**Algoritmi pentru redimensionarea imaginilor**

**Stoian Vlad 461**

**Scorbureanu Sergiu 461**

**Ghergu Nicolae-Marius 462**

**Facultatea de Matematica și Informatica,**

**Universitatea din București**

**Capitole:**

1. **Importanța redimensionării imaginilor**

**1.1. Demonstrarea redimensionării unei imagini in Python**

1. **Conceptul de pixel și imagine**
2. **Determinarea factorilor de scalare**
3. **Eșantionarea pixelilor pentru redimensionare**
4. **Calculul noilor coordonate ale pixelilor**
5. **Aplicarea interpolării**
6. **Alocarea valorilor pixelilor**
7. **Interpolarea “Nearest Neighbor”**

**8.1. Demonstrarea interpolării “Nearest Neighbor” in Python**

1. **Interpolarea bilineară**

**9.1. Demonstrarea interpolării bilineare in Python**

**10. Evaluarea sistematică a metodelor prezentate**

**10.1. Pentru Metoda Nearest Heighbor**

**10.2. Pentru Metoda Bilineara**

1. **Importanța redimensionării imaginilor**

În multe domenii, cum ar fi social media și web, redimensionarea imaginilor este un proces esențial pentru a accelera încărcarea paginilor și a asigura că imaginile se potrivesc pe diferite dispozitive, precum și pentru analiza de imagini și învățarea automată, unde reducerea dimensiunilor ajută la procesarea mai rapidă și mai eficientă.

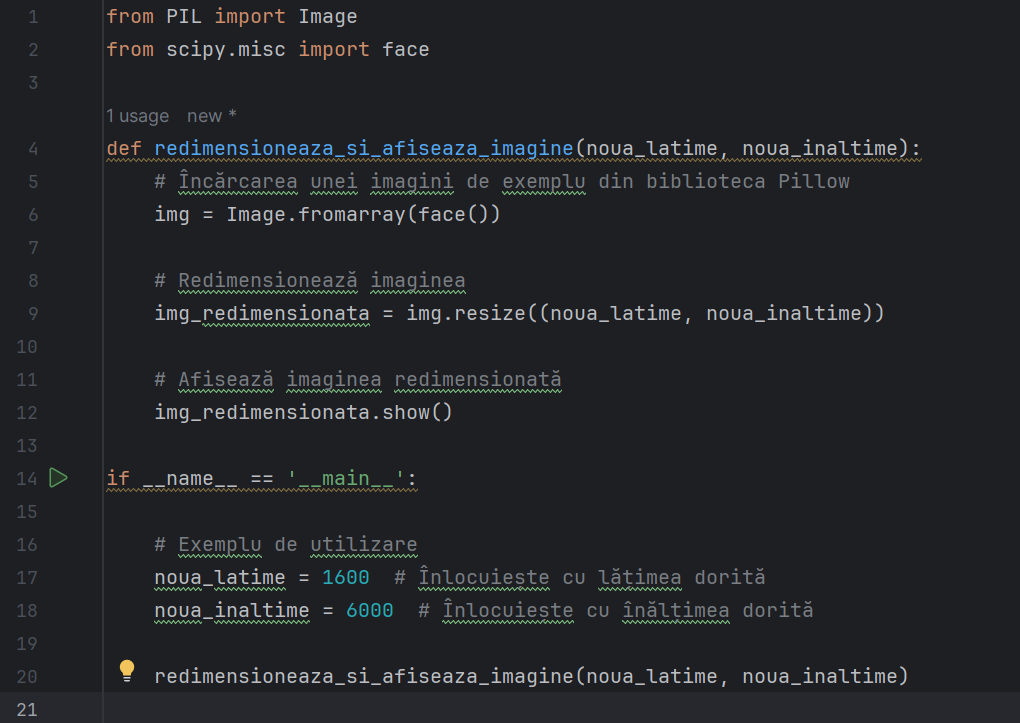
În designul grafic și editarea fotografiilor, este de asemenea crucială pentru adaptarea fotografiilor la diferite formate și scopuri creative. Algoritmii de redimensionare, cum ar fi cei liniari, cubici sau Lanczos, sunt esențiali pentru a echilibra calitatea imaginii și atingerea obiectivelor, inclusiv reducerea dimensiunii fișierului.

Acest lucru afectează calitatea vizuală și utilitatea imaginilor redimensionate.

**1.1. Demonstrarea redimensionării unei imagini in Python**

Codul este destul de ușor de înțeles, iar logica de redimensionare se afla în spatele metodei resize; vom începe prin a explora logica matematică din spatele acestei metode, prin analiza mai multor concepte:

* conceptul de pixel și imagine;
* determinarea factorilor de scalare;
* eșantionarea pixelilor pentru redimensionare;
* calculul noilor coordonate ale pixelilor;
* aplicarea interpolării;
* alocarea valorilor pixelilor;



1. **Conceptul de pixel și imagine**

O imagine digitală este formată dintr-o grilă de pixeli în două dimensiuni, cu fiecare pixel că cea mai mică unitate de imagine controlabilă individual.

În formatul RGB, fiecare pixel al unei imagini este reprezentat de trei intensități de culoare: roșu, verde și albastru.

Există milioane de nuanțe diferite care pot fi create combinând proporțiile diferite ale acestor trei culori de bază. De exemplu, o combinație egală a tuturor celor trei culori creează alb, în timp ce o combinație complet absentă creează negru.

Practic, fiecare valoare a unei culori este reprezentată de un număr care se întinde între 0 și 255, oferind astfel o gamă largă de opțiuni cromatice.

În tehnologiile digitale, acest tip de reprezentare a imaginilor este răspândit.

1. **Determinarea factorilor de scalare**

Procesul de redimensionare a unei imagini implică determinarea factorilor de scalare. Acest proces începe prin calcularea raportului dintre dimensiunile imaginii originale și cele ale noii imagini (lățimea și înălțimea).

Pentru a înțelege cât de mult trebuie mărită sau micșorată imaginea, trebuie să luam în considerare acești factori de scalare. De exemplu, dacă dorim să redimensionăm o imagine care are o lățime de 1000 de pixeli la 500 de pixeli, factorul de scalare pentru lățime va fi 0,5, ceea ce înseamnă că lățimea nouă este jumătate din cea originală.

În mod similar, factorul de scalare al unei înălțimi se găsește comparând înălțimea inițială cu înălțimea dorită.

1. **Eșantionarea pixelilor pentru redimensionare**

Eșantionarea pixelilor, care diferă în funcție de dacă imaginea este mărită sau micșorată, este un proces esențial pentru redimensionarea imaginilor.

În subeșantionare, când imaginea este redusă, unii pixeli sunt identificați și eliminați sau combinați pentru a reduce dimensiunile dorite.

Aceasta înseamnă că un singur pixel poate reprezenta un grup de pixeli din imaginea originală în imaginea redimensionată, ceea ce reduce numărul total de pixeli. În schimb, în supereșantionare, atunci când imaginea este extinsă, se stabilesc zonele în care trebuie adăugați pixeli suplimentari.

Acest lucru se realizează prin interpolarea valorilor pixelilor actuali pentru a crea pixeli suplimentari, care umple spațiul extins al imaginii.

1. **Calculul noilor coordonate ale pixelilor**

După ce pixelii au fost eșantionați în procesul de redimensionare al unei imagini, următorul pas este de a determina coordonatele noi ale pixelilor.

Asta înseamnă că trebuie să găsim locul corect în imaginea originală pentru fiecare pixel din imaginea redimensionată. Acest lucru se face folosind factorii de scalare care au fost calculați anterior.

În imaginea redimensionată, coordonatele axei x și y ale fiecărui pixel sunt înmulțite cu factorii de scalare pentru a obține coordonatele din imaginea originală. Acest proces garantează că fiecare pixel al noii imagini este mapat corect în zona potrivită an imaginii originale.

Acest lucru permite interpolarea sau alte procesări ulterioare să fie efectuate corect în conformitate cu schimbările de dimensiune dorite.

1. **Aplicarea interpolării**

După ce s-au găsit noile coordonate ale pixelilor în procesul de redimensionare al unei imagini, pasul următor crucial este utilizarea interpolării pentru a determina valorile reale ale pixelilor din imaginea redimensionată.

Valorile pixelilor care rămân în subeșantionare, unde imaginea este redusă, sunt găsite folosind interpolarea, care ia în considerare datele din pixelii eliminați sau combinați. Chiar și cu un număr mic de pixeli, aceasta garantează o tranziție precisă și o reproducere fidelă a detaliilor originale.

Cu toate acestea, în supereșantionare, atunci când imaginea este extinsă, interpolarea ajută la determinarea valorilor pentru pixelii suplimentari.

Pentru a asigura o integrare armonioasă și o calitate vizuală adecvată, aceasta se bazează pe valorile și pozițiile pixelilor din jur.

1. **Alocarea valorilor pixelilor**

Următorul pas este aplicarea valorilor pixelilor după rezultatele utilizarii interpolării.

În acest moment, valorile obținute prin interpolare sunt efectiv atribuite pixelilor din imaginea redimensionată.

O valoare de culoare (sau intensitate) este atribuită fiecărui pixel din nouă imagine în urma calculelor de interpolare, care țin cont de pozițiile și valorile pixelilor din imaginea inițială.

Acest lucru garantează că imaginea redimensionată reproduce în mod corespunzător caracteristicile și detaliile imaginii originale, după ce au fost ajustate la noua dimensiune.

Prin acest proces, imaginea redimensionată este completată cu pixelii necesari, fie că aceasta a fost mărită și necesită pixeli suplimentari, fie că a fost redusă și pixelii au fost calculați din combinații ale celor existenți.

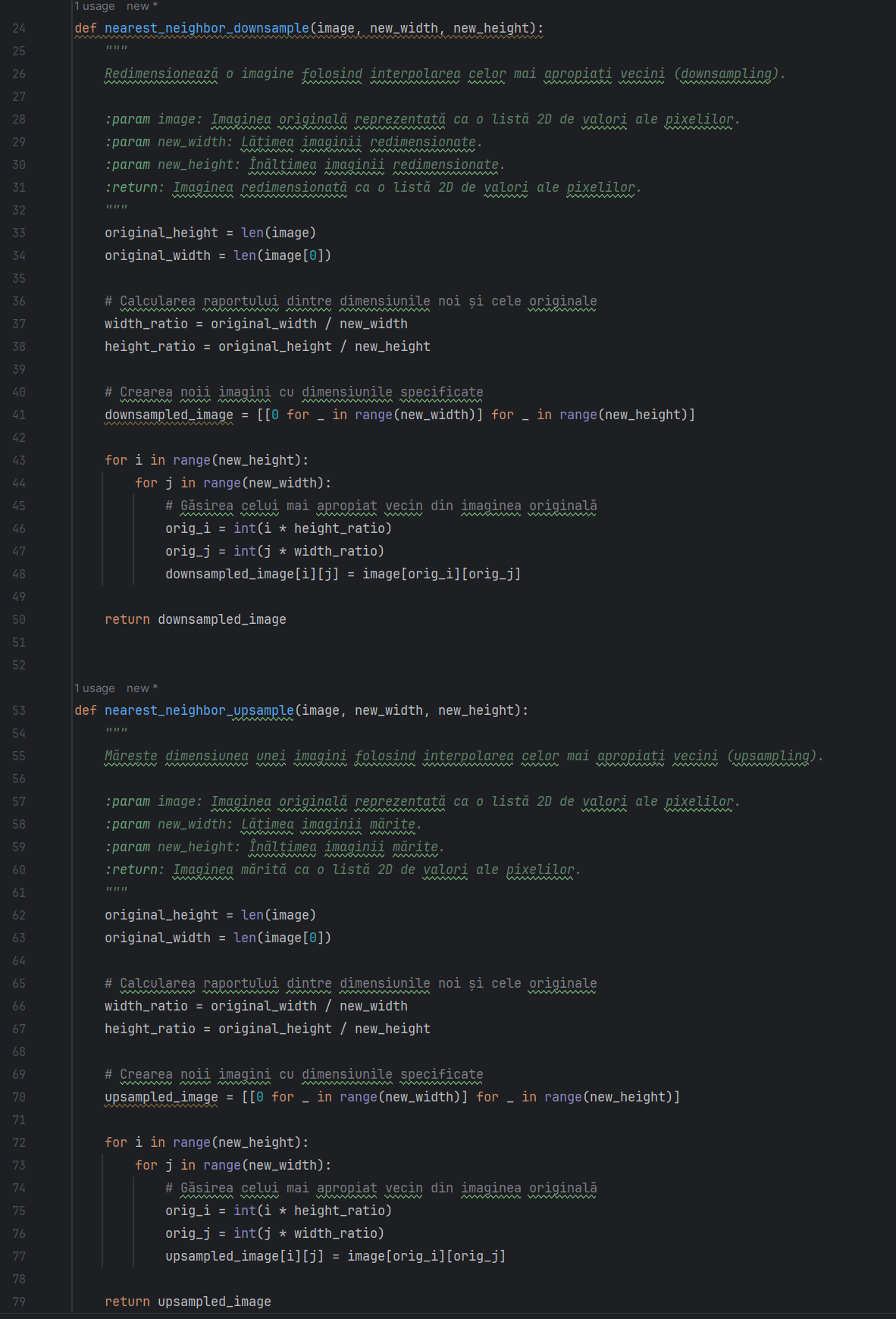
1. **Interpolarea “Nearest Neighbor”**

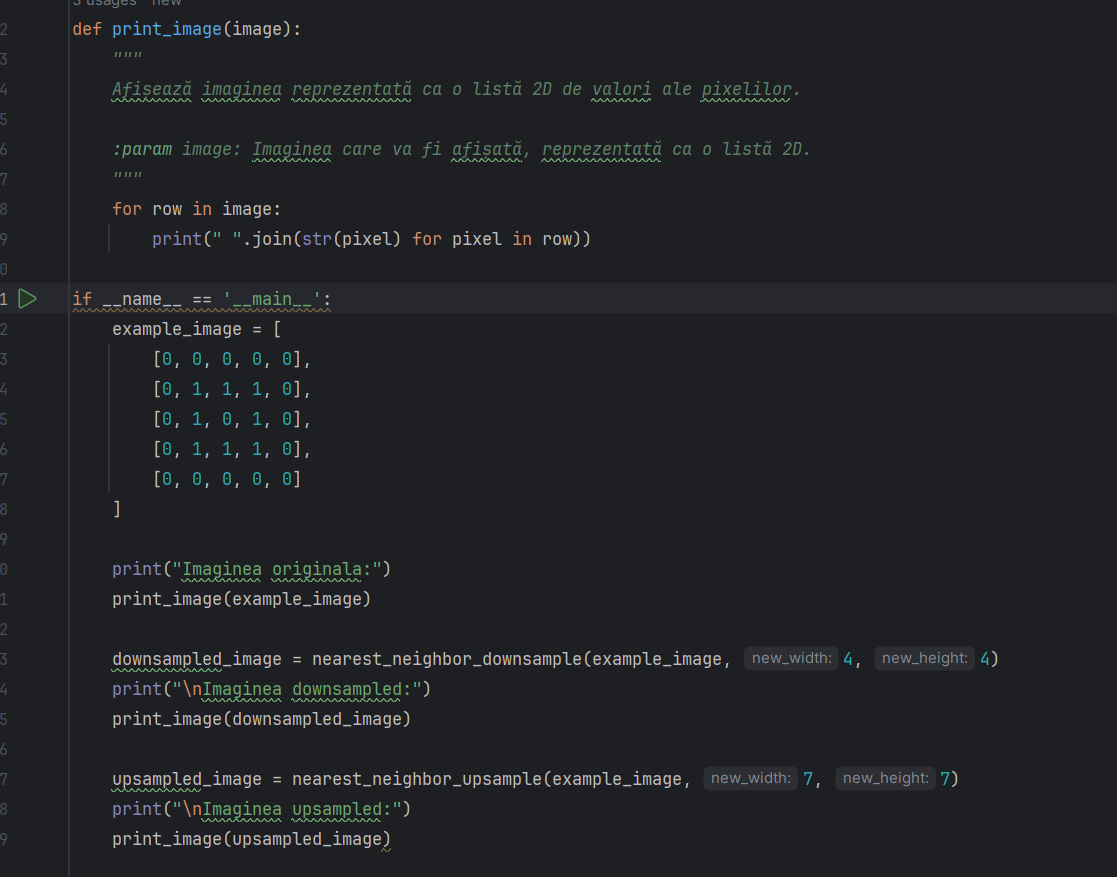
Metoda de interpolare Vecinul Cel Mai Apropiat este folosită atât în subeșantionare, cât și în supereșantionare în procesul de redimensionare al imaginilor; cu toate acestea, metodele sunt diferite în fiecare caz.

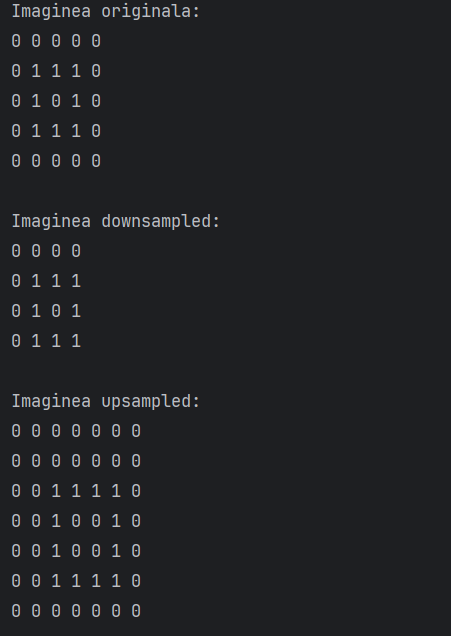
In cazul subeșantionarii, care implica reducerea dimensiunii imaginii, selectează un pixel specific dintr-o parte specifică a imaginii originale pentru fiecare pixel al imaginii redimensionate.

De obicei, acest pixel este cel mai apropiat de centrul zonei, ceea ce permite o abordare directa, fără a calcula medii sau combinații ale valorilor din jur.

În schimb, în supereșantionare, atunci când imaginea este extinsă, Vecinul Cel Mai Apropiat alege valoarea pixelului cel mai apropiat pentru a ocupa spațiile care au fost create.

**8.1. Demonstrarea interpolării “Nearest Neighbor” in Python**





1. **Interpolarea bilineară**

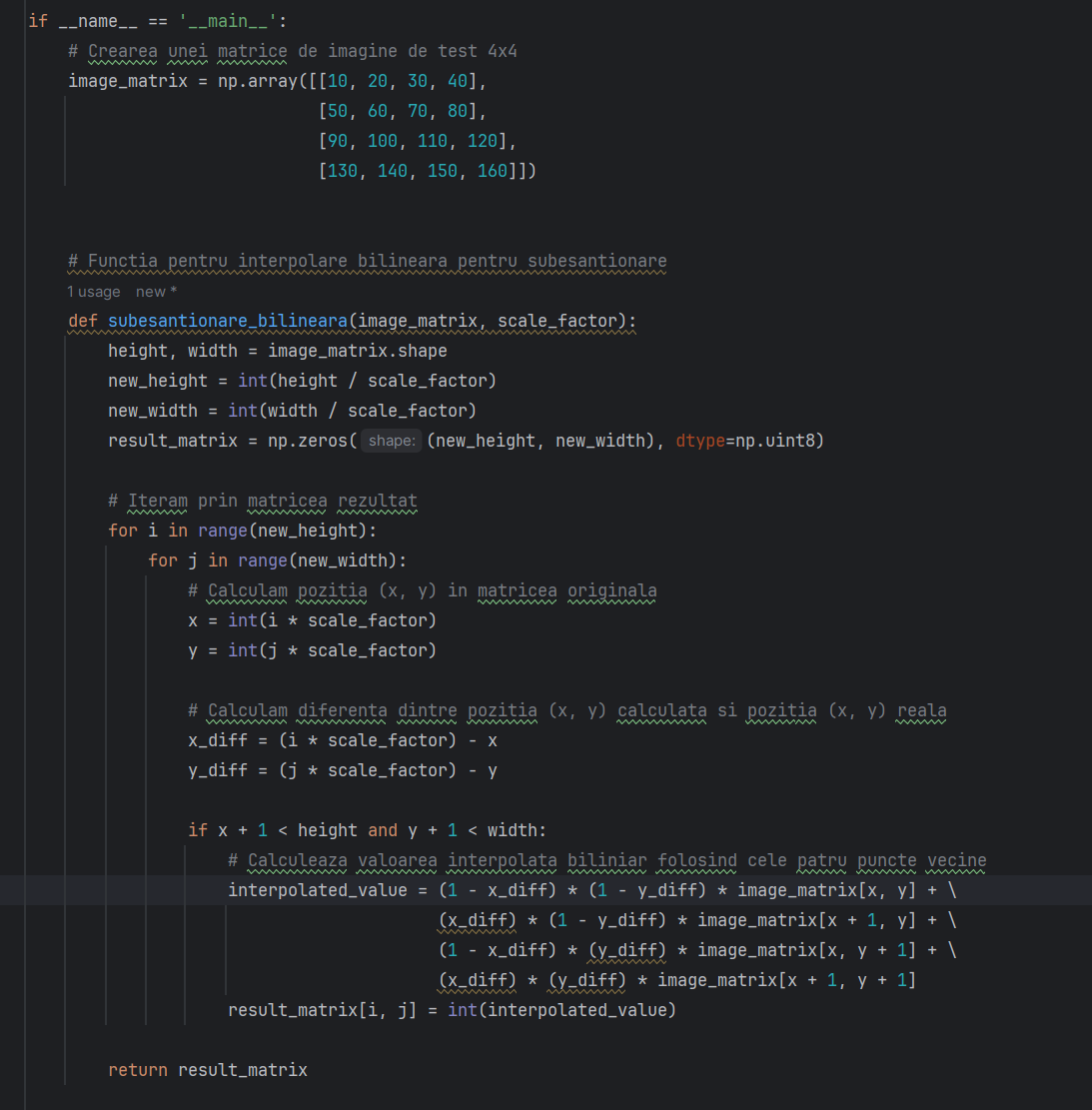
Interpolarea bilineară, care este mai complicată decât cea a Vecinului Cel Mai Apropiat, funcționează bine atât în subeșantionare, cât și în supereșantionare, oferind rezultate mai netede și mai naturale.

Interpolarea bilineară folosește media ponderată a celor patru pixeli cei mai apropiați din imaginea originală pentru a determina valorile fiecărui nou pixel în subeșantionare, când dimensiunea imaginii este redusă.

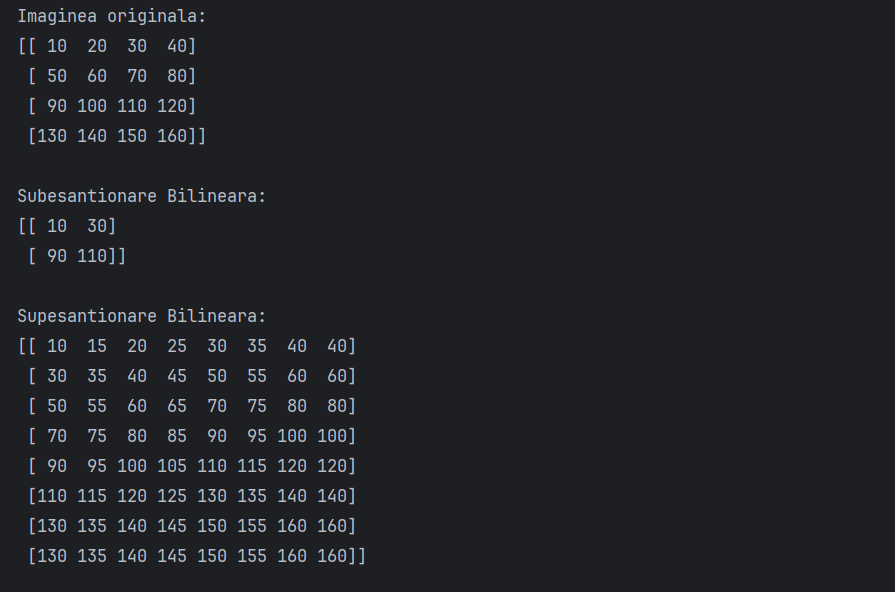
Datorită mediei ponderate, detaliile și gradientele de culoare sunt păstrate mai bine în comparație cu Vecinul Cel Mai Apropiat. Acest lucru reduce riscul apariției efectelor de aliasing sau margini zimțate.

În supereșantionare, când imaginea este mărită, metoda bilineară estimează valorile noilor pixeli adăugați folosind aceeași logică de medie ponderată a pixelilor înconjurători, asigurând o tranziție mai lină între zonele de culoare și evitând astfel apariția pixelilor izolați.

**9.1. Demonstrarea interpolării bilineare in Python**

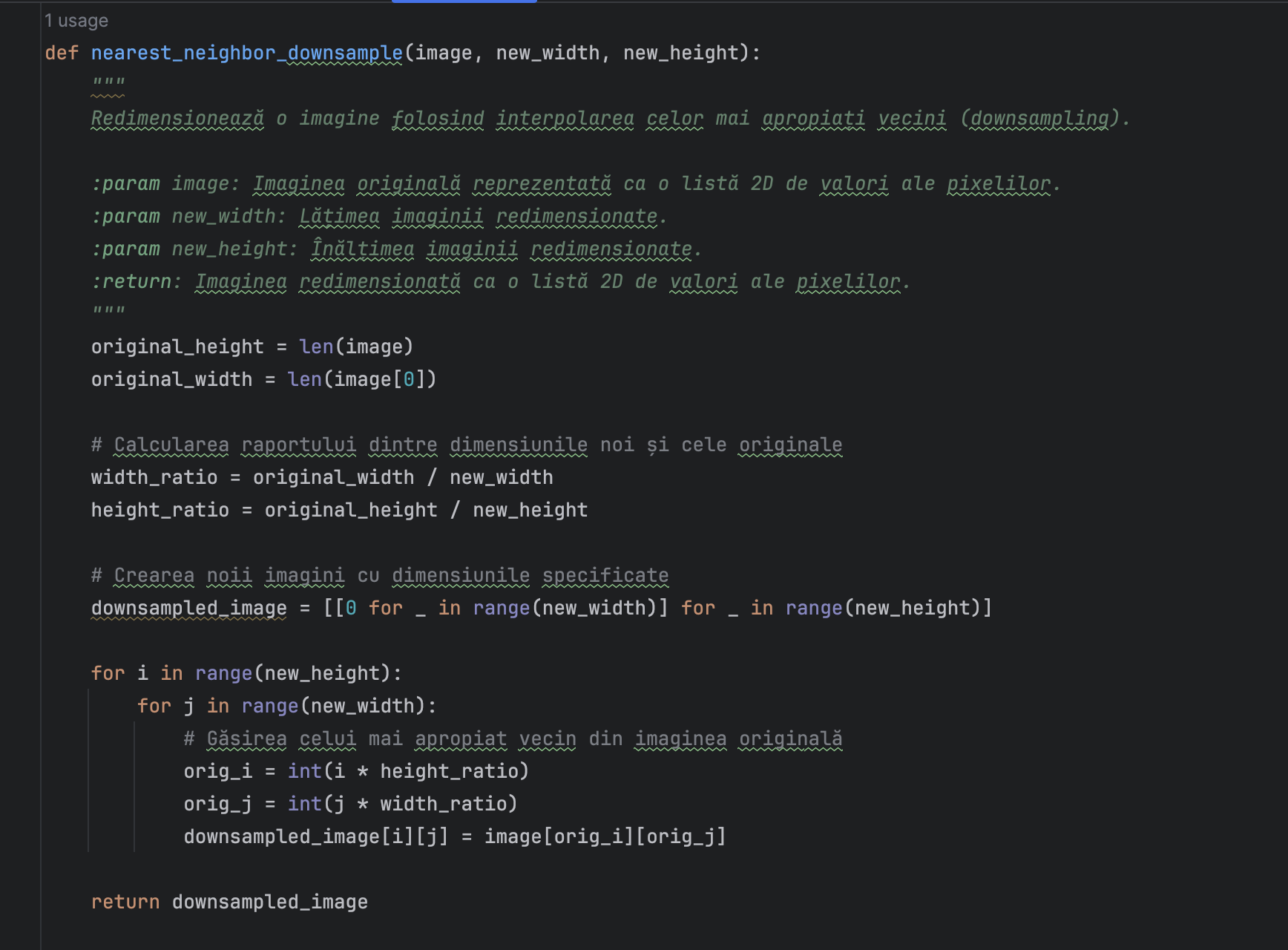


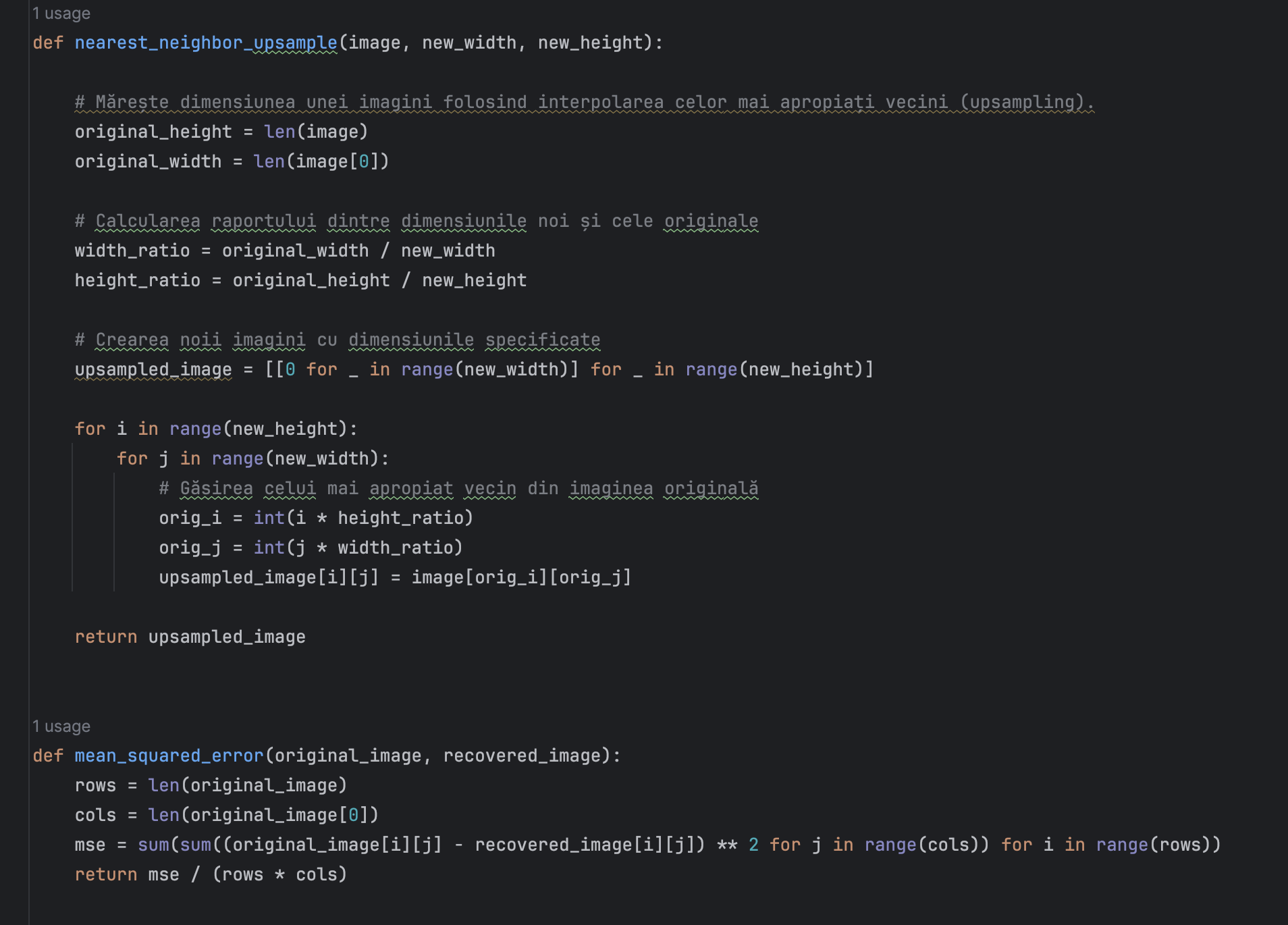


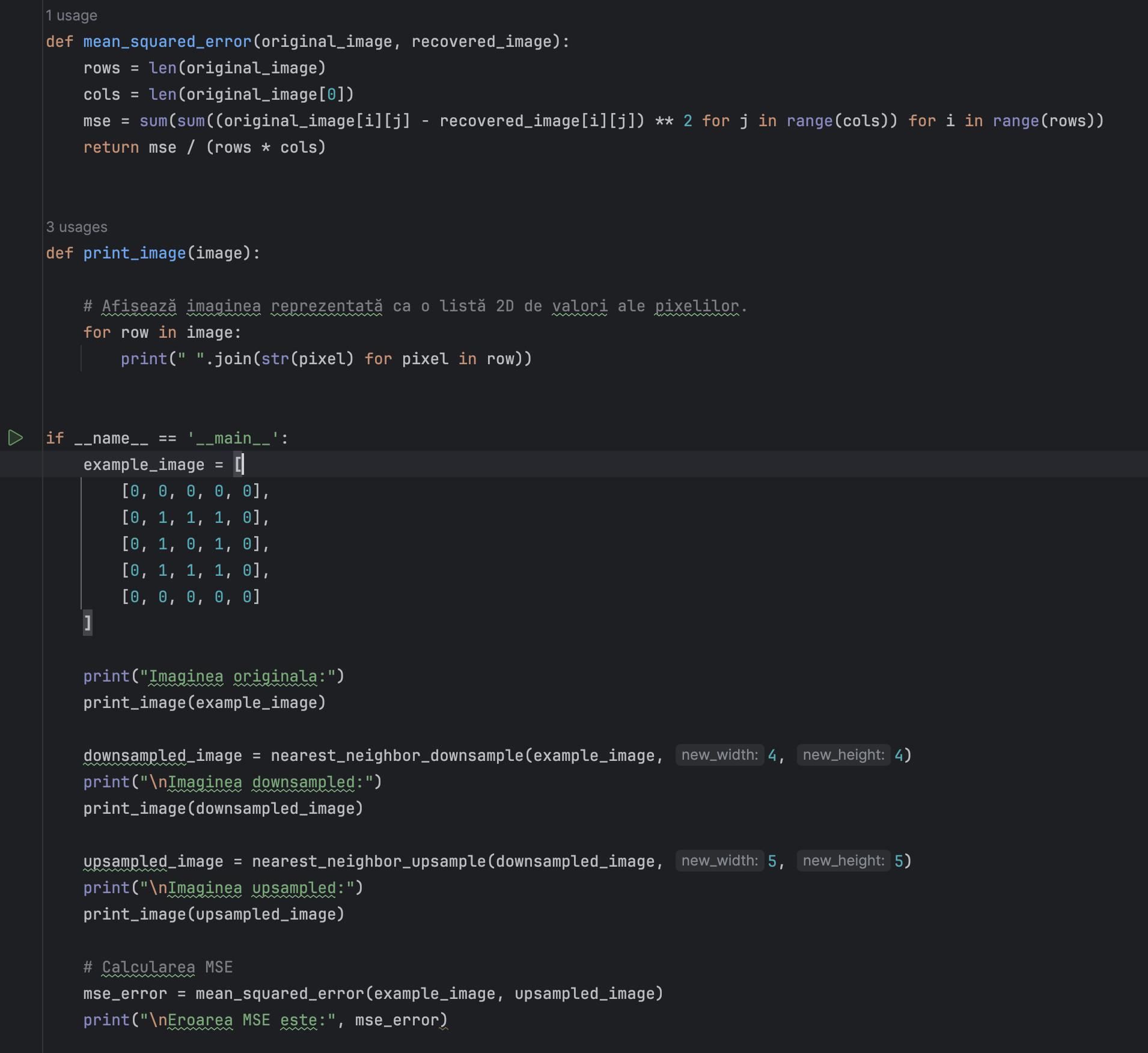


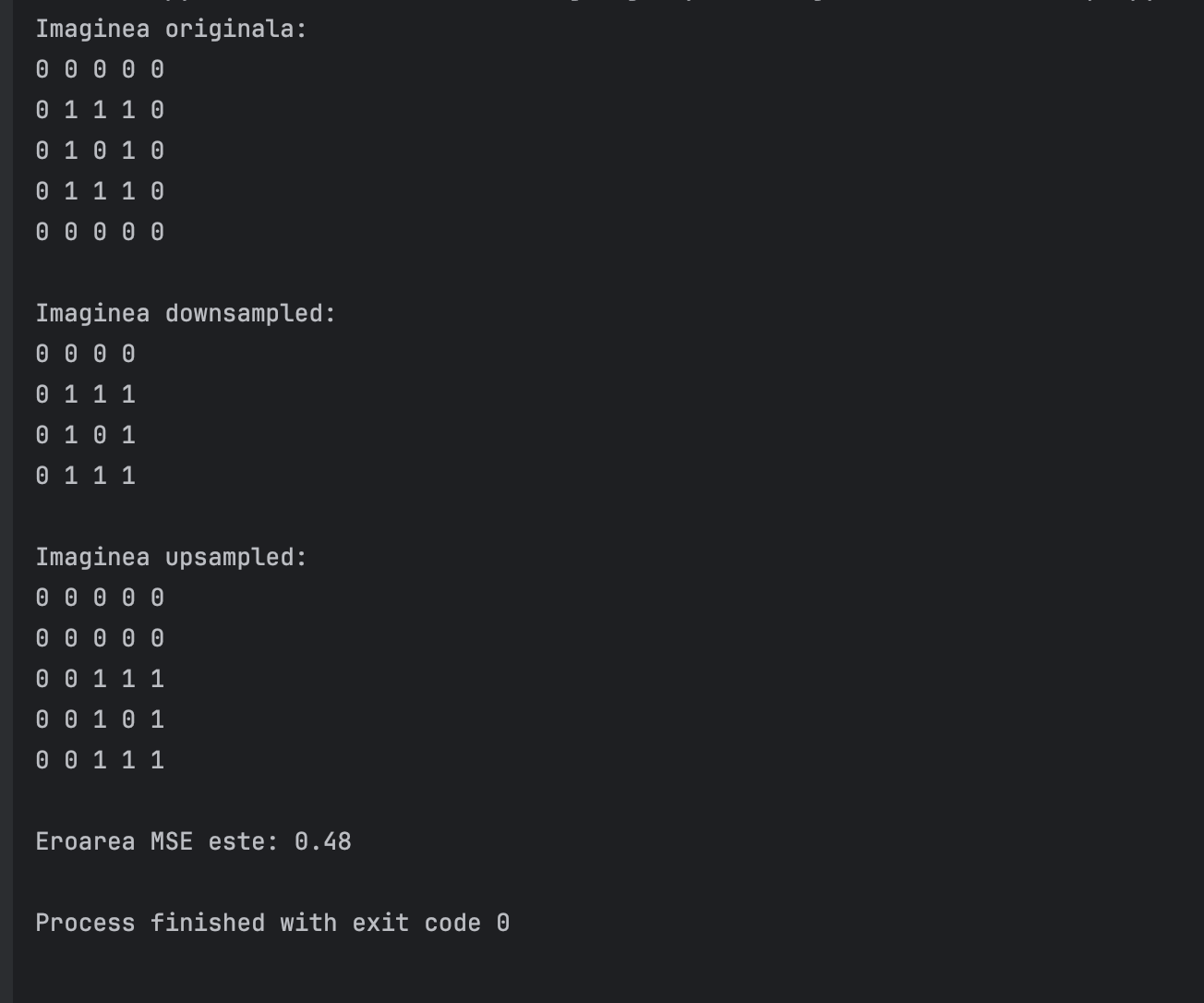
1. **Evaluarea sistematică a metodelor prezentate**

**10.1. Pentru Metoda Nearest Heighbor**

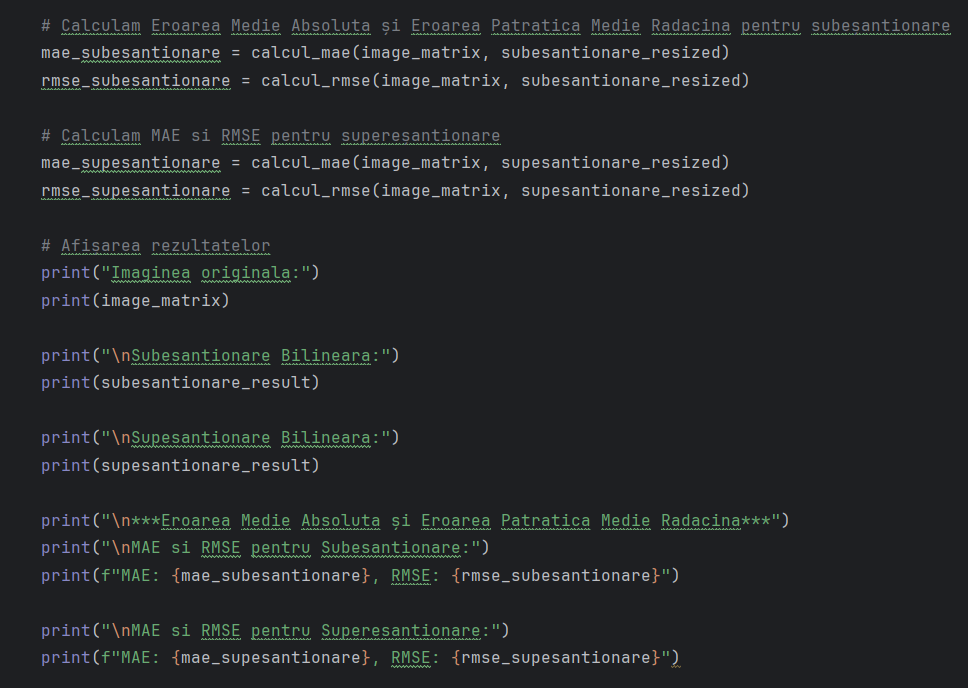


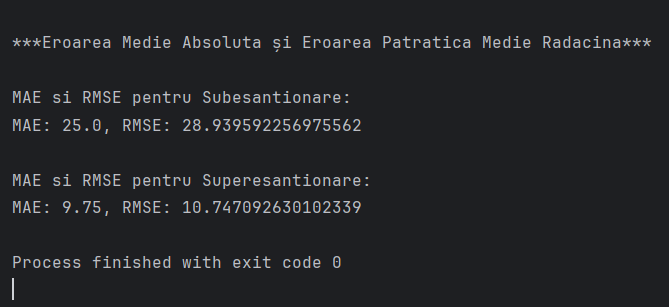






**10.2. Pentru Metoda Bilineara**





**Link cod:**

https://github.com/vladstoian98/semnale/blob/master/proiect\_redimensionare\_imagine.py