

# Kínai karakterek felismerése generált minták alapján

Szilvási Péter

TDK konferencia, Miskolci Egyetem, 2018. április 25.



## Vonások, vonás sorrend

- 1 A vízszintes vonások megelőzik a függőleges vonásokat.
- 2 A balra lejtő vonások megelőzik a jobbra lejtő vonásokat.
- 3 Az írásjegyek írását felülről kell kezdeni.
- 4 Az írásjegyet balról jobbra haladva építik fel.
- 5 A felülről keretezett írásjegyeknél előbb a keretet kell meghúzni.
- 6 Az alulról keretezett írásjegyeknél a keretet legvégül kell meghúzni.
- 7 A teljes keretet mindig legvégül kell bezárni.

# OCR megvalósítások

- Dokumentumok digitalizálása
- OCR részei: szkennelő fej + szoftver
- Feldolgozási szintek:
  - Alacsony szintű: zajos kép → előfeldolgozás → javított kép
  - Középső szintű: kép → szegmentálás → kép jellemzők
  - Magas szintű: jellemzők → osztályozás → osztálycímke

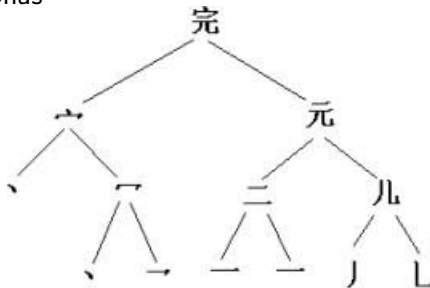


# OCR megvalósítások

- OCR típus:
  - Online
  - Offline

## Kínai karakter felismerés

- Zaj szűrés: pontszerű, elmosódás, forgatás, kontraszt
- Jellemző kivonás



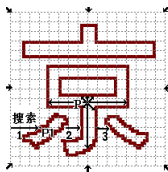
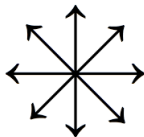
# OCR megvalósítások

Song	多	体	汉	字	识	别
Fang	多	体	汉	字	识	别
Kai	多	体	汉	字	识	别
Hei	多	体	汉	字	识	别

## Használt algoritmus

### ■ Funkció kivonás

$$d_i = \frac{l_i}{\sqrt{\sum_{k=1}^8 l_k^2}}$$



### ■ Tanítás

### ■ Tesztelés

Font	Song	Fang	Kai	Hei
Train	99.82	99.64	99.81	99.57
Test	99.71	99.50	99.80	99.09

# Minták generálása

- Tanító mintapontok előállítás (sorrend, vonal vastagság)
- Képernyő

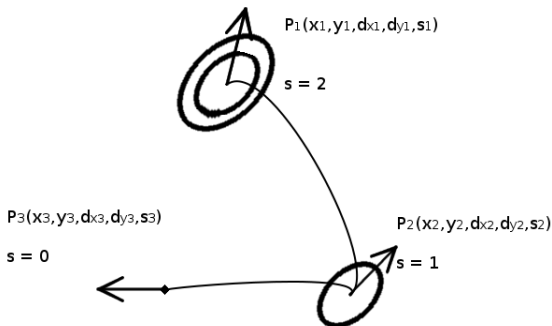
```
img = np.zeros((512,512,3),np.uint8)  
img[0:512] = (255,255,255)
```

- Ecset

```
cv2.ellipse(img,center,axes,angle,start_angle,  
end_angle,color,thickness=1,lineType=8,shift=0)
```

# Karakterek kirajzolása

- 1 Poligonos közelítés
- 2 Procedurális rajzolás



■ Hermit ív:  $H(u) = a_0 u^3 + a_1 u^2 + a_2 u + a_3$

- 3 Pontonkénti színszámítás

# Tanítóminták zajosítása

- 1 Pontszerű: zaj mátrix ( $M \oplus N$ )
- 2 Elmosódás: normalizált szűrő, gauss szűrő, medián szűrő  
 $g(i, j) = \sum_{k, l} f(i + k, j + l) h(k, l)$
- 3 Forgatás:  $M = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ . (cv2.getRotationMatrix2D())
- 4 Vágás: `crop_img = img[y:y+h, x:x+w]`
- 5 Takarás: festő algoritmus





# Mesterséges neurális hálók

Tanítóalgorithmus, backpropagation

# Konvolúciós neurális háló

A hálózat architektúrája, használt topológia

# A háló felépítése

4.10-es ábra

# A hálózat architektúrája

5.2 ábra

# Az offline adatbázis

Kb. 3x4 karakter az 5.1 ábrából

Tesztelés módja, helyesség ellenőrzése

# A felismerés hatékonysága

Eredmények konkrét százalékokkal

# Összegzés



# Hivatkozások

**Köszönöm szépen a figyelmet!**