Proiect CUDA

**Enunt problema:**

Realizati un proiect in care se foloseste framework-ul de CUDA.

Problema pe care am ales-o noi este aplicarea unui filtru pe o imagine, iar filtrul ales de noi este un filtru albastru.

**Solutia abordata:**

Ca si limbaj de programare am folosit Python si modulul Numba pentru Cuda. Partea de cuda in Python este mai simpla decat in C++, in ideea ca am folosit annotarea @vectorize de functioneaza in urmatorul fel:

* Primeste tipurile de date ce vor fi folosite, dar si tipul de date ce va fi returnat; aceste tipuri trebuie sa fie scalare, si sa faca parte din libraria NumPy
* Primul parametru al adnotarii @vectorize este un string ce descrie signatura functiei; in cazul nostru, avem o functie ce returneaza un Int32, si primeste ca parametru 3 valori de tipul Int32, deci string-ul va fi: “int32(int32, int32, int32)”
* Pentru a paraleliza aceasta functie folosind CUDA, se va adauga un al doilea parametru, si anume: target = “cuda”
* Astfel, in momentul in care programul este rulat, functia noastra va fi compilata si trimisa catre GPU

Structura solutiei noastre este urmatoarea:

* Citim imaginea folosind libraria OpenCV2
* Parcurgem linie cu linie matricea de pixeli corespunzatoare imaginii, pentru a crea vectorii de pixeli albastri, verzi si, respectiv, rosii
* Apoi, apelam functia adnotata cu @vectorize, ce calculeaza noile valori ale pixelilor
* La final, reconstruim imaginea, folosind valorile noi ale pixelilor
* Imaginea rezultata este salvata, ulterior, intr-un fisier

**Testare:**

* Am observat ca programul CUDA este mai rapid de 8.2 ori decat rularea pe CPU (pentru o rezolutie de 1536x2048 pixeli)

**Rezultate:**

****

****