

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6120

**DETEKCIJA ZUBA U
ORTOPANTOMOGRAMIMA**

Miljenko Šuflaj

Zagreb, lipanj 2019.

IZVORNIK

Uvod

Problem detekcije zuba u ortopantomogramima je naizgled jednostavan problem riješavaju li ga ljudi. Osobe koje nemaju kvalifikacije, tj. nisu stomatolozi mogu iznimno lagano pronaći zube te ih razrediti znaju li neke osnovne stvari o ljudskim zubima. Problemi nastaju kada trebao uočiti zube u slikama loše kvalitete ili uočiti zube loše kvalitete. U tom slučaju čak će i stručnjaci s vremena na vrijeme završiti s češkanjem po tjemenu. Kao i kod svakog problema, ljudi su vični tražiti spas u računalima koja u trenutku pisanja ovog rada posjeduju zaista zavidnu snagu obrade te su zasigurno izum koji je najviše pomogao razvoju znanosti u 21. stoljeću. Problem uočavanja zuba je problem čiji su ulazni parametri vizualni podražaji. Znanost još uvijek nije standardizirala pretvorbe vizualnih signala u jednoznačne tekstualne podatke, pa se stoga moramo baciti u granu neuroračunarstva, koja za jednu od podgrana svojata računalni vid. Problem detekcije zuba u području neuroračunarstva može se raščlaniti na 2 potproblema; problem pronalaska zuba (*engl. regression*) i problem razredbe zuba (*engl. classification*). Ovi problemi se uobičajeno rješavaju konvolucijskim neuronskim mrežama (*engl. convolutional neural networks*) koje imaju sposobnost učenja damo li im dovoljno veliki skup označenih podataka. U neuroračunarstvu postoje još mnogi pristupi koje bismo mogli iskoristiti za rješavanje ovih problema, no u vrijeme pisanja ovog rada, konvolucijske neuronske mreže su de facto standard rješavanja problema nalik ovome.

Usprkos tome što smo već rano odabrali tehniku rješavanja ovog problema, konvolucijske neuronske mreže su vrlo općenita tehnika, korištena za mnogo različitih tipova problema predloženih vizualnim podražajima, pa ćemo, nakon što objasnimo osnovne pojmove potrebne za usvajanje osnovnog znanja o konvolucijskim neuronskim mrežama pričati i o različitim inačicama i arhitekturama sustava za pronalazak i razredbu zuba.

Sadržaj

Uvod	4
Sadržaj	5
Uvod u konvolucijske neuronske mreže	7
Umjetne neuronske mreže	7
Konvolucijske neuronske mreže	8
Konvolucijski sloj	9
ReLU sloj	11
Sloj združivanja	11
Poptuno povezani sloj	13
Sloj gubitka	13
Razmatrane arhitekture neuronskih mreža	14
Jednoprolazne i dvoprolazne neuronske mreže	14
Arhitektura RetinaNet	16
Komponente	17
Izlučivanje značajki	17
Razredbena konvolucijska neuronska mreža	18
Pronalazna konvolucijska neuronska mreža	19
Princip rada	20
Računanje gubitka	21
Razredbeni gubitak	21
Pronalazni gubitak	22
Zaključivanje	23
Postupak pripreme i interpretacije podataka	24
Rezultati	25
Usporedba s drugim arhitekturama srodnog zadatka	26

Zaključak	27
LITERATURA.....	28
NASLOV, SAŽETAK I KLJUČNE RIJEČI	29
DODATAK.....	30

Uvod u konvolucijske neuronske mreže

Konvolucijske neuronske mreže je jedna od mnogobrojnih inačica umjetnih neuronskih mreža (*engl. artificial neural networks*). Umjetne neuronske mreže su računalni sustavi koji oponašaju živčane sustave u mozgovima životinja. Specifičnost takvih sustava je što uče promatranjem primjera bez da su unaprijed uhodani u neki specifični postupak rješavanja zadanog problema. U pogledu računalnog vida, umjetne neuronske mreže uobičajeno uče promatranjem označenih objekata na foto ili videosadržaju dok se ne postigne zadovoljavajuća učestalost točnog zaključka za objekte koji se razmatraju.

Umjetne neuronske mreže

Općenito, umjetne neuronske mreže sastoje od mnoštva osnovnih jedinica koje nazivamo neuronima koji su analogni s neuronima u mozgu životinja. Svaka veza između neurona, nalik sinapsama, prenosi signale iz jednog neurona u drugi. Uobičajeno je da neuroni prenose realne brojeve, dok se izlaz neurona računa primjenom odabrane funkcije, obično nelinearne, uvrštavanjem sume ulaza. Veze između neurona nazivamo rubovi (*engl. edges*) te je uobičajeno da oni sadrže koeficijent koji podešava snagu emisije signala. Taj koeficijent se najčešće mijenja tijekom učenja. Osim toga, moguće je postaviti prag ispod kojega se signal neće poslati.

Neurone uobičajeno ne gledamo kao autonomne jedinice, već u pogledu slojeva. Slojevi se razlikuju po funkcionalnosti, oni na različite načine preoblikuju ulaze, a u osnovnom slučaju imamo 2 sloja; ulazni i izlazni sloj. Broj neurona u ulaznom sloju mora se poklapati sa brojem značajki ulaza, dok se broj neurona u izlaznom sloju mora poklapati sa brojem značajki izlaza. Konkretno, želimo li neku rastersku sliku veličine 32x32 razvrstati u slike mačke, psa, kornjače ili zeca, naša neuronska mreža imati će 1024 neurona u ulaznom sloju te 4 neurona u izlaznom sloju. Ograničenja osnovne arhitekture je što neuronska mreža sa samo ulaznim i izlaznim

slojem nema mogućnost rješavanja problema nelinearnih problema. Ovo je iz razloga što je neuronska mreža tada samo reprezentacija jedne linearne kombinacije parametara iz ulaznog sloja koji se preslikavaju u 4 vrijednosti. Zbog toga se uvodi pojam skrivenih slojeva (*engl. hidden layers*).

Skriveni slojevi su svi slojevi između ulaznog i izlaznog sloja. Koriste se kada je problem prekompleksan za jednu linearnu kombinaciju parametara. Dok bi nam, u računalnom vidu, mreža bez skrivenih slojeva mogla prepoznati neke vrlo primitive uzorke, trebali bi jedan skriveni sloj za detekciju rubova. Dodavanjem dodatnih skrivenih slojeva možemo dobiti sofisticiranije uočavanje složenijih značajki. Međutim, dodavanje skrivenih slojeva nije nešto što treba raditi preko mjere. Dodavanjem skrivenih slojeva povećavamo složenost, ali i specifičnost neuronske mreže. Previše neurona može rezultirati preprilagođenošću (*engl. overfitting*). Isto tako, premalo neurona može rezultirati nedovoljnom prilagođenošću (*engl. underfitting*).

Konvolucijske neuronske mreže

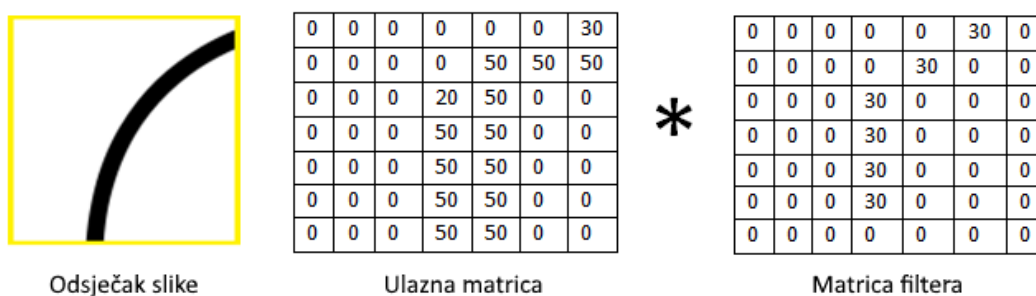
Konvolucijske neuronske mreže razlikuju se od tradicionalnih neuronskih mreža po tome što se tijekom učenja podešavaju vrijednosti tzv. filtera. Filteri su zapravo vektori realnih vrijednosti koji se učenjem podešavaju, a tijekom zaključivanja (*engl. inference*) konvoliraju sa ulaznim podacima. Filteri mogu biti različitih veličina, a u praksi su najuobičajeni 1x1 i 3x3 filteri.

Konvolucijske neuronske mreže su prve neuronske mreže učenja s učiteljem (*engl. supervised learning*) koje su postigle rezultate usporedive s ljudima za probleme koji se njima rješavaju.

Klasično, konvolucijske neuronske mreže su razred dubokih neuronskih šalji-naprijed (*engl. feed forward*) mreža. Uobičajeno je da imaju 1 ili više potpuno povezanih konvolucijskih slojeva. Također je uobičajeno da se težine (*engl. weights*) dijele između slojeva i združivanje (*engl. pooling*), tj. združivanje po maksimumu (*engl. max pooling*).

Konvolucijski sloj

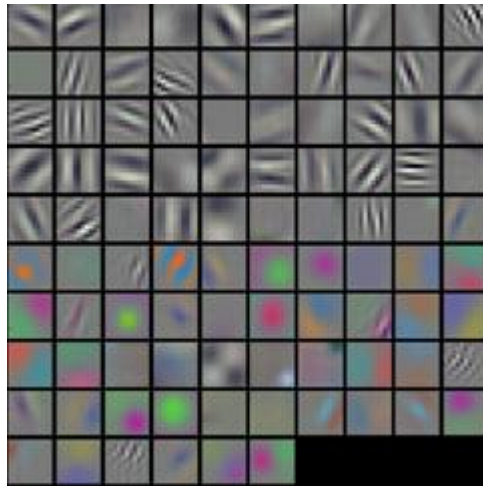
Konvolucijski sloj vrši konvoluciju određenog dvodimenzionalnog podatka s filterom konvolucijskog sloja. Učenjem mreže parametri filtera se podešavaju ne bi li dali što točniji zaključak o priloženom podatku iz baze za učenje. Konvolucijski filteri su obično matrice dimenzija $N \times N$, te u praksi uglavnom govorimo o 1×1 , 3×3 , 5×5 itd filterima. Ideja konvolucijskog sloja je preslikati dvodimenzionalne podatke, obično odsječak slike u jednodimenzionalnu vrijednost. Ovo se postiže konvolucijom matrica ulaza i filtera, tj. množenjem vrijednosti matrica po elementima (*engl. element-wise matrix multiplication*).



Slika 1: Rad konvolucijskog sloja

Na slici iznad izlaz glasi $50 \cdot 30 + 50 \cdot 30 + 50 \cdot 30 + 50 \cdot 30 + 20 \cdot 30 = 6600$

Korištenjem različitih filtera možemo dobiti različitu funkcionalnost konvolucijskih slojeva, tj. oni će zapažati različite stvari (npr. lijeve rubove, desne rubove, određene vrste krivulja itd.). Filteri vuku analogiju različitih vrsta češljeva, jer oni „češljaju“ ulazne podatke u oblike koji bolje ili lošije prikazuju neke značajke danih podataka. Moguće ih je vizualizirati kao sitne sličice:



Slika 2: Vizualizacija filtera

Neuroni u konvolucijskom sloju posloženi su u 3 dimenzije; širini, visinu i dubinu. Svaki sloj povezan je samo sa dijelićem sloja prethodnika. Taj djelić zovemo polje osjeta (*engl. receptive field*). Udruživanjem raznih konvolucijskih slojeva, nekih lokalno povezanih, nekih globalno povezanih, dobivamo konvolucijsku neuronsku mrežu.

Razlog zašto sada odjednom imamo lokalno, a ne globalno povezivanje je jer želimo da naši neuroni stvaraju najsnažniji odziv na prostorno lokalne uzorke. Slaganjem mnogo lokalno povezanih slojeva dobivamo nelinearne filtere koji veću sliku rekonstruiraju slaganjem manjih dijelova.

Finalno, prije smo spomenuli da se filtri ponavljaju jer se svakom neuronu u istom sloju pridružuju iste težine, tj. isti filtri. Ovo rezultira time da svaki neuron u takvom sloju biva pobuđen istim značajkama za svaku pobudu u svojem polju osjeta. Ovo omogućuje da uočavanje značajki ne ovisi o lokaciji na slici, tj. da se svaka značajka koju želimo uočiti može uočiti gdje god se nalazila na slici.

Ova tri svojstva omogućavaju bolju generalizaciju. Dodatno, dijeljenje težina smanjuje snagu obrade te broj parametara koje mreže treba podesiti, što smanjuje zahtjevnost cijelog algoritma te omogućuju treniranje većih mreža.

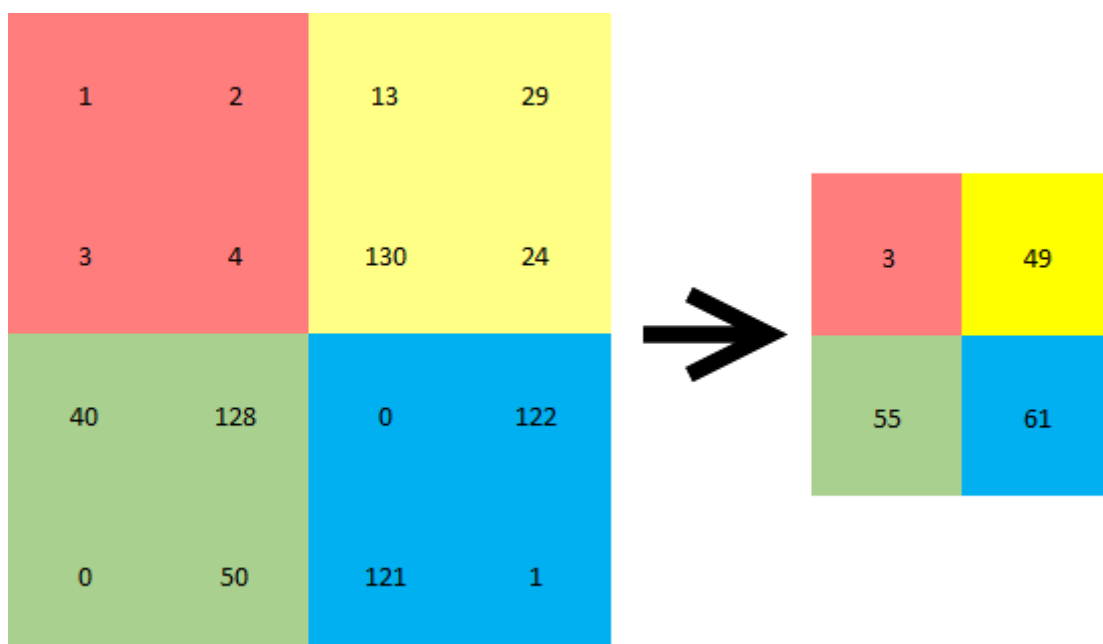
ReLU sloj

ReLU (engl. Rectified Linear Unit) je funkcija oblika $\max(0, x)$. Ova funkcija, kao što se može vidjeti izdaleka, ruši negativne vrijednosti u 0, dok čuva pozitivne vrijednosti onakvim kakve jesu. ReLU sloj u konvolucijskoj neuronskoj mreži povećava nelinearna svojstva funkcije odluke (engl. decision function), ali i cijele mreže bez izmjene podataka polja osjeta konvolucijskog sloja.

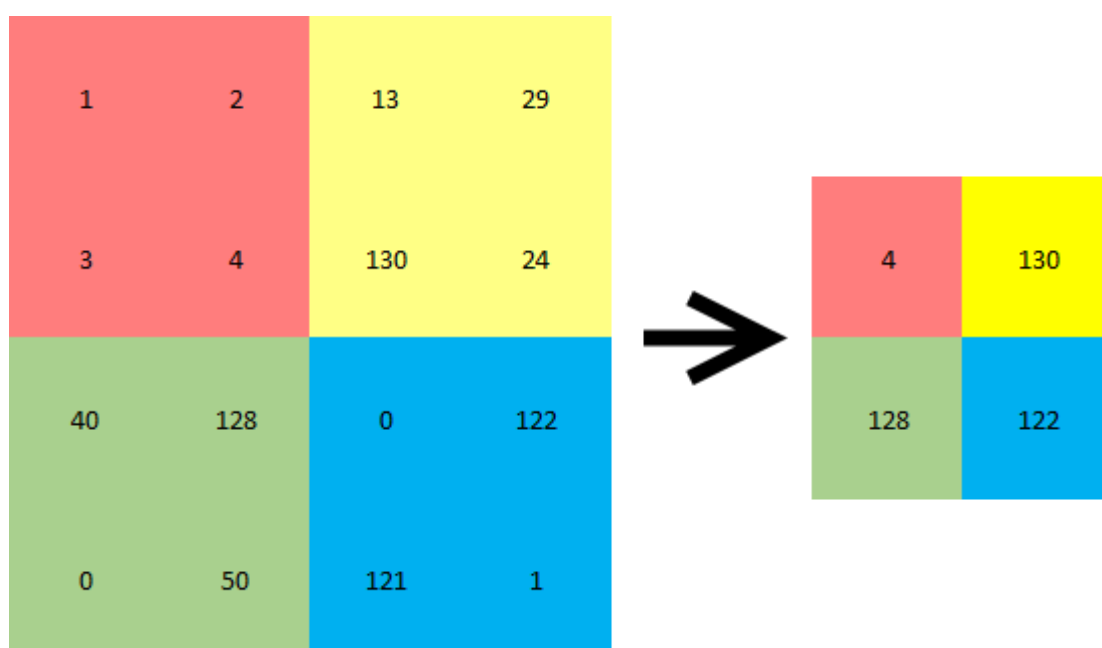
Za sloj iste funkcionalnosti kao ReLU postoje i druge poznate funkcije kao što su hiperbolni tangens ($\tanh(x)$) i sigmoid ($\frac{1}{1+e^{-x}}$). Razlog zašto konvolucijske neuronske mreže uglavnom biraju ReLU za funkciju uključanja (engl. activation function) je zato što se postiže nekoliko puta brže učenje bez značajnih kazni u smislu općenite preciznosti mreže.

Sloj združivanja

Slojevi združivanja vrše združivanje na podacima. Združivanje je jedna od metoda nelinearnog poduzorkovanja (engl. downsampling), čime možemo smanjiti dimenzionalnost podataka. Postoji nekoliko funkcija kojima možemo postići združivanje, npr. združivanje po prosjeku i združivanje po maksimumu. Iako bi laik rekao da je združivanje po prosjeku logičniji izbor za funkciju združivanja, u praksi se najviše koristi združivanje po maksimumu. Kada smanjujemo podatke, bitnije nam je sačuvati bitne podatke sa slike. Unaprijed pretpostavimo da je podatak bitniji što mu je vrijednost kojom je iskazan veća. Kad bi se radilo o doljezorkovanju slike s vizualnog aspekta, bolje bi bilo koristiti združivanje po prosjeku, no u slučaju konvolucijskih neuronskih mreža, pristup s združivanjem po maksimumu pokazao se boljim. Kao parametar pri združivanju koristi se veličina klizećeg prozora (engl. sliding window size), dvodimenzionalna vrijednost koja je uobičajeno 2x2, te korak (engl. stride), što je udaljenost koju će u sljedećoj iteraciji združivanja prozor prijeći kako bi došao do sljedećeg skupa vrijednosti koje treba združiti.



Slika 3: Združivanje s prosjekom, 2x2 prozor, korak 2



Slika 4: Združivanje s maksimumom, 2x2 prozor, korak 2

Motivacija kod združivanja nije samo jedna. S jedne strane, ahilova peta umjetnih neuronskih mreža je dugo vrijeme treniranja. Ono može biti takvo zbog veličine ulaznih podataka, broja iteracija ili epoha, te veličine same mreže. Cilj nam je

smanjiti podatke koliko god možemo, a opet sačuvati informaciju koja se nalazi u tim podacima koliko god dobro možemo.

Potpuno povezani sloj

Nakon slaganja nekoliko konvolucijskih i slojeva združivanja, konvolucijska mreža sadržavati će jedan ili više potpuno povezanih slojeva. Potpuno povezani slojevi su slojevi neurona koji primaju sve izlaze prethodnog sloja. U matematici bi potpuno povezanim slojevima bilo analogno matrično množenje, samo što u ovom slučaju možemo imati neki parametar koji će nam određivati pomak (engl. *bias*), realan broj koji ćemo zbrojiti i time dobiti izlaz neurona.

Sloj gubitka

Jedan od bitnih slojeva u konvolucijskoj neuronskoj mreži je sloj gubitka. Ovaj sloj je zaslužan za računanje pogreške u zaključivanje neke konvolucijske neuronske mreže. Kako je funkcija gubitka (*engl. loss function*) proizvoljno odabrana funkcija, nama je u interesu da ona što vjernije prikazuje inverz točnosti za problem koji pokušavamo riješiti našom konvolucijskom neuronskom mrežom. Ovisno o rasponu vrijednosti koje nam vraća ostatak mreže, prikladno je odabati Softmax gubitak (koristimo kad predviđamo K klasa s međusobno isključivim vjerojatnostima), Sigmoidalnu križnu entropiju (*engl. sigmoid cross-entropy*) kad predviđamo K klasa, svaka vjerojatnosti u rasponu $[0, 1]$, te Euklidski gubitak (*engl. Euclidean loss*) kad su nam vjerojatnosti za svaku od K klasa realni brojevi.

Razmatrane arhitekture neuronskih mreža

Prilikom rješavanja problema uočavanja zuba u ortopantomogramima, zbog toga što je računalni vid kao grana relativno razvijeno područje, mogli smo birati između puno vrsta neuronskih mreža. Na ovu temu postoji već nekoliko radova u kojem je korišten pristup dvoprolaznih neuronskih mreža (*engl. two pass neural network*), specifično arhitekture Faster R-CNN. U tom radu (treba citat) uspoređivana je arhitektura Faster R-CNN s drugim jednoprolaznim neuronskim mrežama (*engl. single pass neural network*), kao što su YOLO i SSD.

Jednoprolazne i dvoprolazne neuronske mreže

Razlika između jednoprolaznih i dvoprolaznih neuronskih mreža nalazi se u broju prolaza ili faza koje neuronska mreža tijekom učenja prođe. Specifično, ono što je posebno kod arhitekture Faster R-CNN je da ona koristi mrežu za preporuku područja (*engl. Region Proposal Network*) kako bi prvo izolirala interesantna područja na slici, a tek potom išla uočavati objekte te ih klasificirati.

Ovakav pristup je potreban jer u općenitom slučaju na slici se može nalaziti puno pozadine. Ovo će dovesti do nejednakosti klasa, tj. s obzirom na to da je na slikama iz baze za učenje najprisutnija pozadina, mreže će zanemarivati objekte koje treba uočiti i označiti. Ovo je rezultiralo redundancijom i lošom općenitom točnošću, pa se tvorci arhitekture Faster R-CNN doskočili potencijalnim uravnoteženjem primjeraka pozadine i razreda objekata koje želimo detektirati.

Jednoprolazne neuronske mreže ovo nisu radile, već su se jednostavno, nakon izlučivanja značajki, bacile na posao treniranja mreže. S jedne strane, zbog toga što su prilikom učenja radile manje posla, mreže su brže učile, no njihova je točnost bila manja, što zbog nejednakosti klasa, što zbog drugih problema, npr. razne skale objekata.

Donedavno, dvoprolazne neuronske mreže bile su *de facto* standard zbog neusporedivo bolje točnosti uz vrlo male kazne na performanse.

Kad smo krenuli rješavati problem uočavanja zuba na ortopantomografima, moje reference bile su upravo mreže bazirane na Faster R-CNN-u. Inicijalno, plan je bio istestirati takvu mrežu, no kada sam preuzimao gotovu implementaciju kao relativno zelen novak u području računalnog vida, poruka da je Faster R-CNN zastario pojavila mi se na ekranu uz poveznicu na navodno bolju arhitekturu.

Arhitektura o kojoj pričamo naziva se RetinaNet. Tijekom istraživanja postalo je jasno da je upravo RetinaNet arhitektura, već tada dosta proslavljena arhitektura nešto čime bi se trebalo pozabaviti.

Arhitektura RetinaNet

Prva od boljki kojom se ponosi arhitektura RetinaNet je da je to jednoprolazna neuronska mreža. Drugo iznenađenje kod mreže jest računanje pogreške. Radi se o nečemu što su originalni autori nazvali žarišni gubitak (*engl. focal loss*), a funkcija gubitka glasi:

$$L(p) = -\alpha(1 - p)^\gamma \log(p)$$

Analizirajmo malo priloženu jednadžbu. Na prvi pogled, ovo izgleda kao križna entropija (*engl. cross entropy*). Razlika je u novim hiperparametrima koje smo uveli, točnije α i γ .

Prvi hiperparametar, α , služi kao broj koji će skalirati ukupan gubitak, te je uobičajeno u rasponu $[0, 1]$. U svojem radu, autori preporučaju da se α postavi na inverz frekvencije neke klase.

Drugi hiperparametar, γ , služi kao faktor kojim će se izraz u zagradi, $(1 - p)$, potencirati, te je to uobičajeno broj u rasponu $[0, \infty >$. Ako analiziramo malo taj izraz, primijetiti ćemo da je on uvijek u rasponu $[0, 1]$, tako da će cijeli taj faktor konvergirati u 0 povećavanjem drugog hiperparametra. S obzirom da zapravo potenciramo vjerojatnost da detektirani objekt nije onaj koji je klasificiran, možemo primijetiti da za γ koji je 1 ili više možemo dobiti faktor koji će smanjivati doprinose gubitka što je klasifikator sigurniji oko zaključka o klasi. Ovo će u prijevodu značiti da će klasifikator biti stroži oko krivih pretpostavki malih sigurnosti, tj. da će se usredotočiti na optimiziranje neuronske mreže za raspoznavanje težih primjera. Uobičajeno je da se za hiperparametar γ uzima vrijednost 2, a ako mu damo vrijednost 0, žarišni gubitak postaje križna entropija.

Naposljetku, parametar p , koji je ujedno i parametar funkcije gubitka je sigurnost s kojom je donesena odluka, a to je uobičajeno vrijednost u intervalu $[0, 1]$.

Ovakvim pristupom želi se boriti protiv nejednakosti razreda. Razredi koji su vrlo zastupljeni će zbog velike sigurnosti manje doprinosti ukupnom gubitku. Mreža će se stoga posvetiti težim, manje zastupljenijim razredima, pa bi si mogli dozvoliti ukidanje procesa pronalaska interesantnih područja, tj. totalno eliminirati mrežu za preporuku područja iz cijele priče.

Komponente

Usprkos upravo navedenim značajkama arhitekture RetinaNet, ona nije obična konvolucijska neuronska mreža uz navedene značajke. Zapravo, radi se u skupu komponenti koje rješavaju neke od problema široko prisutnih u računalnom vidu. Općenito, RetinaNet se sastoji od jedne kičmene mreže (*engl. backbone network*) koja izlučuje značajke te vraća konvolucijsku mapu značajki (*engl. convolutional feature map*), te je ta mreža implementirana kao gotovo rješenje. Zatim postoje dvije jednostavne konvolucijske neuronske mreže koje redom obavljaju razredbu nad izlazom prve mreže, te pronalazak okvira graničnika (*engl. bounding box*).

Izlučivanje značajki

Izlučivanje značajki radi se korištenjem mreže piramide značajki (*engl. Feature Pyramid Network*). Mreža piramide značajki je uobičajena konvolucijska neuronska mreža puta od gore prema dolje (*engl. top-down pathway*). Ono što radimo tijekom izlučivanja značajki je da konvolucijskim filtrima smanjujemo dimenzije slike. Iako konstruiramo 7 slojeva iz originalne slike, za konstrukciju mapa značajki (*engl. feature map*) koristit ćemo slojeve C_3 do C_7 . Brojevi ispod slova označavaju smanjenje dimenzija originalne slike za 2^3 do 2^7 puta. Nakon generacije doljeuzorkovanih slojeva, konvoliramo svaki sloj 1×1 filtrom, generirajući značajke M_7 . Značajke M_7 generirat će našu najnižu razinu piramide, P_7 . Nakon toga, spajanjem slojeva M_7 i 1×1 konvolucije C_6 dobivamo sloj P_6 , i tako redom do P_3 . Spajanje slojeva se vrši na način da se manji sloj preuzorkuje (*engl. upsampling*) funkcijom najbližeg susjeda (*engl. nearest neighbour*), vrijednosti se po elementima

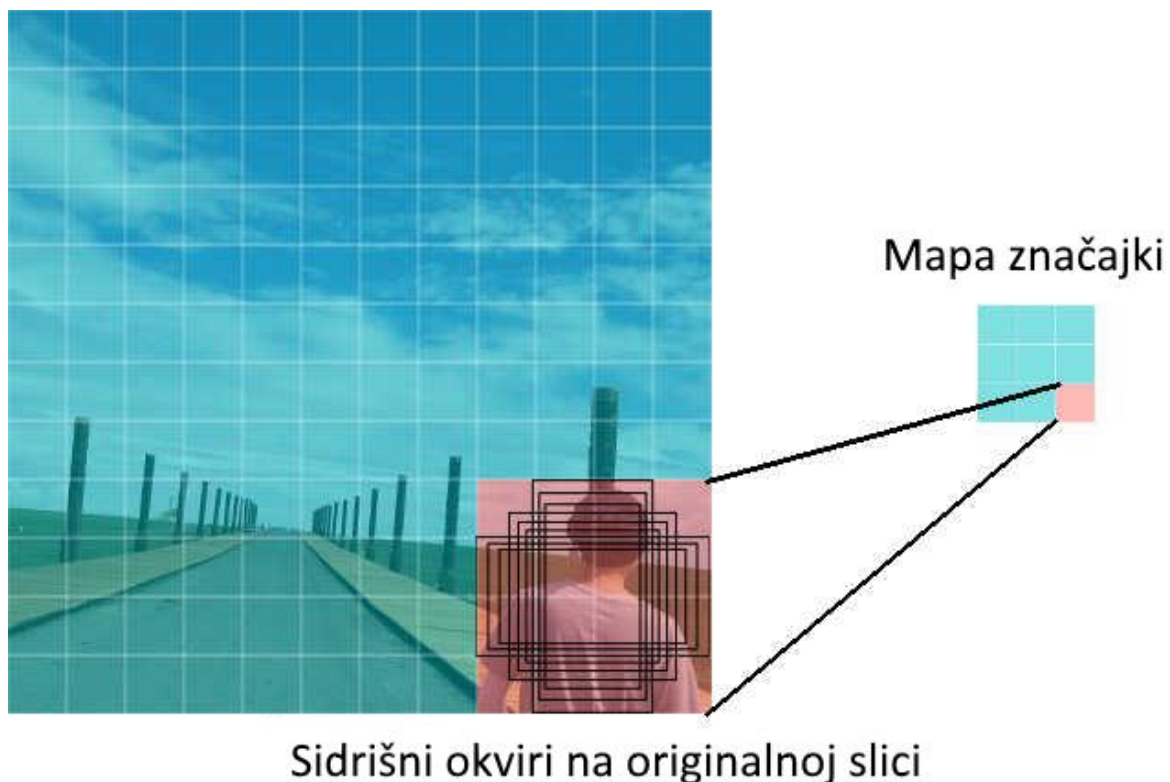
zbroje, te se svaki sloj koji nije onaj najniže rezolucije, koji je nastao bez spajanja, još jednom pročešlja 3x3 konvolucijom kako bi se smanjio *aliasing*. Ovom generacijom slojeva smo stvorili mape značajki koje mogu predviđati objekte različitih razmjera, a po testovima znatno ubrzavaju proces izlučivanja značajki te znatno povećavaju točnost neuronske mreže.

Iako se razredbene i pronalazne mreže u literaturi obično nazivaju podmrežama, u radu će se nadalje referencirati kao samo razredbene ili pronalazne konvolucijske neuronske mreže.

Razredbena konvolucijska neuronska mreža

Razredbu u RetinaNet arhitekturi obavlja potpuno povezana konvolucijska mreža (*engl. fully connected network*), priključena na svaki sloj značajki izlučenih pomoću mreže piramide značajki. Osnovna implementacija se sastoji od 4 sloja konvolucijska sloja dimenzija 3 x 3, te sadrži 256 filtera koji se aktiviraju ReLU funkcijom uključenja, koji vraćaju izlaz na posljednji konvolucijski sloj, također dimenzija 3 x 3 koji ovog puta sadrži $K \times A$ filtera, gdje je K broj razreda u koje ćemo svrstati neki objekt, dok je A broj sidrišnih okvira (*engl. anchor box*), čiji izlaz će se uključivati funkcijom sigmoida. Svaka razina ove mreže ima dijeljene težine. U broj razreda ne ulazi i implicitni razred pozadine, a sidrišni okviri su koncept koji smo uveli za ovu mrežu.

Sidrišni okviri su područja originalne slike različitih omjera širine i visine koje koristimo za označanje jedne ćelije izlaza mreže piramide značajki. Zamislimo da je izlaz naše mreže piramide značaki $N \times N$ dimenzija. Tada ćemo za svaku ćeliju (N^2 njih) definirati A sidrišnih okvira. U slučaju RetinaNet arhitekture, $A = 9$, te su omjeri 1:1, 1:2 i 2:1. Za svaki omjer imamo 3 sidrišna okvira različitih veličina čija su središta u središtu područja slike koje analiziramo.



Slika 5: Primjer sidrišnih okvira

Svaki sidrišni okvir u sebi sadrži vjerojatnosti za K razreda. Sidrišnim okvirima postizemo da uočeni predmet na slici može biti različitog oblika i veličine.

Izlaz razredbene konvolucijske neuronske mreže je oblika (W, H, KA) , gdje W i H označavaju širinu i visinu ulazne mape značajki. Svaka kombinacija širine i visine na mapi značajki odgovara jednom odjeljku originalne slike, za koji ćemo onda imati KA parametara koji će nam govoriti o vjerojatnostima za svaki od K razreda u svakom od A sidrišnih okvira.

Pronalazna konvolucijska neuronska mreža

Pronalazak objekata vrši potpuno povezana konvolucijska neuronska mreža koja je gotovo identična razredbenoj, a jedina razlika se nalazi u zadnjem konvolucijskom sloja; naime, iako je također dimenzija 3×3 , on sadrži $4A$ filtera. Stoga, izlazne dimenzije mape značajki je $(W, H, 4A)$. Razlog zašto imamo $4A$ filtera je vrlo jednostavan; kod uočavanja objekata želimo označiti pomak od središta sidrišnog okvira, te veličinu uočenog objekta. Ovo činimo s 4 vrijednosti: x, y, w, h , te je zato

dimenzionalnost mape značajki ($W, H, 4A$). Semantika prve dvije dimenzionalnosti je jednaka kao i za razredbenu mrežu.

Princip rada

Princip rada ove arhitekture je, srećom, vrlo jednostavan. Prvo, ulazna slika se koristi kako bi se izgenerirale mape značajki koje zadržavaju semantički bitne značajke u nekoliko slojeva, te olakšavaju uočavanje objekata različitih veličina, omjera, osvjetljenja i sl. Ovo se radi mrežom piramide značajke, a u našem slučaju, za to koristimo mrežu ResNet 50.

Nakon što smo izgenerirali mape značajki, razredbena i pronalazna mreža uzima u obzir sve slojeve mapa značajki te na izlazu vraća mape značajki dimenzija (W, H, KA) i ($W, H, 4A$). Nakon zaključivanja, potrebno je izračunati gubitak koji na prvi pogled nije trivijalan postupak.

Kod računanja gubitka ideja je usporediti označene podatke s zaključkom mreže te evaluirati koliko je zaključak mreže „promašio“ oznaku. U našem slučaju postoji velik broj zaključaka, pa prvo moramo odabrati one zaključke koje ćemo prihvatiti kao takve, a ostale odbaciti. S obzirom da se naši zaključci nalaze u tzv. izlaznim tenzorima (*engl. output tensors*), iz oznaka bi trebalo generirati ciljne tenzore (*engl. target tensors*). U općenitom slučaju, stvoriti ćemo tenzore istih veličina kao i izlazni tenzori te na mjestima gdje smo u skupu za učenje označili neki predmet postavljamo vrijednosti oznaka u ciljni tenzor.

Kažemo da je sidrišni okvir odgovara okviru oznake (*engl. ground truth box*) ako je njihov presjek po uniji (*engl. intersection over union*) veći od 50%. Tada će se za sve klase za koje ovo vrijedi u ciljnom tenzoru postaviti vektor duljine K s jedinicama na mjestima koje predstavljaju razrede koji su uspješno spareni u ovom procesu, a 0 na svim ostalim mjestima. Ako se dogodi da za neki sidrišni okvir ne postoji presjek po uniji s okvirom oznake veći ili jednak 40%, taj sidrišni okviri tada smatramo pozadinom, te ga u ciljnom tenzoru interpretiramo kao vektor duljine K sa 0 na svakom mjestu.

Međutim, postoji i treći slučaj – ako se dogodi da sidrišni okvir ima presjek po uniji između 40% i 50% za sve okvire oznake, smatramo da on nema para.

Računanje gubitka

Gubitak u RetinaNetu računa se kao suma dva člana; razredbenog gubitka (*engl. classification loss*) i pronalaznog gubitka (*engl. regression loss*). Možemo ga pisati kao:

$$L = L_{razredbe} + \lambda L_{pronaska}$$

gdje λ označava hiperparametar s kojim se uravnotežuju razredbeni i pronalazni gubitci. Ovaj parametar je potreban s obzirom da se za gubitke koriste 2 bitno različite funkcije za računanje gubitka. Valja napomenuti da se u našem slučaju nije podešavao parametar λ , pa su gubitci bile neuravnoteženi. Srećom, ova arhitektura ne prioritizira napredak ovisno o iznosu pojedinog gubitka, pa smo izgubili isključivo na estetici konačnog iznosa gubitka.

Razredbeni gubitak

Kao što smo prethodno naglasili, jedan od aseva ove arhitekture je računanje razredbenog gubitka. Podsjetimo se, izraz glasi:

$$L_{razredbe}(p) = -\alpha(1-p)^{\gamma}\log(p)$$

Također se podsjetimo da ovom funkcijom želimo riješiti problem neuravnoteženosti razreda (*engl. class imbalance*) koji je veliko ograničenje detektora u praksi. Ovo se postiže parametrom γ koji usredotočuje mrežu na ispravljanje zaključivanja za

zaključke niskih vjerojatnosti, dok parametar α pokušava uravnotežiti utjecaj nekog mnogobrojnijeg razreda na ostale uravnoteživanjem doprinosa kod gubitka, te je on obično inverz frekvencije pojavljivanja razreda za koji se zaključivalo.

Pronalazni gubitak

Pronalazni gubitak je, srećom, otprije poznata funkcija gubitka. Naime, radi se o glatkom L1 gubitku (*engl. smooth L1 loss*) koji se može predložiti sa sljedećom funkcijom:

$$L_{pronalaska} = \sum_{j \in \{x, y, w, h\}} s_{L1}(P_j^i - T_j^i)$$

gdje se funkcija s_{L1} definira kao:

$$s_{L1}(x) = \begin{cases} 0.5x^2 & \text{za } |x| < 1 \\ |x| - 0.5 & \text{za } |x| \geq 1 \end{cases}$$

a parametri T_x^i , T_y^i , T_w^i , T_h^i se računaju na sljedeći način:

$$T_x^i = (G_x^i - A_x^i)/A_w^i$$

$$T_y^i = (G_y^i - A_y^i)/A_h^i$$

$$T_w^i = \ln\left(\frac{G_w^i}{A_w^i}\right)$$

$$T_h^i = \ln\left(\frac{G_h^i}{A_h^i}\right)$$

Parametri koji su označeni s G predstavljaju vrijednosti okvira oznake, dok parametri označeni s A predstavljaju vrijednosti sparenog sidrišnog okvira. Laički rečeno, T_x^i i T_y^i su omjeri diferencijala označenog i predviđenog odmaka od centra sidrišnog

okvira i duljine ili visine sidrišnog okvira, te su stoga vrijednosti u rasponu od $[-0.5, 0.5]$, dok su T_w^i i T_h^i prirodni logaritmi omjera širina i visina okvira oznake i širina i visina sidrišnih okvira, te su, ukoliko nije došlo do pogreške u računanju G_w^i i G_h^i u rasponu od $\langle -\infty, +\infty \rangle$, no zbog prirode ovih mjera, uobičajeno je da su brojevi u neposrednoj okolini 0. Indeksi i označavaju indeks para okvira oznake i sidrišnog okvira, s obzirom da se računanje pronalaznog gubitka svodi na računanje glatkog L1 gubitka za svaki par.

Zaključivanje

Iz svih slojeva piramidalne mape značajki, naša mreža imati će $\sum_{l=3}^7 W_l H_l A$ sidrišnih okvira. Razredbena mreža će za svaki okvir donijeti odluku o K brojeva, 1 za svaki razred, dok će pronalazna mreža svakom sidrišnom okviru pridodati 4 vrijednosti koje će opisivati pomak okvira graničnika od sidrišnog okvira. Radi performansi, RetinaNet uzima u obzir najviše 1000 sidrišnih okvira s najvećim vjerojatnostima za sve razrede za svaki sloj mreže piramide značajki, ali tek nakon što su se maknuli sidrišni okvir za koje je sigurnost manja od 5%.

Nakon odabira sidrišnih okvira, primjenjuje se potiskivanje nemaksimuma (*engl. non-maximum-supression*), svakom razredu posebno na način da se odabere sidrišni okvir najveće sigurnosti te se iz skupa sidrišnih okvira uklone svi sidrišni okviri kojima je presjek po uniji veći od 50%.

Nakon što su se filtrirali svi redundantni sidrišni okviri, mreža zaključuje od pomaku u odnosu na sidrišni okvir preko kojeg se može doći do vrijednosti koje bi predstavljale okvir graničnik za uočeni objekt.

Postupak pripreme i interpretacije podataka

Rezultati

Usporedba s drugim arhitekturama srodnog zadatka

Zaključak

LITERATURA

NASLOV, SAŽETAK I KLJUČNE RIJEČI

DODATAK