

PREPOZNAVANJE PISANIH OFLAJN MATEMATICKIH JEDNACINA

Filip Bajraktari
30. oktobar 2020.

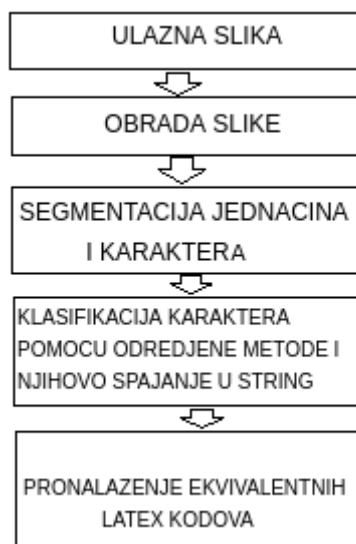
1 Uvod

Kako je tehnologija pocela naglo da se razvija tokom proslog veka, postala je jeftinija samim tim i pristupacnija za veliku vecinu ljudi. Danas imamo pametne telefone(smartphone) I licne racunare(Personal Computer) koji nam omogucavaju da skoro sve informacije dobijamo u veoma kratkom vremenskom intervalom u svega par klikova. S toga se sredinom proslog veka javio problem digitalizacije dokumenata iz arhiva I biblioteka. Bilo je logicno da nije moguće prekucavati sve sto je covecanstvo ikada stvorilo, pa su ljudi odluciti da to rese pravljenjem odredjenog algoritma koji ce to uraditi umesto njih. Za razliku od teksta, problem prepoznavanja pisanih matematickih jednačina je znatno tezi. Prvenstveno zbog kolicne mogucih znakova koji treba da se digitalizuju (kod prepoznavanja teksta imamo oko 30-ak slova I desetak interpunkciskih znakova, dok broj matematickih simbola dolazi I do par stotina). Takodje, reci za razliku od jednačina nemaju visedimenzionalnu ugnjezdenost kao sto su stepen i indeks. To predstavlja problem zato sto svaki znak jednačine mora se prvo izdovjiti, a zatim kasnije u programu klasifikovati. Zbog toga je prepoznavanje jednačina trenutno jedan od najtezih problema kojim se polje kompjuterske vizije bavi.

Postoje dva tipa prepoznavanja jednačina: oflajn, tj. sistem za staticko prepoznavanje I onlajn, tj. sistem za dinamicko prepoznavanje. Onlajn verifikacija se oslanja na digitalnu površinu, tj. uzima podatke u vremenu izvršavanja programa (run time), dok oflajn verifikacija se izvršava na već skeniranoj slici.

2 Metod

Ceo proces se sastoji iz cetiri glavna dela: pripremna obrada slike(pre-processing of image), segmentacija/izdvajanje jednačina sa slike I matematickih simbola iz tih istih jednačina, klasifikacija svakog karaktera pojedinačno I njihovo ponovno grupisanje u string(niz karaktera). U cetvrtom tj. poslednjem delu metoda mozemo da izvršimo pronalazenje ekvivalentnih latex kodova.



2.1 Obrada slike

Proces obrade slike obuhvata sve one promene I modifikacije na slici koje cine sliku pogodnijom za dalju obradu. Ovaj deo mozemo podeliti na tri procesa: pretvaranje RGB slike u Gray-Scale sliku(slika koja sadrzi iskljucivo nijanse sive), uklanjanje sumova I binarizacija slike.

2.1.1 Pretvaranje slike iz RGB u Gray-Scale

Ukoliko bismo radili sa RGB slikom morali bismo da za svaki piksel na slici vodimo racuna o kolicini crvene, zelene I plave boje. Zato uproscavamo sliku tako sto cemo tri vrednosti zameniti jednom vrednoscu, tj. Grayscale vrednoscu. Formula koju cemo koristiti je:

$$Y = 0.999R + 0.587G + 0.114B,$$

gde su Y,R,G,B redom oznake za kolicinu sive, crvene, zelene I plave boje.

2.1.2 Uklanjanje sumova sa slike

Kako je hardver nesto sto je covek stvorio, logicko je da nece uvek raditi u potpunosti kao sto je zamisljeno. Greske hardvera se nazivju sumovi. Kako sa razvojem u samoj elektronici te greske su sve manje I redje, ali I dalje su prisutne I mogu praviti probleme prilikom dalje obrade. Postoje dve najosnovnije vrste suma: Gausov sum I so I biber sum, ali je so I biber sum zanemarljiv na slikama sto I jeste tema ovog rada, te se njime necemo baviti. Bavicemo se samo izolovanjem Gausovog suma[4].

2.1.3 Binarizacija

Binarizacija je proces u kome se svakoj vrednosti Gray-Scale pixela dodeljuje jedna od dve moguće vrednosti, kao sto I ime aludira, te vrednosti su 0 I 1. Da bismo to uradili potrebno je da za svaki piksel znamo prag na osnovu koga cemo datom pixelu dodeliti adekvatnu vrednost. Prag ili granicnu vrednost mozemo izabrati tako da bude jedinstvena i unpred odredjena, ali postoje I pametnije metode koje ce nam pomoci da optimizujemo nasu binarizaciju kao sto je Otsov metod. Otsov metod se bazira na statistickim karakteristikama slike. Na ovaj nacin kompjuter umesto nas obavlja slozenu analizu I bira najoptimalniju globalnu granicnu vrednost.

2.2 Segmentacija jednacina I karaktera

Za pocetak bih koristio podatke koji u sebi sadrže samo jednu jednacinu, tj. jedan matematicki izraz (spreman sam, ukoliko završim ranije sa projektom, da unapredim moje istrazivanje time sto cu ubaciti taj jedan dodatan korak – segmentacija jednacina).

Da bi izdvojili tu jednu jednacinu, potrebno je da nadujemo gornju I donju granicu(upper and lower bound), tj. potrebno je da nadujemo neke dve linije pixela na slici takve da je svaki element te jednacine izmedju te dve linije. Metod koji cemo koristiti je kompaktna horizontalna projekcija, tj. za svaki red piksela racunacemo koliko ima crnih pixela I pronaci odgovarajuc prag(threshold) koji ce nam sluziti za odredjivanje da li je data linija pixela deo jednacine ili nije. Prag racunamo po formuli:

$$T = \Sigma \max(X_i) / n,$$

gde je X_i ukupan broj crnih pixela u datoj liniji, a n ukupan broj horizontalnih linija slike koja se trenutno obrađuje.

Segmentacija karaktera je proces koji nam služi da sliku jednacine podelinom na više manjih slika na kojim će se nalaziti po jedan simbol(promenljiva, znakovi racunske operacije, slova grckog alfabeta,..). Metod koji ćemo koristiti se naziva analiza povezanih komponenti. Ovaj metod se u kompjuterskoj viziji koristi za pronalazenje povezanih delova na binarnoj slici, zato je binarizacija sastavni deo preliminarne obrade slike. Svaki pixel se granici sa još četiri pixela(ne racunajući granicne pixele), tj. tako binarizovanu sliku možemo da predstavimo kao bestezinski graf sa disjunktним komponentama. Algoritimima kao što su DFS I BFS se lako mogu izdvojiti komponente [3]. Svaka slika određene komponente se ostavlja dalje na klasifikaciju.

2.3 Klasifikacija karaktera I njihovo spajanje u string

U datom istraživanju koristio bih dve metode klasifikacije karaktera: CNN(convolutional neural network) I neku drugu metodu. Ovaj deo ostavljam namerno prazno, zato što smatram da ću posle radionice imati bolje razumevanje o ovoj temi pa će I samim tim ovaj deo biti bolje napisan, što I jeste cilj(A malo je I do nedostatka vremena).

2.4 Pronalazenje ekvivalentnog latex koda

Pronalazenje odgovarajućeg latex koda možemo izvršiti u složenosti algoritma $O(n)$. Pretpostavimo da će se koeficijenti nalaziti uvek pre promenljive uz koju stoje, a stepeni posle promenljive. Prvi korak tog algoritma bi bila grupacija jednocifrenih brojeva u visecifrene, na primer "123" → 123, a nakon toga bi se elementi jednacine zamenjivali redom odgovarajućim izrazima.

3 Baza podataka

Baza podataka nam služi da bismo dobili detaljne eksperimentalne rezultate, jer par test primera nije dovoljan za ovakav tip istraživanja. Koristimo dve vrste baza podataka. Prva će služiti za trening karaktera [2]. Koristimo Kaggle-ovu bazu podatak za pisane matematicke simbole. Dok će druga baza podataka zapravo služiti za testiranje tačnosti prepoznavanja samih jednacina. Zbog nedovoljnog broja test primera kod nekih baza ili privatnih baza podataka, trenutno još nisam našao odgovarajuću.

4 Cilj istraživanja

Cilj istraživanja je prepoznavanje offline pisanih matematickih jednacina, a zatim određivanje njihovih ekvivalentnih latex kodova. Time bismo drastično ubrzali proces izrade naučnih radova koji u sebi sadrže kompleksna izracunavanja. Pored toga morao bih I poredio tačnost metoda (jedna će biti konvolucione neuronske mreže a drugu ću osmisliti nakon radionice iz masinskog učenja, imam par ideja ali sam ostavio taj deo za kasnije kada mi neke stvari budu bile jasnije). Tačnost prepoznavanja karaktera bih računao po formuli:

$$\text{Tačnost od karaktera} = \frac{\text{Broj tačno prepoznatih karaktera}}{\text{Ukupan broj karaktera}} * 100\%$$

kao tačno prepoznatu jednacinu bih računao onu kod koje su svi karakteri dobro prepoznati I tačnost prepoznavanja jednacina bih računao po slicnoj formuli:

Tacnost od jednacina = Broj tacno prepoznatih jednacina / Ukupan broj jednacina * 100%.

5 Hipoteza

Prva hipoteza je da ce CNN(convolutional neural network) imati vecu tacnost u odnosu na tu drugu metodu koju budem izabrao.

Druga hipoteza je da ce obe metode dostici tacnost prepoznavanja karaktera vecu od 90%.

6 Literatura

- [1] Md Bipul Hossain, Feroza Naznin, Y.A. Joarder, Md Zahidul Islam, Md Jashim Uddin. Recognition and Solution for Handwritten Equation Using Convolutional Neural Network. Information & Communication Engineering Islamic University, Kushtia, Bangladesh
- [2] Handwritten math symbols dataset, Kaggle, Over 100 000 image samples
- [3] Connected Components in an undirected graph, GeeksforGeeks
- [4] Gaussian noise, Wikipedia