Szegedi Tudományegyetem

Informatikai Intézet

SZAKDOLGOZAT

Szelepcsényi Dávid

2025

Szegedi Tudományegyetem

Informatikai Intézet

**E-K diagram elemeinek felismerése és digitalizálása**

Szakdolgozat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Készítette: |  | Témavezető: |  |
|  | Szelepcsényi Dávid |  | Dr. Kardos Péter |  |
|  | Programtervező Informatikus BSc szakos hallgató |  | adjunktus |  |

Szeged

2025

## Feladatkiírás

A feladat egy olyan program készítése, ami egy raszteres képet kap, és azon végzett műveletekkel meghatározza a diagram elemeit, majd egy XML nyelvű, vektoros rajzolóprogram fájlformátumába átkonvertálni. Ez az OpenCV Python csomag segítségével érjük el. A program képes kell, hogy legyen felismerni az egyed kapcsolat diagram elemeit, azok közt a kapcsolatokat meghatározni. A program, a PyTesseract csomag segítségével képes a szövegek felismerésére. A program futása fél-automatikus, a felhasználó képes átírni a talált szöveget, illetve a felhasználó határozza meg, hogy melyik elem számít gyenge elemnek vagy gyenge kapcsolatnak.

## Tartalmi összefoglaló

* A téma megnevezése:

E-K diagram raszteres képének beolvasása, majd a diagrammot leíró, vektorgrafikus diagram szerkesztő program által szerkeszthető fájl készítése.

* A megadott feladat megfogalmazása:

Egy olyan program készítése, amely képes E-K diagram elemeit, és a köztük lévő kapcsolatokat felismerni egy, a programnak megadott kép alapján, majd az így összegyűjtött információ alapján egy XML forráskódú fájlt állít elő

* A megoldási mód:

Python programozási nyelv használatával valósítottam meg a feladatot. Egy kép megadása után, amennyiben az megfelel bizonyos előkövetelményeknek, a program elvéges a képen előkészítési lépéseket. Majd elsőnek leolvassa az elemek adatait, ezek segítségével, illetve az eredeti kép alapján meghatározza a köztük lévő kapcsolatokat, azokat eltárolja. Ezen adatok alapján, illetve a felhasználó által megadott adatok alapján előállít egy drawio kiterjedésű fájlt.

* Alkalmazott eszközök, módszerek:

Az implementációhoz Python 3.12.3-as verziót, és OpenCV, kép kezelésre kitalált Python könyvtárat használta. A vizuális felülethez PyQt6 grafikus csomagot, a szövegek felismerésére PyTesseract függvény könyvtáratat használta. Az XML fájlok a Python beépített, erre a célra szolgáló xml csomag segítségével készültek. A szükséges számítások elvégzéséhez a program NumPy függvény könyvtárat, és a beépített math csomagokat alkalmaz.

* Elért eredmények:

A program képes felismerni az E-K diagram különböző elemeit, és az azok között fennálló kapcsolatokat. A szövegfelismerés hibája esetén a felhasználó képes javítani a hibákat, illetve képes meghatározni gyenge elemeket, kapcsolatokat, valamint megadni a diagram kulcsait.

* Kulcsszavak:

E-K diagram, Python, OpenCV, XML, PyTesseract,

## Tartalomjegyzék

[Feladatkiírás 2](#_Toc80443487)

[Tartalmi összefoglaló 3](#_Toc80443488)

[Tartalomjegyzék 4](#_Toc80443489)

[Bevezetés 5](#_Toc80443490)

[1. Téma háttere 5](#_Toc80443491)

[1.1. Digitális képfeldolgozá 5](#_Toc80443492)

[1.2. Diagrammok 5](#_Toc80443493)

[1.2.1. Diagrammok, mik azok, miért van rájuk szükség 5](#_Toc80443498)

[1.2.2. E - K diagram 5](#_Toc80443499)

[2.Használt eszközök 6](#_Toc80443505)

[2.1 Python. 6](#_Toc80443506)

[2.2. OpenCV 6](#_Toc80443507)

[2.3. PyQt6 6](#_Toc80443508)

[2.4. PyTesseract 6](#_Toc80443507)

[2.5. Numpy 6](#_Toc80443508)

[3. Diagram felismerés és Xml generálás 5](#_Toc80443495)

[3.1. Adattárolás 5](#_Toc80443496)

[3.2. Előfeldolgozás 5](#_Toc80443497)

[3.2.1. Képek tisztítása 5](#_Toc80443498)

[3.2.2. Hibajavítás 5](#_Toc80443499)

[3.3. Alakzatfelismerés 5](#_Toc80443497)

[3.3.1. Első alakzat felismerés 5](#_Toc80443498)

[3.3.2. Hibaellenőrzés 5](#_Toc80443499)

[3.4. Kapcsolatok keresése 5](#_Toc80443497)

[3.4.1. Elemek eltávolítás és képtisztítás 5](#_Toc80443498)

[3.4.2. Vonalak keresése 5](#_Toc80443499)

[3.4.3. Vonalak elemekhez kapcsolása 5](#_Toc80443498)

[3.4.4. Validáció 5](#_Toc80443499)

[3.4.5. Komplex vonalak 5](#_Toc80443498)

[3.4.6. Nyilak definiálása 5](#_Toc80443499)

[3.5. XML fájl készítése 5](#_Toc80443497)

[4. Vizuális felület és szövegfelismerés 5](#_Toc80443501)

[4.1 A vizuális felület alapjai. 5](#_Toc80443502)

[4.2.Szövegkeresés 5](#_Toc80443503)

[4.3 Gyenge elemek és kulcsok. 5](#_Toc80443504)

[5. Eredmények és hibák 5](#_Toc80443501)

[5.1 Alkalmatás használata. 5](#_Toc80443502)

[5.1.1. Főablak és szabályok 5](#_Toc80443498)

[5.1.2. Szövegfelismerő ablak és felhasználói javítás 5](#_Toc80443499)

[5.2.Elért eredmények 5](#_Toc80443503)

[5.3.Hibák és megoldások 5](#_Toc80443503)

[6. Összegzés 5](#_Toc80443501)

[Irodalomjegyzék 7](#_Toc80443509)

[Nyilatkozat 8](#_Toc80443510)

[Köszönetnyilvánítás 9](#_Toc80443511)

# BEVEZETÉS

Miért fontosak a diagrammok? Mint programozók az első lépes a kódoláshoz, vagy egy projekthez egy terv kialakítása. Itt jönnek elő a diagrammok. Használatuk segít vizualizálni a folyamat elemeit, azokat kialakítani, megtervezni, és javítani. A diagrammok arra is szolgálnak, hogy egyszerű leírást adjanak akár a velünk dolgozóknak, akár olyanoknak, akik nincsenek tisztában az egész eddigi munkafolyamattal. Természetesen, ehhez kell egy bizonyos megértés a diagram felépítéséhez, annak elemeihez, és azok jelentéséhez. Mielőtt elkezdnénk készíteni egy diagrammot, fennállhat a kérdés, hogy ezt milyen eszközzel is tegyük. Csináljuk kézzel esetleg, vagy erre kifejlesztett rajzoló programmal? Ha rajzoló programmal készítjük, onnan kinyomtatni egyszerű már, illetve a későbbi szerkesztés is sokkal könnyebb, de fordítva nem ez a helyzet. Szakdolgozatom célja ennek a folyamatnak a megkönnyítése, lehetővé tevése.

Az én esetemben az egyed – kapcsolat diagrammon dolgoztam, aminek a főbb célja az adatbázis modelljének az elkészítéséhez egy vizuális ábrát adni. A program ennek a diagram típusnak az elemeinek felismerésére lett beállítva. Amennyiben bizonyos feltételek teljesülnek, a program egy egyszerű Python grafikus felületen keresztül képes a diagramról készült képet feldolgozni, azt vektorgrafikus rajzoló és szerkesztő program által kezelhető formába hozni. Ebben a felhasználónak is szerepe van, bizonyos beállításokat neki kell megtennie.

A dolgozatom első részében szeretnék egy bevezetést nyújtani a téma hátterébe, a digitális képfeldolgozásba és a E-K diagram felépítésébe. Ezek szükségesek ahhoz, hogy a témát megértsük. Ezt követően bemutatom a használt eszközöket, amikre szükségem volt a program megírásához, így meglesz minden háttértudás a folyamat megértéséhez. A harmadik részben magáról a kódról beszélünk, azon belül is a képfeldolgozásról, és annak segédeszközeiről. Itt elsőnek az adattárolás, majd a folyamat maga következik. A képfeldolgozás három főbb részből áll. Előfeldolgozás, alakzatfelismerés, és kapcsolat keresés. Végül az így megszerzett adatkból felépül a vektorgrafikus kép. A negyedik rész szól a felhasználó által kezelt részről. Ez magába foglalja a szövegfelismerést, a speciális kapcsolatok és elemek beállítását, illetve itt szeretném magának a vizuális felületnek a felépítését, működését ismertetni. Az ötödik rész az használati útmutatóról, az eredményekről és a hibákról szól. Érdekesnek tartom az, hogy milyen hibák jöttek fel a szakdolgozatom folyamán, és az eredmények megértése, ezek nélkül nem teljes. Végül a hatodik fejezeben összefoglalom az eddig beszélteket.

**1 A TÉMA HÁTTERE**

Ebben a fejezetben a szakdolgozatom témájának két háttérterületébe, a digitális képfeldolgozásba, és a diagrammok, azon belül is az egyed – kapcsolat diagram, megértésébe nyújtok betekintést. A digitális képfeldolgozással kezdünk, ami egy elég tágas téma, és vizuális és absztrakt jellege miatt igen változatos. Majd a diagrammok szükségletessége, szerepe, és felépítésük, és maga az E-K diagram ismertetése következik.

* 1. **Digitális képfeldolgozás**

A digitális képfeldozás

## Irodalomjegyzék

## Nyilatkozat

Alulírott ………………..………… szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet ……………………….. Tanszékén készítettem, …………………….…….…… diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat / diplomamunkámat a Szegedi Tudományegyetem Diplomamunka Repozitóriumában tárolja.

Dátum

Aláírás

## Köszönetnyilvánítás

Mellékletek

A.

B.

C.