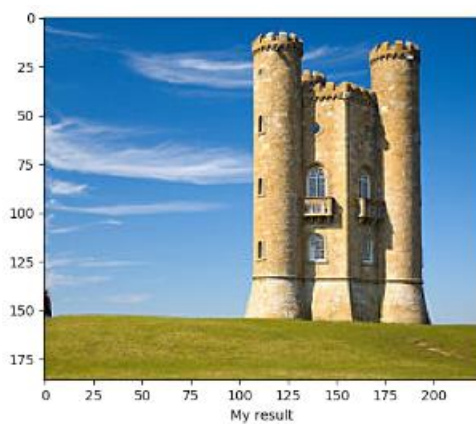
Poza originala



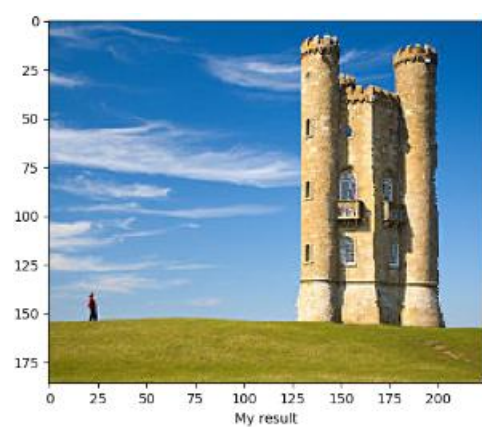
Resize din OpenCV



Programare Dinamica

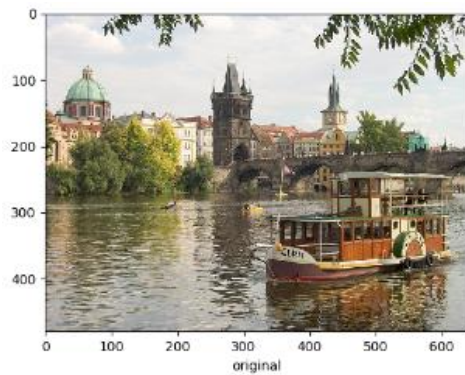


Greedy

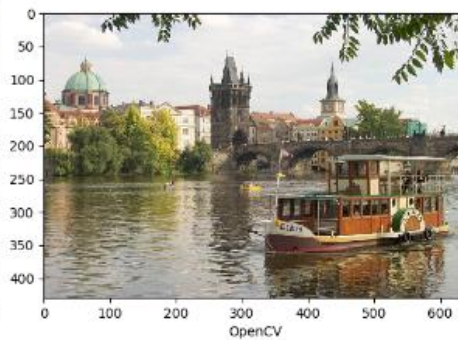


Aleator

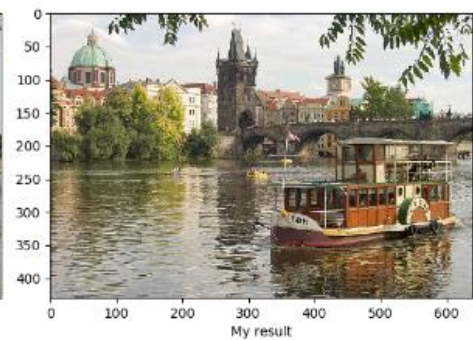
(b) Rezultatele obtinute la punctul (1.2) – micsorarea imaginii pe inaltime – cu 50 pixeli pentru imaginea 'praga.jpg':



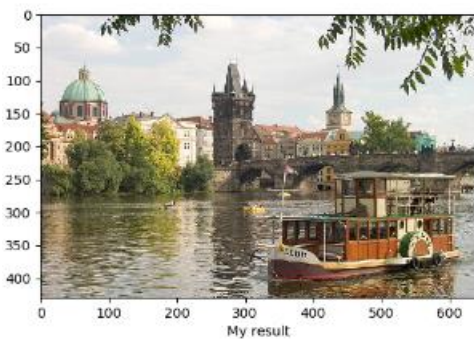
Poza originala



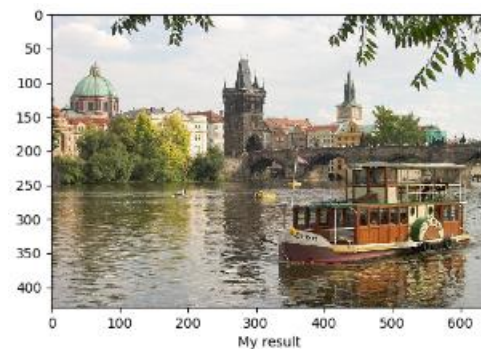
Resize din OpenCV



Programare Dinamica

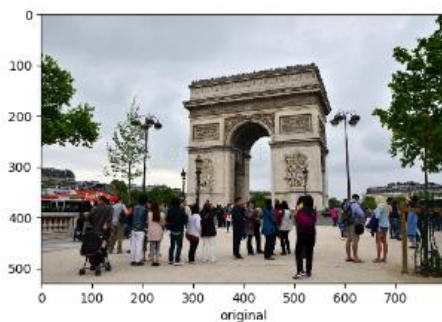


Greedy

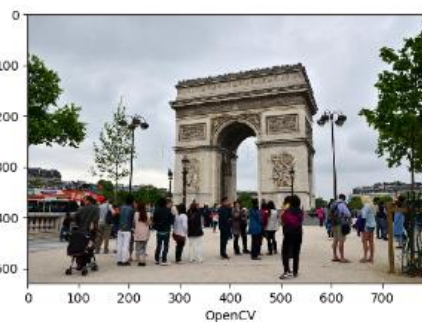


Aleator

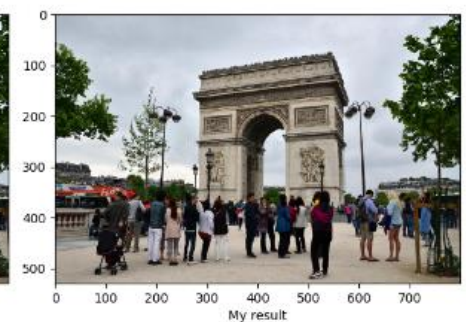
(c) Rezultatele obtinute la punctul (1.3) – amplificarea continutului imaginilor – (105%) pentru imaginea 'arcTriumf.jpg':



Poza originala

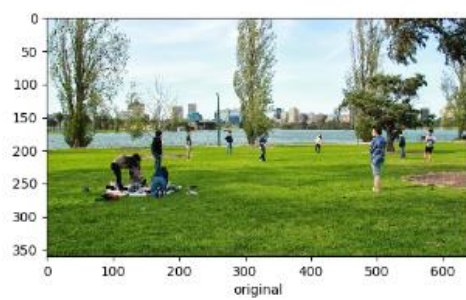


Resize din OpenCV

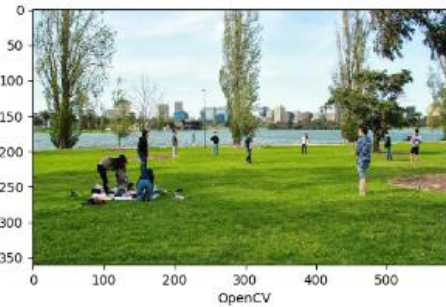


Amplificare continut

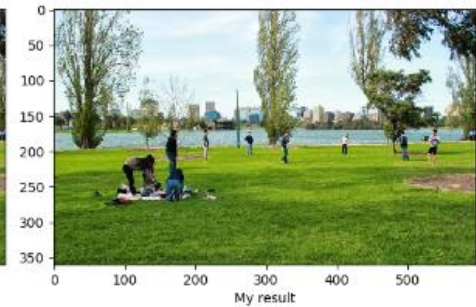
(d) Rezultatele obtinute la punctul (1.4) – eliminarea unui obiect din imagine – pentru imaginea 'lac.jpg':



Poza originala



Resize OpenCV



Eliminare obiect



Obiectul selectat pentru eliminare

(e) 1. Rezultatele obtinute dupa eliminarea a 100 pixeli pe latime dintr-o poza cu un caine:

Observam ca varianta cu programare dinamica merge bine, deoarece in stanga si in dreapta cainelui se afla in mare parte iarba blurata (diferente mici de culoare intre pixeli vecini) deci drumurile eliminate au putut fi alese fara a "taia" din caine. Varianta greedy arata si ea ok deoarece selecteaza mereu drumuri din partea stanga a imaginii (iarba din background).



Poza originala



Resize OpenCV



Programare Dinamica



Greedy



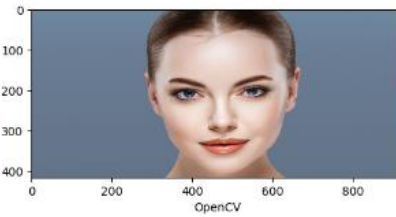
Aleator

2. Rezultatele obtinute dupa eliminarea a 100 pixeli pe inaltime dintr-o poza cu o fata:

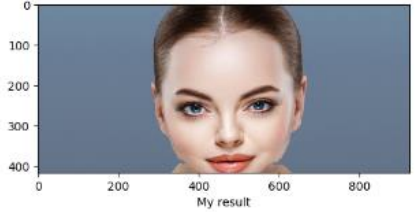
Observam ca varianta cu programare dinamica nu arata tocmai bine. Acest lucru se datoreaza faptului ca toate drumurile orizontale trec prin fata, deci cu toate ca algoritmul alege drumurile cu gradientii cei mai mici, ele vor trece prin zona barbie/gat, rezultatul final nefiind unul reusit. Varianta greedy alege mereu drumuri care incep cu pixelul stanga jos deci taie tot din barbie/gat.



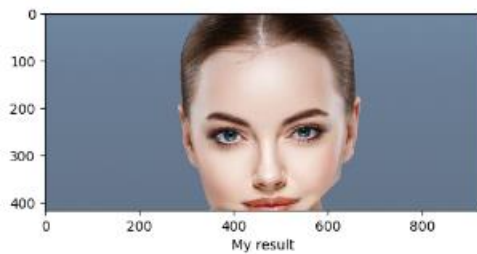
Poza originala



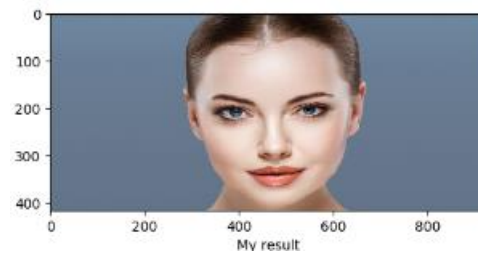
Resize OpenCV



Programare Dinamica



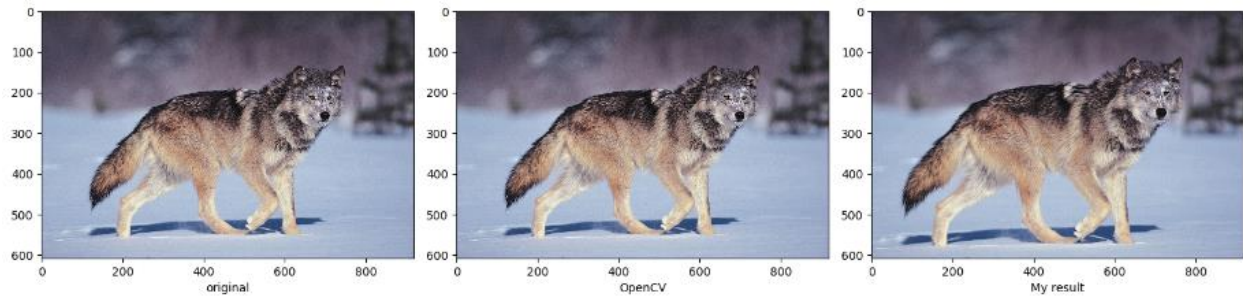
Greedy



Aleator

3. Rezultatele obtinute dupa amplificarea continutului cu 10% pe o imagine cu un lup:

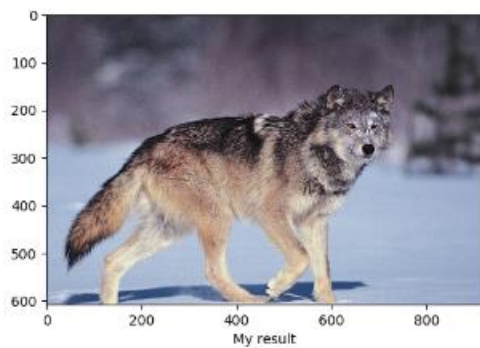
Observam ca lupul este mai mare, deoarece imaginea a fost marita (cu resize) si apoi au fost eliminate drumuri care treceau doar prin zona de background, acolo fiind gradientii mai mici. Am pus si imaginile obtinute cu greedy si cu selectarea aleatoare a drumurilor.



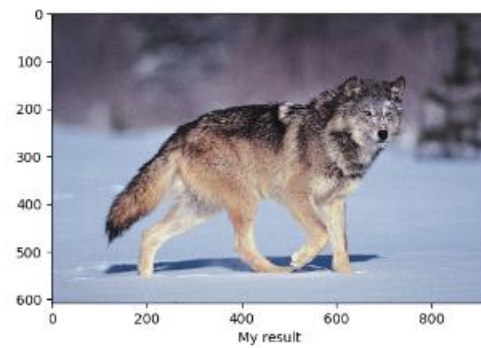
Poza originala

Resize OpenCV (tot poza originala)

Programare dinamica



Greedy



Aleator

4. Rezultatele obtinute dupa eliminarea unui urs dintr-o poza cu 3 ursi:

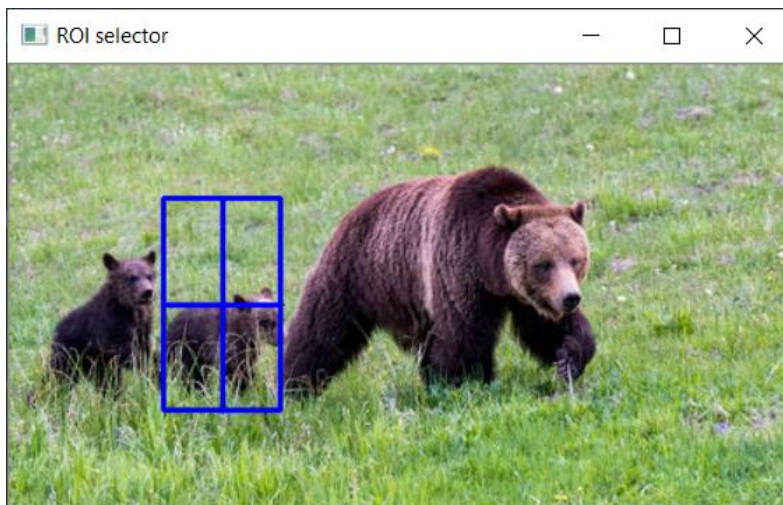
Observam ca rezultatul obtinut este unul reusit, deoarece am selectat o regiune de interes cu inaltimea mai mare decat latimea, astfel s-au eliminat drumuri verticale care treceau doar prin ursul ce trebuia eliminat si prin iarba de deasupra si dedesubtul lui. Daca selectam o zona cu latimea mai mare decat inaltimea (cum este si forma ursului ce doream sa fie eliminat), algoritmul nu ar fi avut de ales decat sa taie si din ceilalti 2 ursi, cu drumuri orizontale.



Poza originala

Resize OpenCV

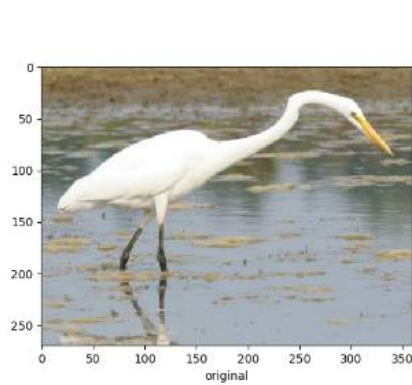
Eliminare obiect
(Programare Dinamica)



Ursul selectat pentru eliminare

5. Rezultatele obtinute dupa eliminarea a 100 pixeli pe latime dintr-o poza cu o barza:

Observam ca rezultatul prin metoda programarii dinamice nu este unul reusit, deoarece sunt selectate pentru eliminare drumuri care trec prin barza, avand gradienti mici (barza este in parte parte alba, fara schimbari de culoare). In varianta greedy, drumurile sunt alese pornind din pixelul stanga sus deci dispare partea din spate a berzei.



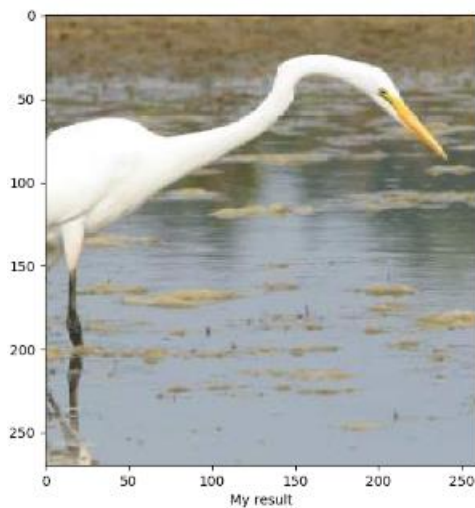
Poza originala



Resize OpenCV



Programare Dinamica



Greedy



Aleator