

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 5709

**Sustav za određivanje strukture
teksta na temelju položaja
pojedinih znakova**

Herman Zvonimir Došilović

Zagreb, lipanj 2018.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA ZAVRŠNI RAD MODULA

Zagreb, 14. ožujka 2018.

ZAVRŠNI ZADATAK br. 5709

Pristupnik: Herman Zvonimir Došilović (0036480275)
Studij: Računarstvo
Modul: Računarska znanost

Zadatak: Sustav za određivanje strukture teksta na temelju položaja pojedinih znakova

Opis zadatka:

Sustavi za automatsko očitavanje teksta sa skeniranih dokumenata imaju nekoliko zadaća koje uključuju lokalizaciju, segmentaciju i prepoznavanje pojedinih znakova te slaganje prepoznatih znakova u složenije strukture poput riječi i linija. To je u praksi vrlo težak problem.

U okviru ovog završnog rada potrebno je proučiti načine za određivanje riječi i linija na temelju položaja individualnih znakova te njihovih omeđujućih pravokutnika. U okviru rada potrebno je pripremiti odgovarajući skup podataka za testiranje te napraviti prototipnu implementaciju sustava.

Radu priložiti izvorni kod razvijenih postupaka uz potrebna objašnjenja i dokumentaciju. Predložiti pravce budućeg razvoja. Citirati korištenu literaturu i navesti dobivenu pomoć.

Zadatak uručen pristupniku: 16. ožujka 2018.

Rok za predaju rada: 15. lipnja 2018.

Mentor:

Doc. dr. sc. Marko Čupić

Predsjednik odbora za
završni rad modula:

Prof. dr. sc. Siniša Srblijić

Djelovodja:

Doc. dr. sc. Tomislav Hrkać

Zahvaljujem svom mentoru doc. dr. sc. Marku Čupiću na dozvoli za odabir vlastite teme i na strpljenju, poticaju i savjetima u razvoju rada.

Zahvaljujem tvrtki Microblink na danim sredstvima bez kojih ovaj rad ne bi bio moguć. Posebno zahvaljujem kolegama Jurici Cerovecu, Nenadu Mikši, Borisu Trubiću, Igoru Smolkoviču i Ivanu Jurinu koji su me svojim bogatim znanjem i iskustvom usmjeravali u razvoju rada.

Tko hoće da među vama bude najveći, neka vam bude poslužitelj! I tko hoće da među vama bude prvi, neka bude svima sluga. - Mk 10,43-44

SADRŽAJ

1. Uvod	1
2. Optičko raspoznavanje znakova	2
2.1. Primjene	2
2.2. Proces izvođenja	4
3. Određivanje strukture teksta	8
4. Opis problema	12
4.1. Primjer ulaza	12
4.2. Zahtjevi sustava	15
5. Skup podataka za testiranje	17
5.1. Slike	17
5.2. JSON datoteke	19
6. Algoritmi za određivanje strukture teksta	21
7. Rezultati i analiza	22
7.1. Metrike	22
7.2. Brojke	22
7.3. Analiza	22
8. Daljnji rad	23
9. Dodaci	24
10. Zaključak	25
Literatura	26

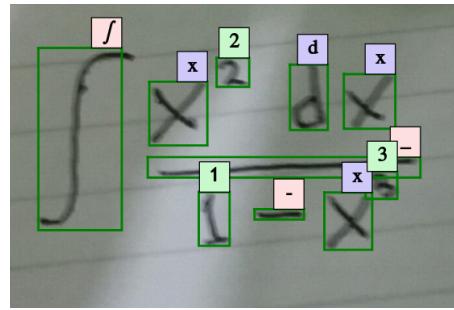
1. Uvod

2. Optičko raspoznavanje znakova

Sustav za optičko raspoznavanje znakova (engl. *optical character recognition*) (u daljnjem tekstu: OCR) pretvara sliku tiskanog teksta u digitalizirani format kojim možemo jednostavno manipulirati na računalu. Iako je to ljudima jednostavan zadatak, računallima nije lako prepoznati tekst i pojedine znakove teksta sa slike zbog velike raznolikosti jezika, fonta i stila kojim tekst može biti napisan. OCR je stoga vrlo zahtjevan problem i mnogo je istraživačkog truda uloženo u pokušaju da se slike teksta pretvore u format koji računalo razumije. (Islam et al., 2017)

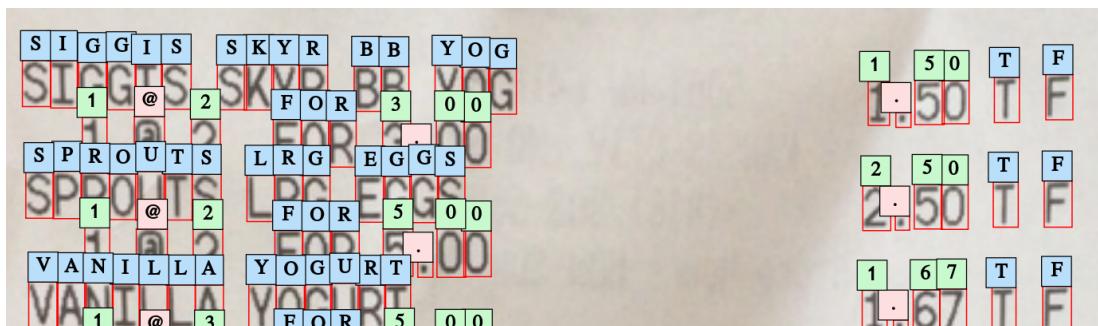
2.1. Primjene

Osim tiskanog teksta, OCR sustavi koriste se i u prepoznavanju znakova rukom pisanih teksta. Prepoznavanje znakova rukom pisanih teksta je teži problem od prepoznavanja tiskanog teksta (Islam et al., 2017) zato jer se oblik znakova i njihov način pisanja razlikuje kod svake osobe (npr. rukopis odrasle osobe potpuno je drugačiji od rukopisa djeteta). OCR sustave za detekciju rukom pisanih znakova možemo podijeliti na dvije potkategorije: *on-line* i *off-line*. *On-line* OCR sustavi detektiraju znakove dok ih korisnici unose i to im omogućuje praćenje parametara poput: brzine pisanja, broj napravljenih poteza, smjer pisanja, itd. *Off-line* OCR sustavi izvode se nad jednom slikom na kojoj se nalazi sav sadržaj nad kojim je potrebno napraviti detekciju. Takvi sustavi nemaju dodatne informacije koje imaju *on-line* sustavi i zato je detekcija znakova komplikiranija (Islam et al., 2017). Slika 2.1 prikazuje primjer rezultata *off-line* OCR sustava za detekciju rukom pisanih znakova.

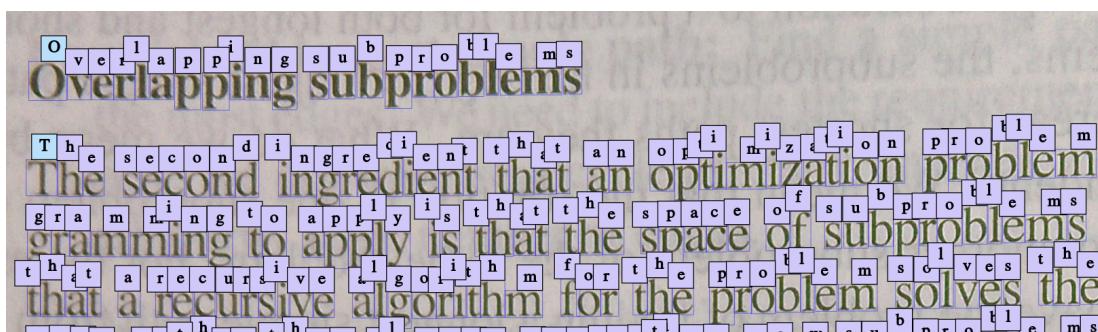


Slika 2.1: Rezultat *off-line* OCR sustava za detekciju znakova rukom pisanog teksta.

OCR sustavi imaju široku primjenu i možemo ih pronaći primjerice u detekciji znakova na registarskim pločicama (Saghaei, 2016), (Laroca et al., 2018), u detekciji znakova na tiskanim knjigama (Wick et al., 2018), (Christy et al., 2017) i detekciji znakova na raznim dokumenatima (Harraj i Raissouni, 2015) (Verma et al., 2016). Na slici 2.2 prikazan je primjer rezultata korištenja OCR sustava za detekciju znakova na računima iz trgovine. Slika 2.3 prikazuje rezultat OCR sustava za detekciju znakova na tiskanim knjigama.



Slika 2.2: Rezultat OCR sustava za detekciju znakova na računima iz trgovine.



Slika 2.3: Rezultat OCR sustava za detekciju znakova na tiskanim knjigama.

2.2. Proces izvođenja

Optičko raspoznavanje znakova provodi se u nekoliko koraka (Islam et al., 2017) (Kaur i Rani, 2016):

1. pribavljanje slike,
2. pretprocesiranje,
3. segmentacija znakova,
4. izdvajanje značajki znakova,
5. klasifikacija znakova i
6. postprocesiranje.

Pribavljanje slike

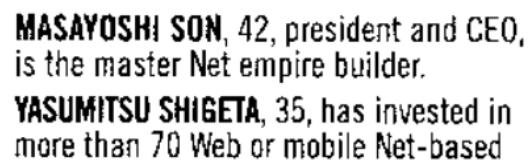
U prvom koraku OCR-a, pribavljanju slike, potrebno je pribaviti sliku nad kojom ćemo provesti ostale korake. Sliku možemo pribaviti s raznih uređaja poput kamere fotoaparata, mobilnog uređaja ili nekog drugog uređaja za digitalizaciju dokumenata (engl. *scanner*). Nakon prvog koraka, slika dokumenta nad kojim provodimo raspoznavanje znakova sastoji se samo od slikovnih elemenata (engl. *pixels*) (Vynckier, 2018). Slika 2.4 prikazuje primjer slike nad kojom možemo provesti postupak raspoznavanja znakova. Primjetimo da slika može sadržati pozadinu koju bi OCR sustav trebao zanemariti.



Slika 2.4: Ulazna slika u OCR sustav pribavljena kamerom mobilnog uređaja.

Preprocesiranje

U koraku preprocesiranja slike OCR sustavi često provode niz morfoloških transformacija i filtra nad pribavljenom slikom. Cilj ovog koraka je povećati kvalitetu slike i smanjiti informacije na slici. Binarizacija je jedan od potkoraka preporocesiranja koji slike u boji ili u nijansama sive pretvara u crno-bijele. Osim binarizacije koriste se neke morfološke transformacije poput dilatacije, rezanja i skaliranja. Slika 2.5 prikazuje primjer slike prije i nakon binarizacije. (Gulan, 2016), (Islam et al., 2017), (Jurin, 2017)



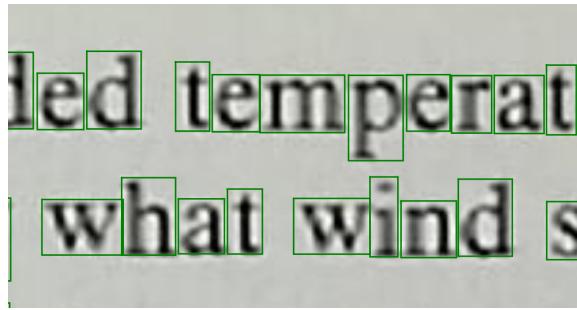
Slika 2.5: Prije binarizacije (lijevo) i nakon binarizacije (desno) (Vynckier, 2018).

Segmentacija znakova

Sljedeći korak, segmentacija znakova, je postupak segmentiranja slike u segmente unutar kojih se nalaze znakovi koje želimo klasificirati. Jedan od pristupa segmentacije izvodi se s vrha prema dnu gdje se najprije segmentiraju linije, zatim riječi i na kraju pojedini znakovi (Jurin, 2017), (Vynckier, 2018). Prednost ovakvog pristupa je da uz lokaciju svakog znaka dobivamo i strukturu cijelog teksta, odnosno, znamo kojoj liniji i kojoj riječi znak pripada. Nedostatak ovakvog pristupa je da ne postoje korekcijski mehanizmi kojima bismo znak pridružili nekoj drugoj liniji ili riječi ako su prva dva koraka segmentacije linije ili riječi neispravni. (Jurin, 2017)

Drugi pristupi poput *ZICER OCR*¹ sustava izravno izvode segmentaciju cijele slike na području koji predstavljaju znakove. Prednost takvog pristupa je da možemo detektirati znakove teksta u kojemu nema riječi i linija, kao što je na primjer matematički izraz. Nedostatak takvog pristupa je da gubimo informaciju o strukturi teksta i zato postoji potreba za razvojem dodatnog sustava koji bi znakove grupirao u riječi, a riječi u linije (Jurin, 2017). Slika 2.6 prikazuje rezultat segmentacije pojedinih znakova.

¹OCR sustav tvrtke *Microblink*, <https://microblink.com>



Slika 2.6: Segmentacija znakova.

Izdvajanje značajki

Izdvajanje značajki pojedinog znaka podrazumijeva odabir značajki prema kojima će se jedinstveno klasificirati svaki znak. Značajke poput geometrijskog oblika ili statističkih svojstava mogu biti uzete u obzir prilikom klasifikacije. Važno područje istraživanja pripada razmatranju koje i koliko značajki je potrebno uzeti u obzir za kvalitetnu i ispravnu klasifikaciju. (Islam et al., 2017)

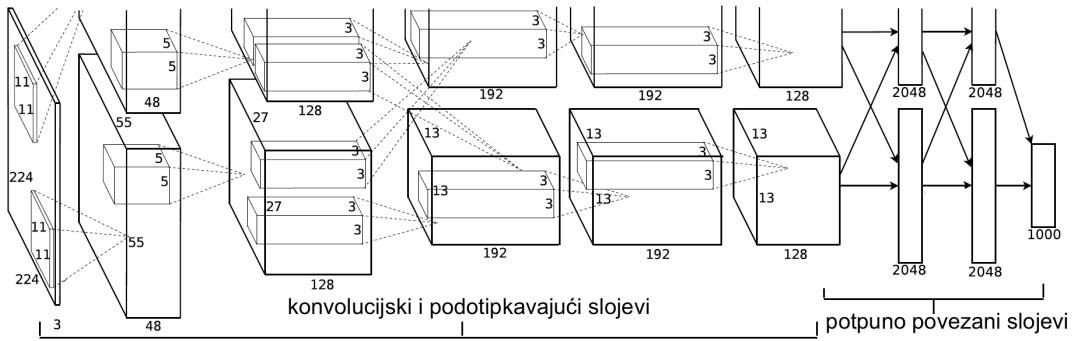
Klasifikacija

Klasifikacija je najvažniji korak optičkog raspoznavanja znakova (Verma i Ali, 2012) (Zhu et al., 2016) koji koristi izdvojene značajke za određivanje klase pojedinog znaka (Lehal i Singh, 1999) (Kaur i Rani, 2016). Statistički pristupi klasifikacije koriste diskriminativne funkcije za određivanje klase znaka (Islam et al., 2017), a u novije vrijeme koriste se duboke neuronske mreže (Jurin, 2017). Neki od statističkih pristupa su: Bayesov klasifikator, klasifikator stablom odluke, umjetne neuronske mreže i metoda k-najbližih susjeda (Islam et al., 2017).

2012. godine Krizhevsky i suradnici (Krizhevsky et al., 2012) objavili su rad koji je označio prekretnicu u klasifikaciji i lokalizaciji objekata (Jurin, 2017). Slika 2.7 prikazuje arhitekturu *AlexNet* koja je pobijedila na natječaju *ImageNet 2012* u području klasifikacije objekata. (Jurin, 2017)

Postprocesiranje

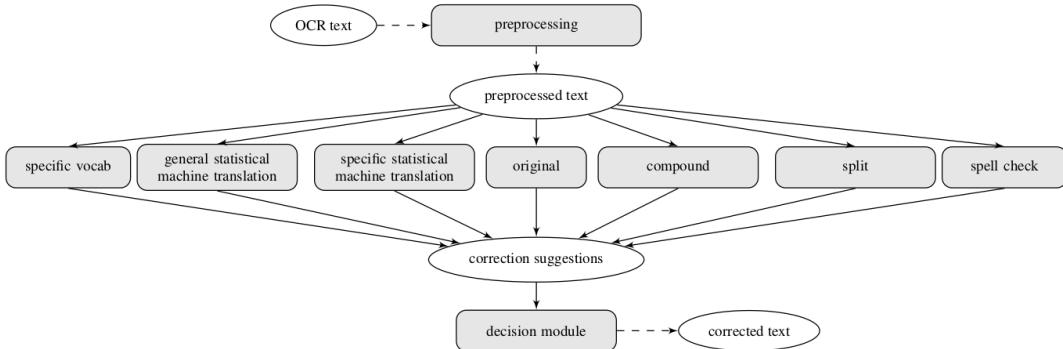
Nakon klasifikacije znakova slijedi njihovo postprocesiranje koje se koristi kako bi se poboljšali rezultati OCR-a. Jedan od pristupa postprocesiranja koristi rezultate više različitih klasifikatora koji mogu biti korišteni slijedno, paralelno ili hijerarhijski. Nakon toga rezultati klasifikatora se kombiniraju različitim pristupima. (Islam et al., 2017) Kao što je spomenuto u 2.2 segmentacija koja se ne provodi s vrha prema



Slika 2.7: Arhitektura *AlexNet* (Jurin, 2017)

dnu nema informaciju o strukturi teksta i zato je potrebno razviti dodatan **sustav za određivanje strukture teksta na temelju položaja pojedinih znakova**.

Schulz i suradnici (Schulz i Kuhn, 2017) 2017. godine predstavili su arhitekturu tzv. *post-correction* OCR sustava kojim su pokazati na koji način su adaptirali generički sustav za postprocesiranje OCR rezultata koristeći domensko znanje za konkretni problem koji su rješavali. Ovim pristupom ostvarili su bolje rezultate za konkretni problem nego što su ostvarili koristeći postojeći generički sustav za postprocesiranje OCR rezultata.



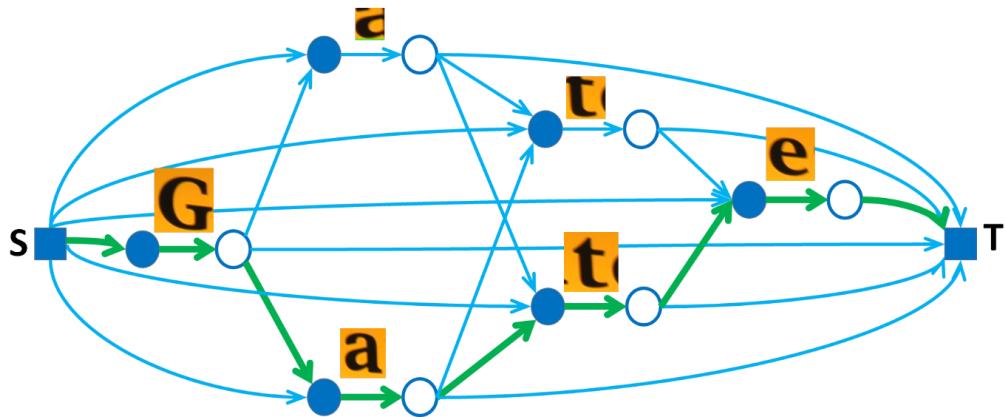
Slika 2.8: Arhitektura *post-correction* OCR sustava (Schulz i Kuhn, 2017)

3. Određivanje strukture teksta

Sustavi za određivanje strukture teksta na temelju OCR rezultata sastavni su dio OCR sustava. Određivanje strukture teksta podrazumjeva segmentaciju linija i segmentaciju riječi unutar linije. Neke tehnike segmentacije OCR znakova i njihove klasifikacije nemaju informaciju o tome kojoj liniji i riječi pojedini znak pripada. Prednost takvog pristupa je da takav OCR sustav možemo koristiti nad slikama koje ne sadrže linije, kao na primjer matematički izrazi (Jurin, 2017). Nedostatak je što nakon klasifikacije moramo razviti sustav koji će OCR znakove dodatno procesirati da bismo dobili strukturu teksta (Jurin, 2017).

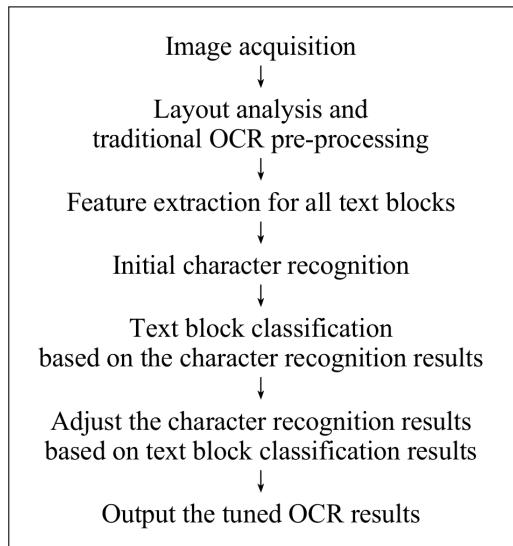
Tekst na slici može biti podijeljen na linije ili blokove, a u bloku tekst možemo podijeliti na linije. Unutar jedne linije znakove možemo grupirati riječi. Način na koji će se odrediti struktura teksta uvelike ovisi o problemu koji riješavamo i kakve rezultate želimo dobiti.

Tain i suradnici (Tian et al., 2016) predlažu sustav za određivanje strukture teksta koji će osim određivanja kojoj liniji pojedini znak pripada znati izbaciti tzv. *false positive* znakove odnosno znakove koje je OCR sustav prepoznao, a zapravo u tekstu ne postoje. Njihov sustav temelji se na *min-cost flow network* modelu koji objedinjuje izbacivanje *false positive* znakova i pronalazak strukture teksta. Na temelju međusobne pozicije između dva prepoznata znaka i dodatnog parametra kojeg dobivaju od klasifikatora, a koji označava vjerojatnost ispravne detekcije, grade težinski usmjereni graf (Slika 3.1) koji svoj problem modeliraju *min-cost flow network* modelom.



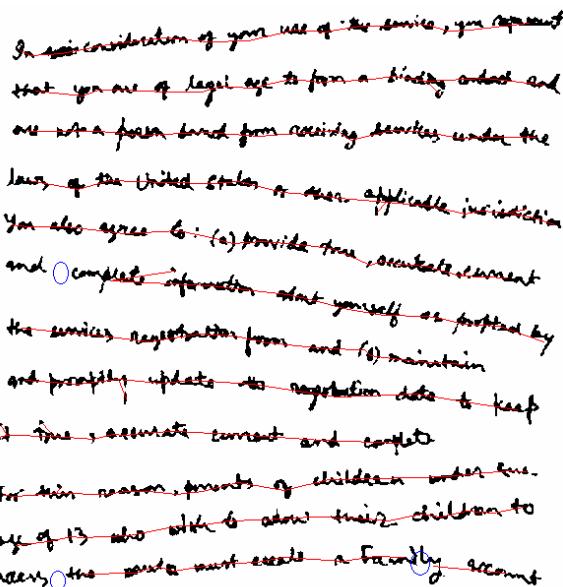
Slika 3.1: Težinski usmjereni graf temeljen na *min-cost flow network* modelu (Tian et al., 2016).

Zhu i suradnici (Zhu et al., 2016) predložili su novu arhitekturu (Slika 3.2) OCR sustava koji se temelji na empirijskim rezultatima koji su pokazali da sadržaj riječi ne ovisi samo o dijelu teksta u kojem se ta riječ nalazi nego i o susjednim dijelovima teksta. Njihov novi OCR sustav radi dvostruku analizu strukture teksta – prije klasifikacije i nakon klasifikacije. Prva analiza strukture teksta omogućuje im da odredi strukturu teksta u blokovima, a druga analiza strukture teksta im omogućuje da poprave pogreške u klasifikaciji. Njihova nova arhitektura predstavlja hibridni OCR sustav koji iskorištava rezultate analize strukture teksta.



Slika 3.2: Arhitektura novog OCR sustava kojeg predlažu Zhu i suradnici (Zhu et al., 2016).

Yin i suradnici (Yin i Liu, 2007) pronalaze linije u tekstu povezujući znakove u težinski graf nad kojim provode Kruskalov algoritam za pronalazak minimalnog razapinjujućeg stabla. Njihov pristup ne koristi rezultate klasifikacije, nego koriste povezane komponente koje im predstavljaju znakove i koje pronalaze koristeći algoritam temeljen na praćenju kontura (engl. *contour tracing*). Slika 3.3 prikazuje rezultat pronalaska linija teksta u rukom pisanim dokumentu na Engleskom jeziku.

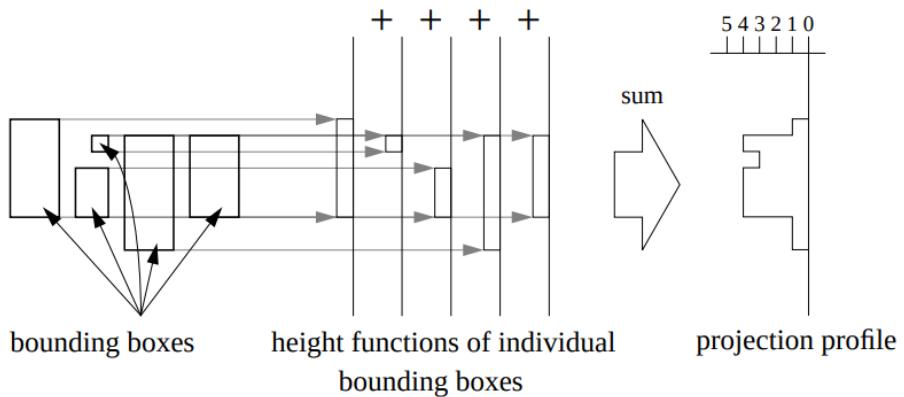


Slika 3.3: Rezultat pronalaska linija teksta u rukom pisanim dokumentu na Engleskom jeziku (Yin i Liu, 2007)

Motivirani njihovim radom Pan i suradnici (Pan et al., 2011) predstavili su sličan pristup koji u težinama grafa uzima u obzir dodatne težine koje su učene MCE (engl. *minimum classification error*) mjerom.

Još jedan pristup predložili su Yin i suradnici (Yin et al., 2013) koji koriste tehniku hijerahiskog grupiranja koji postupno spaja linije koje dijele znakove dok god postoje linije koje se mogu spojiti. (Tian et al., 2016)

Liang i suradnici (Liang et al., 1996) predlažu heuristički algoritam za određivanje strukture teksta. Algoritam radi horizontalnu projekciju (Slika 3.4) omeđujućih pravokutnika na jednu ravnicu i pronalazi vrhove i doline u histogramu koji prikazuje frekvencije pojavljivanja projektiranih pravokutnika. Osim ovog pristupa predložili su još jedan koji spaja dva znaka u jednu cjelinu ako i samo ako su dva znaka dovoljno blizu da ih ima smisla spojiti. Gupta i suradnici (Gupta et al., 2006) također koriste razne heuristike prema kojima povezuju susjedne omeđujuće pravokutnike.



Slika 3.4: Histogram dobiven horizontalnom projekcijom omeđujućih pravokutnika (Liang et al., 1996)

Određivanje strukture teksta težak je postupak koji uvelike ovisi o problemu koji rješavamo. U ovom poglavlju smo pokazali da strukturiranje teksta može biti izvođeno u raznim fazama izvođenja OCR sustava. Trenutak u kojem ćemo pokrenuti analizu strukture teksta ovisi o načinu na koji smo označili segmente znakova, koje informacije o segmentu imamo i koji problem rješavamo. Pokazali smo da su neki pristupi postigli bolje rezultate kada se iskoristilo domensko znanje i kada su se u nekim heurističkim pristupima koristili parametri koji su bili pomno izabrani za dani problem. U nastavku ovog rada predložiti ćemo nekoliko pristupa za određivanje strukture teksta na temelju položaja omeđujućih pravokutnika nakon provedene klasifikacije znakova. Analizirati ćemo njihove prednosti i nedostatke i cijelo vrijeme ćemo znati koji točno problem rješavamo.

4. Opis problema

Problem koji rješavamo je određivanje strukture teksta na temelju položaja pojedinih znakova. Od OCR sustava dobijemo listu znakova i za svaki znak znamo sljedeće informacije:

- x - horizontalnu poziciju gornjeg lijevog kuta,
- y - vertikalnu poziciju gornjeg lijevog kuta,
- $width$ - širinu znaka,
- $height$ - visinu znaka i
- $value$ - Unicode vrijednost znaka.

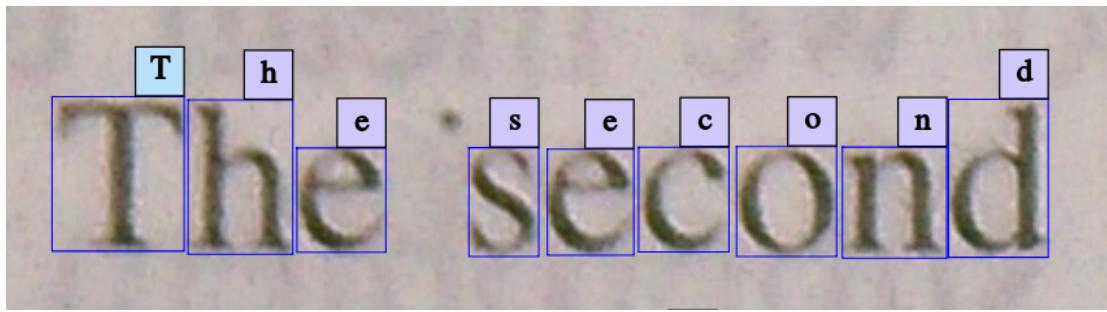
U sklopu ovog završnog rada rješavati ćemo problem određivanja strukture teksta na računima iz trgovine i sadržaja iz knjiga. Stup podataka za testiranje sustava biti će detaljno objašnjen u poglavlju 5.

4.1. Primjer ulaza

Slika 4.1 prikazuje vizualizaciju rezultata OCR sustava koji nam je za svaki znak vratio omeđujući pravokutnik (označeno plavom bojom) i vrijednost znaka odnosno klasu kojoj znak pripada. Slike 2.2 i 2.3 također prikazuju vizualizaciju rezultata OCR sustava i primjer podataka s kakvim će se sustav za određivanje strukture teksta susresti.

Primjer pojednostavljenog (detaljnije u poglavlju 5) OCR rezultata koji će sustav za određivanje strukture teksta dobiti kao svoj ulaz prikazan je Isječkom 4.1. Ulagani OCR rezultat u JSON¹ formatu uvek se sastoji od jedne linije u koju su smješteni znakovi nasumičnim redoslijedom.

¹JSON format, <https://json.org>



Slika 4.1: Primjer vizualizacije rezultata koji vraća OCR sustav.

```
1  {
2      "ocr_result": {
3          "lines": [
4              {
5                  "chars": [
6                      {
7                          "x": 25.95604,
8                          "y": 17.30562,
9                          "width": 10.64438,
10                         "height": 16.60289,
11                         "value": 48
12                     },
13                     {
14                         "x": 19.77133,
15                         "y": 1.28793,
16                         "width": 16.07777,
17                         "height": 10.76925,
18                         "value": 77
19                     },
20                     {
21                         "x": 5.50248,
22                         "y": 2.84320,
23                         "width": 12.13375,
24                         "height": 15.60966,
25                         "value": 73
26                     },
27                     {
28                         "x": 3.19550,
29                         "y": 19.67606,
30                         "width": 14.94088,
31                         "height": 20.78798,
32                         "value": 91
33                 ]
34             }
35         ]
36     }
37 }
```

```
33         } ,  
34  
35     ]  
36     }  
37     ]  
38 }  
39 }
```

Isječak 4.1: Primjer OCR rezultata u JSON formatu. Ulaz u sustav za određivanje strukture teksta.

4.2. Zahtjevi sustava

Sustav za određivanje strukture teksta (u daljem tekstu: *Sustav*) treba za dobiveni OCR rezultat u JSON formatu vratiti novi rezultat u JSON formatu koji će znakove grupirati u linije i koji će unutar linija biti poredani s lijeva na desno. Također, linije moraju biti sortirane tako da se najviša linija u dokumentu nalazi na prvom mjestu.

Osim grupiranja linija, sustav treba odrediti gdje se nalazi razmak između riječi. Od sustava se očekuje da između dva znaka, gdje smatra da završava prethodna i počinje nova riječ, ubaci novi znak bjeline čija je vrijednost (engl. *value*) 32 dok ostale informacije mogu biti prozivoljne.

Isječak 4.2 prikazuje primjer izlaza iz sustava za dani ulaz iz Isječka 4.1. Sustav je znakove grupirao u dvije linije i između prvog i zadnjeg znaka u drugoj liniji je ubacio znak bjeline. Vrijednost znaka bjeline je zahtijevana vrijednost 32. Ostale informacije znaka bjeline mogle su biti proizvoljne, međutim, sustav im je dodijelio sljedeće smislenije vrijednost:

- *x* - horizontalna pozicija gornjeg desnog kuta lijevog znaka,
- *y* - vertikalna pozicija gornjeg lijevog kuta desnog znaka,
- *width* - horizontalna udaljenost između gornjeg desnog kuta lijevog znaka i gornjeg lijevog kuta desnog znaka,
- *height* - visina lijevog znaka.

Pod *lijevi znak* podrazumijeva se na znak koji se nalazi prije znaka bjeline, a pod *desni znak* podrazumijeva se na znak koji se nalazi nakon znaka bjeline.

```
1 {
2     "ocr_result": {
3         "lines": [
4             {
5                 "chars": [
6                     {
7                         "x": 5.50248,
8                         "y": 2.84320,
9                         "width": 12.13375,
10                        "height": 15.60966,
11                        "value": 73
12                    },
13                    {
14                         "x": 19.77133,
```

```

15         "y": 1.28793,
16         "width": 16.07777,
17         "height": 10.76925,
18         "value": 77
19     }
20 ]
21 },
22 {
23     "chars": [
24     {
25         "x": 3.19550,
26         "y": 19.67606,
27         "width": 14.94088,
28         "height": 20.78798,
29         "value": 91
30     },
31     {
32         "x": 18.13638,
33         "y": 19.67606,
34         "width": 7.81966,
35         "height": 20.78798,
36         "value": 32
37     },
38     {
39         "x": 25.95604,
40         "y": 17.30562,
41         "width": 10.64438,
42         "height": 16.60289,
43         "value": 48
44     }
45 ]
46 }
47 ]
48 }
49 }
```

Isječak 4.2: Primjer izlaza iz sustava za određivanje strukture teksta.

5. Skup podataka za testiranje

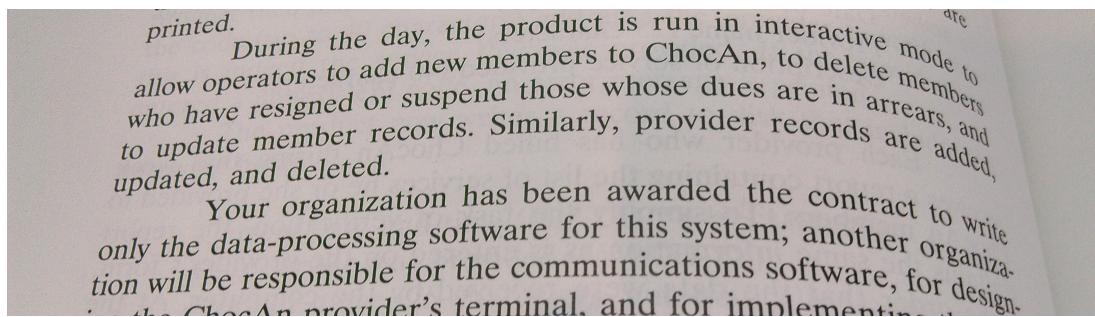
Skup podataka za testiranje (u dalnjem tekstu: *podatci*) sustava sastoji se od slike, JSON datoteka i tekstualnih datoteka. U sklopu ovog završnog rada razvijeni sustav za određivanje strukture teksta rješavati će problem određivanja strukture teksta na računima iz trgovine i sadržaja iz knjiga.

5.1. Slike

Podatci za testiranje sustava sadrže 100 slika računa (Slika 5.1) i 34 slike sadržaja iz knjiga (Slika 5.2).



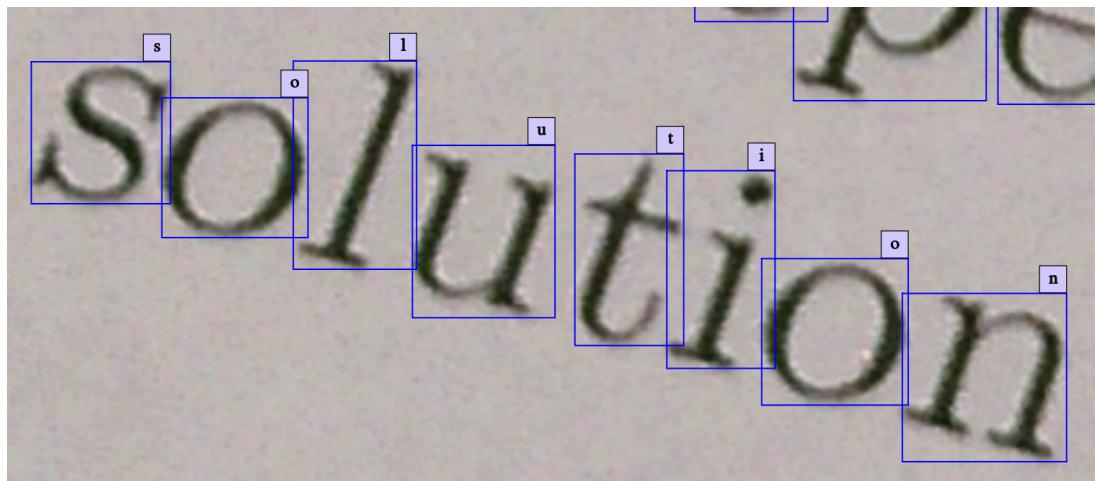
Slika 5.1: Primjer slike računa za koji će se rješavati problem određivanja strukture teksta.



Slika 5.2: Primjer slike sadržaja iz knjige za koji će se rješavati problem određivanja strukture teksta.

Svaki znak na svakoj slici je **ručno** označen i klasificiran. Na slikama računa označeno je ukupno 85068 znakova, a na slikama sadržaja iz knjiga označeno je ukupno 25092 znaka. Ručno označeni podatci oponašaju rezultate OCR sustava koji predstavljaju ulaz u sustav za određivanje strukture teksta.

Omeđujući pravokutnici označenih znakova predstavljaju područje koje označeni znak zauzima na slici. Stranice omeđujućih pravokutnika su uvijek paralelne sa rubovima slike. Slika 5.3 prikazuje isječak slike, sadržaja iz knjige, na kojoj su znakovi ukošeni, a stranice njihovih omeđujućih znakova paralelne su sa rubovima slike. Možemo uočiti kako je moguće da se dva susjedna omeđujuća pravokutnika preklapaju.



Slika 5.3: Primjer slike s kosim tekstrom. Vizualizacija OCR rezultata. Omeđujući pravokutnici su uvijek paralelni s rubovima slike.

5.2. JSON datoteke

Slike opisane u 5.1 **ne predstavljaju** ulaz u sustav za određivanje strukture teksta. Nakon označavanja slika podatci o označavanju svake slike se izvoze i pohranjuju u JSON datoteke. JSON datoteka u kojoj su zapisani podatci o označavanju Slike 5.1 prikazana je u Isječku 5.1. Početna tri ključa `meta`, `tags` i `crop` predstavljaju dodatne informacije o označenoj slici i mogu se zanemariti. Zbog specifičnosti sustava za označavanje slika i načina na koji on pohranjuje informacije o označavanju, svi označeni znakovi će biti smješteni u jednu liniju u nasumičnom poretku, a ta linija će biti smještena u jedan blok. Za svaki znak dostupna je informacija o Unicode vrijednosti znaka koja je smještena pod ključem `value`. Dodatno, za svaki znak dostupna je informacija o poziciji i veličini njegovog omeđujućeg pravokutnika (engl. *bounding box*). Za svaki omeđujući pravokutnik poznate su sljedeće informacije:

- x - horizontalna pozicija gornjeg lijevog kuta,
- y - vertikalna pozicija gornjeg lijevog kuta,
- $width$ - širina i
- $height$ - visina.

Kako vrijednost x omeđujućeg pravokutnika raste tako je znak bliže desnom rubu slike. Čim je vrijednost y omeđujućeg pravokutnika veća time je znak niže na slici. Sve informacije o omeđujućem pravokutniku su vrijednosti iz skupa nenegativnih realnih brojeva.

Dodatne informacije dostupne za svaki znak smještene u ključeve `font` i `quality` mogu se zanemariti.

```
1 {
2     "meta": {
3         "retailer": "Fred Meyer",
4         "validation_type": "turk-multiple",
5         "group": "ibotta-june-turk-multiple"
6     },
7     "tags": [
8         "ibotta-june-turk-multiple"
9     ],
10    "crop": {
11        "x": 90.0,
12        "y": 209.0,
13        "width": 496.0,
```

```

14         "height": 485.0
15     },
16     "ocr_result": {
17         "blocks": [
18             {
19                 "lines": [
20                     {
21                         "chars": [
22                             {
23                             "value": 42,
24                             "quality": 100,
25                             "font": "unknown",
26                             "bounding_box": {
27                                 "x": 55.678818,
28                                 "y": 1.7737274,
29                                 "width": 10.0,
30                                 "height": 12.452545
31                             }
32                         },
33                         {
34                             "value": 42,
35                             "quality": 100,
36                             "font": "unknown",
37                             "bounding_box": {
38                                 "x": 66.28284,
39                                 "y": 3.0,
40                                 "width": 10.434326,
41                                 "height": 12.565659
42                             }
43                         },
44                         // ostalih 585 znakova...
45                     ],
46                 }
47             ],
48         }
49     ]
50 }
51 }
```

Isječak 5.1: JSON datoteka s podatcima o označavanju Slike 5.1.

6. Algoritmi za određivanje strukture teksta

U ovom poglavlju trebaš napisati na koji način je podijeljen sustav za određivanje strukture teksta (*aligner* i *spacer*). Opisati način rada svakog i njihovu suradnju. Opisati što se očekuje od *alignera*, a što od *spacera*.

7. Rezultati i analiza

7.1. Metrike

Opisati metrike koje su korištene za određivanje *fitnesa* layoutera. Zašto baš ta metrika, a ne neka druga. Objasniti prednosti i mane takve metrike i koje su posljedice njezina korištenja.

7.2. Brojke

Brojke rezultata, grafovi ili tablice.

7.3. Analiza

Analiza rezultata. Zašto *aligner* radi, a zašto ne. Isto tako i za *spacer*. Pokazati gdje *aligner* griješi i analizirati zašto. Isto i za *spacer*.

8. Daljnji rad

Opisati što se može napraviti u bližoj budućnosti da rezultati budu bolji. Definitivno spomenuti potrebu za izgradnjom boljeg skupa podataka i definiranja metrike. Istaknuti prednosti takvog novog skupa podataka.

9. Dodaci

U dodacima rada opisati strukturu *test-data* direktorija. Opisati na koji način čitatelj može reproducirati rezultate i kako može isprobati rad *Layouter* na nekom svom skupu podataka. Pogledaj kako se ispravno dodaju dodaci i u kojem dijelu rada dodatak mora biti.

10. Zaključak

LITERATURA

Matthew Christy, Anshul Gupta, Elizabeth Grumbach, Laura Mandell, Richard Furuta, i Ricardo Gutierrez-Osuna. Mass digitization of early modern texts with optical character recognition. *J. Comput. Cult. Herit.*, 11(1):6:1–6:25, Prosinac 2017. ISSN 1556-4673. doi: 10.1145/3075645. URL <http://doi.acm.org/10.1145/3075645>.

Filip Gulan. Očitavanje rukom pisanih slova. Završni rad, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Unska 3, 10000 Zagreb, Republika Hrvatska, Lipanj 2016.

Gaurav Gupta, Shobhit Niranjan, Ankit Shrivastava, i R Mahesh K Sinha. Document layout analysis and classification and its application in ocr. U *Enterprise Distributed Object Computing Conference Workshops, 2006. EDOCW'06. 10th IEEE International*, stranice 58–58. IEEE, 2006.

Abdeslam El Harraj i Naoufal Raissouni. OCR accuracy improvement on document images through a novel pre-processing approach. *CoRR*, abs/1509.03456, 2015. URL <http://arxiv.org/abs/1509.03456>.

Noman Islam, Zeeshan Islam, i Nazia Noor. A survey on optical character recognition system. *CoRR*, abs/1710.05703, 2017. URL <http://arxiv.org/abs/1710.05703>.

Ivan Jurin. Višeobjektni modeli detekcije za raspoznavanje teksta dubokim učenjem. Diplomski rad, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Unska 3, 10000 Zagreb, Republika Hrvatska, Lipanj 2017.

Sukhpreet Kaur i Simpel Rani. A survey on feature extraction and classification techniques for character recognition of indian scripts. 2016.

Alex Krizhevsky, Ilya Sutskever, i Geoffrey E Hinton. Imagenet classification with

deep convolutional neural networks. In *Advances in neural information processing systems*, stranice 1097–1105, 2012.

Rayson Laroca, Evair Severo, Luiz A. Zanlorensi, Luiz S. Oliveira, Gabriel R. Gonçalves, William R. Schwartz, i David Menotti. A robust real-time automatic license plate recognition based on the YOLO detector. *CoRR*, abs/1802.09567, 2018. URL <http://arxiv.org/abs/1802.09567>.

Gurpreet S Lehal i Chandan Singh. Feature extraction and classification for ocr of gurmukhi script. *VIVEK-BOMBAY-*, 12(2):2–12, 1999.

Jisheng Liang, Jaekyu Ha, Robert M Haralick, i Ihsin T Phillips. Document layout structure extraction using bounding boxes of different entitles. In *Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on*, stranice 278–283. IEEE, 1996.

Yi-Feng Pan, Xinwen Hou, i Cheng-Lin Liu. A hybrid approach to detect and localize texts in natural scene images. *IEEE Transactions on Image Processing*, 20(3):800–813, 2011.

Hamed Saghaei. Proposal for automatic license and number plate recognition system for vehicle identification. *CoRR*, abs/1610.03341, 2016. URL <http://arxiv.org/abs/1610.03341>.

Sarah Schulz i Jonas Kuhn. Multi-modular domain-tailored ocr post-correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, stranice 2716–2726, 2017.

Shangxuan Tian, Yifeng Pan, Chang Huang, Shijian Lu, Kai Yu, i Chew Lim Tan. Text flow: A unified text detection system in natural scene images. *CoRR*, abs/1604.06877, 2016. URL <http://arxiv.org/abs/1604.06877>.

Abhishek Verma, Suket Arora, i Preeti Verma. Ocr-optical character recognition. In *7th International Conference on Recent Innovations in Science, Engineering and Management*, 2016.

Rohit Verma i Jahid Ali. A-survey of feature extraction and classification techniques in ocr systems. *International Journal of Computer Applications & Information Technology*, 1(3):1–3, 2012.

Ivo Vynckier. How ocr works, a close look at optical character recognition, 2018. URL <http://how-ocr-works.com/OCR/OCR.html>. Přistupano: 01.06.2018.

Christoph Wick, Christian Reul, i Frank Puppe. Improving OCR accuracy on early printed books using deep convolutional networks. *CoRR*, abs/1802.10033, 2018. URL <http://arxiv.org/abs/1802.10033>.

Fei Yin i Cheng-Lin Liu. Handwritten text line extraction based on minimum spanning tree clustering. U *Wavelet Analysis and Pattern Recognition, 2007. ICWAPR'07. International Conference on*, svezak 3, stranice 1123–1128. IEEE, 2007.

Xu-Cheng Yin, Xuwang Yin, i Kaizhu Huang. Robust text detection in natural scene images. *CoRR*, abs/1301.2628, 2013. URL <http://arxiv.org/abs/1301.2628>.

Weiheng Zhu, Yuanfeng Liu, i Liang Hao. A novel ocr approach based on document layout analysis and text block classification. U *Computational Intelligence and Security (CIS), 2016 12th International Conference on*, stranice 91–94. IEEE, 2016.

Sustav za određivanje strukture teksta na temelju položaja pojedinih znakova

Sažetak

Ključne riječi:

Text Layout Analysis System Based on Individual Character Positions

Abstract

Keywords: