

## Aplicații în Vederea Artificială - Tema 2

### Redimensionarea imaginilor cu păstrarea conținutului

#### Subpunct a - micșorarea imaginii pe lățime

Imaginea inițială



Imaginea inițială redimensionată la o imagine cu 50 de pixeli mai puțini în lățime cu algoritmul de redimensionare uzuală (folosind funcția 'resize' din OpenCV).



Imaginea inițială redimensionată la o imagine cu 50 de pixeli mai puțini în lățime folosind algoritmul de păstrare al conținutului, utilizând metoda programării dinamice.



## Subpunct b - micșorarea imaginii pe înălțime

Imaginea inițială



Imaginea inițială redimensionată la o imagine cu 100 de pixeli mai puțini în înălțime cu algoritmul de redimensionare uzuală (folosind funcția 'resize' din OpenCV).





Imaginea inițială redimensionată la o imagine cu 100 de pixeli mai puțini în înălțime folosind algoritmul de păstrare al conținutului, utilizând metoda programării dinamice.



## Subpunct c - amplificarea conținutului imaginii

Imaginea inițială



Imaginea inițială, amplificând conținutul acesteia cu factorul  $f=1.15$ , păstrând dimensiunile, utilizând metoda programării dinamice.





Imaginea inițială, amplificând conținutul acesteia cu factorul  **$f=1.05$** , păstrând dimensiunile, utilizând metoda programării dinamice.



## Subpunct d - eliminarea unui obiect din imagine

Imaginea inițială



Imaginea inițială, folosind algoritmul de eliminare al unui obiect din imagine (este eliminată din poza inițială femeia cu bluza albă)

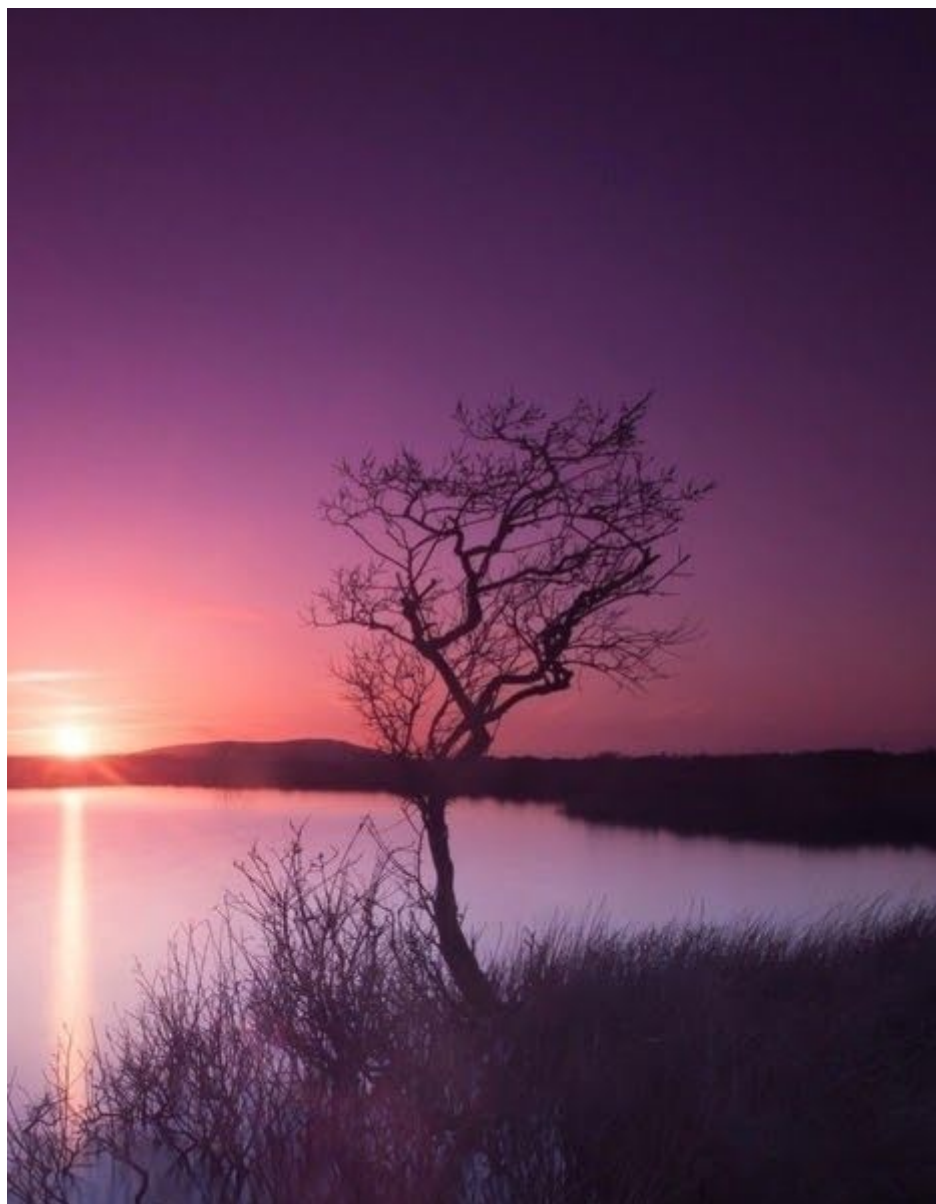




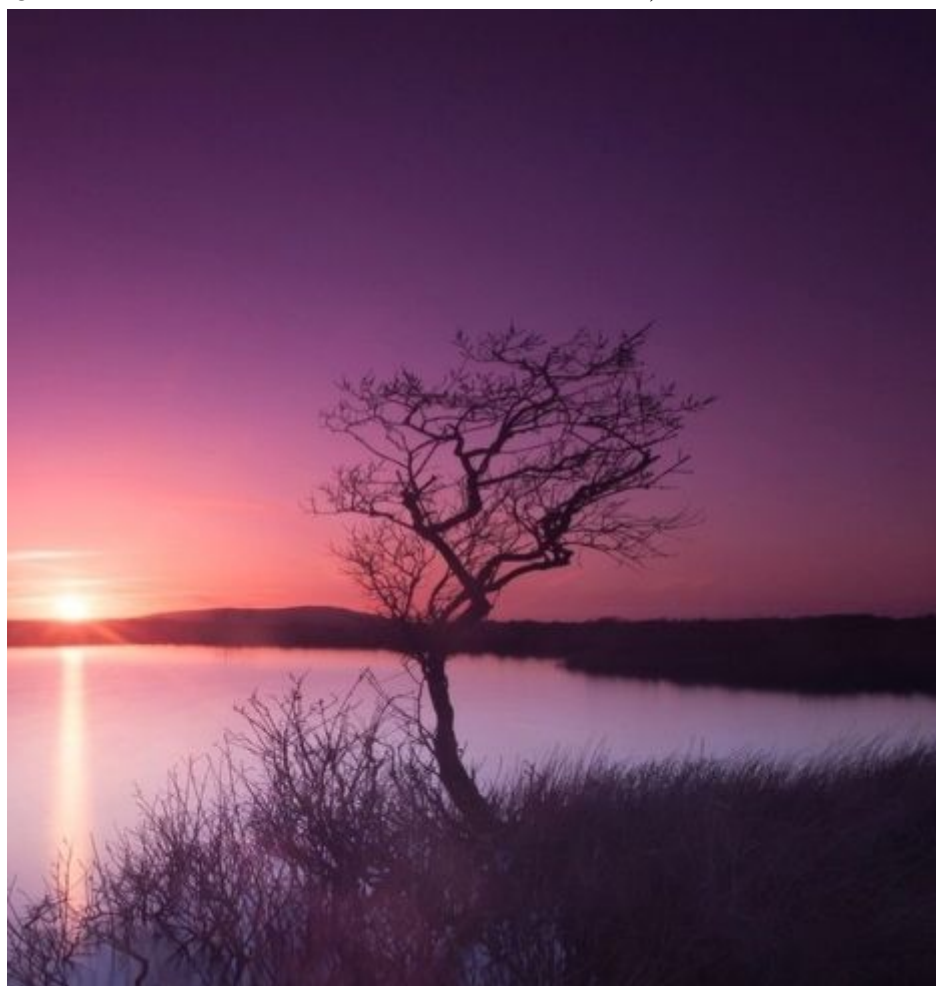
## Subpunct e

**IMAGINEA 1 (caz reușit) - micșorarea imaginii pe înălțime cu 110px**

Imaginea inițială

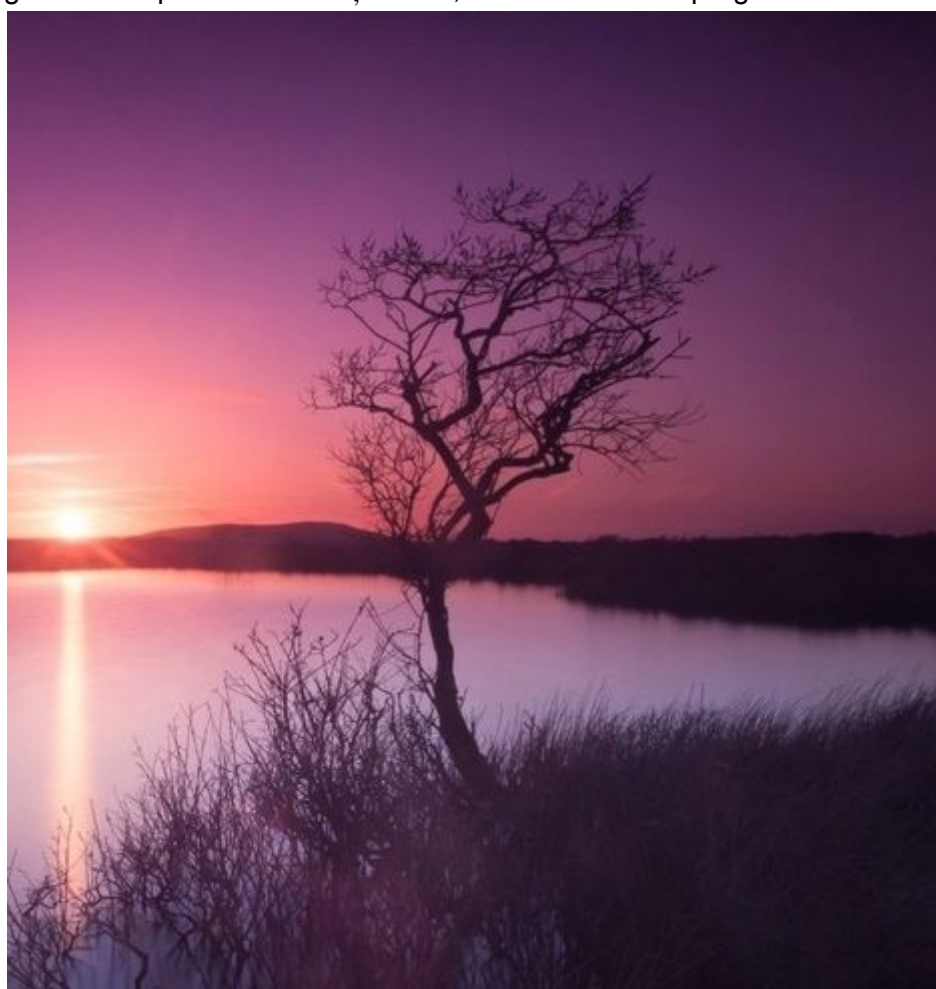


Imaginea inițială redimensionată la o imagine cu 110 de pixeli mai puțini în înălțime cu algoritmul de redimensionare uzuală (folosind funcția 'resize' din OpenCV).

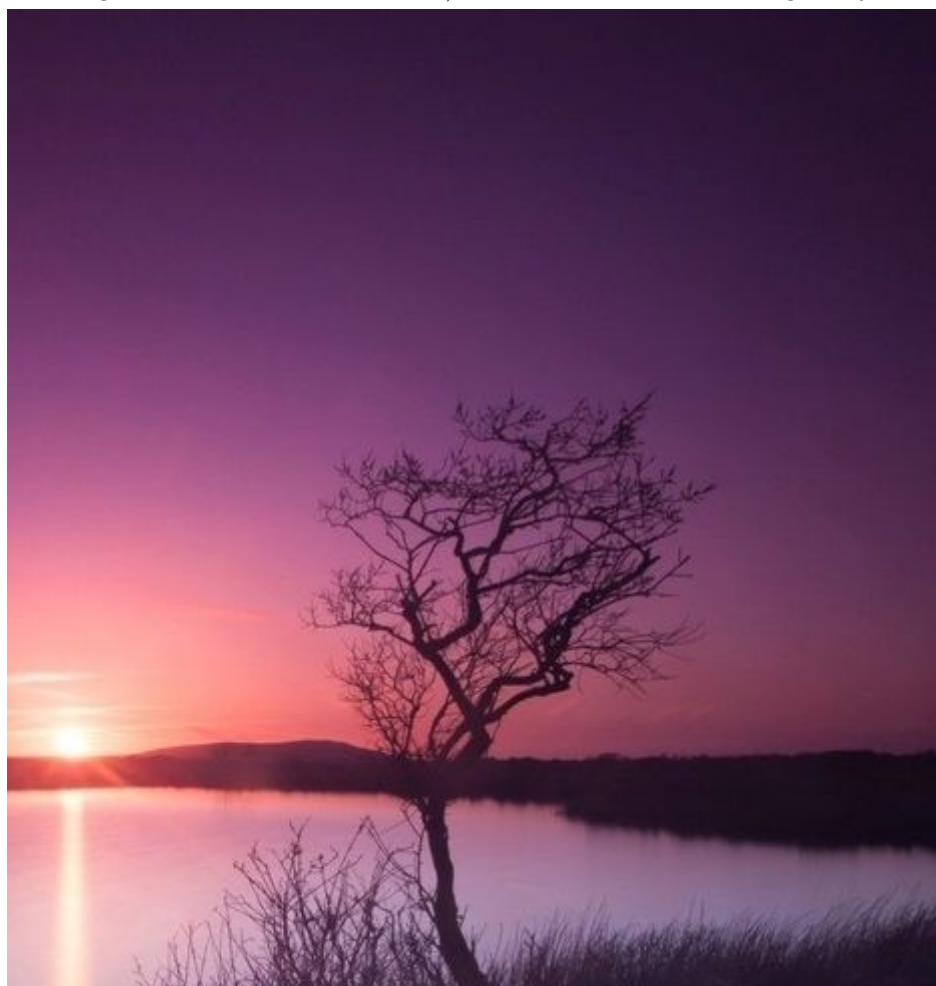




Imaginea inițială redimensionată la o imagine cu 110 de pixeli mai puțini în înălțime folosind algoritmul de păstrare al conținutului, utilizând metoda programării dinamice.

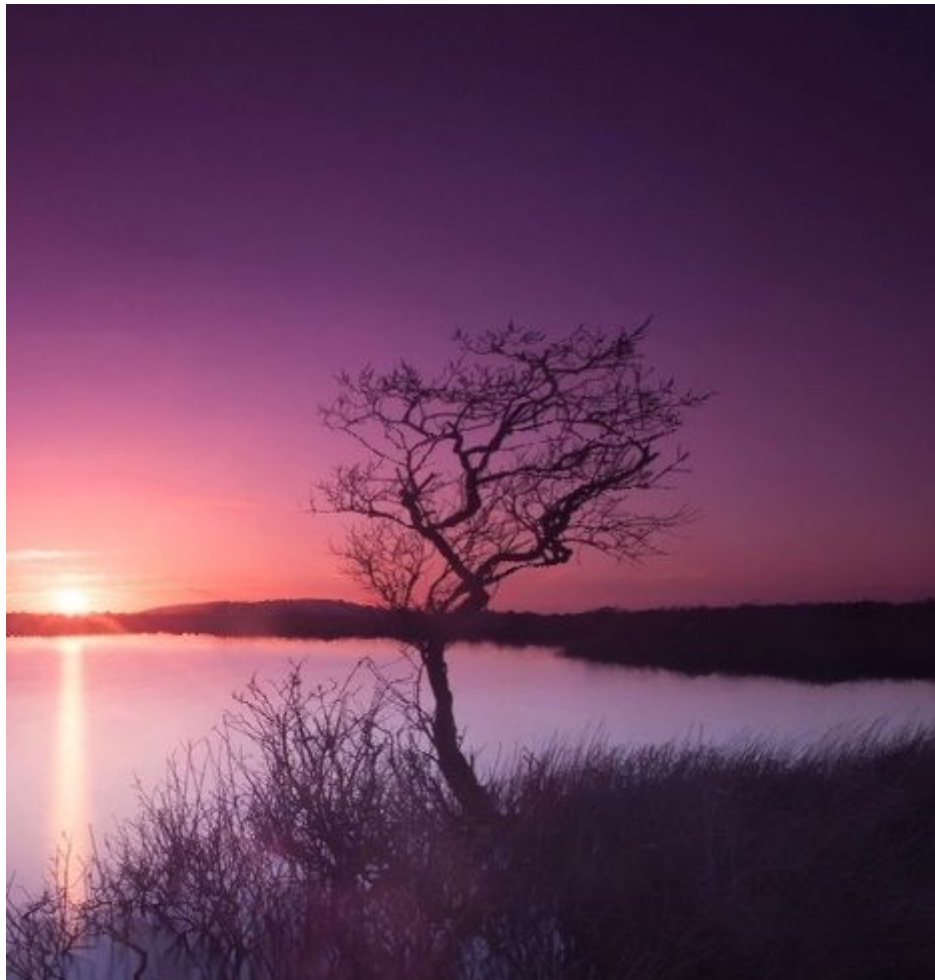


Imaginea inițială redimensionată la o imagine cu 110 de pixeli mai puțini în înălțime folosind algoritmul de păstrare al conținutului, utilizând metoda greedy.





Imaginea inițială redimensionată la o imagine cu 110 de pixeli mai puțini în înălțime folosind algoritmul de păstrare al conținutului, utilizând metoda aleatoare.



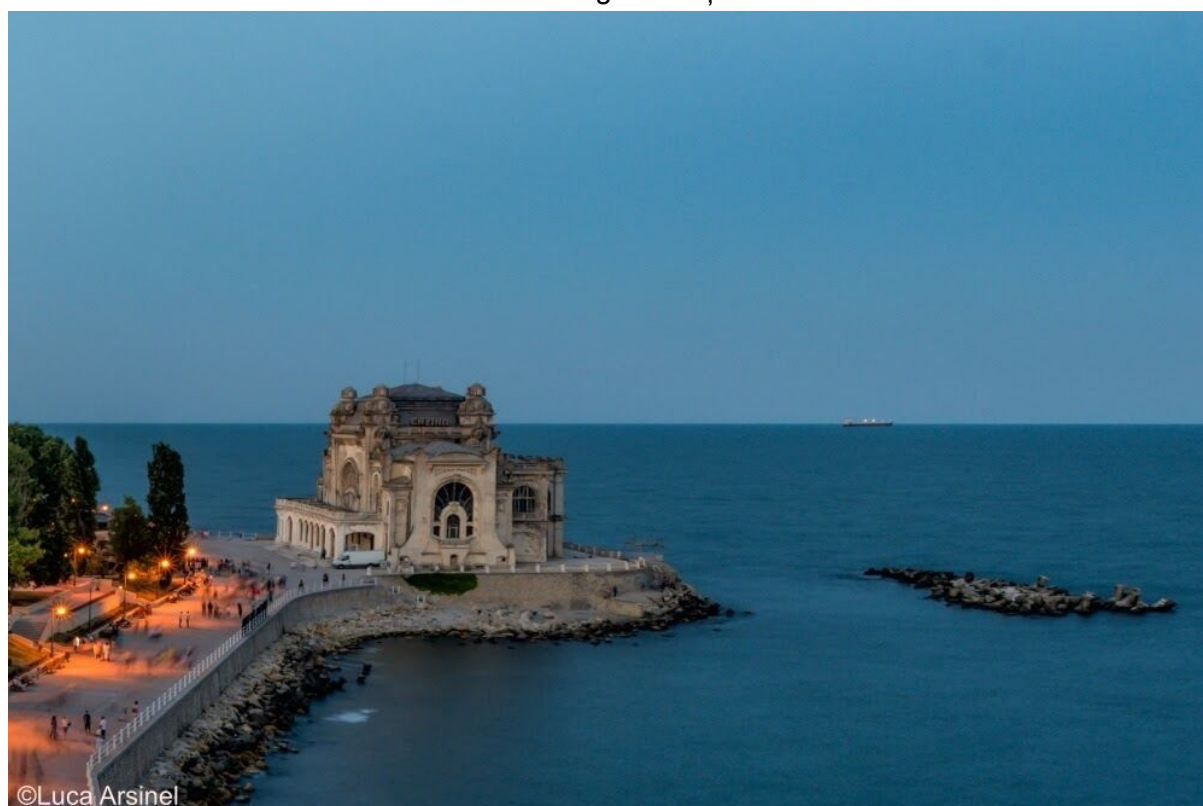
### Concluzii:

Pozele rezultate prin toate cele 3 metode de selectare a drumurilor au reușit pentru imaginea inițială.

- Metoda de selecție aleatoare: în mod surprinzător, deși această metodă nu se bazează pe alegerea unui drum de cost minim, rezultatul algoritmului este aproximativ reușit, observându-se o ușoară deformare la nivelul crengilor copacului.
- Metoda greedy: datorită faptului că această metodă implică alegerea optimelor locale, apoi combinarea acestora pentru a crea un drum optim (care însă nu va fi optim global), algoritmul elimină drumuri din partea de jos a imaginii (lac) întrucât pixelii cu gradientul cel mai mic de pe prima coloană de pixeli a pozei se află în această zonă. Astfel, algoritmul elimină o parte din lac.
- Metoda programării dinamice: în acest caz, majoritatea drumurilor eliminate se află în partea superioară a imaginii (cer), întrucât drumurile de cost minim din matricea de gradienti se află în această zonă, iar un număr mic de drumuri eliminate se află în partea inferioară a imaginii (lac).

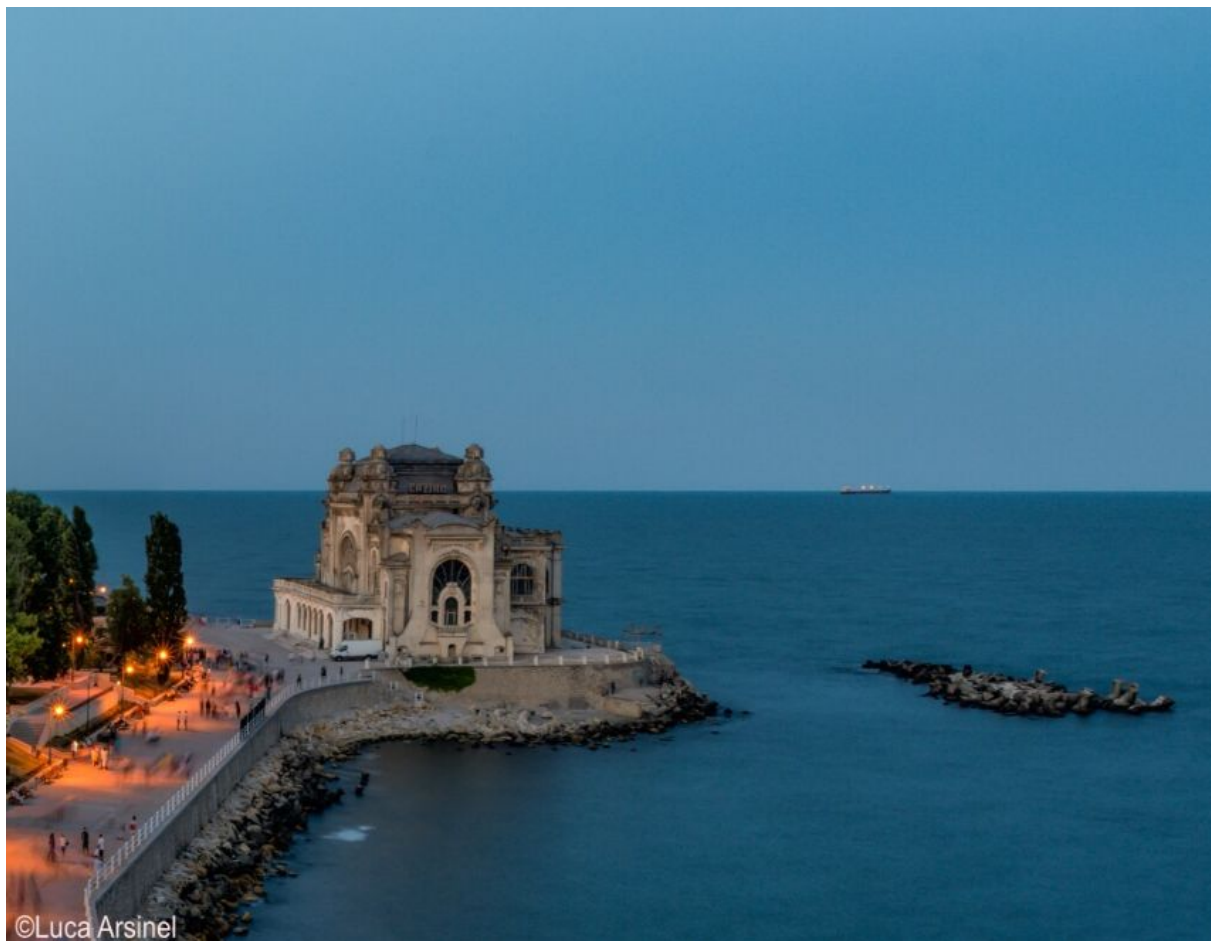
**IMAGINEA 2 (caz reușit) - micșorarea imaginii pe lățime cu 150px**

Imaginea inițială

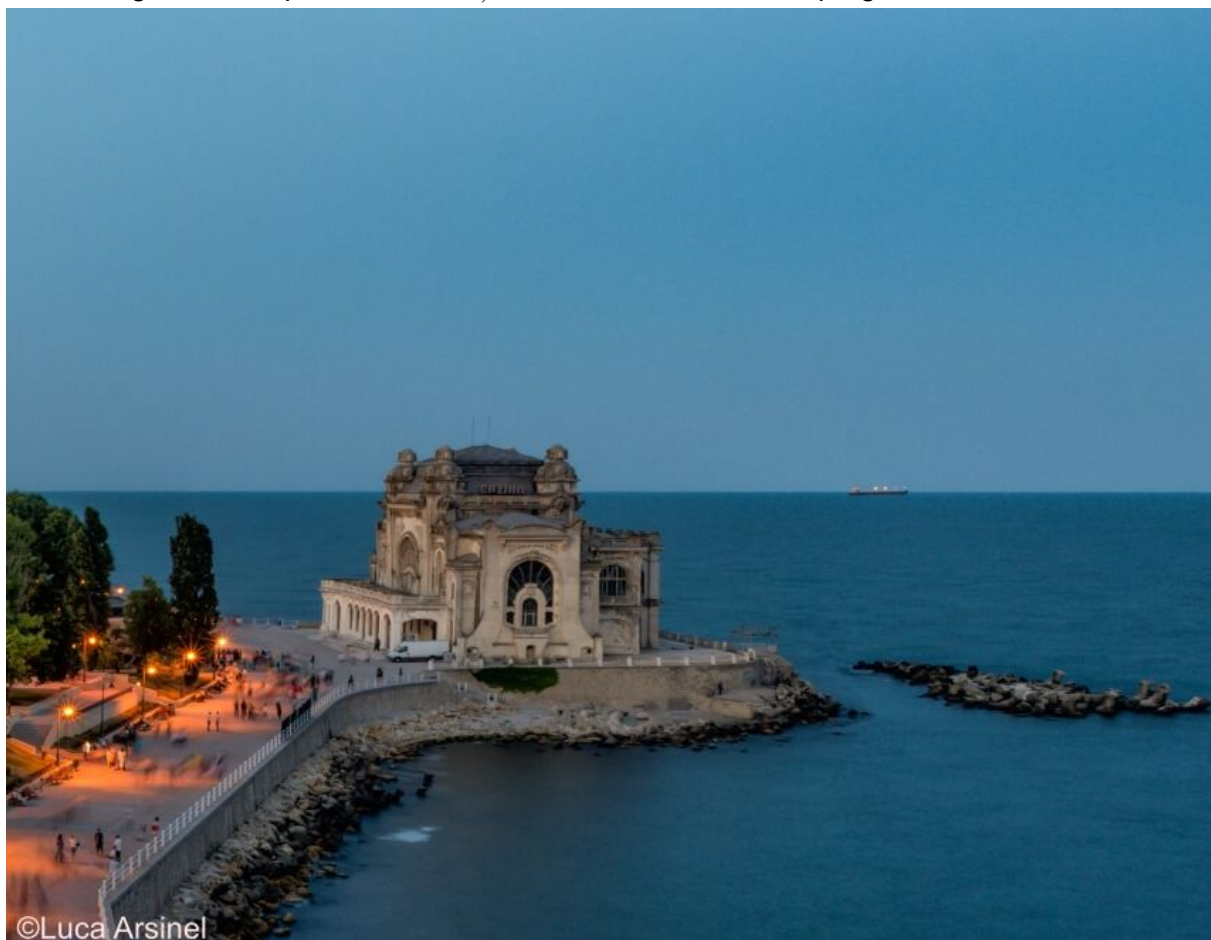




Imaginea inițială redimensionată la o imagine cu 150 de pixeli mai puțini în lățime cu algoritmul de redimensionare uzuală (folosind funcția 'resize' din OpenCV).

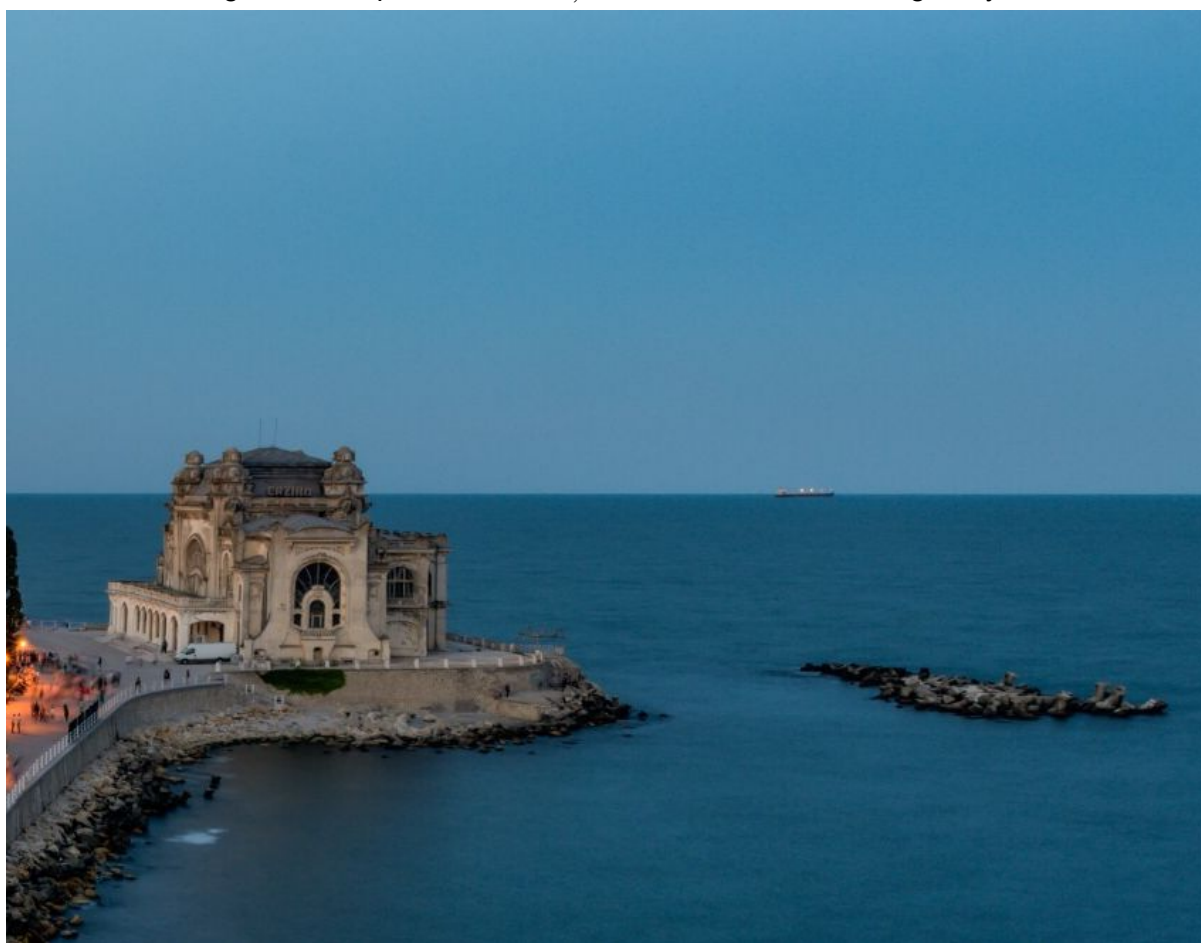


Imaginea inițială redimensionată la o imagine cu 150 de pixeli mai puțini în lățime folosind algoritmul de păstrare al conținutului, utilizând metoda programării dinamice.

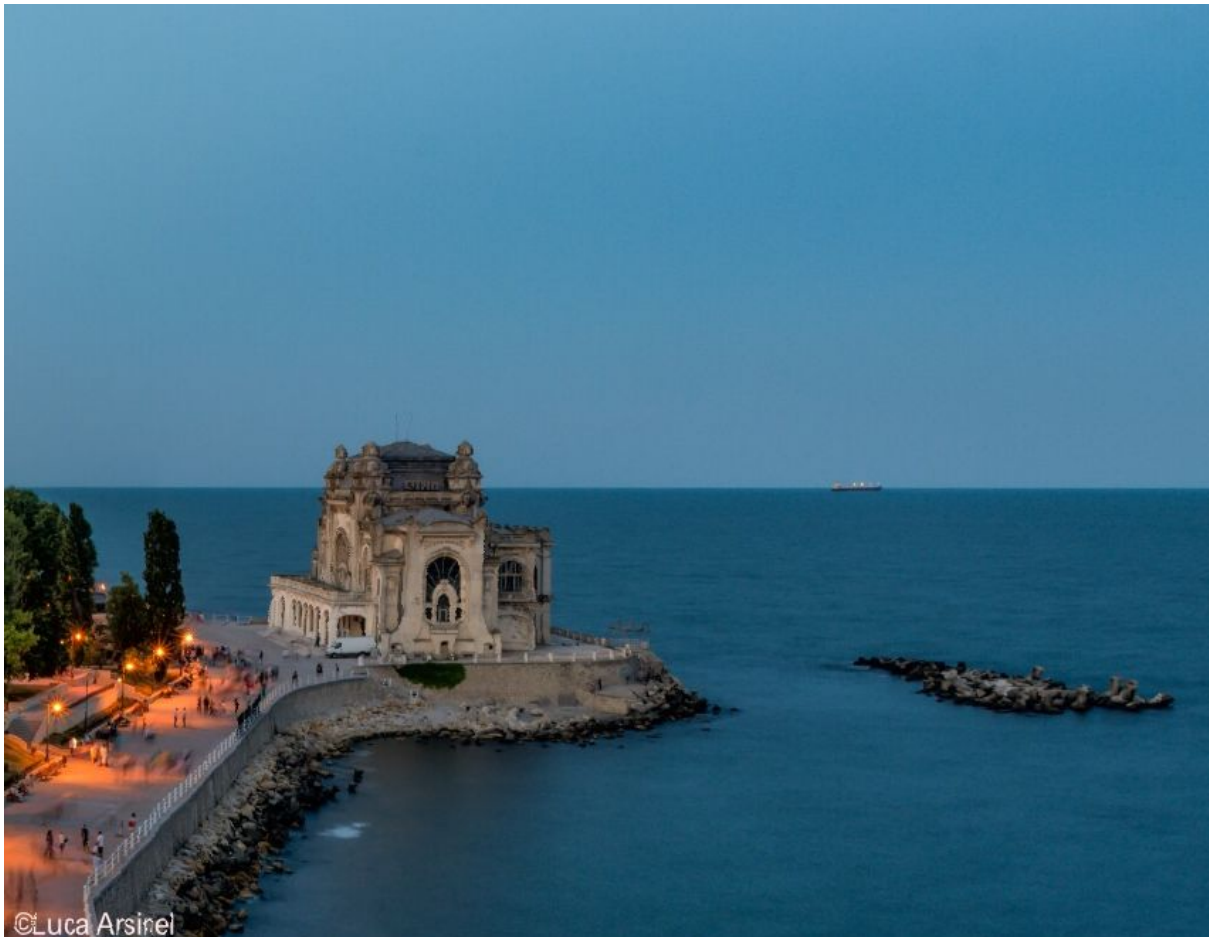




Imaginea inițială redimensionată la o imagine cu 150 de pixeli mai puțini în lățime folosind algoritmul de păstrare al conținutului, utilizând metoda greedy.



Imaginea inițială redimensionată la o imagine cu 150 de pixeli mai puțini în lățime folosind algoritmul de păstrare al conținutului, utilizând metoda aleatoare.

**Concluzii:**

Pozele rezultate prin toate cele 3 metode de selectare a drumurilor au reușit pentru imaginea inițială într-o mare măsură.

- Metoda de selecție aleatoare: rezultatul algoritmului este aproximativ reușit, observându-se o ușoară deformare a străzii și a casinoului.
- Metoda greedy: algoritmul elimină drumuri din partea de stânga a imaginii (stradă, copaci) întrucât pixelii cu gradientul cel mai mic de pe primul rând de pixeli a pozei se află în această zonă. Astfel, algoritmul elimină o bună parte din stradă și copaci.
- Metoda programării dinamice: în acest caz, majoritatea drumurilor eliminate se afla în partea care conține apă, lăsând aproape intacte strada, casinoul, vaporul din depărtare și Insulița Cazino(conglomeratul de stabilopozi); majoritatea drumurilor de cost minim din matricea de gradienți se află între casino și insulă(marea și cerul)

**IMAGINEA 3 (caz reușit) - amplificarea conținutului imaginii cu factor  $f=1.3$**

Imaginea inițială



Imaginea rezultată de pe urmă funcției 'resize' din OpenCV nu a suferit nicio transformare față de cea originală.





Imaginea inițială amplificând conținutul acesteia cu factorul  $f=1.3$ , păstrând dimensiunile, utilizând metoda programării dinamice.



Imaginea inițială amplificând conținutul acesteia cu factorul  $f=1.3$ , păstrând dimensiunile, utilizând metoda greedy.



Imaginea inițială amplificând conținutul acesteia cu factorul  $f=1.3$ , păstrând dimensiunile, utilizând metoda aleatoare.

**Concluzii:**

Pozele rezultate prin metoda greedy și a programării dinamice a drumurilor au reușit pentru imaginea inițială într-o mare măsură.

- Metoda de selecție aleatoare: rezultatul algoritmului nu a reușit, fluturele fiind deformat.
- Metoda greedy: algoritmul elimină atât drumuri din partea stângă a imaginii (floarea din stânga jos nu mai este vizibilă deloc), cât și din partea de jos a imaginii (o bună parte din tulpina florii nu mai este vizibilă). Acest lucru este datorat faptului că, pixelii cu gradientul cel mai mic de pe primul rând de pixeli a pozei se află în partea din stânga a imaginii, pixelii cu gradientul cel mai mic de pe prima coloană de pixeli a pozei se află în partea de jos a imaginii.
- Metoda programării dinamice: în acest caz, drumurile verticale eliminate se afla atât în partea din stânga a florii și a fluturelui, cât și în partea de dreapta, algoritmul lăsând intactă atât floarea din stânga jos, cât și fluturele și floarea pe care este așezat. Majoritatea drumurilor orizontale eliminate se află în partea de sus a fluturelui, însă există câteva drumuri eliminate care trec prin floarea pe care se află fluturele, deformând-o ușor.

**IMAGINEA 4 (caz nereușit) - micșorarea imaginii pe lățime cu 100px**

Imaginea inițială



Imaginea inițială redimensionată la o imagine cu 100 de pixeli mai puțini în lățime cu algoritmul de redimensionare uzuală (folosind funcția 'resize' din OpenCV).





Imaginea inițială redimensionată la o imagine cu 100 de pixeli mai puțini în lățime folosind algoritmul de păstrare al conținutului, utilizând metoda programării dinamice.



Imaginea inițială redimensionată la o imagine cu 100 de pixeli mai puțini în lățime folosind algoritmul de păstrare al conținutului, utilizând metoda greedy.



Imaginea inițială redimensionată la o imagine cu 100 de pixeli mai lățime în lățime folosind algoritmul de păstrare al conținutului, utilizând metoda aleatoare.

**Concluzii:**

Pozele rezultate prin cele 3 metode nu au reușit pentru imaginea inițială.

- Metoda de selecție aleatoare: rezultatul algoritmului nu a reușit, însă dintre toate cele 3 metode a dat rezultatul cel mai bun deoarece nu se bazează pe alegerea unui drum de cost minim.
- Metoda greedy: algoritmul elimină drumuri din partea de stânga a imaginii (partea din stânga a chipului femeii), întrucât pixelii cu gradientul cel mai mic de pe primul rând de pixeli a pozei se află în această zonă. De asemenea, se observă că trăsăturile din partea din stânga sunt deformate.
- Metoda programării dinamice: se observă că imaginea își pierde din veridicitate, trăsăturile faciale ale femeii sunt deformate deoarece drumurile de cost minim, din matricea gradientilor aferentă, sunt eliminate din aceeași zonă.

**IMAGINEA 5 (caz nereușit) - micșorarea imaginii pe înălțime cu 100px**

Imaginea inițială



Imaginea inițială redimensionată la o imagine cu 100 de pixeli mai puțini în înălțime cu algoritmul de redimensionare uzuală (folosind funcția 'resize' din OpenCV).



Imaginea inițială redimensionată la o imagine cu 100 de pixeli mai puțini în înălțime folosind algoritmul de păstrare al conținutului, utilizând metoda programării dinamice.





Imaginea inițială redimensionată la o imagine cu 100 de pixeli mai puțini în înălțime folosind algoritmul de păstrare al conținutului, utilizând metoda greedy.



Imaginea inițială redimensionată la o imagine cu 100 de pixeli mai puțini în înălțime folosind algoritmul de păstrare al conținutului, utilizând metoda aleatoare.



### Concluzii:

Pozele rezultate prin cele 3 metode nu au reușit pentru imaginea inițială.

- Metoda de selecție aleatoare: rezultatul algoritmului nu a reușit, este vizibilă deformarea trăsăturilor oamenilor.
- Metoda greedy: algoritmul elimină drumuri din partea de jos a imaginii (afectând în principal primul rând de oameni), întrucât pixelii cu gradientul cel mai mic de pe primul rând de pixeli a pozei se află în această zonă. De asemenea, se observă că trăsăturile persoanelor din josul imaginii suferă de o ușoară deformare.
- Metoda programării dinamice: se observă că imaginea își pierde din veridicitate, trăsăturile oamenilor sunt distorsionate, în principal ale celor din primul rând: chipul bărbatului din stânga jos și al doamnei din mijloc, trupurile celor două fete din dreapta jos.