Gépi látás

Diagram szkennelése

Tartalom

[*Projekt leírása:* 1](#_Toc120464709)

[*Felhasználói dokumentáció:* 1](#_Toc120464710)

[Fejlesztői dokumentáció: 1](#_Toc120464711)

# Projekt leírása:

Különböző összetevőkből álló diagramm (körök, nyilak, téglalapok…) digitalizációja fényképről, szöveges tartalom nélkül.

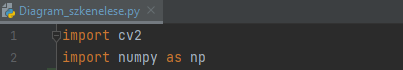
# Felhasználói dokumentáció:

A projektet elérni a <https://github.com/tornyilaci/Diagram-szkenel-se> linken lehet elérni, ahol a src mappában érhetőek el a forrás fájlok. A kódot szerkeszteni a repository klónozásával, vagy a kódrészletek kimásolásával lehetséges. A jegyzetek mappában a projekthez felhasználandó különböző ötleteket tartalmazzák. A src mappában pedig a forráskód, a tesztkód, továbbá a teszteléshez szükséges képek találhatóak. A Documentation mappában található az itt olvasható leírás.

A képen szöveg, képernyőkép, monitor, képernyő látható

Automatikusan generált leírás

A szerkesztést és a futtatást a PyCharm IDE-ben javaslom, amiben előzetesen szükséges a következő packages-eket telepíteni: numpy és opencv-python.



# Fejlesztői dokumentáció:

A fejlesztés során leginkább a különböző formák felismerése volt a cél. Ehhez az opencv-python könyvtár által nyújtott beépített függvényeket felhasználva jött létre a projekt. Ehhez a képek előzetes feldolgozására volt szükség melyben felhasználásra kerültek az alábbi algoritmusok:

A képen szöveg látható

Automatikusan generált leírás

GaussianBlur (Gauss-simítás):

A legáltalánosabb megoldási lehetőség a ***konvolúció*** használata, ami a képpontok egy környezetében elvégzett súlyozott átlag számítását jelenti. A súlyokat egy ***kxk*** méretű mátrixban adjuk meg. A mátrix rendszerint páratlan számú sorból és oszlopból áll, hogy a középső elem egyértelműen meghatározható legyen. Ez a középső elem illeszkedik az éppen vizsgált képpontra. **Nagyobb k értékek esetén erősebb a simítás hatása, mert nagyobb területen számolódik az átlag, és a végrehajtási idő is jelentősen megnő!**

**Simítás esetén elvárás, hogy kép összfényessége ne változzon.** Ezt úgy érhetjük el, ha a maszkelemek összege 1. Tört számok helyett a könnyebb átláthatóság miatt sokszor egész számokkal adjuk meg a maszk elemeit, a szükséges normalizáló osztás kiemeljük a mátrix elé. Ez csak jelölésmódbeli különbség.

A konvolúciós mátrixot Numpy tömbként definiálhatjuk, például a sima átlagoló szűrő esetén:

A képen szöveg látható

Automatikusan generált leírás

Canny (éldetektálás):

Dilate:

Erode: