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 generise java programski kod. Tekst koji se nalazi na slikama je ispisan štampanim slovima engleske latinice. Osim slova detektuju se i karakteri na osnovu kojih se određuju modifikatori pristupa atributa i metoda, kao i da li je u pitanju metoda ili atribut.

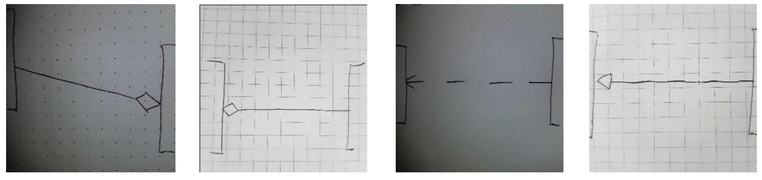
Motivacija za kreiranje projekta leži u potrebi da se od skiciranog klas dijagrama kreira prototip koji je moguće koristiti u ranim fazama razvoja softvera.

SKUP PODATAKA

Postoje tri skupa podataka za obučavanje.

Prvi skup podataka se sastoji od slika klasa, kao i tekstualnih fajlova za svaku sliku u kojima se nalaze koordinate klase na slici. Skup podataka sadrži 50 slika, primenom data augmentationa i selective search-a trening skup podataka je proširen na 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 jednog tipa veza, za svaki tip veze su kreirana 2 tipa veze. Kao što je prikazano na slikama ispod (prve dve slike sa leva) raspoređene su u različite tipove veze, ikao su istog tipa. Svaki skup primera jednog tipa veze podeljen je na trening i test skup u razmeri 90:10. Ovaj skup podataka sadrži 528 slika. Na slikama ispod je prikazano nekoliko primera veza.



Pri kreiranju slike iz prvog i drugog skupa podataka korišćeno je različito osvetljenje, kao i ugao slikanja kako bi se povećala robusnost modela.

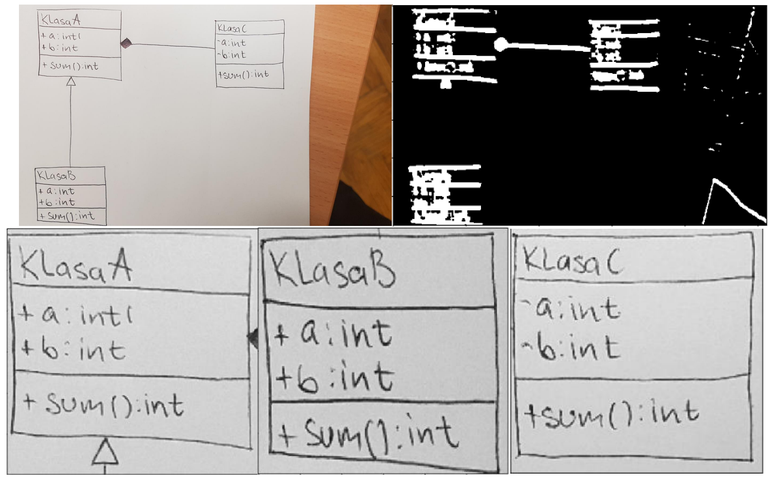
Treći skupu podataka sadrži slike karaktere koje su podeljene na trening i test skup. Skup podataka za karaktera je preuzet sa interneta, dok su slike specijalnih karaktera koji se koriste u klasama ručno kreirane. Model za klasifikaciju karaktera obučen na ovom skupu podataka postizao je lošije rezultate u odnosu na tesseract, pa se u konačnom rešenju koristi tesseract.

ISPROBANE METODOLOGIJE

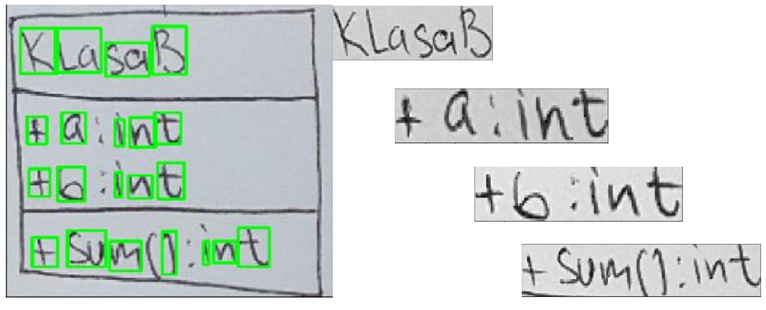
Problem detekcije klasa i veza između njih rešen je obučavanjem zasebnih modela za klasifikaciju. Za detekciju regiona koji sadrže klase i veze isprobano je više postupaka. Prvi način je pomoću sliding window i piramide slika, metod je vremenski zahtevan ali ne pruža najbolje performanse pa je odbačen. Drugi isprobani način se zasniva na selective search-u koji izdvaja regione od interesa i prosleđuje ih konvolutinoj mreži koja kreira feature vektor i prosleđuje ga SVM-u klasifikatoru. Problem ovog pristupa su lošije performanse ako na slici postoji senka kao i taj što bi prepoznavao deo veze, što rezultuje lošijim klasifikatorom za veze. Prednost navedenog postupka je brza detekcija klasa u uslovima dobre osvetljenosti. Problem detekcije karaktera smo pokušali da rešimo sa OCR-om, detekcija regiona od interesa je realizovana upotrebom biblioteke OpenCV. Samo prepoznavanje karaktera je realizovano upotrebnom konvolutivne mreže kao metode izdvajanja bitnih osobina i zatim klasifikacijom koristeći SVM. Dobijeni rezultati na test skupu karaktera su bili zadovoljavajući ali rezultati na klas dijagramima nisu. Jedan od razloga za to je što metod izdvajanja regiona neretko spoji više slova u jedan region, stoga je odabrana metoda za čitanje redova teksta na klas dijagramu.

IZABRANE METODOLOGIJE

Odabrana metodologija za detekciju klasa se zasniva na sobelu. Uz pomoć sobela detektuju se linije klasa i ukoliko su linije u neposrednoj okolini spajaju se njihovi regioni i zajedno čine jednu klasu. Svaki od potencijalnih regiona klase se proverava da li predstavlja klasa ili pozadinski objekat. Metod provere potencijalne klasa se sastoji od kreiranja feature vektora uz pomoć konvolutivne mreže i klasifikacije upotrebnom SVM-u klasifikatora. SVM klasifikator je obučavan na skupu podatka koji sadrži primere klasa. Primeri klasa su predprocesirani na opisan način a zatim uz pomoć konvolutivne mreže prevedeni u format pogodan za obučavanje SVM klasifikatora. Na slikama ispod su prikazane početna slika, slika procesirana sobelom, i klase koje su detektovane kao krajnji rezultat.



Za detekciju karaktera iz klase korišćena je metoda iz cv2 biblioteke MSER. Nakon detekcije karaktera u zavisnosti od njihovog položaja izdvajani su redovi karaktera koje prosleđujemo tesseract-u na detekciju. Razlog odabira ovog pristupa su bolje performanse u odnosu na prosleđivanje jednog karaktera ili cele slike. Drugi razlog su problemi pri izdvajanju pojedinačnih karaktera, kao što je spajanje više slova u jedan region. Na slikama ispod je prikazana detekcija karaktera klase KlasaB (levo) i redovi karaktera koji se prosleđuju na detekciju (desno).



Detekcija veza između dve klase, zasniva se na isecanju regiona između klasa zatim se izdvajaju bitne osobine regiona uz pomoć konvolutivne mreže i prosleđuju SVM klasifikatoru. Isečeni region se smatra da je veza ukoliko je rezultat klasifikatora verovatnoća veća od 0.5, u suprotnom smatra se da veza ne postoji. Klasifikator je obučen na ručno kreiranom skupu podataka. Zbog mogućih različitih pravaca veza, kako bi se smanjila raznolikost unutar istog tipa veze tipovi veza su podeljeni na dva tipa levo i desno. Rotacijom vertikalnih veza pre prosleđivanja mehanizmu za detekciju veza omogućena je i detekcija vertikalnih veza.

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. Korišćenjem Paint-a generisano je još 5 slika i generisana je jedna slika uz pomoć alata (LucidChart) za on-line skiciranje UML dijagrama. Metrika za evaluaciju generisanog koda uključuje:

1. Procenat uspešno pronađenih klasa
2. Procenta uspešno prepoznatih veza
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 dijagramom). 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 su prikazani 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

ZAKLJUČAK

Jedna od načina da se poboljša opisano rešenje je povećavanjem skupa podataka.

OCR je moguće unapredniti kombinacijom tesseract-a i klasifikatora obučenog na skupu podataka karaktera koji je ranije pomenut. Kombinacijom modela bi se prevazišli nedostaci tesseract-a, koji se ogledaju u prepoznavanju ručno pisanih slova. Postprocesingom, na primer korišćenjem Levenštajnovog rastojanja moguće je poboljšati generisani kod .

Modele za klasifikaciju veza i klasa moguće je unapredniti upotrebom nekog od modela ansambla, kao i povećanjem obima trening skupa za veze i klase.