1. Creati o functie care primeste ca parametrii o imagine si dimensiunea vecinatatii unui pixel . Pentru fiecare pixel din imagine comparati intensitatea lui cu intensitatiile pixelilor din vecinatatea a acestuia. Din operatia anterioara va rezulta pentru fiecare pixel o matrice cu valori binare. Liniarizati fiecare matrice sub forma de vector. Pentru imaginea data creati o histograma a vectorilor unici obtinuti in urma operatiei anterioare. Folositi aceste histograme pentru a antrena un model de machine learning. Alegeti voi valorile hyperparametrilor.
2. Creati o functie care calculeaza magnitudinea gradientului unei imagini date, , unde este gradientul imaginii pe directia x si este gradientul imaginii pe directia y. Definitia gradientului pentru o functie este:

Impartiti imaginea in regiuni de dimensiune care nu se suprapun si folositi-va de rezultatul pentru a gasii primele regiuni din imaginea originala cu magnitudinea medie cea mai mare. Pastrati doar aceste regiuni in antrenarea unui model de machine learning. Alegeti voi valorile hyperparametrilor.

1. Similar cu exercitiul anterior calculati magnitudinea gradientului unei imagini, dar determinati si directia acestuia . Folosindu-va de aceasta directie comparati fiecare pixel din magnitudinea gradientului cu vecinii corespunzatori. Creati o noua matrice care sa contina valorile magnitudinilor pentru pixelii a caror magnitudine este mai mare decat ale ambilor vecini sau 0 in caz contrar. Folositi imaginile rezultate intr-un algoritm de machine learning.
2. Impartiti fiecare imagine in regiuni distincte si binarizati-le conform compararii de la exercitiul 1. Concatenati vectorii binari rezultati si folositi-va de distanta Hamming implementati metoda celor mai apropiati vecini pentru un k ales de voi.
3. Folosind histogramele de la exercitiul 1, antrenati un model SVM cu functia kernel intersectie.