

# Pénzérme keresés python programozással

Sós Nikolett

Mérnök Informatikus BSc, 2. évfolyam

Széchenyi István Egyetem, GIVK, Informatika Tanszék

2018. december 13.

# Kivonat

Az általam választott feladat pénzürmék felismerése és megtalálása volt egy megadott fényképen.

A megoldó algoritmus python nyelven lett megvalósítva, opencv, matplotlib és numpy könyvtárak kiegészítésével.

Munkám során foglalkoztam mintakereséssel és foltdetektáló algoritmus létrehozásán is.

**Kulcsszavak:** pénzürmé keresés, python programozás, mintakeresés, foltdetektálás

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>3</b>
1.1. Python programozás bemutatása . . . . .	3
1.1.1. Opencv és numpy könyvtárak . . . . .	3
1.2. Feladvány bemutatása . . . . .	4
<b>2. Feladat megoldása</b>	<b>5</b>
2.1. Mintakeresés . . . . .	5
2.2. Foltdektálás . . . . .	5
<b>3. Összefoglalás</b>	<b>6</b>
<b>Irodalomjegyzék</b>	<b>6</b>

# 1. fejezet

## Bevezetés

Beadandóm során a pénzérme számláló program megoldásával foglalkoztam. A feladvány megoldásához használt python nyelvet és a hozzá használt könyvtárakat az 1.1. fejezetben mutatom be. A megoldott feladat bemutatása az 1.2. alfejezetben olvasható. A 2. fejezetben a programom első felét, azaz a mintakeresést fogom bemutatni. A 3. fejezetben pedig a feladatom fő része, a folt-detektálás található meg.

### 1.1. Python programozás bemutatása

A pythonról, mint programozási nyelvről az 1980-as évek végétől beszélhetünk. A fejlesztést Guido van Rossum kezdte meg 1989-ben az amszterdami Matematikai és Informatikai Központban, a CWI kutatóközpontban. A név öt brit színésztől származik, akik a csoportjukat hívták így (Monty Python-csoport, közkeletű nevén The Pythons), valamint a több részes műsoruk címében is szerepelt a python szó. Az első verzió a 0.9.0.-es 1991 februárjában került a közönség elé, 1994-ben jelent meg az 1.0 verzió, a 2.0-ás, és végül a 3.0-ás verzió (amit én is használok) 2008-ban látott napvilágot[1].

A python magasszintű programozási nyelvként szerepel a köztudatban, azonban itt a futási gyorsaság helyett a programozó munkájának megkönnyítése kerül előtérbe.

#### 1.1.1. Opencv és numpy könyvtárak

Programom működéséhez két könyvtárat szükséges meghívni, az opencv-t és a numpy-t.

Az opencv kimondottan a számítógépes, azaz a gépi látásra kell kitalálva, megkönnyítve a vizuális elemek feldolgozását, értelmezését és felhasználását. Ez a könyvtár python-on kívül más magas szintű programozási nyelvekben is megtalálható, mint például a C++-ban is[2].

A numpy python nyelvre lett kifejlesztve, hogy a programban numerikus számításokat is végre tudjunk hajtani, tehát a több dimenziós tömbök és mátrixok használatát teszi lehetővé. Programomban a képek vannak pixelenként tárolva egy két dimenziós tömbben, így számos művelet végrehajtása lehetséges egyszerű módszerekkel[3].

## 1.2. Feladvány bemutatása

A beadandóm témája a következő volt: Pénzérme számláló alkalmazás: Egy bekért képen megállapítani, hogy hány pénzérme található meg rajta.

## 2. fejezet

# Feladat megoldása

A programom elején meghívom a használt könyvtárakat és opencv-vel beolvasatok két képet. Az egyik kép amelyiken össze akarjuk számolni az érméket, a másik képre a mintakeresésnél fogok részletesebben kitérni.

### 2.1. Mintakeresés

A beadandóm a továbbiakban két fő részből áll. Az első, ahogy a címben is áll az a minta keresés. Ebben a részben lesz jelentősége a már korábban is megemlített második képnek. Ezen egy darab érme látható melynek jellemző pontjait szeretném megtalálni a másik vizsgált képen is, melyen több érme is van. Ezt egyszerű Bruce-Force algoritmussal hajtom végre. Ez a programrész úgy dolgozik, hogy mindkét képen megkeresi a jellemző pontokat és ha egyezést talál, akkor összeköti őket, majd kimenetben ábrázolja ezeket a kapcsolatokat. Jelenleg az első 10 találatot jeleníti meg az algoritmus.

### 2.2. Foltdetektálás

A második fő rész a pénzérmék megszámlálása. Ezt az eredeti kép átszerkesztett verzióján hajtom végre. Gauss szűrővel a zajokat megszüntetem, majd "Otsu threshold-al" az egész képet két színnel jelenítem meg. Az egyik szín a háttér, a másik pedig a pénzérmék színe. Ezek után eróziót és dilataciót is használok az esetleges kép hibák kijavítására. A következő lépés a "SimpleBlobDetector" paramétereinek beállítása, majd ezekkel a konkrét foltdetektálás végrehajtása. Az érzékelt foltokat a kimeneten pirossal bekarikázza a program, majd pedig "len" függvénnyel meg is kapjuk ezeknek a számát amiről e-mailben is értesítést kapunk.

## 3. fejezet

# Összefoglalás

Összefoglalva munkám során megismerkedtem a python programozói nyelvvel és a hozzá használt könyvtárakkal, illetve mintakereső és foltdetektáló algoritmusok létrehozásával foglalkoztam.

# Irodalomjegyzék

[1] *[https : //hu.wikipedia.org/wiki/python\\_\(programozasi\\_nyelv\)](https://hu.wikipedia.org/wiki/python_(programozasi_nyelv))*.

[2] <https://en.wikipedia.org/wiki/opencv>.

[3] <https://hu.wikipedia.org/wiki/numpy>.