

Detekcija vegetacije uz prometnice koristeći konvolucijske neuronske mreže

Toni Antunović, Mladen Kukulić, Una Pale

Sveučilište u Zagrebu,
Fakultet elektrotehnike i računarstva
Zavod za elektroničke sustave i obradu informacija

Siječanj 2015.

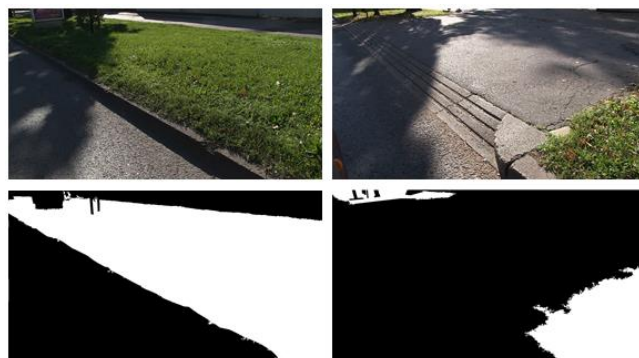
1 Sažetak

Znanost i tehnologija danas se kreću u smjeru automatizacije i autonomnosti sustava. Detekcija vegetacije uz prometnice jedna je od niza ideja kojima se želi ostvariti autonomnost kretanja vozila u prometu bez prisutnosti vozača. Analizom slike snimljene okoline potrebno je utvrditi koji dijelovi slike su vegetacija, a koji nisu. Zbog složenosti problema korištena je duboka konvolucijska neuronska mreža koja je trenirana na velikom broju primjera. Nakon toga cilj je bio ispitati kvalitetu mreže na nekoliko slika koje su dobivene kao rezultat mreže. Na taj način omogućeno je stvaranje vizualnog dojma o kvaliteti mreže.

2 Uvod

Snimanjem okoline kamerom potrebno je moći utvrditi što je vegetacija, a što nije. Ovaj zadatak nije moguće ostvariti samo obradom i analizom slike jer ne postoje specifičnosti koje je se nalaze na svakoj slici. Iz tog razloga krenulo se u smjeru neuronski mreža koje su sposobne same prepoznati specifičnosti na

temelju velikog broja primjera. U ovom slučaju zbog kompleksnosti zadatke broj primjera je veoma velik (oko 500000) te su stoga bile nužne duboke konvolucijske neuronske mreže kako bi obrada mogla biti u nekom konačnom vremenu. Treniranje i testiranje potrebno je vršiti na grafičkom procesoru kako bi trajalo što kraće.



Slika 1. Primjeri slika i pripadnih maska

Složenost problema sastoji se u tome da vegetacija ne znači uvijek samo „zelena trava“. Veliki je broj varijacija koje su moguće; npr. različito godišnje doba čime vegetacija mijenja boju, prisutnost lišća u jesen, sjene, kora drveća, raznobojno cvijeće itd. Također uz rub ceste često su parkirani auti ili se u prolazu snime i ljudi. Iz toga razloga boja nije dovoljno

dobar kriterij za prepoznavanje vegetacije.

Za treniranje mreže potreban je velik broj slika što različitijih mogućih situacija te također ručno označene maske. Maske su željeni rezultati mreže, odnosno jednodimenzionalne crno bijele slike na kojima je vegetacija označena sa 1, a sve ostalo sa 0.

Primjeri nekoliko slika i pripadajućih masaka prikazani su na slici 1.

3 Konvolucijske mreže

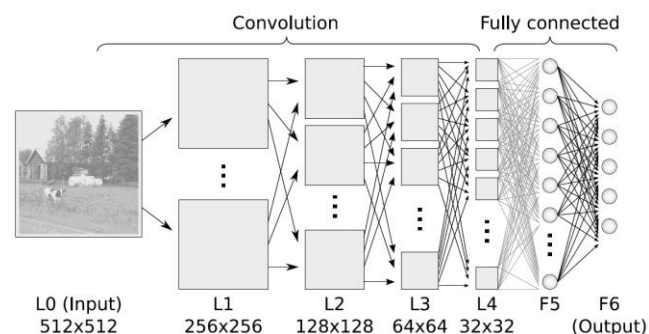
3.1 Duboke konvolucijske mreže

Duboke mreže su mreže sa mnogo skrivenih slojeva između ulaznog i izlaznog sloja. Konkretno u našem slučaju se koriste duboke konvolucijske neuronske mreže koje rade na konvoluciji i sažimanju slike. Na kraju same mreže se radi potpuno povezana mreža, koja je u stvari perceptron. Značaj ove mreže je velik jer pomaže da se izvade ključne značajke za klasifikaciju svakog pojedinog piksela.

Na slici 2. prikazana je tipična struktura jedne duboke neuronske mreže. Na ulazu može biti jedna monokromatska slika ili višekanalna slika u boji. Zatim slijede naizmjenice konvolucijski slojevi i slojevi sažimanja (engl. pooling). Na samom kraju se nalazi nekoliko potpuno povezanih slojeva (klasični perceptron) koji su dvodimenzionalni, uključujući i izlazni sloj. Tipična duboka konvolucijska mreža ima destak slojeva. Konvolucijski slojevi i slojevi sažimanja imaju dvodimenzionalne "neurone" koji se nazivaju mapama značajki (engl. feature maps) koji u svakom sloju postaju sve manjih dimenzija. Konvolucijski slojevi su povezani s prethodnim slojevima te učenjem

prilagođavaju težine veza. Slojevi koji sažimaju ne uče nikakve težine.

Ove mreže je lakše za trenirati od ostalih dubokih metoda učenja, također je moguće naći više značajki, koje se propuštaju kroz filtere, koji praktički jesu slojevi duboke konvolucijske neuronske mreže. Također zahtijevaju relativno malo predobrade. U nekim slučajevima se pokazalo da rade bolje od čovjeka. No često im se zamjera slabo shvaćanje što se ustvari događa unutar skrivenog sloja.



Slika 2. Tipična struktura duboke konvolucijske neuronske mreže

3.2 Caffetool

Korišten je alat Caffe. To je alat koji vrti treniranje duboke neuronske mreže umjesto nas. Dakako da bi se koristio alat Caffe, moramo prilagoditi ulazne podatke u odgovarajući format (LMDB). Nakon toga moramo definirati mrežu; imenovati slojeve, definirati veličine te međusobnu povezanost (u .prototxt formatu). Potrebno je predati lokaciju testnih i validacijskih setova slika.

Također je potrebno definirati parametre treniranja (solver.prototxt), odnosno koliko je maksimalni broj iteracija, nakon koliko iteracija radi validaciju, kolika je stopa učenja itd. Nakon treniranja mreže, potrebno ju je testirati.

4 Slike

4.1 Predobrada slike

Dvjestotinjak slika sa različitim prizorima koristi se kao set za treniranje. No slike prije toga potrebno je napraviti nekoliko koraka kako bi se dobio konačni set podataka za treniranje. Slike su dimenzija 1080*1920 px što je preveliko, te ih je potrebno smanjiti u obje dimenzije na četvrtinu odnosno na dimenzije 270*480 px.

Ulaz u mrežu nije cijela slika već neki izrezani okvir od $DX*DY$ piksela oko središnjeg $[cx,cy]$ piksela. Koristeći $DX*DY$ okolnih piksela mreža treba zaključiti da li je središnji piksel vegetacija: postoje dva moguća izlaza, 1 ili 0. Za svaki taj okvir potrebno je znati točan odgovor te ga također predati mreži. Točan odgovor je vrijednost tog istog $[cx,cy]$ piksela na maski.



Slika 3. Prikaz okvira, središnjeg piksela te problema rubova

Kako bi treniranje bilo što bolje, potrebno je sa svake slike izrezati veliki broj okvira sa nasumičnih mjesta te ih zajedno sa točnim rezultatom pohraniti u datoteku u obliku:

```
crops_train/3_1.jpeg 0  
crops_train/3_2.jpeg 1
```

...

Taj format zove se LMDB. Bitno je da u testnim podacima bude podjednak broj pozitivnih i negativnih primjera, te da oni budu uniformno raspoređeni.

S obzirom da je potrebno moći odrediti i status (vegetacija ili ne) i rubnih piksela potrebno je proširiti sliku kako bi oko njih postojalo također $DX*DY$ piksela. Proširivanje je napravljeno zrcaljenjem. Odnosno sa svake strane slike rubnih $DX/2$ ili $DY/2$ piksela je zrcaljeno te su slike spremljene. Tek nakon ove obrade je napravljeno izrezivanje okvira i pretvaranje u LMDB format.

Prikaz okvira, središnjeg piksela te problema rubova može se primijetiti na slici 3. Na slici 4. prikazana je slika nakon zrcaljenja rubova.



Slika 4. Slika nakon zrcaljenja rubova

S obzirom da je većina mreža konstruirana za vrijednosti ulaznih podataka u rasponu $[0,1]$, a vrijednosti boja piksela na slikama su u rasponu $[0,255]$ bilo je potrebno skalirati vrijednosti svih piksela.

Za navedenu predobradu napravljena je Matlab skripta koja iz zadanog foldera sa svim slikama i foldera sa pripadajućim maskama radi sljedeće:

- smanji slike
- zrcali rubove
- skalira vrijednosti na $[0,1]$
- izrezuje na slučajnim mjestima okvire
- sprema okvir kao posebnu sliku
- ime slike okvira i rezultat sprema u LMDB datoteku

Osim što su okviri izabrani sa nasumičnih mjesta na slici, oni su naknadno izmiješani. Ovaj postupak bilo je potrebno napraviti za slike koje služe za treniranje, te također za slike koje će služiti za validaciju unutar treniranja. Nije potrebno to raditi za slike za konačno testiranje jer je kod njih potrebno to provesti za svaki piksel slike kako bi se kasnije mogla rekonstruirati slika koja predstavlja rezultat mreže.

4.2 Postobrada slike i testiranje

Provjera kvalitete treniranja mreže napravljena je na nekoliko slika iz test seta. Napravljena je Matlab skripta koja uzima jednu po jednu sliku iz foldera sa test slikama, smanjuje ih na 270*480 px, skalira na vrijednosti [0,1], proširuje na jednak način kao i u predobradi te zatim za svaki piksel slike po redu izrezuje okvir te da šalje u mrežu.

Mreža izbacuje 2 broja; vjerojatnost da je rezultat 1 (vegetacija) i vjerojatnost da je rezultat 0 (ostalo). Nakon što su rezultati mreže pohranjeni u niz za svaki od piksela te slike potrebno je odrediti graničnu vrijednost iznad kojeg postotka će se reći da je to stvarno 1. Određivanje postotaka radi se primjenom ROC krivulja.

Nakon određenog optimalnog postotka, niz sa izlaznim vjerojatnostima mreže se uspoređuje sa tom graničnom vrijednošću te se konačno odlučuje da li je mreža rekla 0 ili 1 za pojedini piksel. Nakon toga rekonstruirana se dobivena slika. Vizualno je moguće usporediti sliku dobivenu mrežom i masku koja predstavlja točno rješenje.

Da bi se odredila numerička točnost mreže, ovaj postupak se ponavlja za nekoliko slika i pri svakoj slici se izračuna broj točno pogodjenih

vrijednosti piksela (oduzmu se rezultat mreže i maske i prebroji broj 0). Točnost mreže je omjer broja točno pogodjenih piksela i ukupnog broja piksela. Točnost se računa za svaku sliku, a na kraju se izračuna prosječna točnost svih slika. Sve se to zapisuje u .txt datoteku.

5 Rezultati

Pri pokušaju evaluacije rada svake od isprobanih neuronskih mreža korištenjem Matlab wrappera za Caffe za svaki dani ulaz (izrezan okvir oko [cx,cy] piksela slike za testiranje) smo dobili isti izlaz iz mreže. Na razne načine smo pokušavali riješiti ovaj problem, a to je uključivalo preispitivanje svega do sada urađenog.

Najprije smo mijenjali arhitekturu mreže nekoliko puta, ali svaki put je problem ostajao prisutan. Zatim smo pokušali sa skaliranjem ulaznih podataka na interval [0,1]. Potom smo vidjeli da je isti problem bio prisutan i kod drugih ljudi (sa Interneta) te smo pokušali primijeniti savjete kojima su oni riješili isti, a to je bilo smanjenje stope učenja i miješanje skupa za treniranje, ali ni to nije pomoglo.

Posljednji pokušaj je bio dodatno uravnoteživanje skupa za treniranje na način da ima točno 50% pozitivnih primjera za učenje. Prethodni način kojim je kreiran skup za treniranje je nasumično izrezivao okvire iz slika ne obraćajući pažnju na njihovu oznaku, tako da je bio znatno veći broj negativnih primjera za učenje. Treniranje nad ovakvim podacima uz stopu učenja 0.0001 uzrokovalo je da funkcija gubitka kroz iteracije učenja blago oscilira oko vrijednosti 0.70 za što pretpostavljamo da je bio lokalni minimum iz kojeg se uz tako nisku stopu učenja funkcija gubitka nije mogla

pomaknuti. No, povećanjem stope učenja na 0.1 i ponovnim pokušajem treniranja mreže funkcija gubitka blago pada, ali i dalje oscilira.

Jia_, E. Shelhamer, J. Donahue, S. Karayev, UC Berkley
[3.]<http://caffe.berkeleyvision.org/> (pristup 20.12.2014.)

6 Zaključak

Proučene su duboke konvolucijske neuronske mreže te je konstruirano nekoliko njih u pokušaju rješavanja problema detekcije vegetacije uz prometnice.

Napisane su Matlab skripte za pred i post obradu slika. Predobrada vrši određene operacije nad ulaznim slikama te od njih stvara set podataka za treniranje i validiranje. Postobrada uzima cijele slike i dio po dio (okvir po okvir) iz provlači kako bi dobila rezultat mreže. Cilj je iz tih rezultata rekonstruirati sliku koja je što sličnija maski, te na taj način osim brojčane točnosti mreže i vizualno doživjeti kvalitetu mreže.

Nažalost do konačne strukture mreže koja uspješno detektira vegetaciju se još uvijek nije došlo. Pojavili su se problemi da loss funkcija prilikom treniranja mreže ne pada kontinuirano nego oscilira. Također nakon završetka treniranja mreže kada se želi testirati mreža, ona vraća gotovo iste vrijednosti za svaki ulazni okvir koji joj se preda, odnosno daje isti rezultat neovisno o ulazu koji joj se preda.

Rad na problemu će se nastaviti.

7 Literatura

- [1.] „Raspoznavanje objekata dubokim neuronskim mrežama“, Vedran Vukotić, Diplomski rad, FER 2014
- [2.] „Caffe: Convolutional Architecture for Fast Feature Embedding“, Yangqing