

# Sustav za određivanje strukture teksta na temelju položaja pojedinih znakova

Herman Zvonimir Došilović

Sveučilište u Zagrebu  
Fakultet elektrotehnike i računarstva

Zagreb, srpanj 2018.

# Sadržaj

## 1 Uvod

- Optičko raspoznavanje znakova
  - Primjene optičkog raspoznavanja znakova
  - Komponente sustava za optičko raspoznavanje znakova

## 2 Određivanje strukture teksta na temelju položaja pojedinih znakova

- Željena funkcionalnost
- Suradnja s OCR-sustavom
- Skup podataka za ispitivanje
- Korištenje skupa podataka za ispitivanje

## 3 Algoritmi za određivanje strukture teksta

- Algoritmi za određivanje linija
- Algoritmi za rastavljanje riječi

## 4 Mjere točnosti algoritama

## 5 Rezultati

## 6 Zaključak

# Uvod

## Optičko raspoznavanje znakova

- Engl. *Optical Character Recognition (OCR)*



Slika 1: OCR-sustav na mobilnom uređaju. [1]

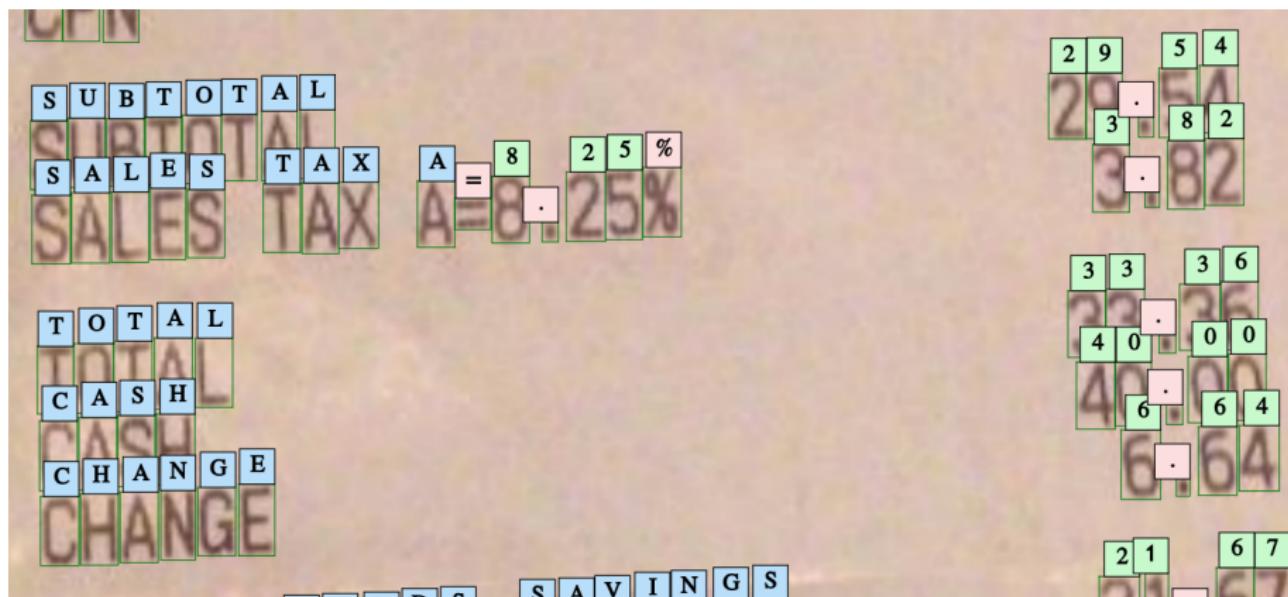
# Optičko raspoznavanje znakova

## Primjene optičkog raspoznavanja znakova (1)

- Bankovne aplikacije za mobilne uređaje
  - ▶ Plaćanje računa
  - ▶ Otvaranje bankovnog računa (npr. Zagrebačka banka, Revolut, N26)
- Turizam i hoteljerstvo
  - ▶ Prijava boravka u hotelima
- Registracija glasovanja na biralištima
- Registracija posjetitelja na raznim događajima
- Granične kontrole
- Upravljanje financijama
- Digitalizacija knjiga
- Detekcija znakova na registarskim pločicama

# Optičko raspoznavanje znakova

## Primjene optičkog raspoznavanja znakova (2)



Slika 2: Vizualizacija rezultata OCR-sustava.

# Optičko raspoznavanje znakova

## Komponente sustava za optičko raspoznavanje znakova

Optičko raspoznavanje znakova provodi se u nekoliko koraka:

- pribavljanje slike,
- predobrada,
- segmentacija znakova,
- izdvajanje značajki znakova,
- klasifikacija znakova i
- **naknadna obrada.**

# Određivanje strukture teksta na temelju položaja pojedinih znakova

Željena funkcionalnost

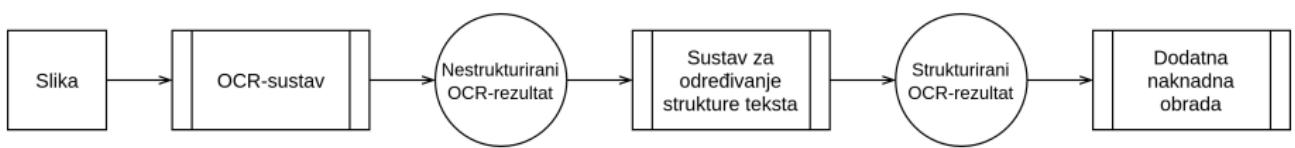
2.5 Reasoning About Efficiency

Gross reasoning about an algorithm's running time of is usually easy given a precise written description of the algorithm. In this section I will work through several examples perhaps in greater detail than necessary.

Slika 3: Vizualizacija rezultata sustava za određivanje strukture teksta.

# Određivanje strukture teksta na temelju položaja pojedinih znakova

Suradnja s OCR-sustavom



Slika 4: Suradnja OCR-sustava i sustava za određivanje strukture teksta.

# Određivanje strukture teksta na temelju položaja pojedinih znakova

Skup podataka za ispitivanje

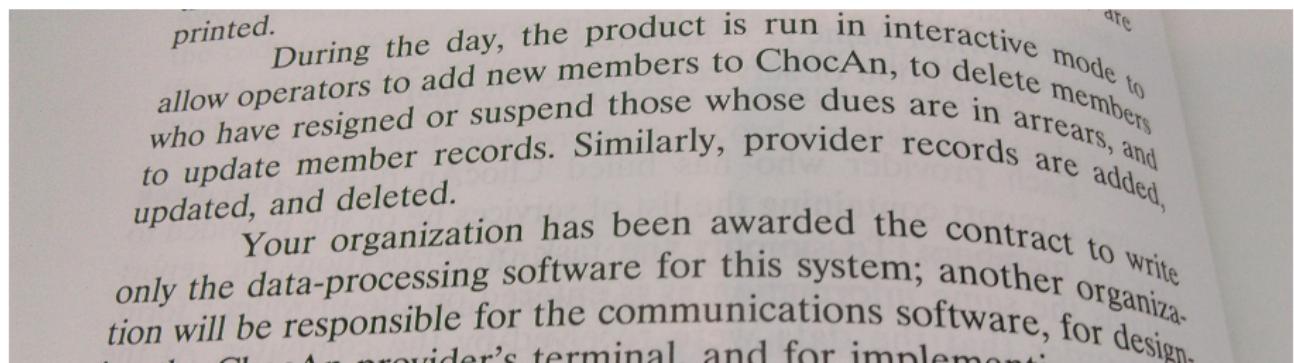
Skup podataka za ispitivanje sastoji se od:

- slika,
- ulaznih datoteka u formatu JSON i
- očekivanih izlaznih datoteka.

# Skup podataka za ispitivanje

## Slike (1)

- **Ručno** označene i klasificirane slike.
- 100 slika računa iz trgovina (ukupno 85068 znakova).
- 34 slike sadržaja iz knjiga (ukupno 25092 znaka).



Slika 5: Primjer slike sadržaja iz knjige.

# Skup podataka za ispitivanje

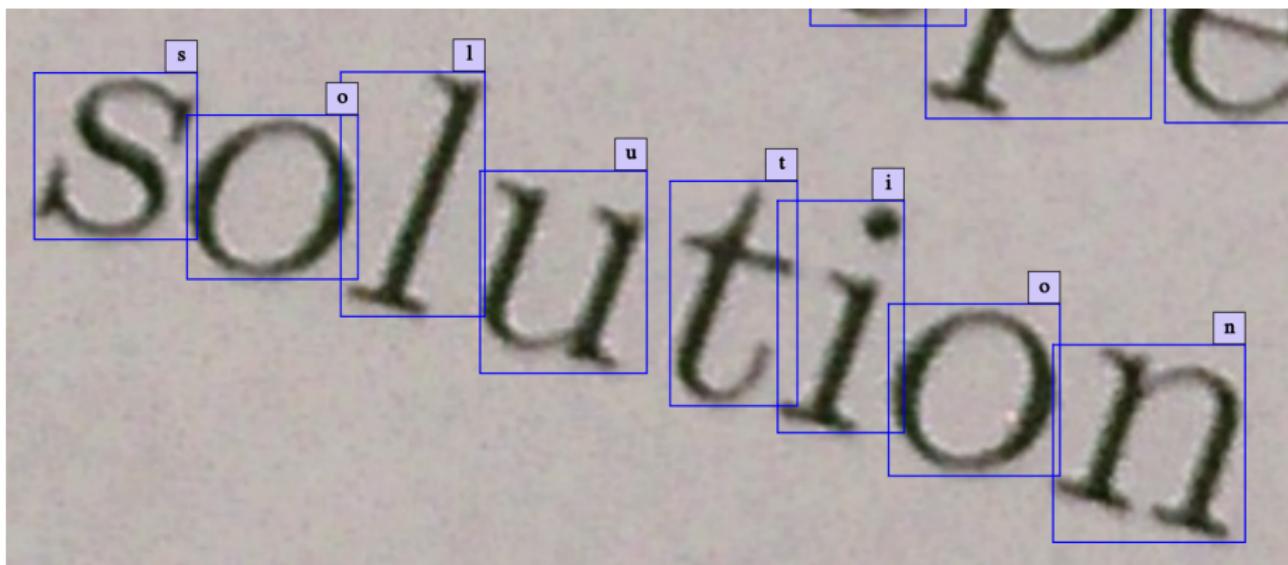
Slike (2)



Slika 6: Primjer slike računa iz trgovine.

# Skup podataka za ispitivanje

Slike (3)



Slika 7: Primjer slike s ukošenim tekstrom.

# Skup podataka za ispitivanje

## Ulagne datoteke

- Slike **ne predstavljaju** ulaz u sustav za određivanje strukture teksta.
- Ulagne datoteke u formatu JSON predstavljaju nestrukturirani OCR-rezultat.
- Za svaki znak poznate su sljedeće informacije:
  - ▶  $x$  - horizontalna pozicija gornjeg lijevog kuta,
  - ▶  $y$  - vertikalna pozicija gornjeg lijevog kuta,
  - ▶  $width$  - širina,
  - ▶  $height$  - visina i
  - ▶  $value$  - Unicode vrijednost.

# Skup podataka za ispitivanje

## Očekivane izlazne datoteke

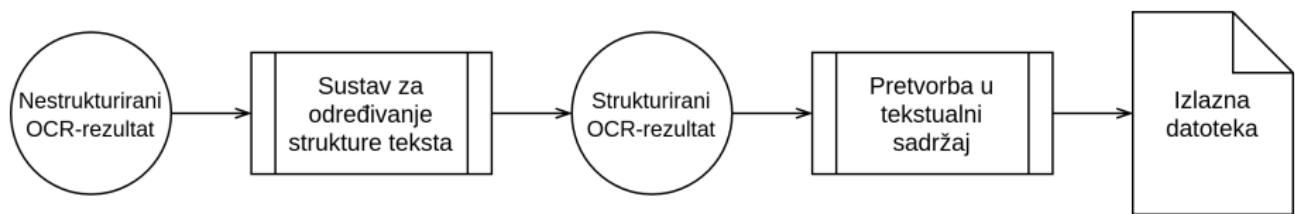
### Isječak 1: Ispravni tekstualni sadržaj slike 6.

```
1 POINTS TO $35 REWARD 8770
2 BALANCE REWARDS ACCT # ****2463
3 OPENING BALANCE 20820
4 EVERYDAY POINTS - RETAIL 410
5 CLOSING BALANCE 21230
6 ****
7 Walgreens 01875
8 ACCT 7681
9 SEQUENCE 1875220350
10 PAYMENT FROM PRIMARY
11 Get the flu shot that helps provide
12 a lifesaving vaccine to a child in need
13 Get a Shot. Give a Shot.® It's that easy
14 Learn more at the pharmacy.
```



# Skup podataka za ispitivanje

## Korištenje skupa podataka za ispitivanje (1)



Slika 8: Postupak dobivanja izlazne datoteke iz strukturiranog OCR-rezultata.

# Skup podataka za ispitivanje

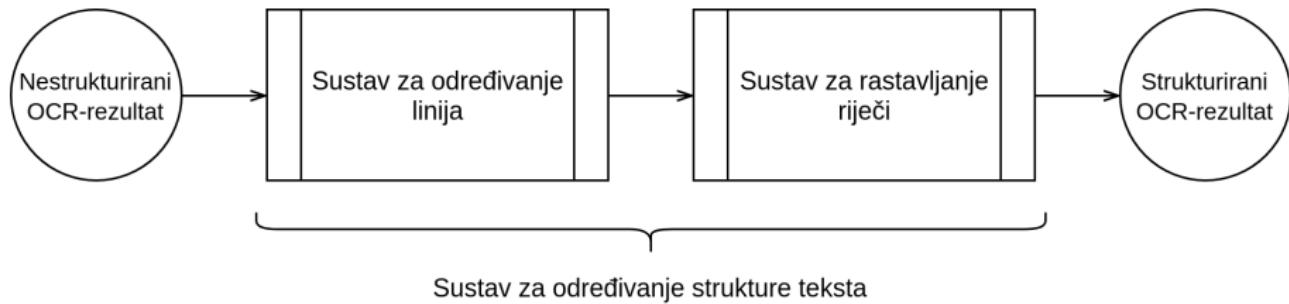
## Korištenje skupa podataka za ispitivanje (2)

Isječak 2: Pseudokôd algoritma za ispis OCR-rezultata.

```
1 def ispisi(ocrRezultat)
2     for linija u ocrRezultat.linije
3         for znak u linija
4             print(znak.value)
5         end
6         print("\n")
7     end
8 end
```

# Algoritmi za određivanje strukture teksta

## Komponente sustava za određivanje strukture teksta



Slika 9: Komponente sustava za određivanje strukture teksta.

# Algoritmi za određivanje strukture teksta

## Algoritmi za određivanje linija

Known Uses

The Interpreter pattern is widely used in compilers implemented with object-oriented languages, as the Smalltalk compilers are. SPECTalk uses the pattern to interpret descriptions of input file formats [Sza92]. The QOCA constraint-solving toolkit uses it to evaluate constraints [HHMV92].

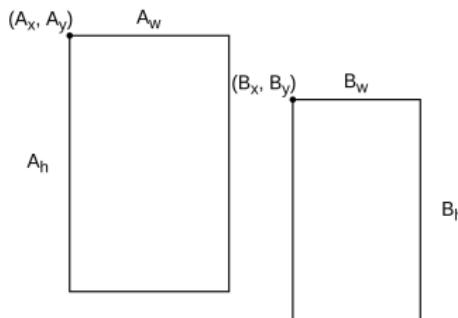
Slika 10: Vizualizacija detektiranih linija u sadržaju iz knjige.

# Algoritmi za određivanje linija

Algoritam temeljen na maksimalnom preklapanju znakova (1)

- Temelji se na pretpostavci da dva susjedna znaka koje se nalaze u istoj liniji ostvaruju maksimalno vertikalno preklapanje.
- Vertikalno preklapanje dva znaka definira se izrazom:

$$\text{overlap}(A, B) = \frac{\max(0, \min(A_y + A_h, B_y + B_h) - \max(A_y, B_y))}{\min(A_h, B_h)} \quad (1)$$



Slika 11: Omeđujući pravokutnici znakova.

# Algoritmi za određivanje linija

Algoritam temeljen na maksimalnom preklapanju znakova (2)

- Algoritam izgrađuje linije.
- Na početku, algoritam uzlazno sortira sve znakove po horizontalnoj  $x$  vrijednost.
- Promatrani znak pridružuje se onoj liniji s kojom ostvari maksimalno preklapanje:

$$I_{max} = \arg \max_{I \in L} \{overlap(A, I_{-1})\} \quad (2)$$

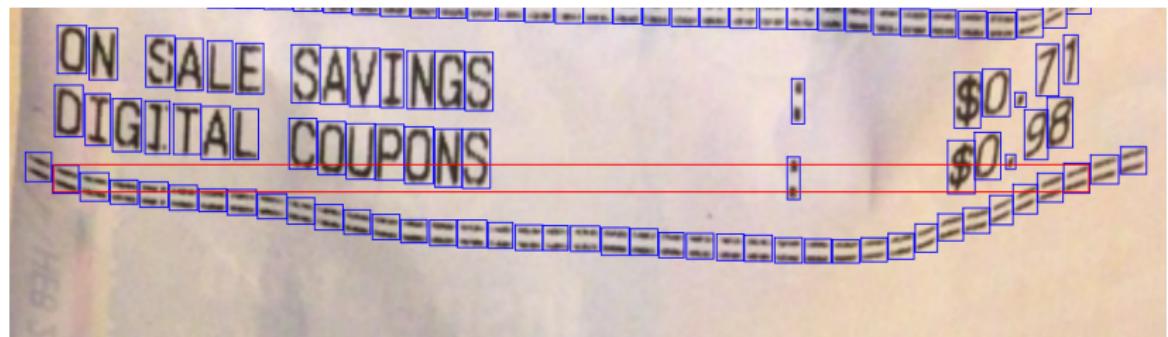
- Za preklapanje vrijednosti 0, u skup  $L$  dodaje se nova linija i promatrani znak postaje početak te linije.

# Algoritmi za određivanje linija

Algoritam temeljen na maksimalnom preklapanju znakova (3)

- Ponekad je poželjno izmjeriti preklapanje ne samo sa zadnjim znakom u liniji, nego i sa zadnjih nekoliko znakova:

$$l_{max} = \arg \max_{I \in L, i \in [1, \min(|I|, c_1)]} \{overlap(A, I_{-i})\} \quad (3)$$



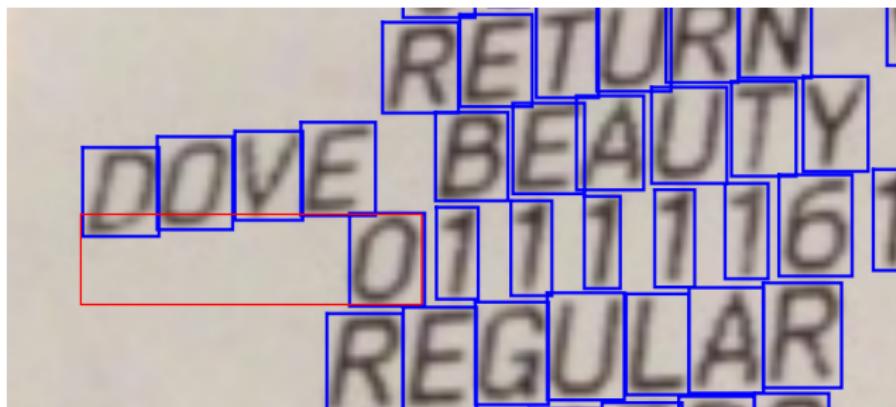
Slika 12: Valovite linije otežavaju određivanje linija.

# Algoritmi za određivanje linija

Algoritam temeljen na maksimalnom preklapanju znakova (4)

- Uvodi se dodatan uvijet koji će odlučiti hoće li promatrani znak pripasti liniji s kojom ostvaruje maksimalno preklapanje:

$$\max_{i \in [1, \min(|I_{\max}|, c_1)]} \{overlap(A, I_{\max-i})\} > c_2 \quad (4)$$



Slika 13: Preklapanje početaka dviju linija.

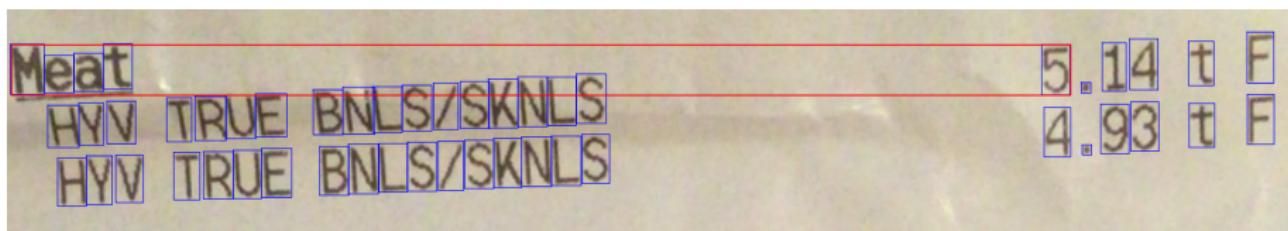
# Algoritmi za određivanje linija

Algoritam temeljen na maksimalnom preklapanju znakova (5)

- Za rješavanje problema lažnog pozitivnog preklapanja definiraju se dvije funkcije:

$$f_1(x) = \frac{1}{1 + c_3 \cdot x} \quad (5)$$

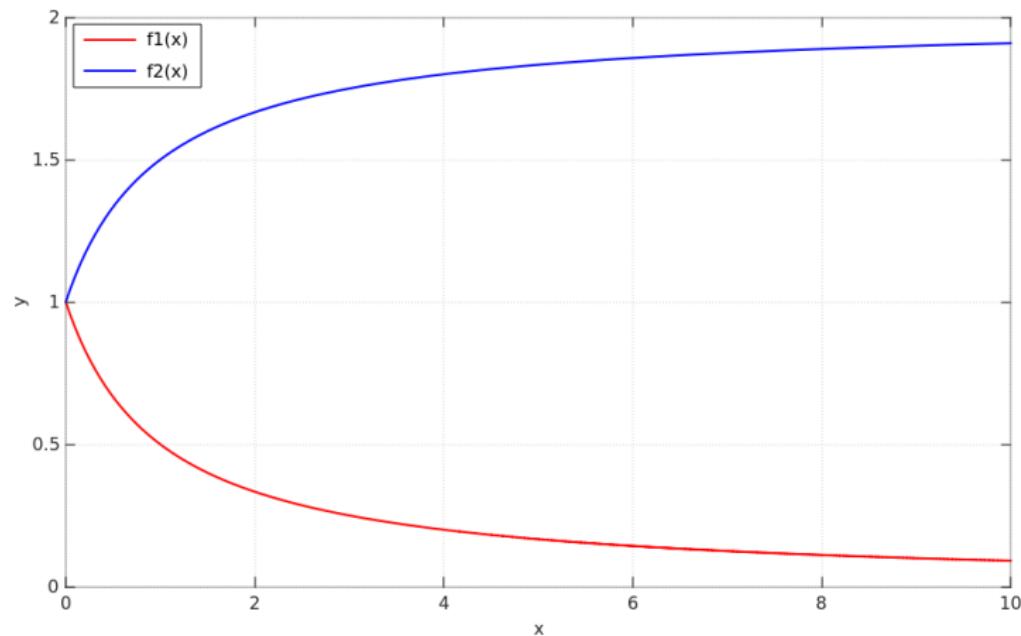
$$f_2(x) = 1 + \frac{c_4 \cdot x}{1 + c_4 \cdot x} \quad (6)$$



Slika 14: Lažno pozitivno preklapanje.

# Algoritmi za određivanje linija

Algoritam temeljen na maksimalnom preklapanju znakova (6)



Slika 15: Grafovi funkcije  $f_1$  (crveno) i funkcije  $f_2$  (plavo).

# Algoritmi za određivanje linija

## Algoritam temeljen na maksimalnom preklapanju znakova (7)

Iterirajući po skupu  $L$  računamo preklapanje sa svakom linijom. Neka znak u jednom trenutku pripada liniji  $l$  s kojom ostvaruje preklapanje  $p_l$ . U idućem koraku iteracije računamo preklapanje znaka s linijom  $k$ . Postavljamo nove uvjete za pridruživanje znaka liniji  $k$ .

Znak će pripasti liniji  $k$  ako je zadnji znak linije  $k$  bliži promatranom znaku od zadnjeg znaka linije  $l$  i ako vrijedi:

$$p_k > p_l \cdot f_1(\hat{d}_l(l_{-1}, k_{-1})) \quad (7)$$

Znak će pripasti liniji  $k$  ako je zadnji znak linije  $k$  dalji od promatranog znaka nego zadnji znak linije  $l$  i ako vrijedi:

$$p_k > p_l \cdot f_2(\hat{d}_l(l_{-1}, k_{-1})) \quad (8)$$

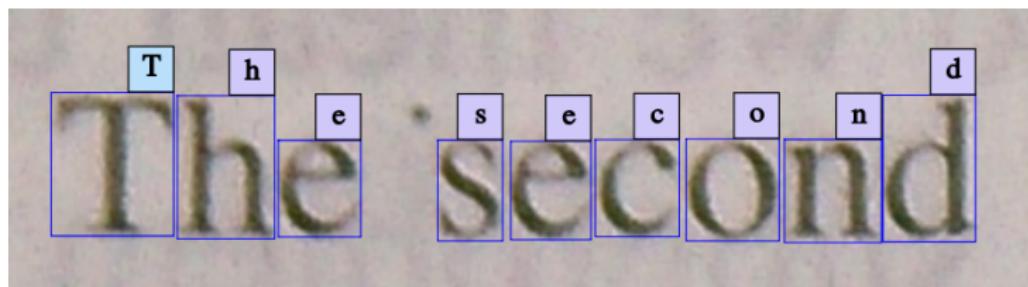


FER

# Algoritmi za određivanje strukture teksta

## Algoritmi za rastavljanje riječi

- Na ulaz primaju OCR-rezultat s grupiranim linijama.
- Trebaju ubaciti znakove bjeline između znakova za koje smatra da su završetak prethodne i početak iduće riječi.
- Razvijena su tri algoritma za rastavljanje riječi:
  - ▶ algoritam temeljen na prosječnoj širini znaka,
  - ▶ algoritam temeljen na prosječnoj relativnoj udaljenosti i
  - ▶ algoritam temeljen na prosječnoj udaljenosti centara.



Slika 16: Linija s dvije riječi.

# Algoritmi za rastavljanje riječi

Algoritam temeljen na prosječnoj širini znaka

- Najjednostavniji razvijeni algoritam.
- Temelji se na pretpostavci da je širina razmaka između riječi proporcionalna prosječnoj širini znakova u promatranoj liniji.
- Prosječna širina znaka u liniji / računa se na sljedeći način:

$$\overline{w_l} = \frac{\sum_{A \in l} A_w}{|l|} \quad (9)$$

- Algoritam ubacuje znak bjeline između znakova  $A$  i  $B$  ako vrijedi:

$$d(A, B) > \overline{w_l} \cdot c_1 \quad (10)$$

# Algoritmi za rastavljanje riječi

Algoritam temeljen na prosječnoj relativnoj udaljenosti (1)

- Temelji se na pretpostavci da je udaljenost znaka  $A$  s vrijednosti (engl. *value*)  $A_v$  proporcionalna prosječnoj udaljenosti koju svi znakovi s vrijednost  $A_v$  ostvaruju sa svojim susjedima.

# Algoritmi za rastavljanje riječi

Algoritam temeljen na prosječnoj relativnoj udaljenosti (2)

ETOPRUCCI	SNDWCH PEP	8.81 A *
ETOPRUCCI	HARD SALAMI	9.34 A *
FL HUT PEPPER CHS	M	2.40 A *
YLW AMERICAN CHSE	M	2.40 A *
FROZEN/DAIRY		
TGIF CHED CH POPPERS	M	7.98 A *
2 @ 3.99		
GROCERY		
SUNCHIPS GARD SALSA	M	3.29 A *
1 @ 2 FOR 6.58	M	3.29 A *
SUN CHIPS FRCH ONION	M	3.29 A *
1 @ 2 FOR 6.58	M	3.29 A *
FL BOLD SNACK MIX		3.58 A *
2 @ 1.79		2.99 A *
WISE ONION GARLIC	M	5.79 B *
12PK CN DT A&W R/BR	M	
PRODUCE		
CLEM TANGERINES 3LB	M	4.99 A *
Savings		
You saved:		
SUNCHIPS GARD SALSA		-0.29
SUN CHIPS FRCH ONION		-0.29
WISE ONION GARLIC		-1.11
TGIF CHED CH POPPERS		-0.60
CLEM TANGERINES 3LB		-1.00
12PK CN DT A&W R/BR		-0.80
FL HUT PEPPER CHS		-0.80

Slika 17: Slika računa iz trgovine.

# Algoritmi za rastavljanje riječi

## Algoritam temeljen na prosječnoj relativnoj udaljenosti (3)

- Skup svih susjeda znaka  $A$  definiramo kao skup svih znakova  $B$  koji su različiti od  $A$ , koji pripadaju istoj liniji kao i  $A$ , i za koje vrijedi:

$$\hat{d}_c(A, B) < c_1. \quad (11)$$

- Skup svih susjeda vrijednosti  $v$  definiramo kao:

$$s(v) = \bigcup_{A \in C} \{S(A) | A_v = v\} \quad (12)$$

- Prosječna udaljenost između znakova  $A$ , za koje vrijedi  $A_v = v$ , računa se na sljedeći način:

$$\overline{d}_c(v) = \frac{\sum_{B \in C, B_v=v} \left[ \sum_{D \in S(B)} d_c(B, D) \right]}{|s(v)|}$$

# Algoritmi za rastavljanje riječi

## Algoritam temeljen na prosječnoj relativnoj udaljenosti (4)

- Algoritam ubacuje znak bjeline između znakova  $A$  i  $B$  ako vrijedi:

$$d_c(A, B) > \overline{d}_c(A_v) \cdot c_2 \quad \vee \quad d_c(A, B) > \overline{d}_c(B_v) \cdot c_2 \quad (14)$$

# Algoritmi za rastavljanje riječi

## Algoritam temeljen na prosječnoj udaljenosti centara (1)

- Temelji se na pretpostavci da je širina razmaka između riječi proporcionalna s prosječnom udaljenosti centara između susjednih znakova.
- Skup svih susjeda znaka  $A$  definira se kao i u algoritmu temeljenom na prosječnoj relativnoj udaljenosti.
- Prosječna udaljenost centara između susjednih znakova u liniji  $I$  definira se na sljedeći način:

$$\overline{d}_c(I) = \frac{\sum_{A \in I} \left[ \sum_{B \in S(A)} d_c(A, B) \right]}{\left| \bigcup_{A \in I} S(A) \right|} \quad (15)$$

# Algoritmi za rastavljanje riječi

Algoritam temeljen na prosječnoj udaljenosti centara (2)

- Algoritam ubacuje znak bjeline između znakova  $A$  i  $B$  ako vrijedi:

$$d_c(A, B) > \overline{d}_c(l) \cdot c_2 \quad (16)$$

# Literatura

[1] Microblink Ltd. DeepOCR Technology, 2018. URL

<https://microblink.com/technology>. Pristupano:  
03.07.2018.