

Paralelna realizacija metoda za obradu slike pomoću NVIDIA CUDA tehnologije

Anabel Dautović, Ita ?, Lana ?, Valeria ?, Marko Haralović, Dino Babić

May 31, 2024

Sažetak

Sažetak od stotinjak riječi spominje najvažnije teme koje su obrađene u daljnjem tekstu. Tu bi vjerojatno ukratko trebalo napisati o implementacijama koje smo napravili

1. Uvod

Uvod sadrži kratki opis područja, zašto je ono važno i koje su primjene. Mislim da bi tu trebalo pisati o području obrade slike, zašto je važno i koje su njegove primjene.

2. Opis područja i aktivnosti u svijetu

Opis područja je detaljniji prikaz područja uljučujući teoretske osnove i primjene. U svakom tehničkom izvještaju ili članku uobičajeno je uz svako navođenje rezultata istraživanja i primjena razmatranih metoda navesti i poznate istraživačke grupe ili laboratoriji u svijetu, odnosno referencirati se na objavljene radove, npr. piramidalne sheme za brzo bogaćenje pokazale su se iznimno uspješne.

Mislim da bi ovdje trebalo pisati o razvoju grafičkih kartica i njihovoj primjeni u području obrade slike, koje su razlike i prednosti u odnosu na CPU. Možda malo predstaviti i NVIDIA kao kompaniju, zašto se ona istakla i navesti neke druge proizvođače grafičkih kartica i zašto oni nisu toliko popularni kada je u pitanju primjena grafičkih kartica u računarske svrhe. Također, možda i detaljnije opisati kako uopće rade grafičke kartice.

3. Pregled implementacija metoda za obradu slike pomoću NVIDIA CUDA tehnologije

Ovdje bi za svaku metodu opisao što ona radi, dakle što na primjer radi Canny edge detector i kako to radi, usporedio vremensku razliku između naše implementacije na CPU i GPU, ali i ako OpenCV ima implementaciju tog algoritma onda bih dodao i njega u vremensku usporedbu. Također, treba provjeriti da naša implementacija radi jednako dobro, znači samo provučemo istu sliku kroz našu i OpenCV implementaciju i prikazemo da su rezultati jednaki ili da se vrlo malo razlikuju.

3.1. Usporedba matričnog množenja na CPU i GPU te primjena nad transformacijama slike

Opis metoda slijedi...

3.2. Implementacija metoda za detekciju rubova u slici

Opis metoda slijedi...

3.3. Implementacija algoritma grupiranja s K srednjih vrijednosti za segmentaciju slike

Opis metoda slijedi...

3.4. Implementacija neuronske mreže za klasifikaciju rukom pisanih znamenaka

U ovom poglavlju razmotrit ćemo implementaciju neuronske mreže za klasifikaciju rukom pisanih brojeva. Obje implementacije napisane su u programskom jeziku Python. Verzija koja se izvodi na CPU zasniva se na biblioteci NumPy, dok se verzija koja se izvodi na GPU zasniva na biblioteci CuPy.

Za treniranje modela korišten je MNIST skup podataka, koji sadrži 60,000 slika rukom pisanih znamenaka. Svaka slika je veličine 28x28 piksela. Prilikom učitavanja skupa podataka, slike su povećane na veličinu 128x128 piksela jer je originalna veličina slika premala da bi se uočile prednosti izvođenja na grafičkim karticama. Naime, treniranje modela s originalnom veličinom slika bilo je sporije na grafičkoj kartici nego na procesoru. Iako se matricne operacije daleko brže izvode na grafičkim karticama, prenošenje podataka na grafičku karticu vremenski je skupo. U slučaju kada imamo malo podataka, vremenski je isplativije izvršiti operacije na procesoru nego prenositi podatke na grafičku karticu i natrag.

Model u ulaznom sloju ima 16,384 neurona. Skriveni sloj sadrži 30 neurona, dok izlazni sloj ima 10 neurona jer toliko imamo mogućih klasa. Za treniranje modela koristili smo stohastički gradijentni spust, dok smo kao aktivacijsku funkciju koristili sigmoidnu funkciju. Oba modela trenirali smo 10 epoha te su im točnosti na testnom skupu podataka bile približno jednake. Treniranje modela koji se izvršavao na procesoru je trajalo 1145 sekundi, dok je treniranje modela koji se izvršavao na grafičkoj kartici trajalo 67 sekundi. Grafička usporedba prikazana je na slici 1..

Vremenska ušteda je značajna. Verzija modela koja se izvršava na grafičkoj kartici također radi brže predikcije, te predikciju za istu sliku izvršava gotovo deset puta brže nego model koji se izvršava na procesoru.

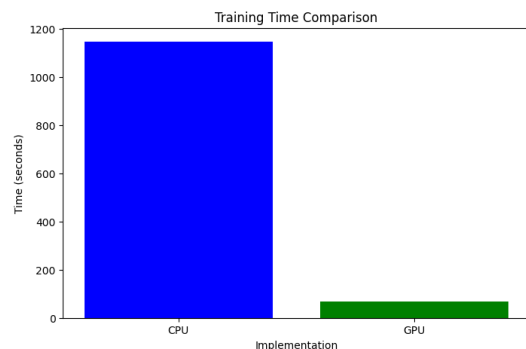


Figure 1.: Prikaz usporedbe vremena treniranja modela koji se izvršavaju na procesoru i na grafičkoj kartici.

3.5. Ostale metode za obradu slike

Opis metoda slijedi...

4. Zaključak

Zaključak treba istaknuti i diskutirati najvažnije rezultate iznesene u tekstu i sugerirati eventualne buduće trendove.

References

- [1] E. R. Davies. Machine vision: Theory, Algorithms, Practicalities. Academic Press, London, 2. edition, 1997.
- [2] J. Lampinen, S. Smolander, O. Silvén and H. Kauppinen. Wood defect recognition: A comparative study, Workshop on Machine Vision for Advanced Production, Oulu, Finland, p. 7, 1994.
- [3] T. Itoh i K. Koyamada. Isosurface generation by using extrema graphs, Proceedings of Visualization '94, 1994, pp. 77-83, IEEE Computer Society Press, Los Alamitos, CA.
- [4] J. User. A new efficient algorithm for making money, <http://www.rich.quick/>.