Generator koda na osnovu skice sa slike

Dušan Bućan, Milica Travica
Fakultet tehničkih nauka, Novi Sad

DEFINICIJA PROBLEMA

Detekcija klas dijagrama, odnosno detekcija klasa, veza između klasa, metoda, atributa unutar klase, na osnovu kojih se kreiraju odgovarajuće java klase. Tekst koji se nalazi na slikama je ispisan štampanim slovima engleske latinice. Osim slova detektuju se i karakteri koji nam pomažu pri određivanju da li je atribut ili metoda klase privatna ili javna, kao i da li je u pitanju metoda ili atribut.

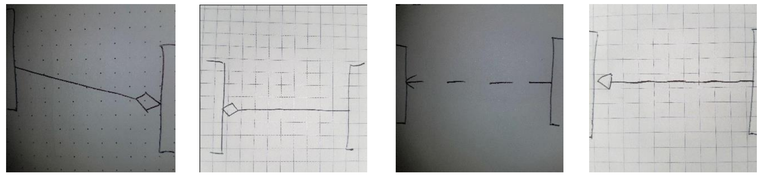
Većina programa za kreiranje klas dijagrama podržava ovu funkcionalnost, odnosno generisanje koda na osnovu klas dijagrama, na primer PowerDesigner. Da ne bi programeri radili dupli posao, odnosno nakon skiciranja slike klas dijagrama kreirali isti u nekom od programa namenjenih za to, mogu da koriste generator koda na osnovu slike.

SKUP PODATAKA

Postoje tri skupa podataka za obučavanje.

Prvi skup podataka se sastoji od slika klasa, kao i tekst fajla u kome se za svaku sliku nalazi naziv onoga šta je prikazano na slici kao i koordinate samog objekta. Ovaj skup podataka sadrži oko 50 slika, ali smo ga proširili primenom selectiv searcha i data augmentation i dobili oko 200 primera.

Drugi skup podataka se sastoji od slika veza, slike su podeljene po folderu u zavisnosti od tipa veze kojoj pripada, kao i od pravca veze, zbog velike varijanse unutar veza. Kao što je prikazano na slikama ispod (prve dve slike sa leva) raspoređene su u različite foldere, ikao su istog tipa. Svaki folder jednog tipa veze podeljen je na dva skupa trening i test u razmeri 90:10. Ovaj skup podataka sadrži 528 slika. Na slikama ispod je prikazano nekoliko primera veza.



Slike iz prvog i drugog skupa podataka smo mi kreirali, vodili smo računa da osvetljenje bude različito, kao i ugao slikanja kako bismo povećali robusnost modela.

U trećem skupu podataka nalaze karakteri, slike karaktera su takođe podeljene na trening i test skup. Slike karaktera slova smo pronašli na internetu, dok smo slike specijalnih karaktera koji se koriste u klasama mi kreirali. Ovaj skup podataka smo koristili za klasifikaciju karaktera ali je davao loše rezultate i na kraju smo se odlučili za gotovo rešenje za detekciju karaktera (tesseract).

ISPROBANE METODOLOGIJE

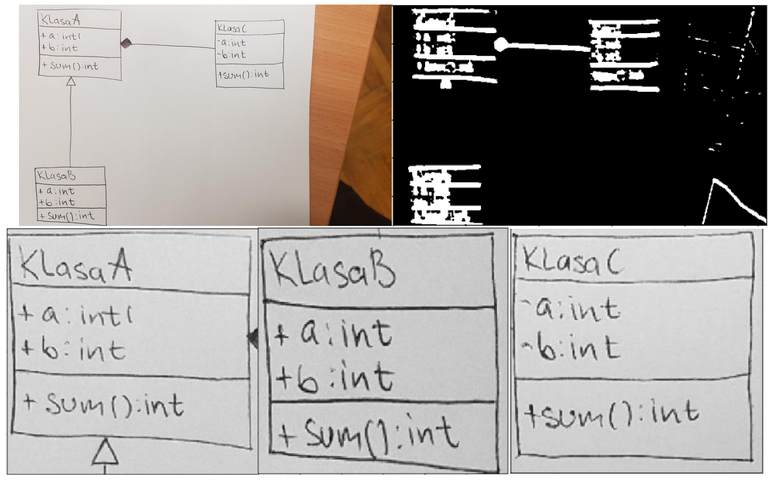
Kako bi rešili problem detekcije klasa i veza između njih pokušali smo da napravimo mrežu koja će da razaznaje vrstu veze kao i klasu. Da bi detektovali same objekte koji se nalaze na slici pokuši smo nekoliko načina. Prvi način je pomoću sliding window i piramide slika, ovaj način nije bio baš uspešan jer klasu ne bi dovoljno dobro isekao, njoj bi pridružio i deo veze ili prazan prostor, ili u gorem slučaju odsekao deo klase, jer se kreće za određeni korak, što bi kasnije otežavalo posao za detekciju veza, kao i određivanje metoda i atributa same klase.

Drugi način na koji smo pokušali da odredimo klase i veze jeste korišćenje mreže uz selectiv search, ovaj metod se pokazao loš za detekciju veza, dok je za detekciju klasa bio dosta uspešniji. Problem sa vezama je bio taj što ne bi prepoznavao celu vezu, što dovodi samim tim i do lošeg klasifikovanja veze ili bi prepoznao celu vezu i u 90% slučajeva loše bi je klasifikovao.

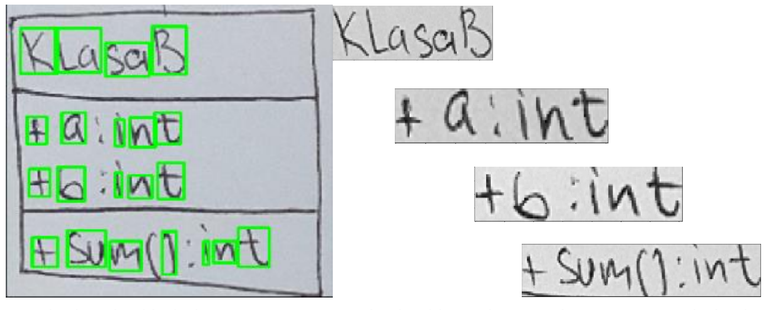
Problem detekcije karaktera smo pokušali da rešimo sa OCR-om, detekcija samih karaktera je u većini slučajeve bila uspešna, jedini razlog zbog kog nismo primenili ovo rešenje je jer smo pronašli bolje. Samo prepoznavanje karaktera smo pokušali da realizujemo predprocesiranjem slike karaktera uz pomoć konvolutivne mreže i klasifikacijom koristeći SVM, dobili smo dosta dobra rešenja na tesnom skupu, ali kada smo ovu metodologiju primenili na karaktere koje smo prepoznali unutar klase nismo dobijali dobra rešenja.

IZABRANE METODOLOGIJE

Metodologija za koju smo se odlčili za detekcju klasa je sobel i selectiv search. Uz pomoć sobela detektujemo linije klasa i ukoliko su linije u neposrednoj okolini spajaju se njihovi regioni i to se smatra jednom klasom. Nakon što se odrede potencijalne klase radi se provera da li je to klasa ili neki drugi deo sa slike, ukoliko zadovolji određeni kriterijum potencijalna klasa se smatra pravom klasom. Potencijalna klasa se uz pomoć konvolutivne mreže pretvara u feature vektor koji se prosleđuje SVM-u na klasifikaciju, gde se donosi odluka da li je klasa ili pozadninski objekat. SVM je obučavan uz pomoć selectiv searcha, odnosno sa slike su detektovan klase i drugi objekti koji su predprocesirani uz pomoć konvolutivne mreže odnosno pretvoreni su u feature vektor i prosleđeni na klasifikaciju. Na slikama ispod su priazane početna slika, sobel, i klase koje je detektovao kao krajnji rezultat.



Za detekciju karaktera iz klase koristili smo metodu iz cv2 biblioteke MSER. Nakon što bi detektovali karaktere u zavisnosi od njihovog položaja izdvajali bi redove karaktera koje prosleđujemo tesseract-u na detekciju. Razlog zbog kog smo se odlučili da tesseract-u prosleđujemo redove karakterea je to što tako daje najbolje rezultate, u odnosu na prosleđivanje jednog karaktera ili cele slike. Drugi razlog je to što je mnogo teže izdvojiti pojedinačne karaktere. Na slikama ispod je prikazana detekcija karaktera klase KlasaB (levo) i redovi karaktera koji se proselđuju na detekciju (desno).



Kako bi odredili da li postoji veza između dve klase i koja je ako postoji, prosleđivali smo sliku između dve klase SVM klasifikatoru za veze. Obučavanje klasifikatora je rađeno na primerima koje smo mi kreirali, sve slike su prvo predporcesirane u konvolutivnoj mreži, a nakon toga prosleđene na obučavanje. Smatra se da veza postoji ukoliko je verovatnoća da je veza koju je vratio klasifikator veća od 0,5, u suprotnom smatra se da veza ne postoji. Zbog mogućih različitih pravaca veza, kako bismo smanjili raznolikost unutar istog tipa veze podelili smo ih na levo i desno, kako ne bismo dodavali i za gore i dole rotiramo sliku veze pre prosleđivanja klasifikatoru kada se klase nalaze jedna ispod druge i time vertikalnu vezu posmatramo kao horizontalnu.

REZULTATI

Test skup se sastoji od 13 slika klas dijagrama, koje su generisane korišćenjem različitih metoda. Ručnim skiciranjem je generisano 6 slika, koje su slikane pri različitim osvetljenjem. Korišenjem Paint-a generisano je još 5 slika i generisana je jedna slika uz pomoć alata (LucidChart) za online skiciranje UML dijagrama. U metriku za evaluaciju generisanog koda smo uključili:

1. Procenat uspešno pronađenih klasa
2. Procenta uspešno prepoznatih veza (pored tipa veze i smer veze ako je bitno)
3. Procenat ukupnog uspešnog pronalaska atributa
4. Procenat atributa kod kojih su pravilno generisani tip i modifikator pristupa
5. Procenat uspešno generisanih imena klasa
6. Procenat uspešno generisanih naziva atributa
7. Procenat uspešno generisanih naziva metoda

Metrike od broja 1 do 4 ukazuju kolika je tačnost generisanog koda i direktno opisuju usklađenost funkcionalnosti generisanog koda sa specifikacijom (UML diagramom). Metrike od broja 5 do 7 pokazuju tačnost OCR-a i imaju manji značaj jer ne opisuju usklađenost funkcionalnosti generisani koda sa specifikacijom, već čitljivost koda, lakše razumevanje koda i održavanje.
U tabeli 1 se mogu videti rezultati dobijeni na test skupu.

Procenat uspešno pronađenih klasa	0.7115384615384616
Procenat uspešno prepoznatih veza	0.303030303030303
Procenat ukupnog uspešnog pronalaska atributa	0.8910256410256411
Procenat atributa kod kojih su pravilno generisani tip I modifikator pristupa	0.5448717948717948
Procenat uspešno generisanih imena klasa	0.5256410256410255
Procenat uspešno generisanih naziva atributa	0.44166666666666665
Procenat uspešno generisanih naziva metoda	0.46969696969696967

Tabela 1

ZAKLJUČAK

Jedna od načina da se poboljša ovo rešenje je povećavanjem skupa podataka. Povećanjem skupa podataka za veze postigli bi se mnogo bolji rezulatati.

Ukoliko bi se odlučili da treniramo svoju mrežu, odnosno klasifikator za slova bilo bi mnogo bolje da se koriste primeri koje bi mi napisali, nego primeri koje smo pronašli na internetu. Razlog zbog kojeg bi trebao da se zameni tesseract je taj što on ne detektuje najbolje slova koja su ručno pisana, makar ona bila i štampana. Samu detekciju karaktera bismo mogli da poboljšamo i postprocesingom, na primer korišćenjem Levenštajnovog rastojanja.

Samu klasifikaciju karaktera ili veza mogli bismo da poboljšamo korišćenjem nekog od modela ansambla, jer bi kombinovanje više modela davalo bolje rezultate.

Takođe ukoliko bi imali više primera za obučavanje detekcije klasa uz pomoć selctiv searcha, davao bi bolje rezultate nego prepoznavanje regiona na osnovu linija klase.