OCR teksta na ruskom jeziku sa slike

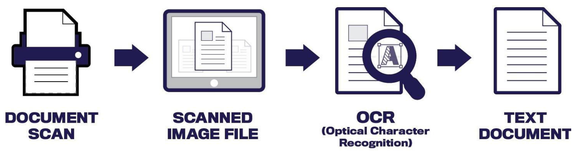
Marieta Rakoš, kontakt: marieta.rakos@gmail.com
Univerzitet u Novom Sadu, Fakultet tehničkih nauka

Apstrakt

Ideja ovog rada je proučavanje algoritama OCR (Optical Character Recognition) nad tekstom na ruskom jeziku.
U ovom radu, algoritam je malo izmenjen u odnosu na onaj sa vežbi.
Za mrežu je korišćena keras biblioteka, a za poređenje rezultata pytesseract biblioteka.
Iz radoznalosti je urađeno prevođenje prepoznatog teksta na srpski pomoću googletrans biblioteke.

Uvod

OCR je algoritam za prepoznacanje karaktera sa slike i njihovo konvertovanje u tekst.



Svodi se na nekoliko koraka:
•Predprocesiranje
•Određivanje regiona od interesa (karakteri)
•Treniranje nad slikom sa alfabetom
•Testiranje nad proizvoljnom slikom (lična ocena)

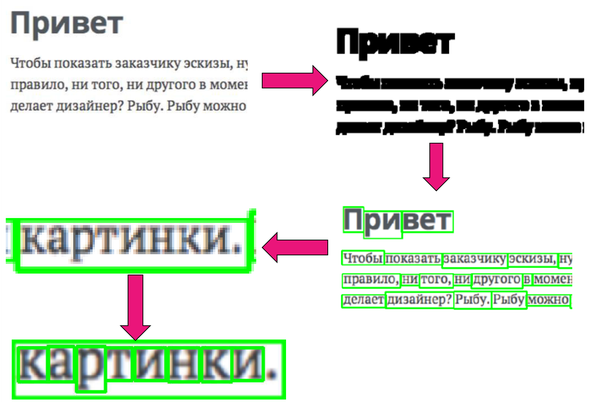
Postupak

Predprocesiranje

Svaku sliku je potrebno propustiti kroz niz filtera koji će je pripremiti za analizu. Neki od filtera su graj scale, invertovanje, binarizacija, zamućivanje itd. Ovime se slika u boji dovodi na crno'belu sliku gde su karakteri crne, a pozadina bele boje.

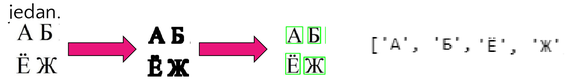
Izdvajanje objekata od interesa

Klasičan ocr algoritam izdvaja karaktere i na osnovu njihovog međusobnog rastojanja ih spaja u reči. Ideja u ovom projektu je bila da se izdvajaju cele reči kao regioni od interesa i onda nad tim izdvojenim rečima izdvajaju pojedinačni karakteri.
Za ovo je potrebno raširiti slova erodiranjem toliko da se slova spoje u reči, ali ne u redove.



Regioni koji predstavljaju reči se sortiraju po y-osi, zatim se određuju redovi i na kraju se u svakom redu sortiraju po x-osi
Prepoynata slova se spajaju u reči, a reči u rečenicu.

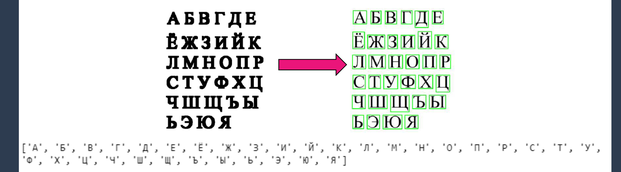
Kod treniranja se koristi slika sa alfabetom gde su sva slova razmaknuta dovoljno da se ne spoje prilikom raširivanja. Raširivanjem se u ovom slučaju karakter iz više regiona spoji u jedan.



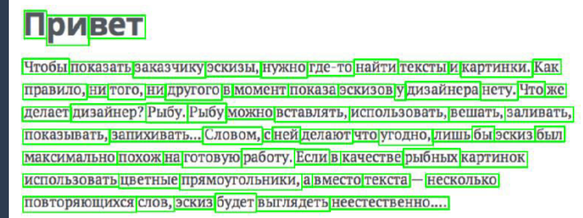
Rezultati

Treniranje

Pri treniranju, mreža tačno prepozna sve karaktere iz alfabeta. Trenirano je nad 1000 epoha.



Testiranje nad proizvoljnim slikama.
Na slikama koje su korišćene za testiranje, u većini slučajeva se tačno izdvoje reči i u njima slova (npr. na slici levo).
Prepoznavanje karaktera se nije pokazalo veoma dobro. Na primer, sa slike ispod



Algoritam prepozna sledeći tekst:

юзэнр стситз бАххфсэ бэйххжеёзр фхйдфлуршэ фдэсжхэихзр бэхзэАяъгъб б Б яэйххэзр сфхвффтАз фф итз зэмэ хёбъшихзр бэфэь щиэзфжт ххйххюффл бэх шф Ажнл шэ сАлмэмэ хдйфх Аяфэлмэсрффйф мэзэхлъ Арнл дсэхэф хдйфхр нлстз х фэьшэ лэфэфз Ахнэз ежмсястзр шАшэ д штф хдзихсязр бэйххх чбф стсёъз х мстзэ хйэдзэ х фттбтбхтшшэ хдффхэх йхютдзит фдбэсрхэхтзр юзэ шэфзф зтйби лэзэ л зтйдзр Арншрь сфхэффэАэ штдфэсрйэ сфар ф етаёзр йхАзфшэй йёАзфшйф нр шэз хдйфе хуфевзр юзэ йэй нра ьэ

Zaključak

Sudeći prema tačnosti pokazanoj pri treniranju, očekivalo bi se da će se algoritam bolje pokazati. Valja napomenuti da algoritam za sortiranje regiona takođe ne daje dobar rezultat te stoga nije baš lako odrediti valjanost algoritma. Ono što je pomalo čudno kod sortiranja jeste to da iako je navedeno da sortira po npr. x-osi, iz sortirane liste se vidi da ipak nije sortirano po datoj osi.

Uz dalja istraživanja, ovaj algoritam bi mogao da daje bolje rezultate. Neke od ideja su: treniranje na ne tako savršenom slikom- koristiti sliku manje rezolucije; uvesti treniranje nad različitim fontovima.
Ono što bi umnogome poboljšalo rad algoritma jeste bolje predprocesiranje- otkloniti šum, zamagliti sliku itd.
Takođe, treba poboljšati algoritam za sortiranje regiona.

Cilj ovog rada je bio napraviti veoma jednostavan algoritam koji će dovoljno dobro da radi.

Dalja istraživanja

•Usavršavanje predprocesiranja slike
•Treniranje nad ne tako savršenim podacima (npr. slika sa alfabetom manje rezolucije)
•Upotreba drugog algoritma
• Pronalaženje najpribližnijih reči Levenštajn metodom