

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

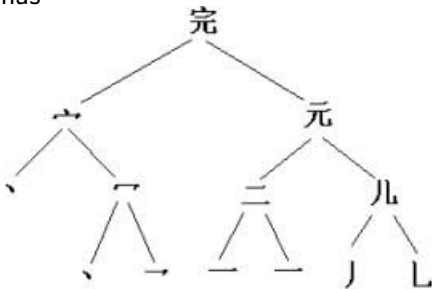


OCR megvalósítások

- OCR típus:
 - Online
 - Offline

Kínai karakter felismerés

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



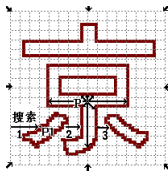
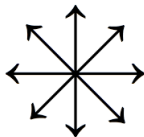
OCR megvalósítások

| | | | | | | |
|------|---|---|---|---|---|---|
| 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

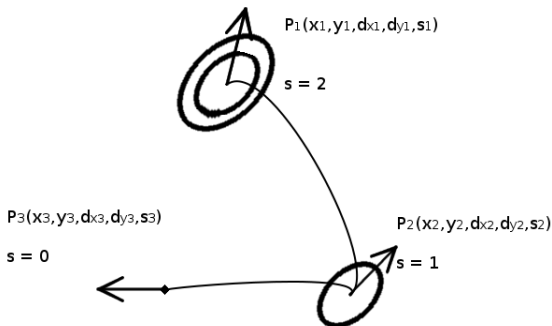
```
img = np.zeros((512,512,3),np.uint8)  
img[0:512] = (255,255,255)
```

- 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: $\mathbf{H}(u) = \mathbf{a}_0 u^3 + \mathbf{a}_1 u^2 + \mathbf{a}_2 u + \mathbf{a}_3$

- 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

- Neurális hálózatok

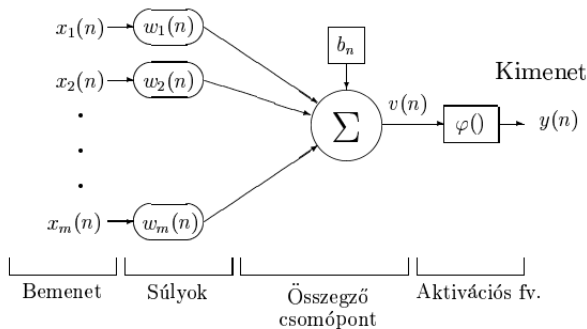
- Rétegek

- Elemei

- Backpropagation

- Hiba $E_{total} = \sum \frac{1}{2} (target - output)^2$.

- Láncszabály $\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \cdot \frac{\partial out_{o1}}{\partial net_{o1}} \cdot \frac{\partial net_{o1}}{\partial w_5}$.



Konvolúciós neurális háló

- Bevezetés
- Hálózat felépítése (konvolúciós rétegek -> hagyományos ANN)
- Bemenet->(Konvolúció->RELU->POOL)->Kimenet(FC)

| | | | | |
|-----|-----|-----|---|---|
| 1x1 | 1x0 | 1x1 | 0 | 0 |
| 0x0 | 1x1 | 1x0 | 1 | 0 |
| 0x1 | 0x0 | 1x1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

Input x Filter

| | | |
|---|--|--|
| 4 | | |
| | | |
| | | |

Feature Map

Single depth slice

| | | | |
|---|---|---|---|
| 1 | 1 | 2 | 4 |
| 5 | 6 | 7 | 8 |
| 3 | 2 | 1 | 0 |
| 1 | 2 | 3 | 4 |

x

y

max pool with 2x2 filters
and stride 2

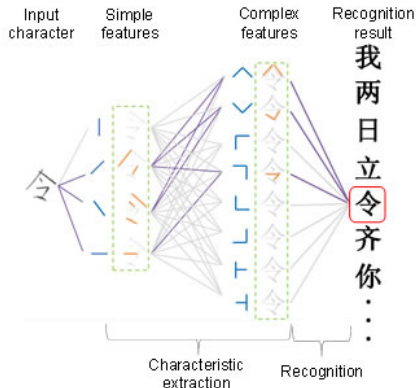
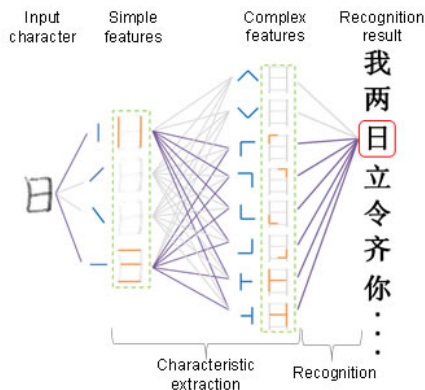
| | |
|---|---|
| 6 | 8 |
| 3 | 4 |

- Hálózat tanítás

1. Előre terjesztés
2. Veszteség számítás
3. Hiba visszaterjesztés
4. Súly frissítés

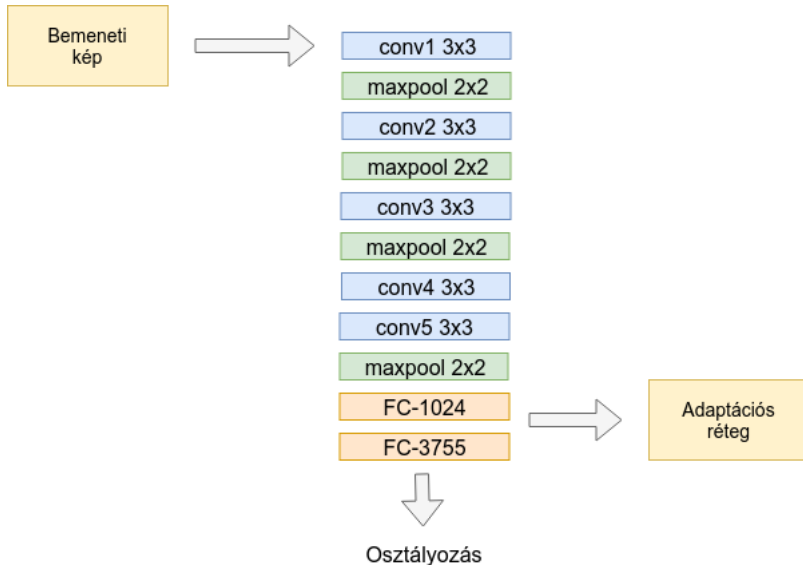
- Dropout

A háló felépítése



- Tesztelés
- Transfer learning

A hálózat architektúrája



A hálózat architektúrája

```
model.add(
    Convolution2D(1,          # filter retegek szama
                  3, 3,       # 3x3 kernel meret
                  strides=(1,1) # lepes
                  input_shape=image))
```

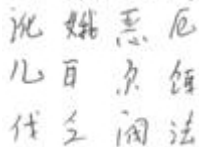
```
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.compile(loss='mean_squared_error', # Hiba
              optimizer='adam', metrics=['accuracy'])
```

```
model.fit_generator(generator=training_data,
                    steps_per_epoch=1000, epochs=10)          # Tanitas
```





Az offline adatbázis

- Adathalmaz: nyomtatott, kézzel írott, generált



- Tanító/Teszt(80/20) `random.shuffle(self.images)`
- Tanító minták változatossága
- Tesztelés módja
- Helyesség ellenőrzése

A felismerés hatékonysága

| | | | | |
|--------|---|---|---|--------|
| 0% | → |  | → | 90-95% |
| 75-80% | → |  | → | 84% |
| 68% | → |  | → | 75% |
| 87% | → |  | → | 93% |

Összegzés

- Kínai karakterek
 - stroke
 - vonásrend
- OCR
 - részei
 - használt OCR bemutatás
- Minta generálás
 - vonal vastagság
 - görbe rajzolás
 - zajok hozzáadása
- Neurális hálózatok
 - hagyományos neurális háló (ANN)
 - konvolúciós neurális háló (CNN)
- Validáció
 - adathalmaz előállítás
 - hálózat osztályozása

Hivatkozások

Köszönöm szépen a figyelmet!