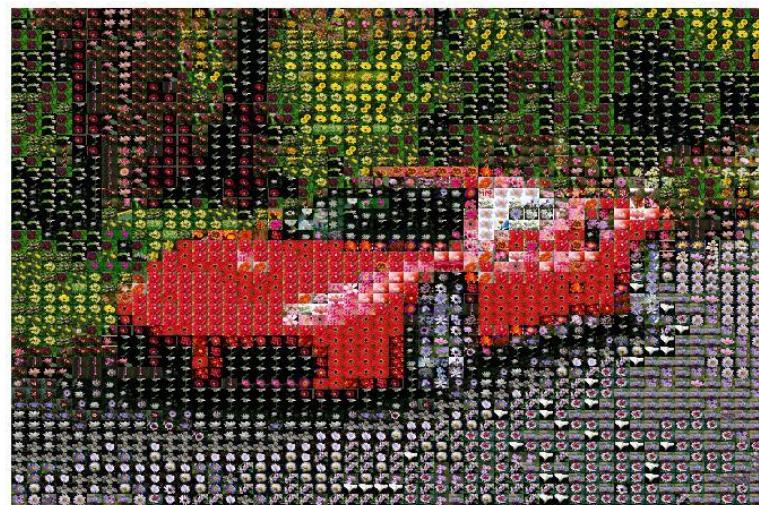
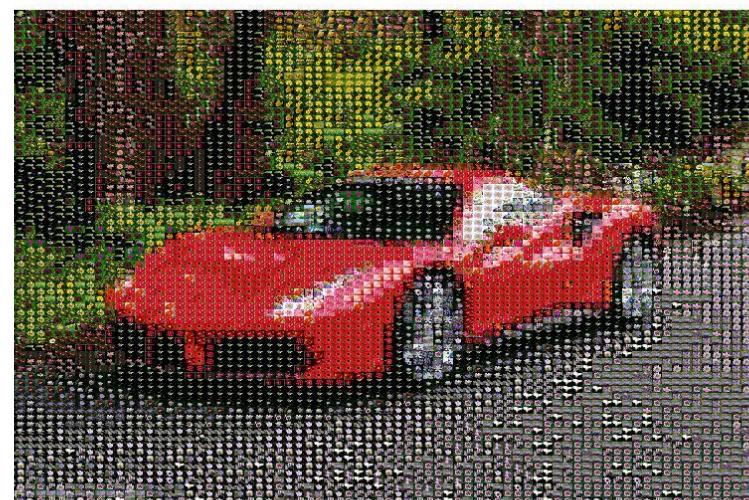
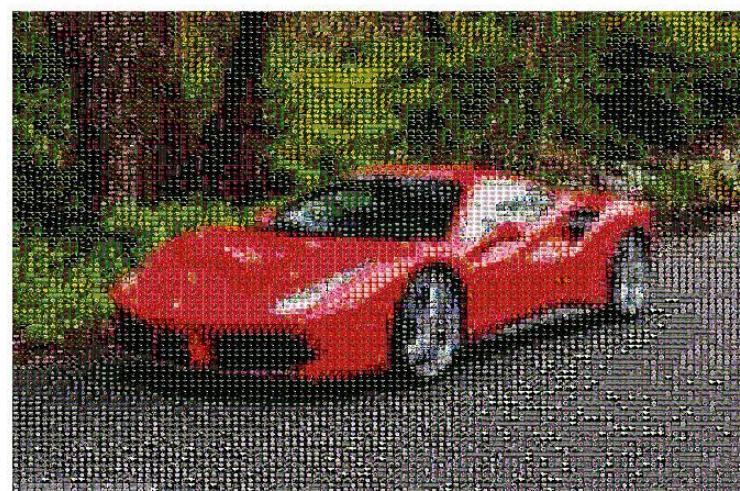


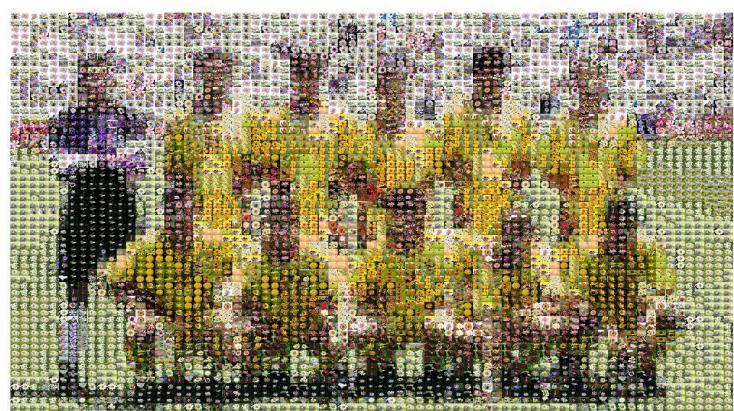
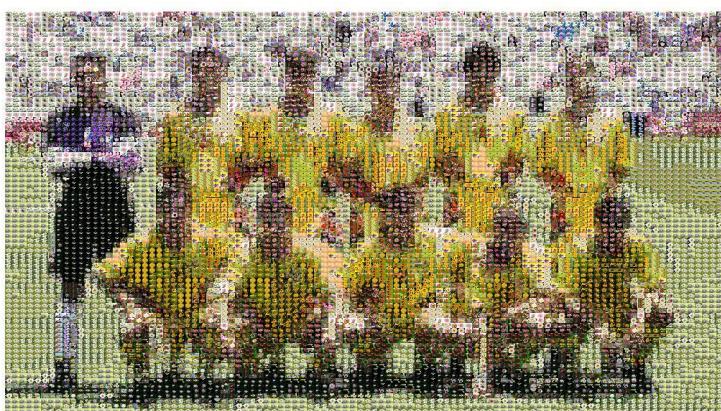
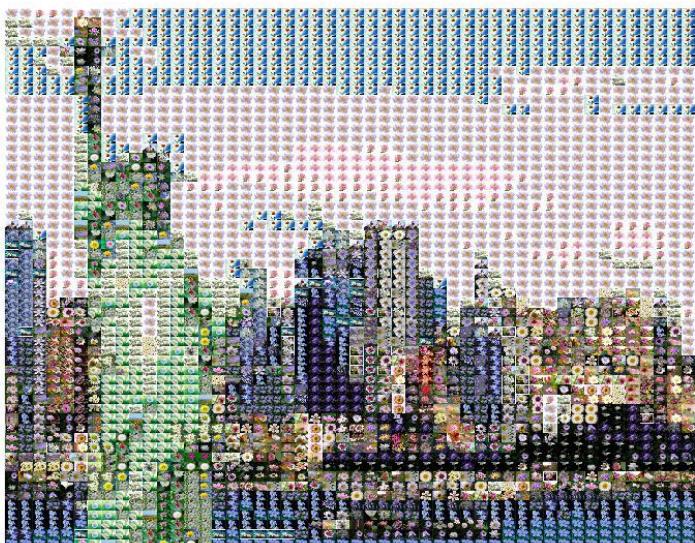
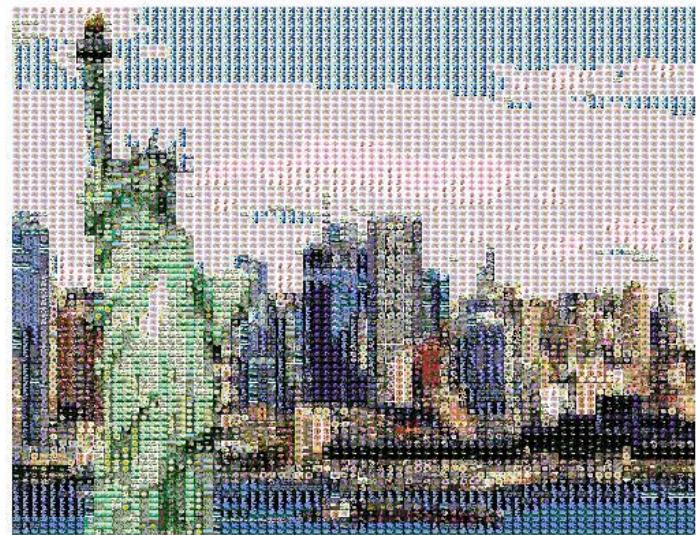
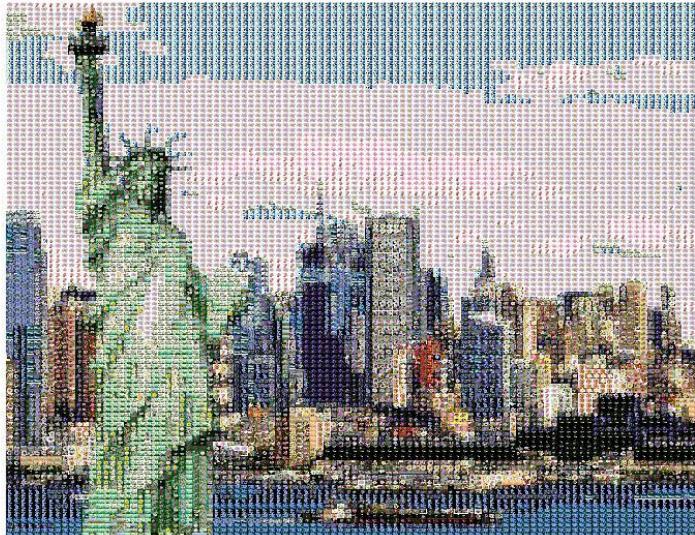


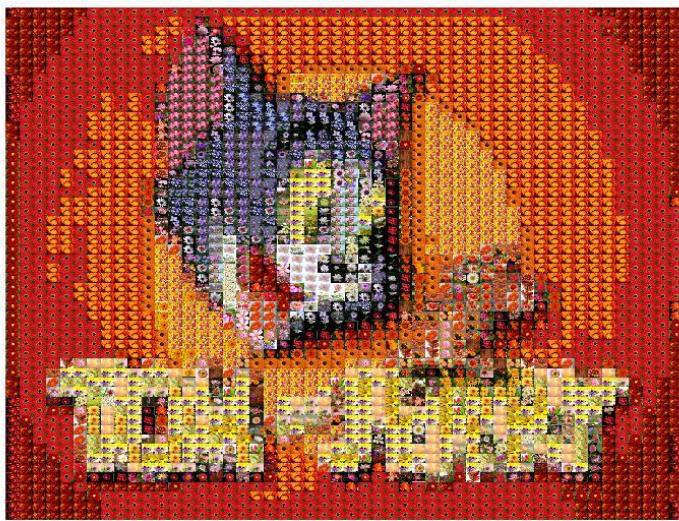
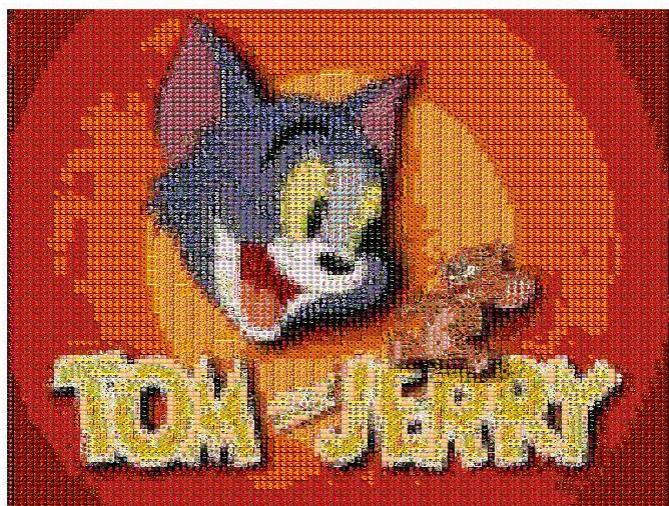
## Inteligentă Artificială - Proiect 1 Realizarea imaginilor mosaic

### 1.4 Predarea proiectului

(a) (2 puncte) mozaicurile obținute pentru imaginile din directorul '.../data/imaginiTest/' la rularea algoritmului vostru pentru diferite valori ale parametrului numarPieseMozaicOrizontala (100, 75, 50, 25) și criteriul distanței euclidiene dintre culorile medii comentând influența lor asupra rezultatelor obținute.

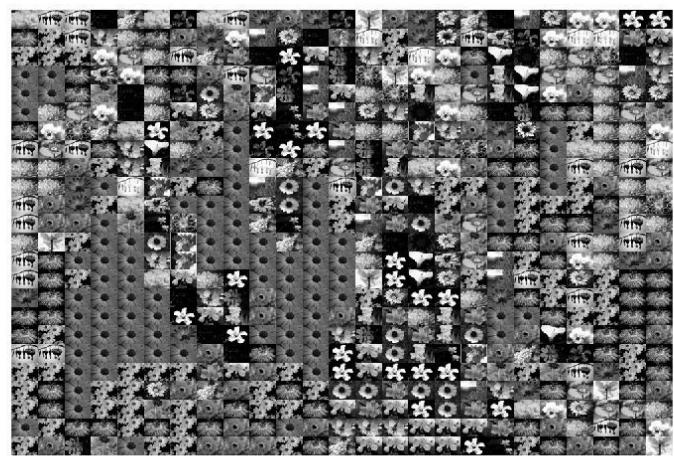
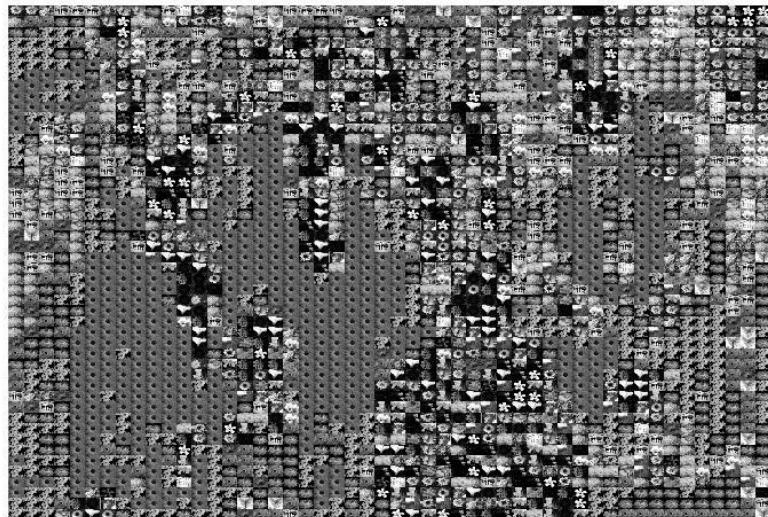
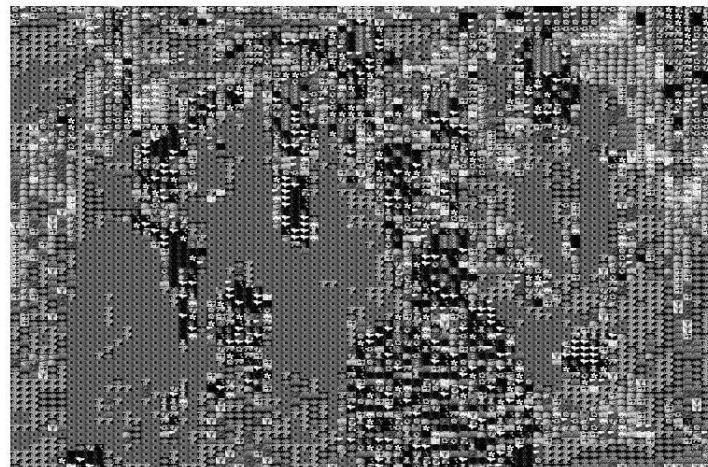
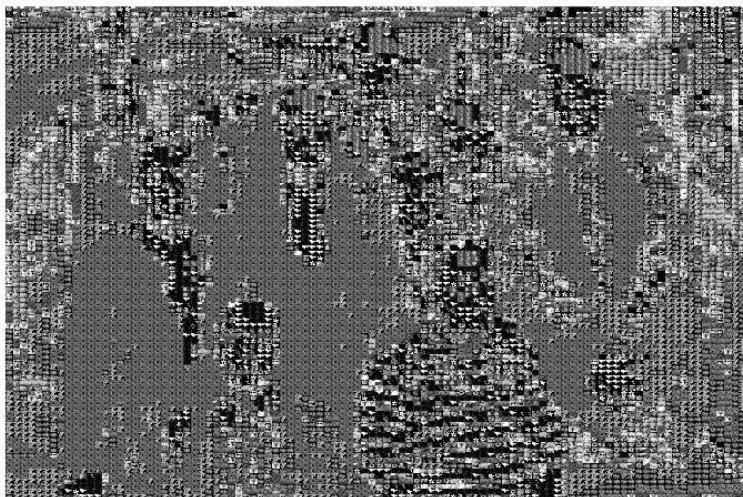




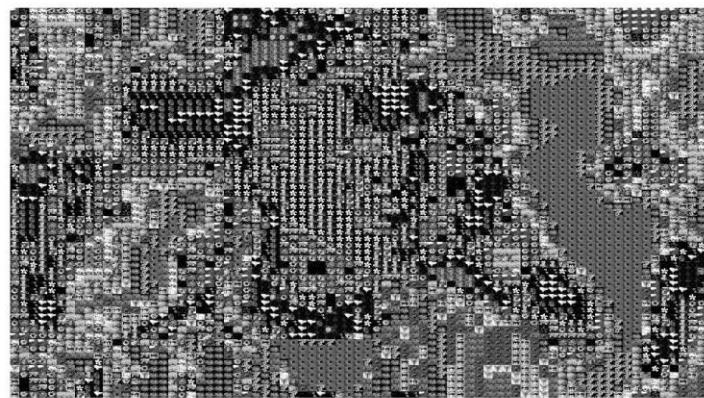
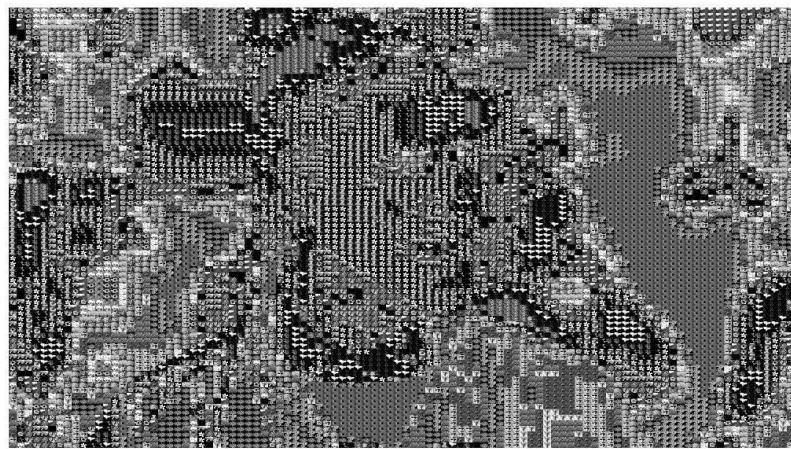


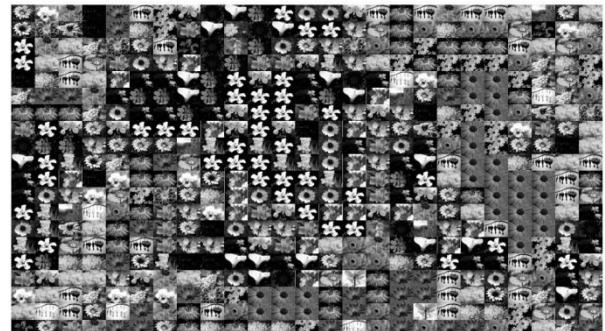
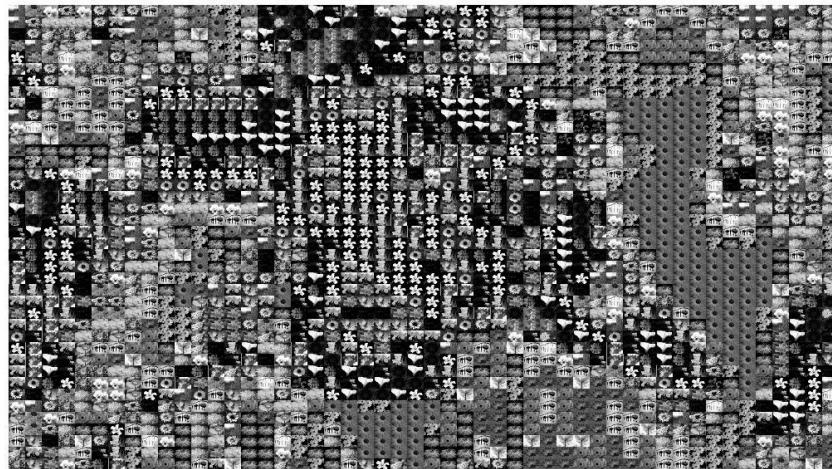


adams.JPG



obama.JPG





(b) (2 puncte) mozaicurile obținute pentru imaginile din directorul '.../data/imaginiTest/' pentru modul de aranjare 'aleator' și criteriu distanței euclidiene dintre culorile medii. Pentru rezolvarea acestui punct trebuie să scrieți funcția adaugaPieseMozaicModAleator.m. Pentru imaginea inițială din Figura 1 ar trebui să obțineți ceva similar cu imaginile din Figura 3.

Pentru rezolvarea problemei "pe baza distanței euclidiene dintre culorile medii"- folosind acest criteriu veți înlocui blocurile de pixeli din imaginea de referință redimensionată cu piese alese astfel încât culoarea medie a piesei să fie cât mai apropiată (în sensul distanței euclidiene) de culoarea medie a blocului.

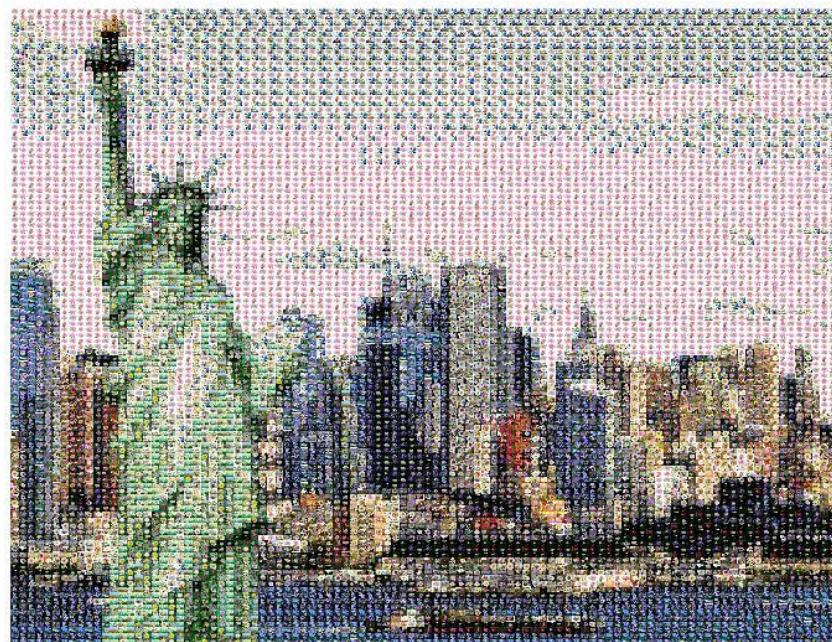
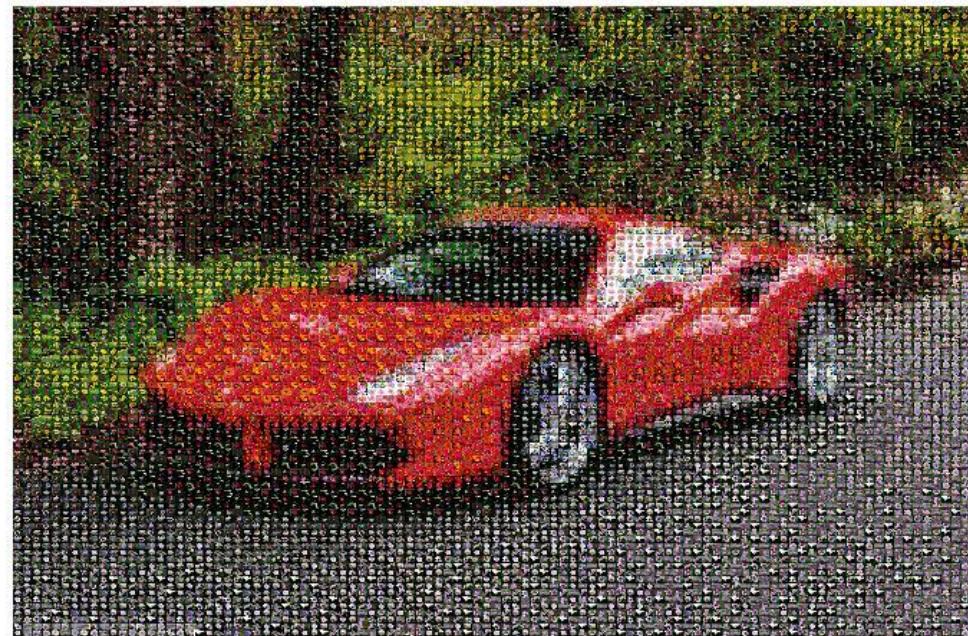
În acest caz, am luat fiecare canal de culoare și am facut media, urmând ca apoi să pun cele 3 canale sub forma unei matrice pe care am introdus-o formulă de calcul pentru distanță euclidiană. Apoi am extras o fereastră (de dimensiunea unei piese de mozaic) din imaginea redimensionată (am redimensionat imaginea pentru a facilita transformarea ei în mozaic), careia i-am facut media culorilor pentru a vedea ce imagine din cele 500 pe care le am, se potrivește cel mai bine pe imaginea mea de referință.

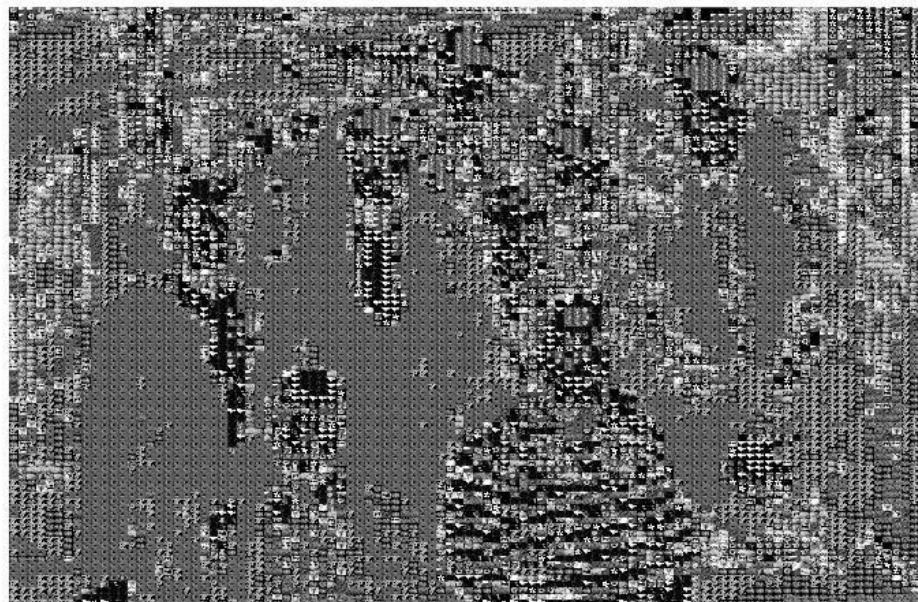
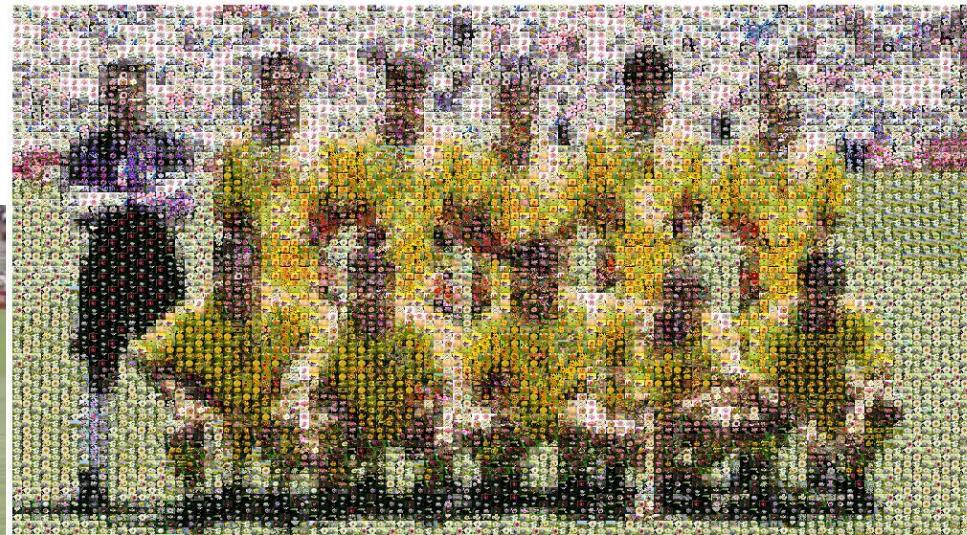
Pentru cazul "grayscale", am urmat aceeași metodă de rezolvare, singura diferență fiind la numărul de parametrii ai matricei ce reținează imaginea, imaginea alb-negru având doar parametrii pentru H și W față de cea color care mai are un parametru pentru canalul de culori RGB.

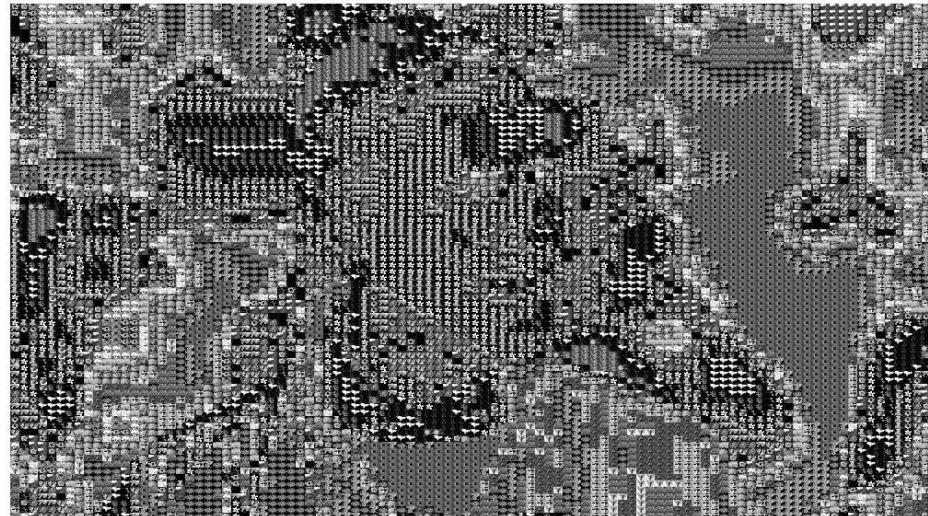


(c) (2 puncte) mozaicurile obținute pentru imaginile din directorul '.../data/imaginiTest/' astfel: implementați o modificare a funcției adaugaPieseMozaicPeCaroiaj.m astfel încât mozaicul obținut să aibă proprietatea că nu există două piese adiacente (stânga, dreapta, jos, sus) identice. Pentru imaginea inițială din Figura 1 ar trebui să obțineți ceva similar cu imaginile din Figura 4.

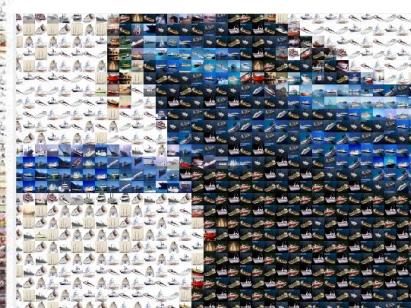
Pentru a rezolva acest subpunct, eu am avut de comparat fiecare piesă cu vecina ei. Și pentru a face lucrul acesta a fost nevoie să aflu și a doua distanță minimă. Cum minimul meu se încadrează între 0 și 255, m-am gândit că pentru a afla al doilea minim, valoarea trebuie să iasă din interval. Și am astfel 3 cazuri, cazul în care piesa se află pe prima linie, pe prima coloană sau nu se află nici pe prima linie, nici pe prima coloană.

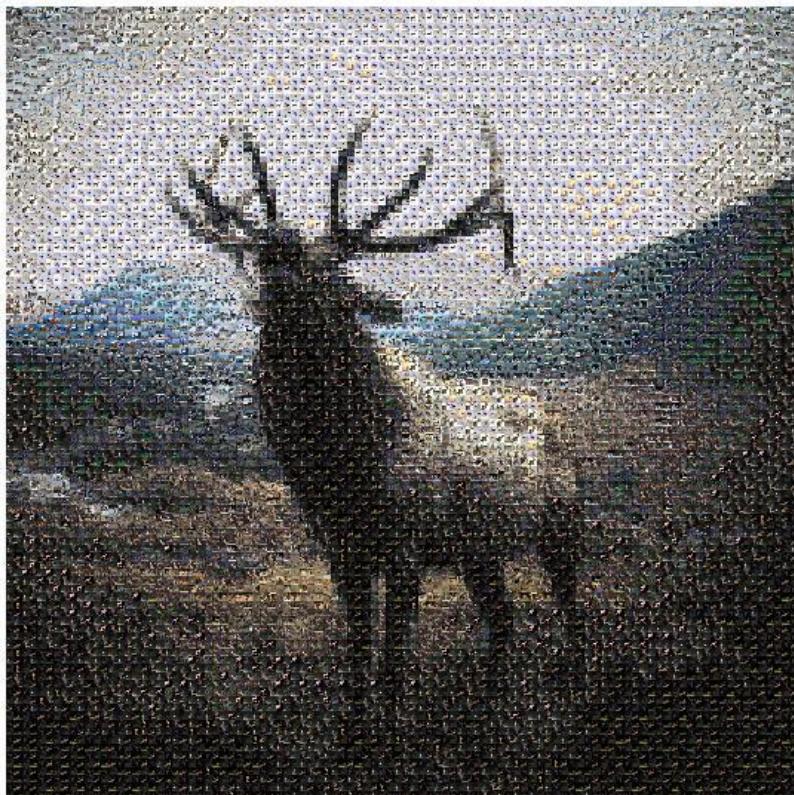
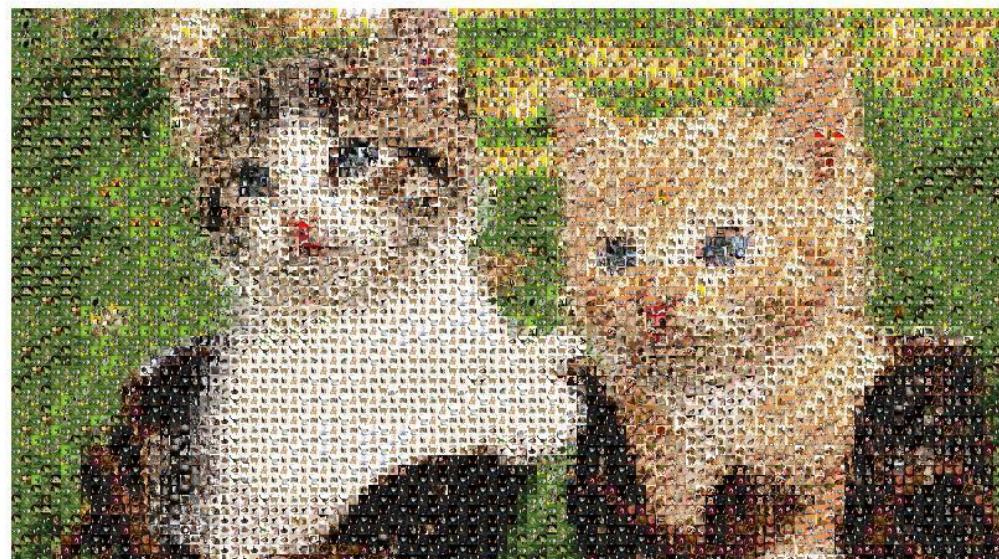






(d) (2 puncte): înlocuiți colecția furnizată de noi (cu cele 500 de piese ce reprezintă flori) cu o colecție proprie. O posibilitate ar fi să downloadați setul de date CIFAR - 10 de la adresa <https://www.cs.toronto.edu/~kriz/cifar.html>. CIFAR - 10 conține 60000 de imagini color de dimensiuni 32 x 32 pixeli cu obiecte din 10 clase: avion, automobil, etc. Realizați mozaicuri tematice, construind mozaicuri pentru imagini conținând obiecte din aceste clase cu piesele corespunzătoare: realizați un mosaic pentru o imagine cu un automobil folosind piese cu automobile. Includeți în pdf-ul vostru cel puțin 5 exemple de mozaicuri tematice.







Pentru a descarca si genera imaginile, m-am documentat pe internet în legătură cu acest lucru și am creat în matlab, un script în care am încărcat categoriile de imagini existente în arhiva CIFAR – 10.