

Софийски университет “Св. Климент  
Охридски”

Факултет по математика и  
информатика



Проект по “Обработка на изображения”

на тема “Сегментиране и нормализиране на символи от  
текстово изображение - по размери и интензитет

”

Изготвен от: Петър Дамянов, №: 9MI3400060, спец. Изкуствен Интелект, I  
курс

# Съдържание:

1. Приложение.
2. Сегментиране на изображение.
  - Сегментиране на редове
  - Сегментиране на думи
  - Сегментиране на символи
  - Сегментиране на обединени символи
  - Сегментация от тетрадка
3. Литература

## 1. Приложения.

В индустрията Оптичното разпознаване на символи(OCR) намира голямо приложение в разпознаването на сериини номера.

Сериен номер - е уникален осемцифрен номер, позволяващ да се идентифицира всяко периодично издание, независимо от това къде е издадено, на какъв език и на какъв носител. Серииният номер се състои от осем цифри, разделени с дефис на две четирицифрени числа. Последната цифра може да е от 0 до 9 или X и служи за контролна цифра.



Заради добрият формат на серрините номера алгоритъма би бил много опростен и оптимизиран. От качеството на снимката би зависело нивото на нужно подобряване на снимката преди сегментирането.

Сериини номера на части.

Те обикновено се гравират върху самата част и е нужна предварителна обработка за да се достигне желаното ниво за сегментация

Пример за сериен номер на двигател.



### 1.1.2 Приложения печата.

Оптичното разпознаване на символи е широко използван за въвеждане на печатни данни от хартия или файл, включително от лични документи, фактури, банкови извлечения, компютърни разпечатки, визитки, поща.

Това е често срещан метод на запис на печатни текстове, за да може текстът да бъде редактиран с текстов редактор, да се търси в него, да се съхранява по-компактно, да се показва онлайн, както и да се използва в компютърни програми, като автоматизиран превод, конвертиране на текст към говор.

State of Alabama Unified Judicial System Form C-10 Rev. 6/88		<b>WRIT OF EXECUTION</b>		Case Number CV 02-5212	
IN THE <u>CIRCUIT</u> COURT OF <u>JEFFERSON</u> , ALABAMA (Circuit or District)		(Name of County)			
<u>WADE TUCKER, ET AL</u> Plaintiff		<u>RICHARD M. SCRUSHY</u> Defendant			
Home Address: <u>c/o John Q. Somerville, Esq.</u> <u>Galloway &amp; Somerville, LLC, 11 Oak Street</u> City/State/Zip Code: <u>Birmingham, AL 35213</u>		Home Address: <u>2406 Long Leaf Street</u> City/State/Zip Code: <u>Birmingham, AL 35243</u>			
FILED IN OFFICE SEP 04 2009 ANNE-MARIE ADAMS Clerk		Date of Judgment/forfeiture: <u>June 18, 2009</u>			
		Judgment amount \$ <u>2,876,103,000.00</u>			
		Court costs <u>970.00</u>			
		Alternate property value			
		Damages/rent			
		Other <u>73,734,311.80 interest</u>			
		TOTAL \$ <u>2,949,838,281.80</u>			
TO ANY LAW ENFORCEMENT OFFICER OF THE STATE OF ALABAMA: You are ordered to perform the action specified.					
<input type="checkbox"/> Seize the property described below which is in the possession of _____ and restore to _____ If this property is not available, seize and sell any personal and real property of _____ for the alternate value of the property. <input type="checkbox"/> Exemptions as to Personal Property waived.					
<input type="checkbox"/> Restore to _____ the described property now in the possession of _____ Collect \$ _____ for detention of the property.					
<input checked="" type="checkbox"/> Seize any real or personal property of Defendant, Richard M. Scrushy, located in Jefferson County, Alabama that will satisfy the total monetary value specified above. <input type="checkbox"/> See description for exemption.					
<input type="checkbox"/> Exemption as to personal property waived.					
<input type="checkbox"/> Hold until further court action <input type="checkbox"/> Sell and return					
<input checked="" type="checkbox"/> Sell property described below previously seized and being held by you.					
<input type="checkbox"/> Collect from _____ the court cost amount. If cash cannot be collected, seize and sell any real or personal form which can be made the sum of the costs.					
Description: <u>Items of personal property set out on Exhibit "A", attached hereto and incorporated herein, located at 2406 Long Leaf Street, Birmingham, AL 35243</u>					
YOU ARE TO MAKE RETURN OF THIS EXECUTION AND EXPLAIN BELOW HOW YOU PERFORMED THE SPECIFIED ACTION.					
Date issued: <u>SEP - 4 2009</u>		By: <u>Anne-Marie Adams</u> Clerk			
Exemption Date _____ Remarks: _____					
Sheriff _____			By Deputy Sheriff _____		
COURT RECORD: Original		ADDRESSEE: Copy			

Добра практика е документите да се сканират защото това по изчистени изображения.

## 2 Сегментиране на изображение

Сегментиране определение: Сегментирането на изображение представлява разделяне на изображението на неговите части.

Сегментирането се прави в 3 основни стъпки:

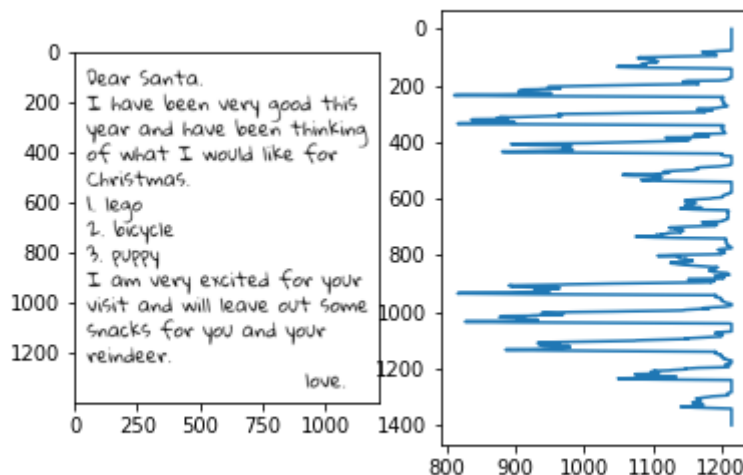
- Сегментиране на редове
- Сегментиране на думи
- Сегментиране на символи

### Сегментиране чрез хистограма.

След като изображението премине през бинаризация, то се разделя на черни и бели пиксели. Тези два типа пиксели формират Полезни пиксели и Фонови пиксели. Тези пиксели са бели или черни зависимост нашият избор. Ако сме избрали бели пиксели на черен фон полезните пиксели са бели а фоновете черни и обратното в обратния случай.

В този метод ние броим броят на фонови пиксели по редове.

### Получаване на хистограма



Ако искаме ползваме за фонови пиксели черно и полезни бяло:

```
horizontal_hist = np.sum(img,axis=1,keepdims=True)/255
```

Връща матрица която е с размер големината на изображението по оста която проверява и ни казва колко полезни пиксела е проброило на този ред. axis = 1 му казва по коя ос да брой, а 255 е цвета на бялото.

За фонови пиксели бяло и полезни черно:

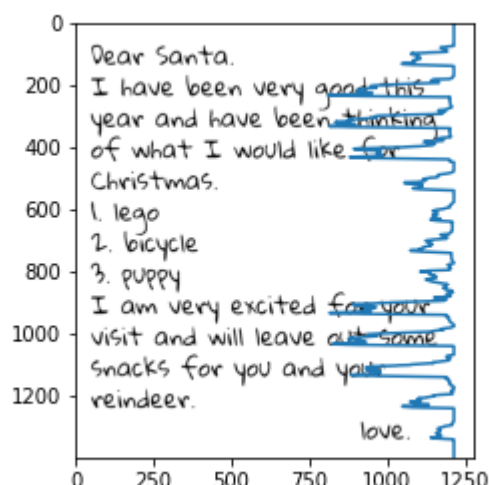
```
horizontal_hist = img.shape[1] - np.sum(img,axis=1,keepdims=True)/255
```

Номер на колони - номера на белите пиксели.

## Изчертаване на хистограма:

В изображенията виждаме как се сравнява хистограмата с изображението от нея се виждат две неща:

- Върховете съответстват на редовете които съдържат символи
- Падините съответстват на празни редове“



## Разделяне на изображението

След построяване на хистограмата можем да използваме ниските области в нея за разделяне на изображението.

Намиране на най - висока стойност(стойността с мин полезни пиксели)

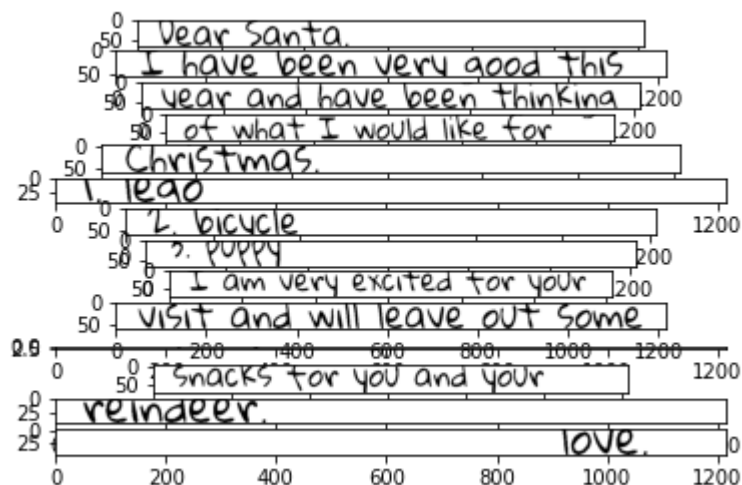
Използване на най - висока стойността като разделител. В едно перфектно разделено изображение можем без проблем да се използва тази променлива, но в повечето случай това няма да е така. Трябва да се избере променлива която да не отрязва полезните части, но и да не добавя множество безполезни защото е засякло че в този ред има 1 пиксел например.

В този пример е разделило изображението като там където са били празни редове е запълнено с черен цвят.



Разделяне на изображението на редове.

Трябва да знаем от къде до къде реално имаме полезна информация от изображението. Използваме списък от списъци в които има номерата на начален и краен ред на частта от изображението което носи полезна информация. След това правим една допълнителна проверка за да изчистим нежелани редове ако са се получили.



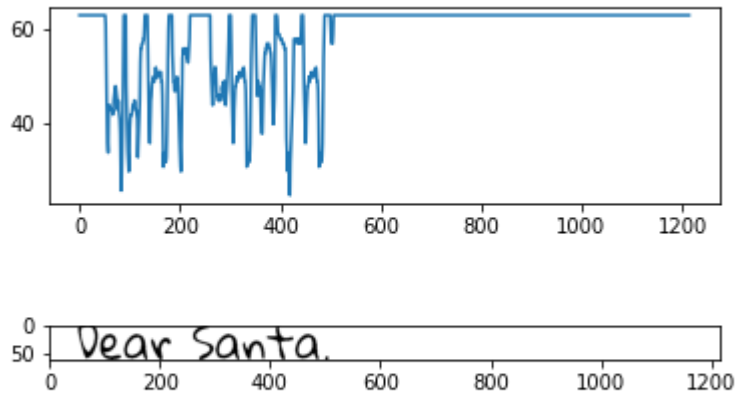
## Сегментация на думи и Сегментация на символи

Сегментацията на думи и символи следва същата процедура като сегментацията на редове с някои малки разлики.

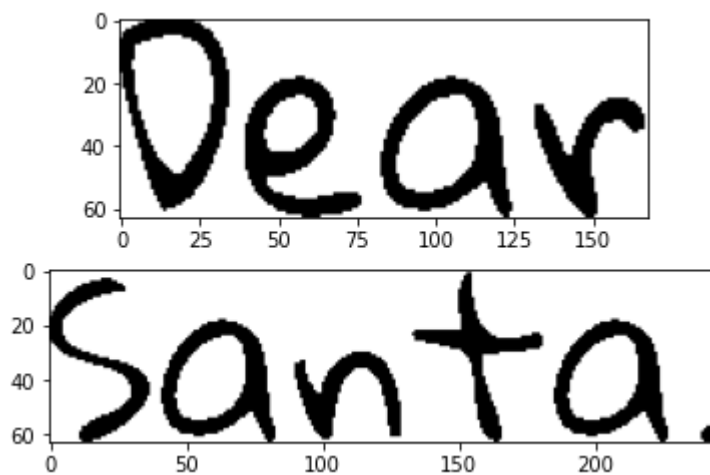
Разлика в Получаване на хистограма:

Главната разлика е че се обръща оста на смятане.

```
vertical_hist = np.sum(img, axis=0, keepdims=True) / 255
vertical_hist = img.shape[0] -
np.sum(img, axis=0, keepdims=True) / 2
```



Един добър подход към разделянето на думи е намирането на интервалите (на празни пространства). И след това използване на края на интервала и началото на другия за границите на дума.

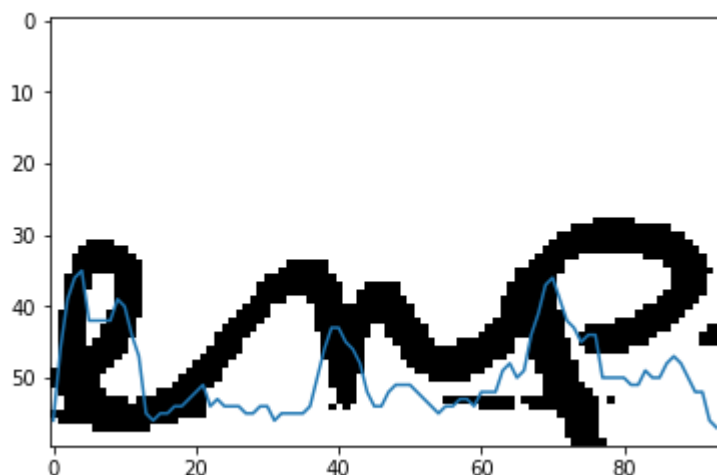


Разделянето на символи следва същата процедура, но използва най - голяма стойност на фоните пиксели за разделител.



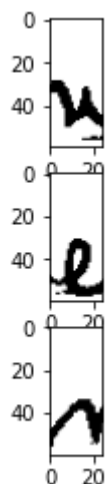


## Обединени символи



Един вариант е добър вариант е да се повтаря горният алгоритъм докато не се раздели напълно. В този случай това би намаляло броя на най - много фонов пиксели.

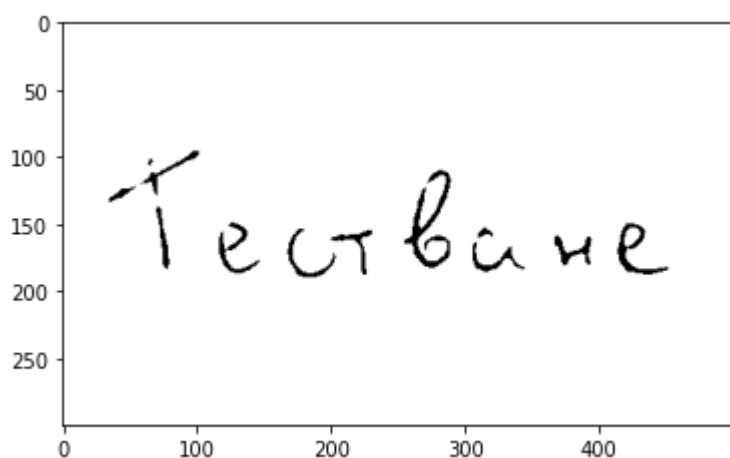
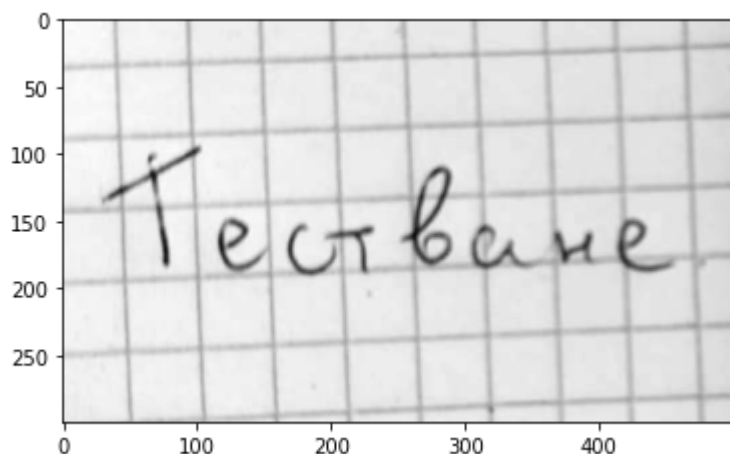
Друг вариант е използването на средна стойност за дължина символ. И разделяна на еднакви интервали на изображението.



Този метод работи добре в повечето случаи, но често води и до грешки. Очевидно е че грешката се натрупва.

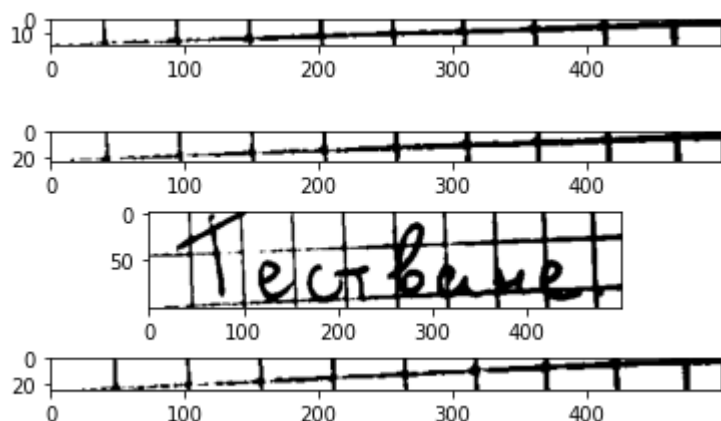
## Разглеждане на фон(тетрадка)

Един от вариантите е да се премахне фона използвайки достатъчен праг за бинаризация.

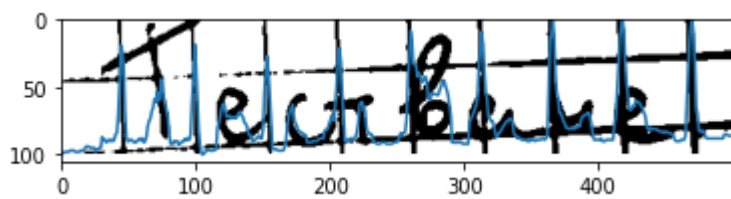


Този метод изисква фонът да е по светъл от самият текст. След премахване на фона алгоритъм си продължава по стандартен начин.

Квадратчетата могат да бъдат използвани за разделяне на символите само в перфектния случай. На практика те добавят повече шум и доста затрудняват работата на алгоритъма.



Дори след обработка се вижда че добавят много шум.



След разделяне алгоритъм за разпознаване на символ би се затруднил заради високите нива на шум

# Литература

Оптично разпознаване на символи - от Уикипедия

[https://bg.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D1%87%D0%BD%D0%BE\\_%D1%80%D0%B0%D0%B7%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B5\\_%D0%BD%D0%B0\\_%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D0%B8](https://bg.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D1%87%D0%BD%D0%BE_%D1%80%D0%B0%D0%B7%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BD%D0%B0_%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D0%B8)

Segmentation in OCR -Susmith Reddy

<https://towardsdatascience.com/segmentation-in-ocr-10de176cf373>