



M Ű E G Y E T E M 1 7 8 2

**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
ETT Tanszék

Hubay Csongor

**CSIPERKEGOMBA SZEDÉSÉNEK  
TECHNOLÓGIAI FEJLESZTÉSE  
INTELLIGENS  
KÉPFELDOLGOZÁSSAL**

Tudományos Diákköri Konferencia Dolgozat

KONZULENSEK

Dr. Géczy Attila - egyetemi docens - BME ETT

Dr. Geösel András - egyetemi docens, intézetigazgató  
MATE, Kertészettudományi Intézet, Zöldség- és Gombatermesztési Tanszék

Dr. Farkas Csaba - adjunktus - BME ETT

BUDAPEST, 2022

# 1 Tartalomjegyzék

<b>1</b>	<b>TARTALOMJEGYZÉK .....</b>	<b>2</b>
<b>2</b>	<b>RESUME.....</b>	<b>4</b>
<b>3</b>	<b>ABSTRACT.....</b>	<b>5</b>
<b>4</b>	<b>BEVEZETÉS, A GOMBATERMESZTÉS AUTOMATIZÁLÁSÁNAK INDOKOLTSÁGA.....</b>	<b>6</b>
<b>5</b>	<b>CÉLKITŰZÉS.....</b>	<b>8</b>
<b>6</b>	<b>MÓDSZERTAN.....</b>	<b>8</b>
<b>7</b>	<b>ELTERJEDT CSIPERKEGOMBA SZEDÉSI MÓDSZEREK ELEMZÉSE .....</b>	<b>9</b>
7.1	SZEDÉS EGYKEZES ÉLŐ MUNKAERŐVEL.....	9
7.2	VÁGÓKÉSES BETAKARÍTÁS.....	11
7.3	FÉLAUTOMATA SZEDÉS – KÉTKEZES MÓDSZERREL, HUMÁN MUNKAERŐVEL.....	12
7.4	TELJESEN AUTOMATIZÁLT SZEDÉS.....	13
7.4.1	<i>Mycionics Inc. vállalat gombabetakarító gépe.....</i>	<i>14</i>
7.4.2	<i>TechBrew vállalat gombabetakarító gépe.....</i>	<i>15</i>
7.5	ELTERJEDT CSIPERKEGOMBA SZEDÉSI MÓDSZEREK ÖSSZEHASONLÍTÁSA.....	17
<b>8</b>	<b>KÉPFELDOLGOZÓ PROGRAM ÍRÁSA PYTHON NYELVEN .....</b>	<b>18</b>
8.1	CÉLKITŰZÉS: .....	18
8.2	A PROGRAM INPUTJÁUL SZOLGÁLÓ KÉP .....	18
8.3	PROGRAMOZÓI FELÜLET ÉS KIEGÉSZÍTŐ CSOMAGOK .....	20
8.4	A KÉPFELDOLGOZÁS LÉPÉSEI.....	20
8.4.1	<i>Szürkeárnyalatossá alakítás és homályosítás.....</i>	<i>21</i>
8.4.2	<i>Binarizálás.....</i>	<i>21</i>
8.4.3	<i>Szegmentálás előtti apró fekete pontok eltávolítása.....</i>	<i>21</i>
8.4.4	<i>Szegmentálás.....</i>	<i>23</i>
8.4.5	<i>Gombok pozíciójának meghatározása.....</i>	<i>27</i>
8.4.6	<i>A képfeldolgozás lépéseinek szemléltetése képekkel.....</i>	<i>28</i>
8.4.7	<i>A képfeldolgozás optimalizálása.....</i>	<i>30</i>
<b>9</b>	<b>AZ AUTOMATIZÁLT GOMBASZEDÉssel SZEMBEN TÁMASZTOTT ELVÁRÁSOK, AZ ADATGYŰJTÉS MEGOLDANDÓ TECHNOLÓGIAI PROBLÉMÁI.....</b>	<b>33</b>
9.1	CÉLKITŰZÉS .....	33
9.2	ADATGYŰJTŐ RENDSZER ILLESZTÉSE A PROFESSZIONÁLIS GOMBATERMESZTŐ ÜZEMEK ADOTTSÁGAIHOZ.....	33
9.3	MAGAS PÁRATARTALOM ÉS KORROZÍV KÖZEG.....	34
9.4	AZ ADATGYŰJTÉS NEHÉZSÉGEI .....	35

9.5	A KÉPKÉSZÍTÉS NEHÉZSÉGEI .....	35
9.6	KÉP SZÉLÉN ELHELYEZKEDŐ GOMBÁK PROBLÉMÁJA .....	36
9.7	ADATGYŰJTŐ EGYSÉG TÁPELLÁTÁSÁNAK ÉS A KOMMUNIKÁCIÓS VEZETÉKEK PROBLÉMÁJA .....	39
<b>10</b>	<b>AZ ADATGYŰJTŐ EGYSÉG PROTOTÍPUSÁNAK MEGTERVEZÉSE.....</b>	<b>40</b>
10.1	CÉLKITŰZÉS.....	40
10.2	AZ ADATGYŰJTŐ KOCSI FELÉPÍTÉSE ÉS A RAJTA LÉVŐ EGYSÉGEK .....	40
10.2.1	<i>Kamera mozgatása és elhelyezése az adatgyűjtő kocsin .....</i>	<i>41</i>
10.2.2	<i>Vision kamera kiválasztása .....</i>	<i>41</i>
10.3	GOMBA POZÍCIÓK ADATFORMÁTUMÁNAK KIDOLGOZÁSA ÉS TOVÁBBÍTÁSA .....	44
10.4	ESZKÖZÖK KÖZÖTTI KOMMUNIKÁCIÓ MEGTERVEZÉSE .....	46
<b>11</b>	<b>ADATGYŰJTŐ KOCSI MEGVALÓSÍTÁSÁNAK MEGTERVEZÉSE.....</b>	<b>48</b>
11.1	ADATGYŰJTŐ KOCSI 3D-S MEGTERVEZÉSE.....	48
11.2	AZ ADATGYŰJTŐ KOCSI ANYAGKÖLTSÉGEINEK BECSLÉSE .....	51
<b>12</b>	<b>TOVÁBBI KUTATÁSI LEHETŐSÉGEK .....</b>	<b>52</b>
<b>13</b>	<b>ÖSSZEFOGLALÁS .....</b>	<b>54</b>
<b>14</b>	<b>KÖSZÖNETNYILVÁNÍTÁS.....</b>	<b>55</b>
<b>15</b>	<b>IRODALOMJEGYZÉK .....</b>	<b>56</b>
<b>16</b>	<b>FÜGGELÉK.....</b>	<b>59</b>

## 2 Resume

A csiperkegomba fogyasztás világszerte egyre nő és a jövőben egyre nagyobb szerepet kap a fenntartható élelmiszertermelésben. A termelési költségek legnagyobb részét a kézi szedés bére teszi ki. Az alacsony bérek és a kellemetlen munkaviszonyok miatt, nehéz a szedésre munkavállalót találni, ezért a termelők egyre inkább érdekeltek lennének abban, hogy áttérjenek a gépi betakarításra. Jelenleg azonban nem létezik megbízhatóan működő, teljesen automata gombaszedési megoldás, mely teljes mértékben helyettesíteni tudná a kézzel végzett gombaszedést. Kutatásom ezért a csiperkegomba automatizált szedésének technológiai fejlesztésére irányul.

Dolgozatomban bemutatom a fehér csiperkegomba eddig alkalmazott betakarítási technológiáit, összehasonlítva az egyes eljárások előnyeit és hátrányait. Célom, hogy olyan berendezést fejlesszek, mely képes a modern gombatermesztő állványzat polcán végig haladni, a termőterületet végig fotózni és a képeket eljuttatni a képfeldolgozó egységnek, úgy, hogy a berendezés integrálható legyen a már elterjedt gombatermesztő rendszerekbe.

Kidolgoztam az automatizált gombaszedés első és legfontosabb állomásának, a gombák intelligens, elektronikai eszközzel történő felismerésének eljárását, illetve a szedőfej pozíciójának meghatározását. Ehhez összegyűjtöttem a képfeldolgozáshoz szükséges adatgyűjtés fontos aspektusait, kritériumait, bemutatva a fejlesztés nehézségeit és megoldandó problémáit, melyhez a MATE Zöldség- és Gombatermesztési Tanszékétől kaptam szakmai támogatást. Ismertettem az általam Python programozási nyelven fejlesztett gombafelismerő szoftver működési elvét, mind elméleti, mind gyakorlati megközelítésben, és a képfeldolgozás során felmerülő feladatokat (pl. összenövő gombák szegmentálása, micéliumok eltüntetése a képről). Megoldottam a leszedendő gombák detektálását. *OpenCV* és *NumPy* programkönyvtárak alkalmazásával nagymértékben lecsökkentettem a képfelismerő szoftver futásidejét.

Megterveztem az adatgyűjtő egység vezérlésének, hajtásának, a képanyag elkészítésének és továbbításának módját. Elkészítettem az adatgyűjtő rendszer prototípusának méretpontos 3D-s tervét, melynek tervezési alapelve a könnyű szerkezet és a lehető legolcsóbb megvalósíthatóság volt. A gombaszedés gépesítésével és technológiájának automatizálásával nagymértékben lecsökkenthető a gombatermesztés költsége, így a fogyasztókhöz olcsóbban juthat el a termék, ami megfékezheti a csiperkegomba árak folyamatos emelkedését, tovább növelheti a fogyasztást.

### 3 Abstract

The consumption of mushrooms is growing worldwide and will become an increasingly important part of sustainable food production in the future. The largest part of the cost of cultivation is the labour for hand picking. Due to low wages and uncomfortable working conditions, it is difficult to find workers for picking and growers would be increasingly interested in switching to automated harvesting. However, there is currently no reliable, fully automated mushroom picking solution which could fully replace manual picking. My research therefore focuses on the technological development of mushroom picking.

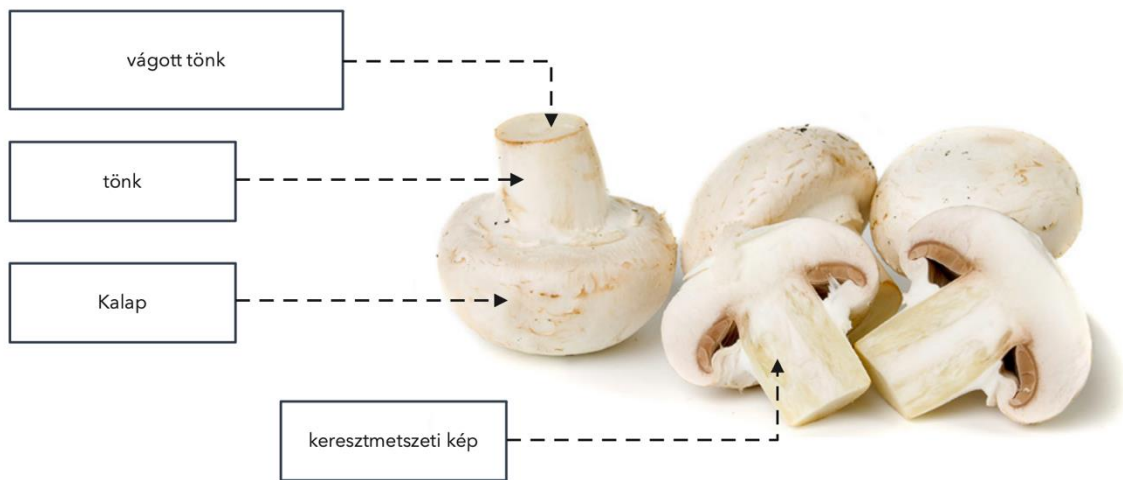
In my thesis I will present the harvesting technologies for white button mushrooms that have been used so far, comparing the advantages and disadvantages of each method. My aim is to develop a device that is capable of moving along the shelf of a modern mushroom harvesting rack, taking images of the entire growing area and sending the data to an image processing unit, in such a way that the device can be integrated into existing mushroom harvesting systems.

I have developed a method for the first and most important step of automated mushroom picking, the intelligent detection of mushrooms by an electronic device and the determination of the position of the picker head. For this purpose, I have collected the important aspects and requirements of data collection for image processing, presenting the difficulties and problems to be solved in its development, for which I have received technical support from the Department of Vegetable and Mushroom Production at MATE. I described the principle of operation of the mushroom detection software that I developed in Python programming language, both in theoretical and practical approaches, and the tasks that occurred during image processing (e.g. segmenting intergrown mushrooms, removing mycelia from the image). I solved the problem of detecting mushrooms to be picked. By using *OpenCV* and *NumPy* libraries, I significantly reduced the runtime of the image recognition algorithm.

I have designed the way to control and drive the data logger unit, to create and transmit the image data. I prepared a scale-accurate 3D design of a prototype data logging system, based on the design principle of lightweight construction and the lowest possible cost. By the automation of the mushroom harvesting process and technology, the cost of mushroom production can be drastically reduced, allowing cheaper products to reach consumers, which can curb the continuous rise in mushroom prices and further increase consumption.

## 4 Bevezetés, a gombatermesztés automatizálásának indokoltsága

Európában a legismertebb és a világon az egyik leggyakrabban fogyasztott gomba az *Agaricus Bisporus*, közismertebb nevén a kétspórás csiperkegomba (1. ábra). Fogyasztása rendkívül egészséges, mivel az összes esszenciális aminosavat tartalmazza, és nagyon alacsony a zsírtartalma. A gombák kalóriaértéke alacsony, viszont telítőértéke az emészthetetlen rosttartalom miatt magas, ezért a kalóriaszegény étrendek fontos részét képezi. Igen népszerű azok körében, akik nem fogyasztanak húst, mert magas a fehérjetartalma a nem állati eredetű táplálékok között, ezen kívül különleges íz és aromaélménye miatt is fogyasztják. Magyarországon folyamatosan növekszik az egy főre jutó elfogyasztott gomba mennyisége. [1] [2]



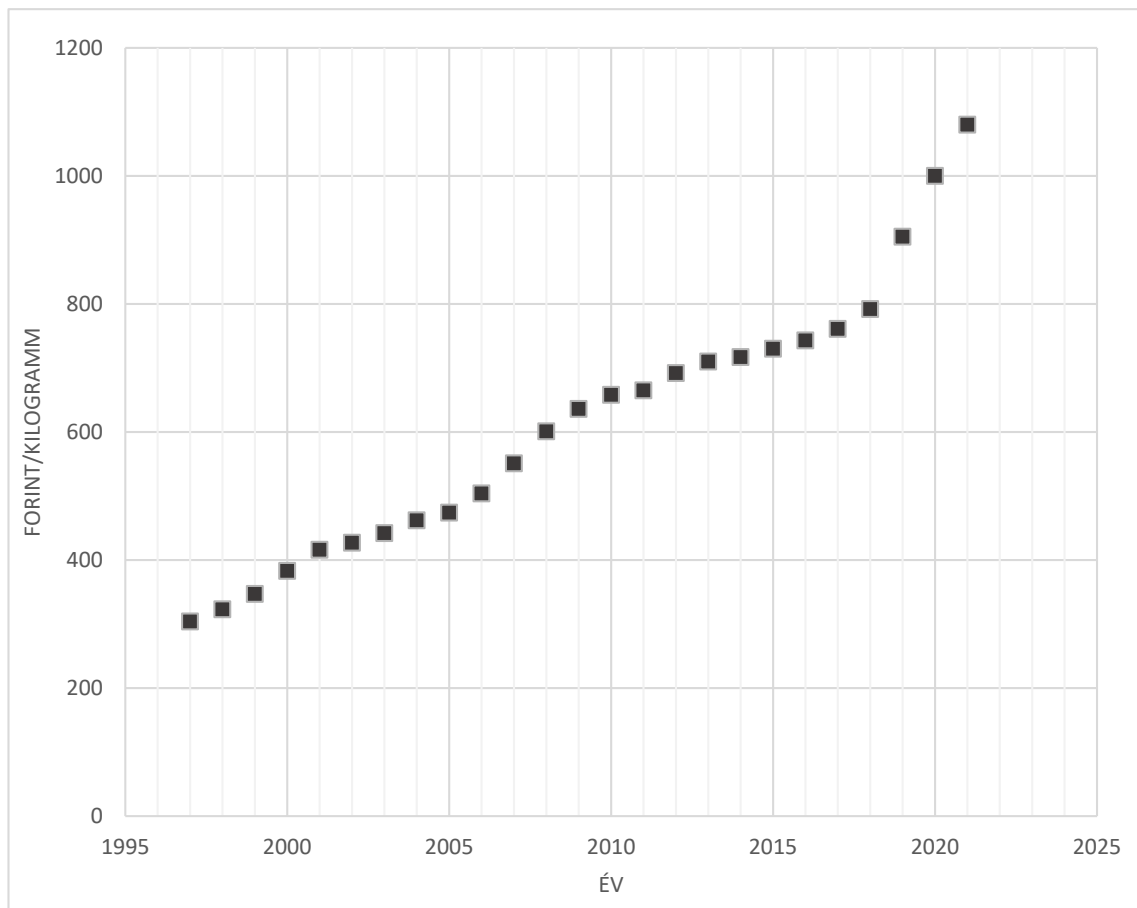
1. ábra A kétspórás fehér csiperke [28] (saját szerkesztésű ábra)

A kereskedelemben általánosan kapható csiperkegombát Európában modern gombatermesztő üzemekben termesztik, komoly tudományos háttérű ipari üzemekben, ahol a gombák növekedését és termesztésének körülményeit számítógépes rendszerek szabályozzák, így biztosítva a minőséget és a terméshozamot. Előre meghatározható a termelt gomba mennyisége, növekedése és a gombák szedési ideje is. [2]

A csiperkegomba termőközege főként a mezőgazdasági ágazatokból származó állati eredetű trágya, szalma, gipsz és víz, amelyet a gomba betakarítása után tovább lehet hasznosítani komposztként a növénytermesztésben. A gombatermesztés majdnem minden mellékterméke felhasználható, visszakerül a természeti körforgásba. [3] [2]

A gombatermesztés legjelentősebb költségeit a termesztő közeg előállítása, a gombaszedők bérezése és a termelési klimatikus viszonyok fenntartása jelenti. Közép és Kelet-Európában egyelőre gyorsabban fejlődik a gombatermesztés, mert alacsonyabb költséget jelent a betakarítás a termelőknek, de így is állandó kihívást jelent a betakarításához szükséges munkaerő felvétele és megtartása. [4] A termelőhelyiségekben való munkavégzés magas koncentrációt és precíz munkavégzést követel meg, de a munka monoton, a munkakörnyezet hűvös, magas a páratartalom, egész nap hajolgatni kell a polcok között, ami kellemetlenné teszi a munkát. Nyugat-Európában már nehéz erre a feladatra munkaerőt találni, az emberek nem szívesen végzik ezt a fajta munkát. [4]

A gombatermesztők évtizedek óta igyekeznek csökkenteni a szedők iránti igényt, mert a gomba előállítási ára folyamatosan nő, ami a fogyasztói árban is megjelenik (2.ábra).



**2. ábra: Csiperkegomba éves fogyasztói átlagárának alakulása Magyarországon [30] (saját ábra)**

## 5 Célkitűzés

A csiperkegomba fogyasztás világszerte egyre nő és a jövőben egyre nagyobb szerepet kap a fenntartható élelmiszertermelésben. A termesztési költségek legnagyobb részét a kézi szedés bére teszi ki. Az alacsony bérek és a kellemetlen munkaviszonyok miatt, nehéz a szedésre munkavállalót találni, ezért a termesztők egyre inkább érdekeltek lennének abban, hogy áttérjenek a gépi betakarításra. Jelenleg azonban nem létezik megbízhatóan működő, teljesen automata gombaszedési megoldás, mely teljes mértékben helyettesíteni tudná a kézzel végzett gombaszedést. Kutatásom ezért a csiperkegomba automatizált szedésének technológiai fejlesztésére irányul.

Dolgozatomban bemutatom a fehér csiperkegomba eddig alkalmazott betakarítási technológiáit, összehasonlítva az egyes eljárások előnyeit és hátrányait. Célom, hogy olyan berendezést fejlesszek, mely képes a modern gombatermesztő állványzat polcán végig haladni, a termőterületet végig fotózni és a képeket eljuttatni a képfeldolgozó egységnek, úgy, hogy a berendezés integrálható legyen a már elterjedt gombatermesztő rendszerekbe. Célom, hogy írjak egy programot, mely képes a termőterületről készített képeken a gombákat felismerni, méretüket meghatározni, a megfelelő méretű gombákat kiválasztani és azok szedési pozícióját egyértelműen meghatározni, továbbá a folyamat ellenőrizhető-ségéhez a képfeldolgozás végeredményét jelenítse meg, jelölve a szedésre alkalmas gombákon lévő szedőfejezéspozíciókat. Cél, hogy a program futása minél gyorsabb legyen. Célom, hogy olyan berendezést fejlesszek, mely képes a gombatermesztő polcokon végighaladni, és a termőterületet végig fotózni, és a képeket eljuttatni a képfeldolgozó egységnek. Cél, hogy az integrálható legyen a modern gombatermesztésben alkalmazott állványzatba.

## 6 Módszertan

A képfeldolgozó szoftvert Python programozási nyelven, NumPy és OpenCV könyvtárak felhasználásával írom meg, Visual Studio Code fejlesztőkörnyezetben. Az adatgyűjtő kocsi megtervezését és 3D modellezését az Autocad 2022 programmal végzem, melyhez az általam megrajzolt elemek mellett a grabcad.com, illetve a termékeket forgalmazó cégek weboldaláról letöltött modelleket is felhasználtam.



## 7 Elterjedt csiperkegomba szedési módszerek elemzése

### 7.1 Szedés egykezes élő munkaerővel



3. ábra: Szedés egykezes humán munkaerővel, liftes gombaszedő kocsik segítségével [31]

A legelterjedtebb gombaszedési módszer az egykezes szedés (3., 4. ábra), melynek lényege, hogy a szedést végző egy szedőállványon helyezkedik el, az egyik kezében a kést fogja, míg a másik kezével benyúl a polcok közé és enyhe csavaró mozdulattal leszed két-három gombát. Késsel levágja azok tönkjét, és dobozokba helyezi méret szerinti osztályozás szerint. A csavarási mozdulat azért szükséges, hogy biztosan elszakadjanak a gombafonalak és a leszedendő gomba körül növvő, de még apró gombák további fejlődése biztosítva legyen. Nagy rutin szükséges a komplex feladat elvégzéséhez, hogy a megfelelő gombákat válassza ki a szedő, ne tegyen kárt a termőfelületen, és a feladatait a lehető leggyorsabban végezze. A gomba leszedése mellett a szedők dolga a levágott tönköket tartalmazó vödör ürítése, a leszedett gombák dobozainak rakodása, új dobozok elővétele, valamint a polcrendszer mellett haladó szedőállványok (kocsik) helyzetének, illetve magasságának állítása is feladatai közé tartozik, hogy ezzel mindegyik polcot elérhesse és a mélyebben lévő gombákat is leszedhesse. [5] Megfelelő logisztikai háttér szükséges ahhoz, hogy a leszedett termőtestek a helyiségből kikerüljenek melyet a szedés körül dolgozó fizikai személyzet

végez, akik új üres rekeszeket hoznak és feliratozzák is a teliket, a teljesítménybér alapú elszámoláshoz.

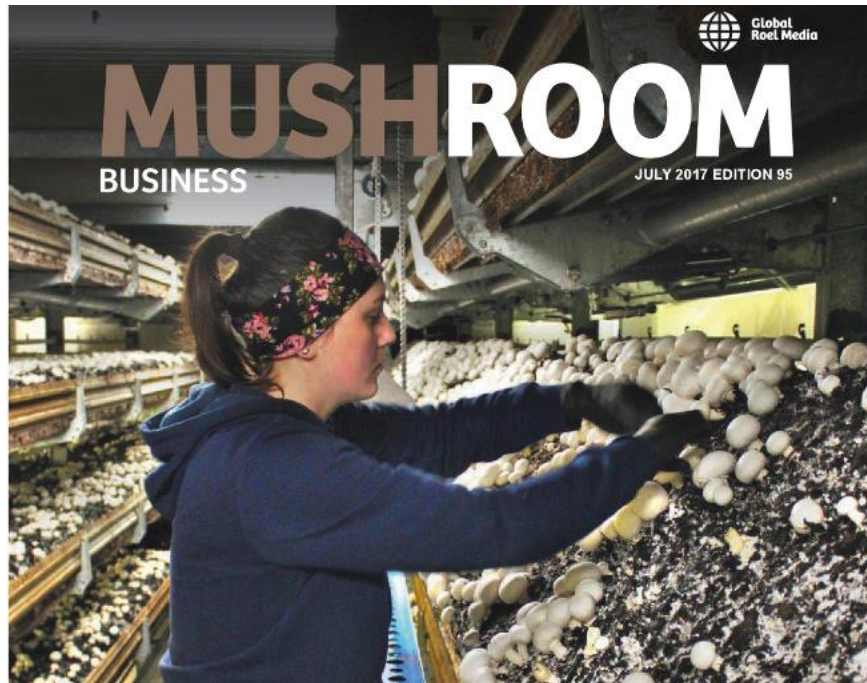


4. ábra: polcok közötti haladást segítő korszerű akkumulátoros gombaszedő kocsi [31]

A szedési körülmények javítására vezették be a dönthető polcrendszert, amelynél a polcok 45 fokos szögben a szedő felé dönthetőek (5. ábra), majd a szedés befejeztével visszaállnak vízszintes helyzetbe [5], ezáltal a szedő sokkal jobban látja a termőterületet, könnyebben tudja leszedni a megfelelő méretű gombát, mert kényelmesebben lehet hozzáférni azokhoz és nem kell annyira a polcok közé hajolni. Így ez a fejlesztés ergonomikusabb, ami felgyorsítja a szedési folyamatot. Hátránya, hogy ez a fajta polcrendszer nagyobb távolságokat igényel a polcok között, így kevesebb gomba termeszthető az állványzaton, valamint drágább, komplexebb polcrendszert igényel, mint a nemdönthető elődje, ezért még nem igazán terjedt el a pár éves fejlesztés.

Ahhoz, hogy ne kelljen lecsökkenteni az állványzaton lévő polcok számát, fejlesztettek úgynevezett mozgatható fiókos rendszert is. A technológia lényege, hogy a polcokon lévő fiókok kihúzhatóak a polcra, a szedést nem az állványoknál végzik el, hanem egy erre kialakított helyiségben, ahova átviszik a szedés idejére a fiókokat. Ennek a megoldásnak előnye, hogy a szedőnek csak egyetlen területre kell fókuszálnia, mert nem kell a szedőállvány mozgatásával foglalkoznia, így sokkal jobban tud összpontosítani, vizuálisan könnyebben azonosítható és leszedhető a megfelelő gomba, mivel nincs polc közvetlenül a termés felett, nincsenek zavaró árnyékok, jobban

megvilágított a terület. Ezzel a technológiával magasabb minőségű munkát tudnak végezni a gombaszedők. Hátránya, hogy drága, komplex polcrendszert igényel, valamint helyet kell biztosítani a szedéshez az állványról leszedett fiókoknak, melyek mozgatása a nagy tömegük miatt nehézkes. [5]



5. ábra: Dönthető polcrendszer a Mushroom Business szakmai magazin címlapján [32]

## 7.2 Vágókéses betakarítás

Már az 1990-es években is használtak gépeket a gomba betakarítására, viszont technológiai akadályok miatt magas minőségű termék nem állítható elő vele. A gépi betakarításra szánt csiperkegombák termesztése során más környezeti paramétereket használnak, mint a kézi szedés esetében. Az utóbbinál a termőhullám széthúzása a cél, hogy így a szedés hatékonysága és az elérhető hozam magasabb legyen. A gépi szedési farmokon egy betakarítás van, ahol a gombák elvárt mérete közel azonos kell legyen. A precíz klímaszabályozásnak köszönhetően a gombák azonos sebességgel és nagyságúra nőnek, tönkjük a magasabb CO<sub>2</sub> szint hatására meghosszabbodik, így el lehet végezni a gombák gépi vágását. A folyamat lényege, hogy végighalad a késekkel rendelkező vágóberendezés a polcrendszeren (6. ábra), elvágja a tönköket, és egy párhuzamos futószalagra kiforgatja a letermelt gombát. A szalag kijuttatja azt a termesztőterületről, majd egy másik futószalag segítségével osztályozó gépbe vagy ládába kerül. A penge magasságának pontos beállítása kritikus, hiszen a termesztőközeg felülete göröngyös és



a gombák nem egyenletes magasságban nőnek. A technológia hátránya, hogy magas a sérült gombák száma. A gomba a futószalagos szállítás során keletkezett ütődések miatt nem sokkal a leszedés után elszíneződik, foltos lesz. A gépiesített vágókéses betakarítással nagy mennyiségű gombát lehet leszedni nagy sebességgel, viszont az így betakarított gomba rosszabb minőségű, és nagyon rövid időn belül fel kell dolgozni, így leginkább csak konzervgomba előállítására vagy szeletelésre használható. [6]



6. ábra: Automata vágókéses betakarítás [6]

### **7.3 Félautomata szedés – kétkezes módszerrel, humán munkaerővel**

Manapság a legmodernebb működő gombaszedő technológia a kétkezes félautomata szedés. A gombatechnikai vállalatok többsége, köztük a *Van den Top Machinebouw* holland cég olyan rendszerek tervezésére összpontosít, amelyek a kétkezes szedést segítik, így növelik a betakarított gomba mennyiségét. A félautomata betakarítási technológiánál a szedést végző mindkét keze szabad, mert a gomba tönkjének levágását a gép végzi. A szedő egyetlen feladata, hogy a megfelelő méretű gombákat leszedje és a mellette haladó futószalagra helyezze. A futószalagon elhelyezett lyukba ültetett gombák a kalapjukon támaszkodnak fel, és a tönkjük kilóg a szalag túloldalán, hogy alulról le lehessen vágni azt. A *Van den Top* félautomata szedőgépe a gombák tönkjét egy előre beállított állandó magasságban vágja le, majd a kalapról, felülről készült kép alapján méret szerint szortírozza azokat egy delta robot. A futószalagról a robot műanyag tálcákba

helyezi a gombákat, melyben fóliázás és címkézés után a boltok polcaira kerül. [5] [7] A *Van den Top* félautomata gépe az automata szortírozáshoz készült fényképen a gomba minőségét nem ellenőrzi. A tönk levágásakor nem vizsgálja, hogy milyen mértékben szennyezett, hanem állandó méretben vágja le a gomba tönkjét. (7. ábra)



7. ábra: Kétkezes humán munkaerővel való szedés félautomata szortírozó híddal [7]

A technológia további hátránya, hogy nehezen integrálható az elterjedt polcrendszerekbe, leginkább egy tálcás termelőpolcokon használják, ami alacsonyabb termelékenységet jelent a többpolcos rendszerekkel szemben. A félautomata rendszer gyorsít a szedés folyamatán, mivel a kedvezőbb környezetben koncentráltabbak a szedők, kevesebb feladatott kell ellátniuk, pontosabban, gyorsabban tudnak dolgozni, viszont továbbra is szedőkre van szükség, ráadásul a beruházás rendkívül magas ára miatt nem elterjedt a gombatermesztők között.

Nagy igény van olyan teljesen automata rendszerre, melyben a gombák pontos leszedése a humán munkaerő alkalmazása nélkül is megvalósítható lenne.

## 7.4 Teljesen automatizált szedés

Jelenleg a *Mycionics Inc.* és a *TechBrew* cégek is kísérleteznek teljesen automatizált gombaszedő gépek fejlesztésével. Ez a technológia folyamatos fejlesztés alatt van, és egyelőre alacsony termelékenysége miatt nem tudta kiváltani a humán munkaerővel végzett kézi szedést.

### 7.4.1 Mycionics Inc. vállalat gombabetakarító gépe

A *Mycionics* gombaszedő rendszere egy oldalsó mozgatható állványzaton áll az emeletes polcrendszerek közötti folyosón, ahonnan benyúlva a polcok közé végzi a betakarítást. (8., 9. ábra) A cég dokumentációi szerint az általuk fejlesztett robot egy óra alatt 20 kg gombát képes leszedni [8], amely nem éri el a kézi szedéssel elérhető óránkénti 30 kg feletti teljesítményt. A gombák felismeréséhez lézeres távolságmérést alkalmaznak, mely végigpásztázza a termőterületet. Háromágú szedőfejet fejlesztettek, ami az ujj mozgását hivatott utánozni, de a szedést nem csavarással, hanem oldalra való „töréssel” valósítja meg, jelentős mennyiségű talaj marad a tönkön. Emellett a cég szerint a gombákon a megfogás helyein foltok jelennek meg, ráadásul a kalap felől nem látható sérülések a termék minőségét, eltarthatóságát nagymértékben lecsökkentik.



8. ábra: Mycionics vállalat automata gombaszedő gépe barna csiperkét szed [22]

A szedőfej a gomba megfogása után kimegy a termelő polc szélére, ahol a leszedett gombát egy másik tartófej átveszi, és egy ollószerű kés segítségével lecsípi a tönköt. Jelentős mennyiségű idő telik el (kb. 4-5 másodperc), míg a gomba megfogását követően az eljut a tönk levágásához. Sokszor szennyezett marad a gomba tönkje, és a vágó szerkezet roncsolja azt. A háromágú szedőfej az egymáshoz túl közel nőtt gombákat



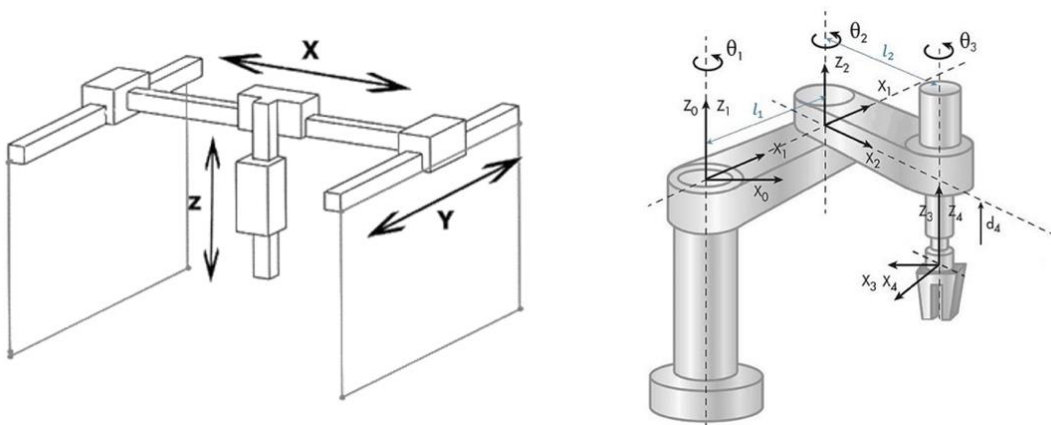
csak a szomszédos gomba megsértésével tudja leszedni. A gombák sérülései miatt egyelőre nem tudja teljes mértékben helyettesíteni a kézi szedést.



9. ábra: Mycionics vállalat automata gombaszedő gépe [22]

#### 7.4.2 TechBrew vállalat gombabetakarító gépe

A cég fejlesztése sok mindenben különbözik, mint az eddigi próbálkozások, egészen más szemszögből közelítették meg a szedőgép megvalósítását. Nem párhuzamos struktúrájú robotot használnak, hanem SCARA csuklós robotot, emiatt rendszerük kisebb helyen elfér, így használható az elterjedt emeletes polcrendszerekbe integrálva is. (10. ábra) [9]



10. ábra: Párhuzamos struktúrájú, illetve SCARA csuklós robot [23]

A gombák pozíciójának meghatározása fénykép alapján történik, a kamera és az ahhoz tartozó megvilágítás a szedőfej mellett helyezkedik el ugyanazon a roboton. Egy vákuumos szedőfejet alkalmaznak, ami a gombát felülről, a kalapján fogja meg. A szedést követően egy kamera megvizsgálja a gombát, hogy mennyire szennyezett, a hibás darabokat a gép selejtezi és félrerakja, ezzel a megoldással javítja a csomagolt termék minőségét. A csuklók és a gép kialakításából adódóan kb. 6 másodpercbe telik [9], míg a gomba a megfogástól a tönk levágásához kerül, emiatt lassú a betakarítás. Eddig két gazdaságban hajtottak végre korlátozott körű teszteléseket, mellyel 95%-os sikeres szedési arányt értek el. Tervezési irányuk, hogy technológiát váltanak, és nagy mennyiségű képet készítve áttérnek a mesterséges intelligenciával és a gépi tanulással támogatott képfeldolgozásra, amivel könnyebben el tudják dönteni, hogy mely gombákat kell leszedni. (11. ábra) [9]



**11. ábra: TechBrew Robotics gombaszedőgépe működés közben [24]**



## 7.5 Elterjedt csiperkegomba szedési módszerek összehasonlítása

Jelenleg nincsen széles körben elterjedt, teljesen automatizált, megbízható gombaszedő rendszer, amely képes a humán munkaerő beavatkozása nélkül önállóan működni. Az elterjedt módszerek összehasonlítását az 1. táblázatban szemléltetem. Egy betakarítást végző robot tervezése összetett feladat a gomba érzékenysége miatt is. Kihívást jelent a csomagolás automatizálása is, hiszen a humán munkaerő sokszor szebben tudja elhelyezni a gombákat a dobozban úgy, hogy az megfelelően kihasználja a rendelkezésre álló teret, de jól is mutasson, a gombák épségének megtartása mellett. Jelenleg nincsen olyan gombaszedő rendszer mely lépést tudna tartani a kézi szedés sebességével. [5]

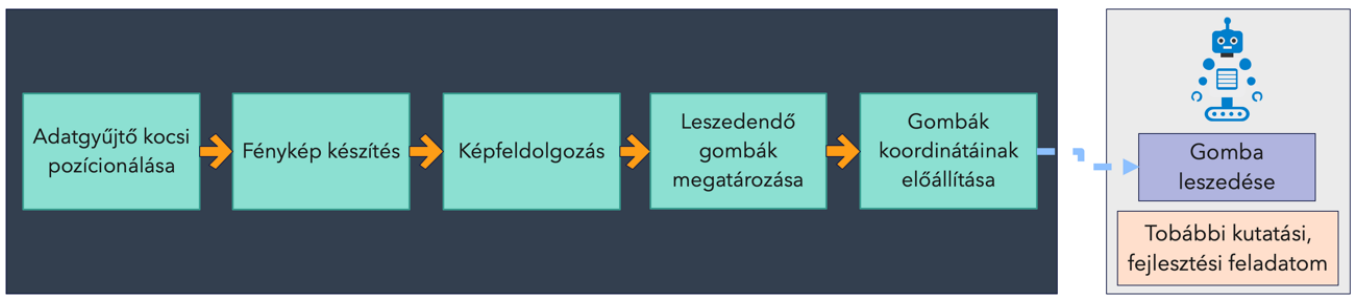
Szedési módszer	sebesség	gomba sérülékenysége	munka-körülmény	üzemeltetési költség	bekerülési költség
Egykezes humán munkaerővel	átlagos	kiemelkedően alacsony	közepes	magas	alacsony
Félautomata szedés (kétkezes humán munkaerővel)	gyors	átlagos	közepes	magas	magas
vágókésés betakarítás	kiemelkedő	nagyon magas, sok a sérült darab	nagyon jó	alacsony	közepes
Teljes automatizált szedés	jelenleg lassú	átlagos	nagyon jó	alacsony	nagyon magas

1. táblázat: Elterjedt csiperkegomba szedési módszerek összehasonlítása

## 8 Képfeldolgozó program írása Python nyelven

### 8.1 Célkitűzés:

Az automatizált gombaszedés első állomása a leszedendő gombák és pozíciójának meghatározása. (12. ábra) Célom, hogy olyan berendezést fejlesszek, mely képes a polcokon lévő gombákról fotókat készíteni, az elkészített képeken lévő gombokat felismerni, méretüket meghatározni, a leszedésre alkalmas gombokat kiválasztani és azok szedési pozícióját egyértelműen meghatározni, valamint, hogy a technológiai folyamat ellenőrizhetőségéhez a képfeldolgozás végeredményét jelenítse meg, jelölve az leszedendő gombokon lévő szedőfejpozíciókat. Célul tűzöm ki, hogy a program futása minél gyorsabban történjen meg.



12. ábra: A folyamat (saját ábra)

### 8.2 A program inputjául szolgáló kép

Szükségem volt fényképekre a program fejlesztése során, melyeken tesztelni tudom annak működését, megfigyelhetem a képfeldolgozás nehézségeit, és javaslatot tehetek, hogy hogyan célszerű azokat elkészíteni, annak érdekében, hogy a gombafelismerés minél hatékonyabb lehessen. A 13. ábrán látható kép a Magyar Agrár- és Élettudományi Egyetem Zöldségtermesztési tanszékének gombatermesztő laboratóriumában készült, Dr. Geösel András, egyetemi docens, tanszékvezető, a Kertészettudományi Intézet igazgatójának segítségével. Több kép közül lett kiválasztva a fejlesztés alapjául szolgáló fénykép. Olyan képet kerestünk, mellyel a képfeldolgozás során fellépő legrosszabb eshetőséget is szimulálni is lehet. A képen olyan különleges problémák jelennek meg, mint például a talajjal szennyezett gombafejek, vagy eldőlvé, oldalról látszódó gombák, valamint összenőtt gombafejek, melyek bonyolítják a digitális képfeldolgozás folyamatát. A továbbiakban a 13. ábrán látható képen fogom bemutatni a képfeldolgozás lépéseit, fontos aspektusait, melynek főbb tulajdonságait a 2. táblázatban foglaltam össze.



13. ábra: A képfeldolgozás alapjául szolgáló fotó (Fotó: Geösel A.)

A kép megvilágítása megfelelő, árnyékmentesnek mondható, de kissé életlen, mert a kép nem pontosan felülről készült. A terület majdnem egésze gombával fedett, némelyik közülük összeért, valamint egyes gombák kalapja alatt kisebb gombák is nőnek. A kép bal szélén, közepén, olyan apró gombák tönkjei látszanak, amik mutatják, hogy a gombák ferdén növést más gomba akadályozta a növekedésben. A gombákon látható fekete pöttyök a talajból származnak. (A termesztés során el kell kerülni, hogy a gombákra szennyeződés kerüljön.) A kiválasztott fényképen szimulálni lehet, hogy az algoritmus hogyan detektálja és kezeli, hogy némely gombafej alatt kisebb gombák helyezkednek el.

	formátum	méret (képpont)	fájl mérete	felbontás	színreprezentáció
kiindulási kép	JPG file	1262 x 842	184 KB	120 dpi	sRGB

2. táblázat: A képfeldolgozás alapjául szolgáló fénykép fontosabb tulajdonságai

### 8.3 Programozói felület és kiegészítő csomagok

A képfeldolgozás megvalósításához a Python 3.10.5 verzióját használtam, Windows 10 64 bites platformon, a Microsoft által fejlesztett Visual Studio Code nyílt forráskódú kódszerkesztőben. A Python nyelv előnye, hogy akár amatőr programozók számára is jól olvasható, mivel nagymértékben egyszerűsített a szintakszisa, de mégis sokoldalú, nagyszámú kiegészítőcsomag érhető el hozzá, melyekből az *OpenCV*-t és a *NumPy*-t is használtam a program megalkotására. (3. táblázat)

Az *OpenCV* (Open Source Computer Vision Library) egy nyílt forráskódú számítógépes látás és gépi tanulási szoftverkönyvtár, mely infrastruktúrát biztosít a számítógépes vision alkalmazások számára, segítségével felgyorsítható a tárgyak érzékelése, felismerése. [10] Az *OpenCV* megkönnyíti a képek feldolgozását, illetve az azokkal való műveletek végzését. A könyvtár több mint 2500 optimalizált algoritmust tartalmaz, melyek közül fel is használtam néhányat az algoritmus fejlesztése során, a képeken végzett manipulációknál: a kép szürkeárnyalatossá alakítása, binarizálás vagy Gauss féle képsimítás alkalmazása során. A debuggolást és a fejlesztést segíti, hogy az *OpenCV* támogatja a program futása közbeni képmegjelenítést, mely hasznos volt ahhoz, hogy ellenőrizni lehessen, hogy megfelelő-e a program működése az egyes képfeldolgozási lépések között.

	Visual Studio Code	Python	NumPy	OpenCV
Verzió	version 1.71	3.10.5	1.23.0	4.6.0.66

3. táblázat: Programozás során felhasznált felületek és csomagok verziói

### 8.4 A képfeldolgozás lépései

A következőkben lépésenként mutatom be, hogy milyen képfeldolgozási lépéseket és algoritmusokat kellett alkalmazni ahhoz, hogy gombok pozíció koordinátáit a végén megkaphassuk. Az egyes lépéseket képeken is szemléltetem a következőkben.

### **8.4.1 Szürkeárnyalatossá alakítás és homályosítás**

A képfeldolgozás első lépéseként a képet szürkeárnyalatossá alakítottam és a fényképen lévő képzajt és szemcsésséget Gauss simítással leszűrtem. A kép apró részletei nem fontosak a gombák detektálása során, a körvonalak, a kontúrok hordozzák a hasznos információkat. A nyers fénykép minden egyes pixele egy RGB vektorból áll, míg a konverzió után egy pixelt csak egy szürke árnyaltos érték jellemez. Innentől egyetlen érték elég az adott pixelt jellemzéséhez, így kevesebb adattal kell dolgozni. A műveletvégzések még hatékonyabbá tehetők a binarizálással.

### **8.4.2 Binarizálás**

A szürkeárnyaltos képen kiválasztva egy fényességi küszöbértéket meghatároztam, hogy mely értékek képviselik a sötét talajt, és melyek a képen lévő fontos objektumokat. A küszöbérték felett a pixel értéke 1, míg a küszöbérték alatt 0 értékű lesz. A binarizálással meg lehet határozni az összefüggő pixeleket a képen, illetve egyértelműen detektálhatóvá válnak az objektumok körvonalai, a kontúrvonalak. Kritikus a fényességi küszöbérték helyes megválasztása, mert túl alacsony értéknél az objektumok összemosódnak, eltorzulnak, míg fordított esetben információk veszhetnek el a képen. Az objektumok kisebbek lesznek a ténylegesnél, vagy a fénykép gyengébben megvilágított területein lévő gombák nem látszódnak, a fekete háttérben „elvesznek”.

A paraméterek optimalizálásánál azt a beállítást kerestem, mikor a felesleges, zavaró részletek eltűnnek a képről, de az alakzatok körszerűek maradnak, illetve a komponensek nemérnek össze. Az elkészült binarizált képen már elkezdhető az objektumok feltérképezése, de előtte pár módosítást kell végezni a képen a hatékonyabb detektálás érdekében.

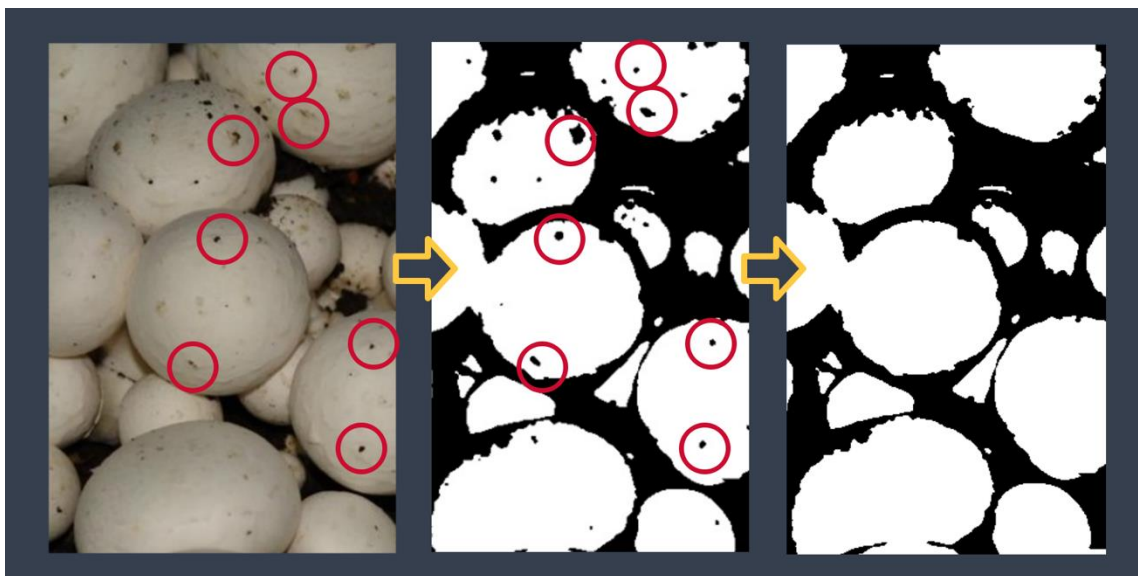
### **8.4.3 Szegmentálás előtti apró fekete pontok eltávolítása**

A nyers kezdeti képen a gombák alapján apró feketés elszíneződések láthatóak, melyek lehetnek apró talajdarabok, vagy a gomba felülete egy kicsit sötétebb ott. A fekete foltok eltávolítása azért szükséges, mert az objektumok szegmentálásánál megzavarhatja az algoritmus működését, valamint a gombák kalapja területének kiszámolásakor a fekete foltok pontatlan értékeket eredményeznének.

A pontok eltávolítására használható az OpenCV *floodFill* függvénye, vagy a OpenCV morfológiai átalakításai közül a „*Morphological Closing*”, ami a fekete egybefüggő alakzatokban lévő fehér foltok eltüntetésére alkalmas. A függvényekkel való kísérletezés során az algoritmusok olyan részleteket is eltávolítottak, amik fontos részét képezik a képnek. Így inkább saját megoldást dolgoztam ki erre a részfeladatra.

#### 8.4.3.1 A fekete pontok eltávolításának módszere

A műveleteket a binarizált kép invertálásával kezdtem, majd összegyűjtöttem, hogy az egyes objektumoknak mely pixelek az elemei, melyet az *OpenCV connectedComponents* függvényével végeztem el. A függvény megvizsgálja a kép minden egyes pixelét, hogy annak környezetében lévő 4 (vagy akár 8) pixel „1” értékű-e, majd az összekapcsolódó pixelek koordinátáit egy objektumként elmenti egy 2 dimenziós tömbben. Az így kapott tömbökön végig járva, azok mérete alapján kiszűrtem azokat az objektumokat, amiket ki kell törölni (az eredeti binarizált képen a fekete foltokat). Ezzel elkerültem, hogy fontos részletek is törölve legyenek a képről. A mátrix elemeiből újra képet alkottam, majd invertáltam.



14. ábra: A szegmentálás előtti apró fekete pontok eltávolításának lépései (Saját ábra)

A 14. ábrán baloldalt az eredeti fénykép egy kivágott és nagyított része látható, melyen piros körökkel jelölve néhány elszíneződött részt emeltem ki. Az eredeti fénykép szürkeárnyalatossá alakítása, Gauss simítása, majd binarizálása után a középső kép keletkezik, melyen továbbra is megfigyelhetők a jelölt foltok. A jobboldali képen a módosított *fillingHoleUpdate* függvény eredménye látható, melynek implementálása a

függelékben tekinthető meg. A micéliumok vagy apró gombák miatt jelenlévő kisebb fehér foltok eltávolítását is a *fillingHoleUpdate* függvénnyel oldottam meg, mint a fekete foltok eltávolítását, csak a kép invertálása nélkül. A programkód bemutatása és értelmezése a következő bekezdésben olvasható.

#### 8.4.3.2 A *fillingHoleUpdate* függvény működésének értelmezése

A függvény hívásakor megkapja a képet, mely már átesett az előfeldolgozási lépéseken, majd a binarizált képet invertálni kell, az az a képet alkotó 0 és 1 értékészletű pixelek negáltját képezem, mely ahhoz szükséges, hogy fel tudjam használni a korábban megírt függvényeket, így elkerüljem a kódismétlést. Az *objectcreator* függvény visszatérési értékei a képen megtalált objektumok száma (*found\_labelsBlack*, ez egy pozitív egész szám), valamint az *objectsBlack* nevű változó, mely a feldolgozandó kép pixeleinek számával megegyező méretű kétdimenziós tömb, melynek elemei egész számok. A tömb elemei azt jelölik, hogy az adott pixel melyik objektumhoz tartozik (az objektumokat egész számokkal sorszámozza az algoritmus, *found\_labelsBlack* változó a legmagasabb sorszámot jelöli, míg a nulla a háttér). Ahhoz, hogy az objektumokat egyben lehessen kezelni, elkészítettem a *collect\_object\_pixels* függvényt, aminek az a feladata, hogy az összes, azonos objektumhoz tartozó pixelek „koordinátáit” összegyűjtse az *objectsBlack* mátrixból. A függvény visszatér egy kétdimenziós tömbbel, mely az objektumokra lebontva tartalmazza a pixelek koordinátáit (a „belső” tömbök mérete eltérő, hiszen az objektumok mérete különböző). Egy ciklusban megvizsgálom, hogy hány elemet tartalmaz a tömb adott eleme, amennyiben a meghatározott *tombrange* változónál kisebb a mérete, akkor törölöm azt a *collectedforfilling* 2D tömbből, különben a következő elemre lépek. Majd az objektumokból újra képet lehet készíteni, melynek működési elve, hogy a *collectedforfilling*-ben tárolt koordináták helyére 1-es értéket helyezünk, egy csupa nulla elemű 2D tömbben (mely az eredeti kép méreteivel megegyező méretű) melyet a *createImageFromObjects* függvény végez, mely visszatér a kész képpel, amit már csak invertálni kell, hogy a gombák alakzata fehér legyen, míg a háttér fekete.

#### 8.4.4 Szegmentálás

A binarizálás során elvesztek részletek a képről, például árnyalt körvonalak két gomba kalapjának összeérésénél, így azok egybefüggő alakzatként jelennek meg a

binarizált képen. A két gomba méretének és azok pozíciójának külön-külön kiszámolásához meg kell határozni azok feltételezett határvonalát és szét kell választani azokat. Ezt a feladatot a *Watershed* algoritmus felhasználásával oldottam meg. Ez a függvény úgy keresi meg egy egybefüggő objektumnak a szétválasztási vonalait, hogy addig csökkenti az objektumot kívülről a középpontja felé, míg az szét nem válik több komponensre. Ez az eljárás kiválóan működik, ha nagy méretű alakzatokról van szó, például szedésre érett gombák esetén. Abban az esetben, ha a szétválasztandó objektum kis méretű komponensekből épül fel, ahol az összenövés mértéke majdnem akkora, mint a komponensek átmérője, nem tudja szétválasztani az algoritmus az objektumot. Az algoritmus a binarizált képen berajzolja a szétválasztott objektumokat. A szétválasztás színét a háttér színének, feketének kell megválasztani a paraméterek között. A következőkben az *OpenCV* könyvtárából a szegmentáláshoz használt *Watershed* algoritmus eredményét vizsgálom, melyhez ki kell nyerni, hogy hány komponens született a szétválasztás során.

#### 8.4.4.1 Szegmentálás előtti és utáni komponensek számának összehasonlítása

Ahhoz, hogy pontosan meghatározható lehessen a szegmentálás előtt, illetve utáni képen lévő objektumok száma, az 1. programkódban látható kódot írtam, melyben a szegmentálás előtt, illetve után is lefuttattam a *connectedComponents* algoritmust, melynek *found\_labels* visszatérési értéke megadja a megtalált komponensek számát, melyet a terminálba írtam ki a programmal. Ennek eredménye a 15. ábrán látható.

```
found_labels,objects=(objectcreator(blackFilledImage))

print("Szekmentálás előtti objektumok száma %d db komponens" %
found_labels)

result = objectSegmentation(img, img_bin)

found_labels,objects=cv2.connectedComponents(result)

print("Szekmentálás utáni objektumok száma %d db komponens" %
found_labels)
```

1. programkód: A szegmentálás eredményének összehasonlítása



### 8.4.4.2 A Watershed Algoritmus alkalmazásának eredményessége

```
hubaycsongor@Hubay-MacBook-Pro Mushroom-Industry % /usr/bin/env /usr/bin/python3 /Users/hubaycsongor/.vs
code/extensions/ms-pyt
hon.python-2022.16.0/pythonFiles/lib/python/debugpy/adapters/../../debugpy/launcher 50215 -- /Users/hubayc
songor/Mushroom\ Indus
try\Mushroom-Industry\repair.py
Szegegmentálás előtti objektumok száma 59 db komponens
Szegegmentálás utáni objektumok száma 74 db komponens
Process finished --- 2.1831629276275635 seconds ---
hubaycsongor@Hubay-MacBook-Pro Mushroom-Industry %
```

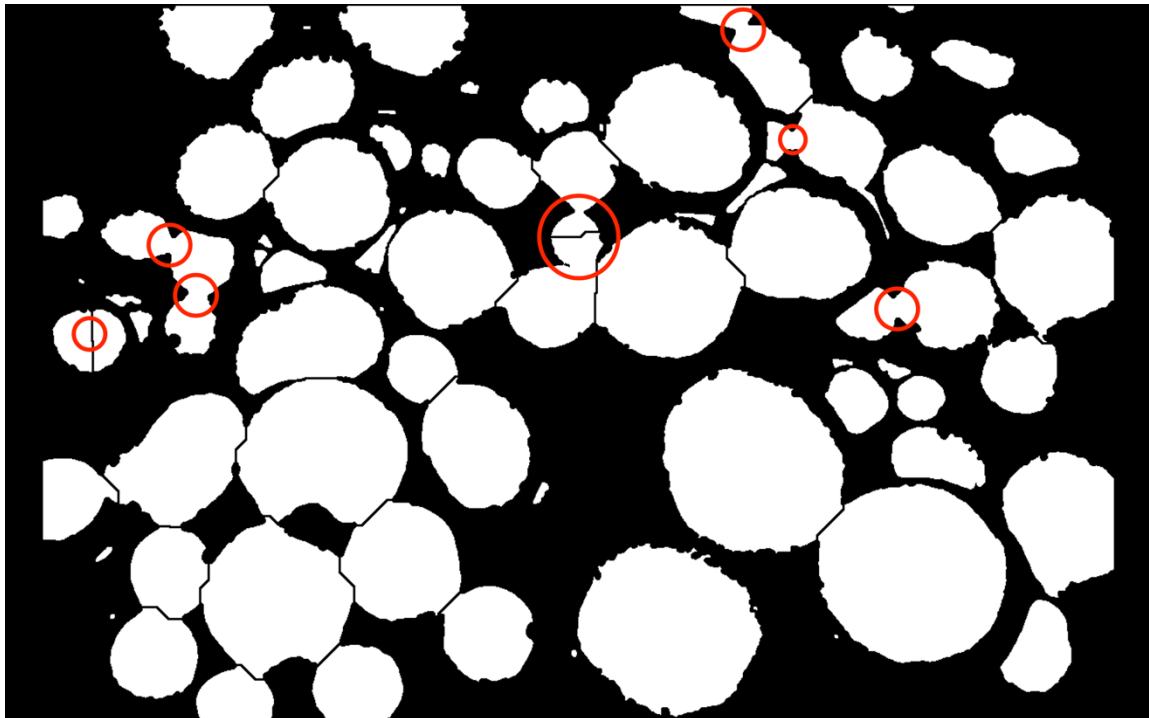
15. ábra: Visual Studio Code program termináljában megjelenített mérési eredmények

A szegegmentáló algoritmus működésének hatékonysága a 16. ábrán látható. Az ábrán bejelöltem piros körrel azokat a részeket, ahol valamilyen hibát vétett az algoritmus, és eltér a kapott eredmény az elvárttól, melynek számszerű eredményét a 4. táblázatban szemléltettem. A ~94%-os eredmény igen jónak tekinthető, hiszen fontos ismét megjegyezni, hogy a kép, melyen az algoritmus le lett futtatva, a worst-case esetet szimulálja, normál, ritkításos szedés esetén nem fordulhat elő ennyi össze- és egymás alatt növekvő gomba.

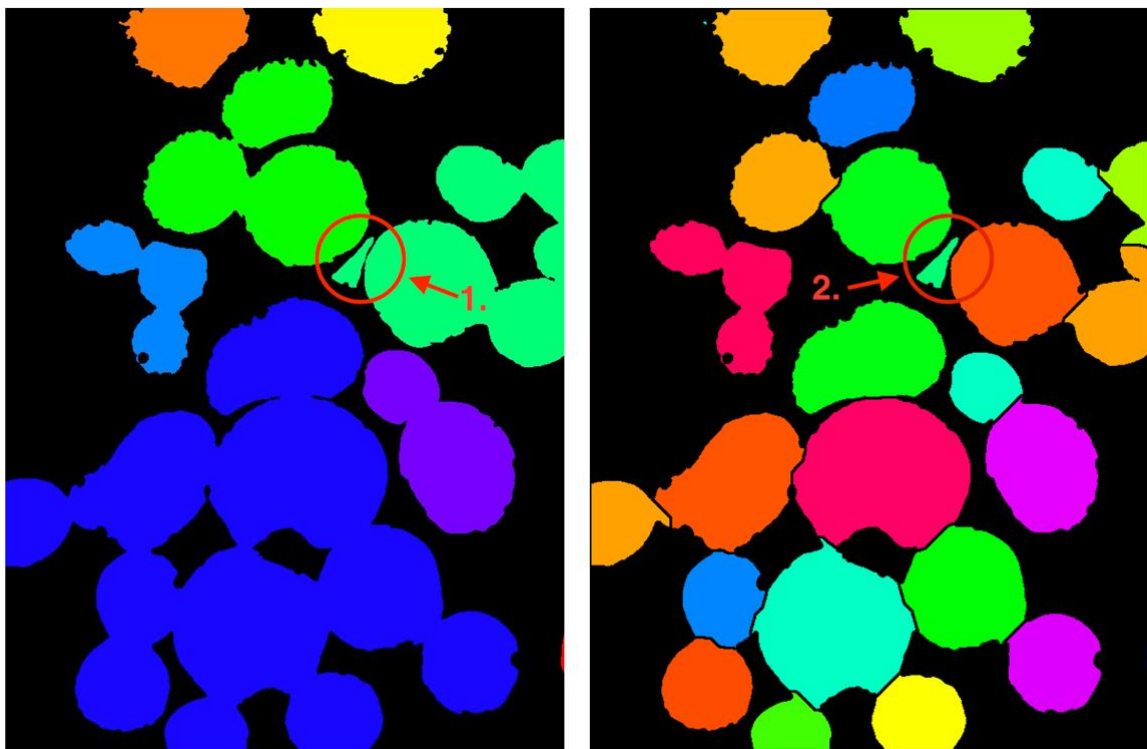
Eredeti objektumok száma	Elvárt komponensek száma	Watershed Algoritmus által kapott komponensek száma
59 [db]	79 [db]	74 [db]
Szétválasztás elmaradásának száma	Hibásan szétválasztott komponensek száma	Elvárt és az algoritmus által kapott komponensek számának százalékos aránya
6 [db]	1 [db]	93,67%

4. táblázat: A szegegmentálás eredményének értékelése

Ahhoz, hogy a program fejlesztése során ellenőrizni lehessen az objektumok szétválasztását, különböző színekkel jelenítettem meg az egymástól független objektumokat. Szabad szemmel ugyanis nehezen lehet észrevenni az akár 1 pixeles hidat két objektum között, így ez a megoldás felgyorsítja a tesztelhetőséget és pontosabb eredményt mutat. Ehhez létrehoztam egy függvényt, ami RGB színes képet készít az összefüggő komponensek megkeresését követően és a különböző objektumokhoz tartozó pixelekhez eltérő színt rendel. Ezt a függvényt a *Watershed* algoritmus futása előtt, illetve utána újból meghívom. A színek egyértelműen megmutatják, ha két objektum összeköttetésben van egymással, mert akkor azonos színűek maradnak.



16. ábra: A szegmentáló algoritmus végeredménye, piros körökkel jelölve a hibás szétválasztásokat (saját ábra)



17. ábra: A baloldali képen a szegmentálandó kép, míg a jobboldalt a szegmentálás eredménye látható, kiemelve a komponensek színezésének fontosságát (saját ábra)

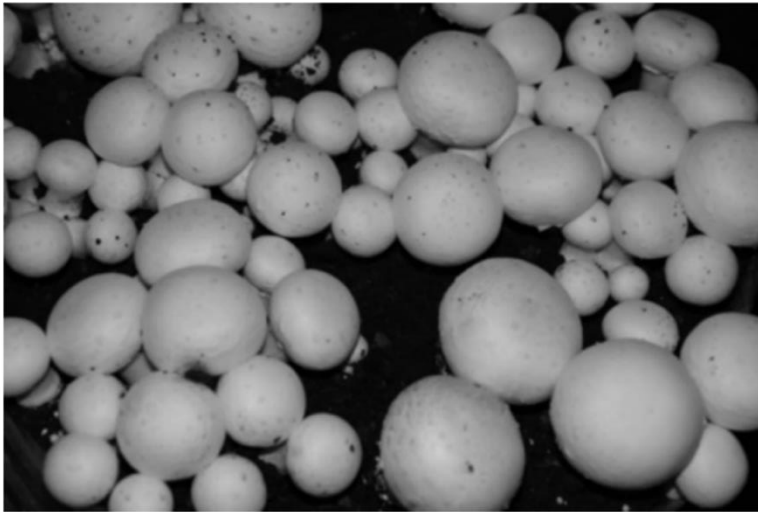
A 17. ábrán megjelölt részeken szemléltetem a szegmentálás végeredményének ellenőrzését segítő algoritmus gyakorlati fontosságát. A képen kiválóan látható, hogy az egyes sorszámmal megjelölt részen, az azonos világoszöld komponensek látszólag nem érnek össze, de valójában pár pixel összeköti azokat, mivel azonos színűek, így azok egy objektumot alkotnak. A jobboldali képen a szegmentálás után, a kettes sorszámmal jelölt részen már eltérő színűek a komponensek, tehát szét lett választva az objektum.

#### **8.4.5 Gombák pozíciójának meghatározása**

Az eddig elvégzett képfeldolgozási lépések után a gombák már különálló objektumok, így azok méret szerint szortírozhatóak. Egy konstans érték változtatásával ki lehet szűrni a kellenél kisebb gombákat, hogy csak a megfelelő, szedésre érett gombák jelenjenek meg a képen, melyeknek meg kell határozni a koordinátáit. A képeken jelölt gombák középpontjai valójában a kép bal felső sarkához viszonyítva vannak meghatározva, így ez az adat csak egy kép inercia rendszerében értelmezhető. mely koordináták egy következő lépésben további feldolgozásra kerülnek. A gombák, a polrendszerhez viszonyított pozíciójának meghatározását, kiszámolását a későbbiekben fogom ismertetni. Az objektumok középpontjának meghatározásához először azok kontúrjait kellett megkeresni, majd az *OpenCV moments* függvényének segítségével meghatároztam az középpontjaikat.

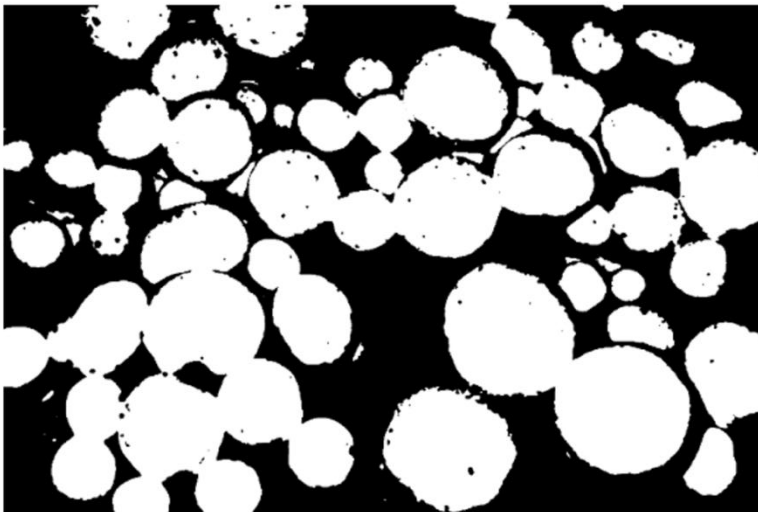
A kontúrok használata azért célszerű, mert nem szükséges egy objektum összes pixelének eltárolása, hanem elegendő annak körvonalának definiálása, ha az objektum nem „lyukas”, de bizonyos esetekben még ez is leegyszerűsödik, ha egy szabályos objektumról van szó. Egy téglalap 4 sarkának koordinátái elegendők az objektum pontos definiálásához.

#### 8.4.6 A képfeldolgozás lépéseinek szemléltetése képekkel



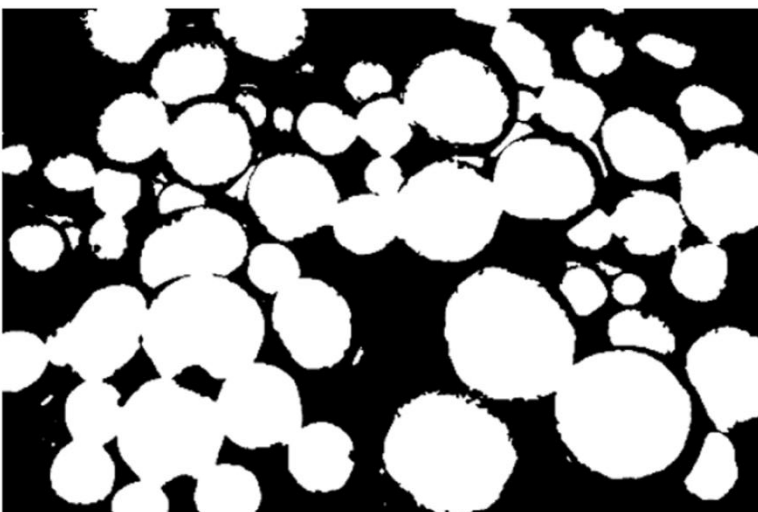
1.

Szürkeárnyalat  
és Gauss  
simítás



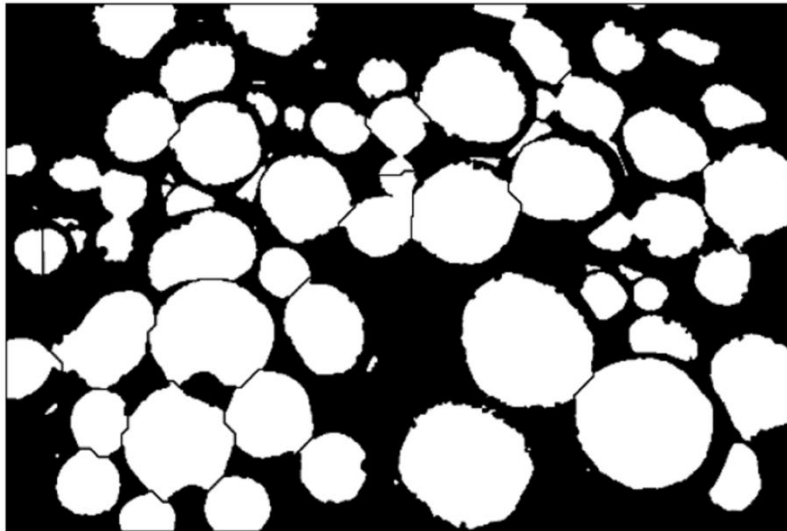
2.

Binarizálás



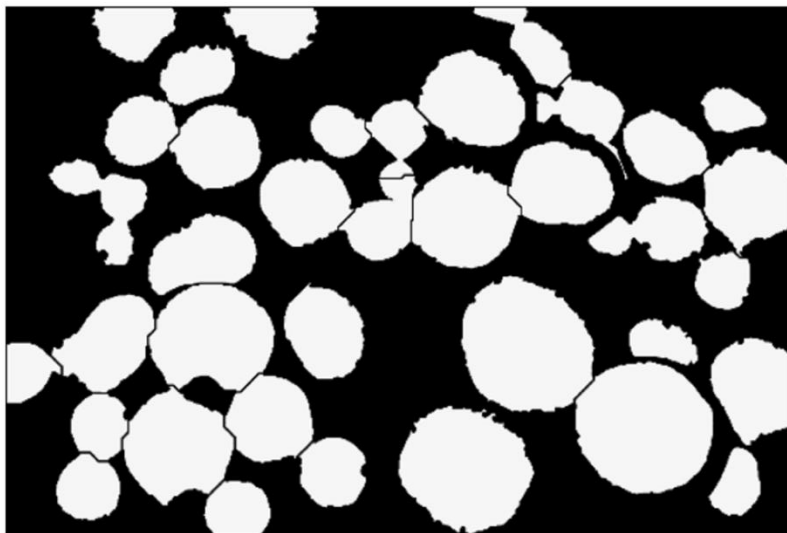
3.

Szegmentálás  
előtti apró  
fekete pontok  
eltávolítása



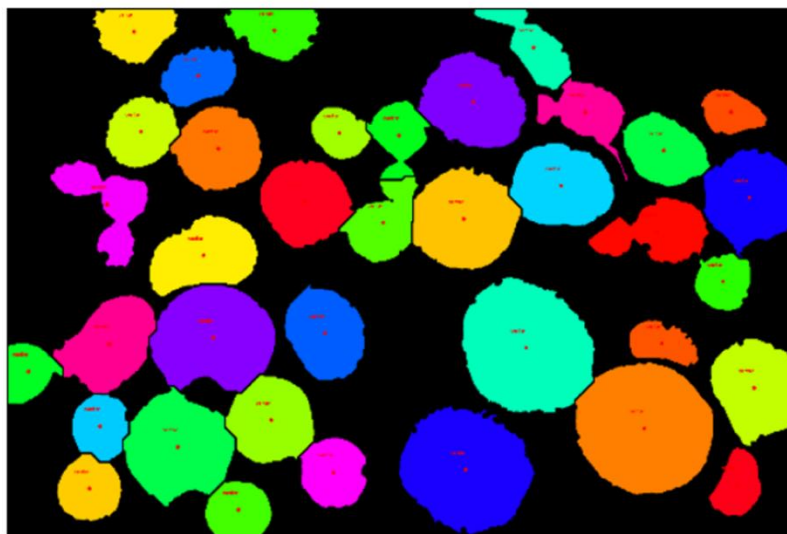
4.

Szegmentálás



5.

Micélium foltok  
eltüntetése



6.

Gombák  
középpontjának  
meghatározása

18 a)-f). ábra: A képfeldolgozás lépéseit szemléltető ábrasorozat (saját ábrák)

## 8.4.7 A képfeldolgozás optimalizálása

A program futása hagyományos Python tömbök implementálásával és bejárásával igen lassúnak bizonyult, így annak optimalizálására volt szükség. Olyan megoldást kerestem, ami elősegíti a mátrixműveletek gyorsabb elvégzését, így a Python tömböket *NumPy* tömbökkel helyettesítettem, valamint a változók létrehozásakor azoknak definiáltam a típusát is (Python nyelvben a változó típusának meghatározása automatikusan történik az futásidőben), ezzel csökkentve a futás során elvégzendő műveleteket.

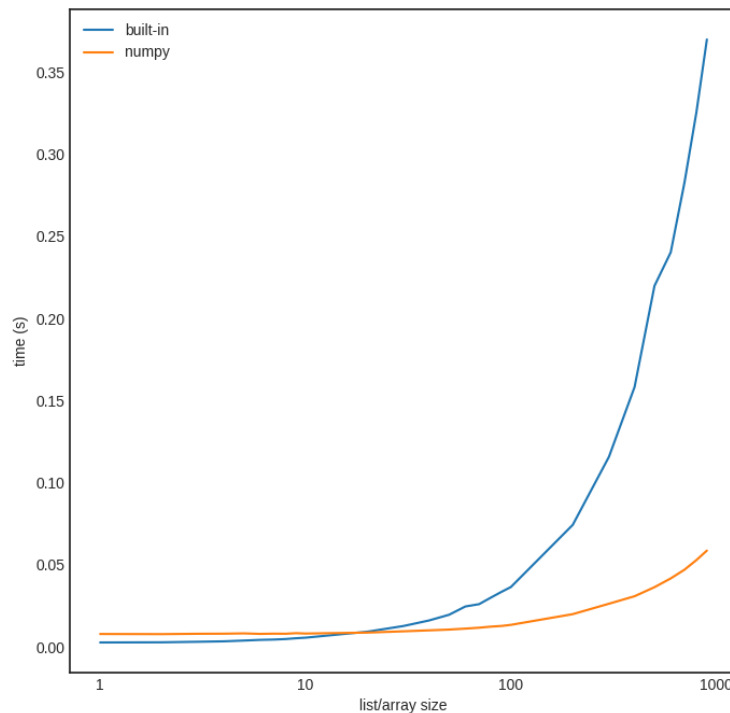
### 8.4.7.1 A *NumPy* Array bemutatása

A Python programozási nyelvhez készített tudományos számításokhoz való kiegészítőcsomag, a *NumPy* lehetővé teszi a többdimenziós mátrixok bejárását és a tömbökkel való matematikai műveleteket. Futásidőben hatalmas különbségek lehetnek a *NumPy* tömbökkel és a „gyári” Pythonban implementált tömbökkel megvalósított programkód futása között. [11] Python programozási nyelv lényege, hogy egyszerűen, könnyedén lehet parancsokat megfogalmazni és műveleteket végezni, így nem kell foglalkozni a változók típusainak definiálásával, vagy a műveletek, függvények közötti értékátadásokkal sem. A Pythonnal megvalósított tömbök a memóriában szétszórva kerülnek tárolásra (ezért is tud dinamikusan növekedni), a *NumPy* esetén egybefüggő memóriaterületek vannak lefoglalva, ezáltal nagyon gyorsan végezhetőek a műveletek, mert képes egy feladatot több részfeladatra osztani és párhuzamosan feldolgozni. A gyorsaságának kulcsa, hogy a függvények C nyelven vannak implementálva, ami gyorsabbá teszi a Python listákhoz képest, mely sebességkülönbség nagy elemszámú tömb esetén számottevő, melyet a 19. ábra szemléltet.

### 8.4.7.2 Algoritmus felgyorsítása *NumPy* tömbök használatával

A képfeldolgozásnál problémát jelent, hogy hatalmas mennyiségű adattal kell dolgozni. Például egy 2 megapixeles kép esetén a felbontás 1600 x 1800 pixelből áll, ami 1.920.000 darab pixelt jelent összesen, míg 5 Mp (2592 x 1944) esetén 5.038.848 elemű tömböt kell bejárni, amit felgyorsít a *NumPy* tömbök alkalmazása. A Python programozási nyelvben nem szükséges egy változó létrehozásakor megadni annak típusát, mert a nyelv az

értékadáskor azt intelligensen kezelni tudja, ellenben a futásidő csökkentése érdekében mégis célszerű definiálni azokat.



**19. ábra: NumPy tömbökkel és beépített Python tömbökkel végzett műveletek sebességének összehasonlítása random számokat tartalmazó tömb generálással [27]**

Az 5. táblázatban az általam írt program futásideje látható, tisztán Python tömbökről NumPy tömbökre való váltás, és további optimalizálás után, melyekkel sikerült 98%-kal lecsökkenteni a futásidőt, ennek mérését a Python *time* moduljának felhasználásával végeztem el.

### 8.4.7.3 Program futásidejének mérése

Az elvégzett méréssel összehasonlítottam, hogy mennyi idő telik a képfeldolgozás lépéseinek elvégzése hagyományos Python, illetve NumPy tömbökkel megvalósított program esetén. Ehhez a 2. számú programkóddal egészítettem ki a programot. A futásidő mérésének kezdetét a program „buildelése” utáni pillanatnak választottam, mikor a számítógép a kód első sorát kezdi kiértékelni, ekkor veszem az első időmintát. Miután az algoritmus meghatározta a leszedendő gombok pozícióit, rögzítettem a másik időpillanat értékét. A két időpont közötti távolság adja meg a program futásának idejét, melynek értékét a Visual Studio Code programban a Terminál felületére írtam ki, ahogy a 15.

ábrán is látható. A mérést két hardverrel is elvégeztem, ennek eredményeit az 5. táblázat szemlélteti.

## 2. programkód: A Program futásidejének méréséhez írt programrészlet

```
#programkód elejére:

import time

start_time = time.time()

#a programkód utolsó sora után:

print("Process finished --- %s seconds ---" % (time.time()-start_time))
```

### 8.4.7.4 Program futásidejének értékelése

A mérési eredmények jónak mondhatók, hiszen a mérés egy 6. generációs processzorral lett elvégezve. Fontos megjegyezni, hogy a program több olyan elemet tartalmaz, ami a debuggolást, illetve a program futásának nyomon követhetőségét és a szoftverfejlesztést segíti, például a szétválasztott objektumok színezése. Ezek a funkciók hosszabb futásidőt eredményeznek, ezek kiiktatásával is csökkenthető a futásidő.

Feladat: Gombok felismerése és pozíciójuk meghatározása	Tiszta Python megvalósítással elért futásidő	Numpy Array-el való megvalósítással elért futásidő	Numpy Array-el való megvalósítással elért gyorsulás
<b>Hardver 1:</b> Intel® Core™ i3-6006U 6. generációs CPU 16GB RAM 64 bites WIN operációs rendszer	228.361 sec	4.22164 sec	98,15%-al csökkent a futásidő
Feldolgozott kép mérete:		1262 x 842 pixel	
<b>Hardver 2:</b> Intel® Core™ i7-6820HQ 6. generációs CPU 16GB RAM MacOS Monterey 12.6	2.164736 sec		

**5. táblázat Leszedendő gombok pozícióját meghatározó program futás idejének összehasonlítása Numpy, illetve tiszta Python tömbökkel való implementációval**



## **9 Az automatizált gombaszedéssel szemben támasztott elvárások, az adatgyűjtés megoldandó technológiai problémái**

### **9.1 Célkitűzés**

Az előzőekben bemutattam az általam fejlesztett gombafelismerő program felépítését és működését, mely a gomba betakarítás automatizálhatóságának fontos elemét képezi. Arra kerestem a választ, hogy hogyan célszerű megoldani az adatgyűjtést. Első lépésként összegyűjtöttem a termőterületről való képkészítés fontos kritériumait és az adatgyűjtés megoldandó technológia problémáit, a modern gombatermesztő üzemek sajátságainak bemutatásával. Céloom olyan adatgyűjtő egység kidolgozása volt, mely alkalmazható üzemi körülmények között is.

### **9.2 Adatgyűjtő rendszer illesztése a professzionális gombatermesztő üzemek adottságaihoz**

A gombatermelő üzemekben főként emeletes polcrendszereket használnak, melyek az egykezes módszerű betakarításhoz helytakarékosan vannak kialakítva. A termesztő állvány polcai rendszerint egymástól 55, ill. 60 cm-re vannak [12], erre kb. 20-23 cm vastag termőtalajt töltenek, amiből a gombák 5-6 cm magasra nőnek. A polcok között helyezik el az automata öntöző rendszer permetező fejeit is. Ezért legfőképpen 26-30 cm [12] áll rendelkezésre a gomba fejek és a felső polc alja, illetve a szórófejek között, ahol el lehet végezni az adatgyűjtést, valamint a gombák betakarítását is. Ahhoz, hogy az automatizált gombaszedés később teret nyerjen, ezen feltételek között lenne jó, hogy működjön. Úgy kell integrálhatónak lennie a meglévő polcrendszerekhez, hogy lehetőleg olcsón lehessen megvalósítani a technológiát.

Ennek érdekében a polcrendszer oldalsó határoló peremét használom fel. A perem alapvető funkciója, hogy megtámasztja a talajt, de a polcok kialakítása miatt kézenfekvő lehetőség, hogy az adatgyűjtő egység ezen a peremen, kerekeken gurulva közlekedjen, sínként használva a megszakítás nélkül végig futó alumínium profilt. Ebben az esetben nem kell a polcrendszerek közé külön állványzatot építeni a gépnek, mely esetleg a termesztési műveleteket is zavarná.

A letermelt a komposztot lehúzzák a polcokról és lemossák, fertőtlenítik a berendezést, mielőtt új termőtalajt helyeznek be, ezért az adatgyűjtő egységet nem lehet állandóan az állvány polcain elhelyezni. (20. ábra)



20. ábra: 6 polcos szabványos gombatermesztő állványzat [35]

### 9.3 Magas páratartalom és korrozív közeg

A termelő helységben körülbelül 90%-os a relatív páratartalom. A termőtalajt a polcrendszeren, a gombák felett elhelyezett víz porlasztó berendezéssel nedvesítik. Az adatgyűjtő egység kialakításakor figyelembe kell venni, hogy csak olyan elemeket lehet felhasználni, melyek magas páratartalmú közegben is működőképeseek, illetve a fröccsenő vízzel szemben is ellenállóak. Különösen érvényes ez a csatlakozókra, hiszen azok fokozottan ki vannak téve a magas páratartalmú levegő korrozív hatásának. A vezérlő elektronika burkolatok, vízálló dobozok segítségével megvédhető ezen hatások ellen, míg a csapágyak, mechanikus részeknek a lehetőségekhez mérten zártnak vagy ellenállónak kell lennie, hogy ne hibásodjanak meg az alkatrészek.

## 9.4 Az adatgyűjtés nehézségei

Az adatgyűjtő kocsit az állványzat polcai között egy lift mozgatná, ezáltal mindegyik polcon lehetne ugyanazt az adatgyűjtő egységet használni. Alap állapotban a liften helyezkedik el a kocsi, majd amint megkapja a parancsot, a kívánt polc magasságába emelkedik, hogy át tudjon gurulni az állványzat két szélén lévő fém sínre, majd megkezdődhet az adatgyűjtés.

Mivel a kamera látótere kisebb, mint a polcon lévő termőterület szélessége (kb. 120 cm), ezért több képet is szükséges egymás mellett készíteni. A kamera mozgását oldal irányba egy léptetőmotor végzi bordásszíj hajtással, lineáris sínpályán guruló futókocsival, melyre rögzíteni lehet a kamerát. Mikor a kamera elkészítette a képeket a termőterület teljes szélességében, a kocsi továbbhalad a következő „sorra”, a kamera látóterének méretével megegyezően, majd itt is elkészíti a képeket, melyek a képfeldolgozó PC-be kerülnek, ami előállítja a szedéshez szükséges koordinátákat. Ez a folyamat addig ismétlődik, míg a polc végére nem ér a kocsi, ami az adatgyűjtés végeztével visszamegy eredeti pozíciójába, az állvány előtt elhelyezett liftre.

## 9.5 A képkészítés nehézségei

A fénykép elkészítését nehezíti, hogy nincs a fényképezéshez elegendő fény a termelő csarnokokban, és a polcok között, mert a gombák növekedéséhez nem létszükséglet a fény. A képek elkészítéséhez megfelelő világítás szükséges, olyan, amely a magas páratartalom ellenére nem reflektálódik a gombákon. A kép készítését zavarhatja a termelőüzem világítása, ennek elkerülése végett, minél jobban ki kell zárni a kívülről érkező fényt, és homogén megvilágítást kell alkalmazni, hogy ne legyenek árnyékok a fotózandó területen. (21. ábra) Az árnyékok elkerülése érdekében a világítást célszerű felülről elvégezni, valamint a kép elkészítésekor ügyelni kell rá, hogy a termőterülethez képest 90 fokot zárjon be a kamera szöge, így pontosan szemből látszódnak a gombák kalapjai a képen.

A termőterületen különböző méretű gombák helyezkednek el, melyek magassága is különbözik, egyrészt a talaj egyenetlensége miatt, másrészt, a gombák különböző magasságából adódóan, ez nehezíti, hogy a kép mindenhol éles legyen.



21. ábra Az adatgyűjtést nehezíti a kívülről érkező fény és a polcok között rendelkezésre álló szűk hely  
[36]

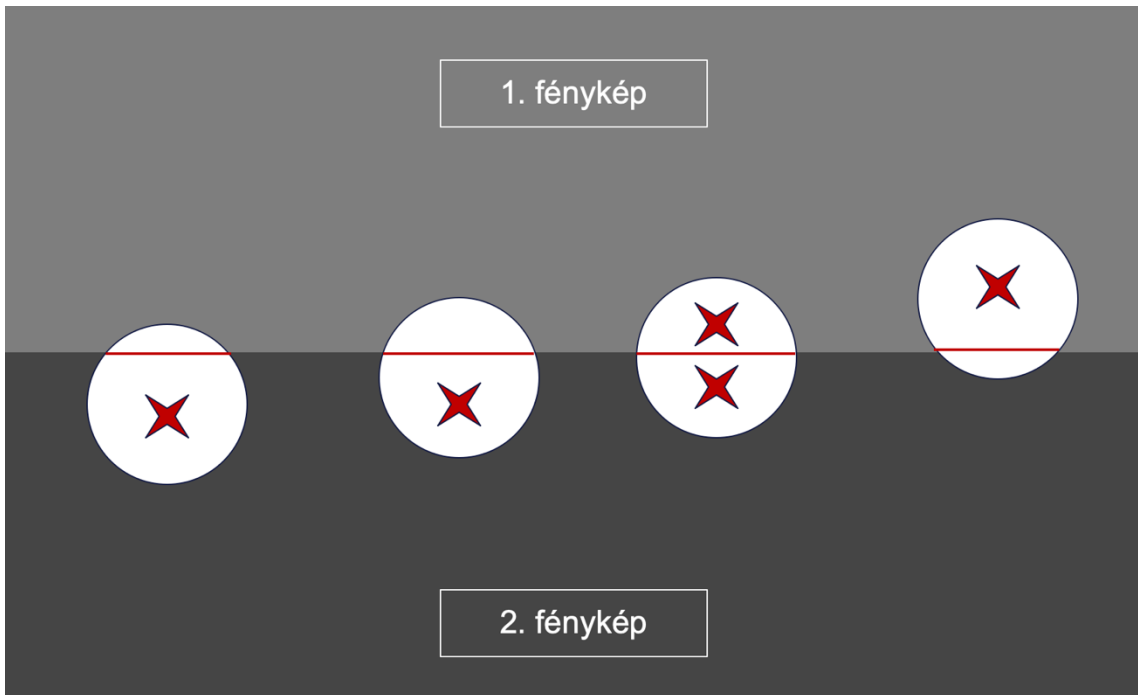
## 9.6 Kép szélén elhelyezkedő gombák problémája

A termőterületről készített fényképek szélén lévő gombák közül nem mindegyiknek kerül rá a képre a kalapjuk teljes területe. Ilyen esetben az algoritmus úgy érzékeli, hogy egy kisebb méretű gomba van a képen, valamint a gomba közepének meghatározása is pontatlan lesz.

Előfordulhat, hogy két külön gombát is detektálhat az algoritmus, ha a kép által „félbevágott” gomba két képen is szerepel, a lehetséges eseteket a 22. ábrán mutatom be. A hibából következhet az is, hogy kétszer próbálna leszedni a gép ugyan azt a gombát, vagy egy megfelelően megnőtt gombát figyelmen kívül hagy az automatika, mert a „vágás” miatt a meghatározott szedési méret alá esik az objektumok területe.

A hiba elkerülhető, ha azokat az objektumokat melyek a fénykép szélét érintik, kiszűri egy algoritmus. A kiszűrt objektumokat piros X-ek jelölik a 23. ábrán. Ez után meg kell vizsgálni, hogy a kép szélén elhelyezkedő pixelek a koordinátaik alapján mely objektumhoz tartoznak, majd azt törölni kell az objektumok listájából. A keresést értelemszerűen csak akkor kell elvégezni, ha a pixel „1-es” értékű. A kép szélén lévő

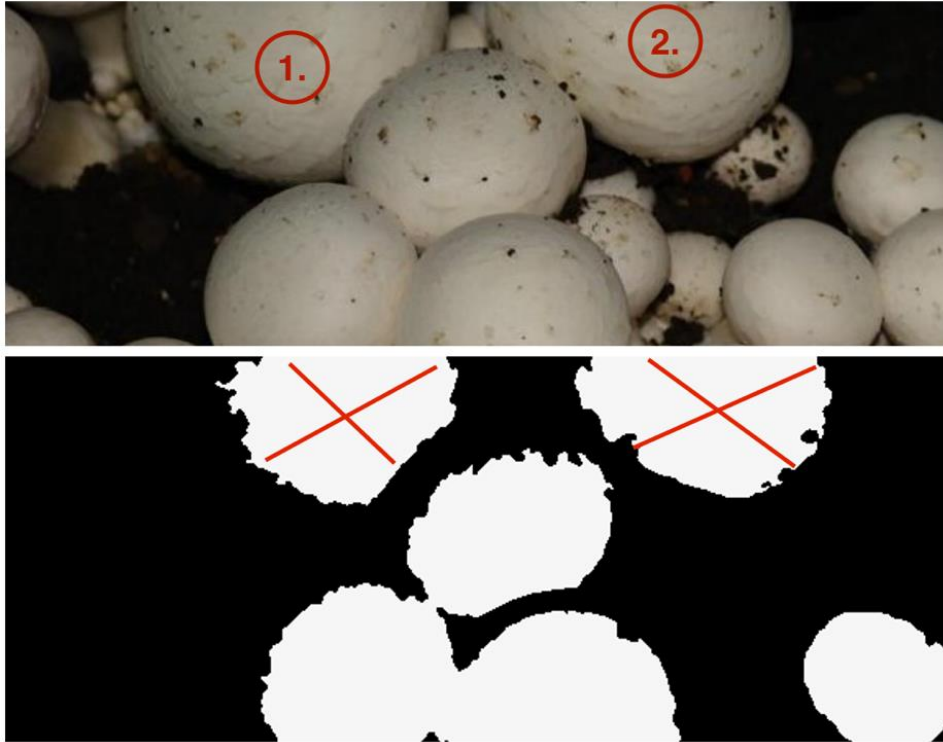
objektumok törlése után nem maradnak hibás pozíciók, viszont az így kiszűrt gombákat is meg kell vizsgálni, hogy megfelelő méretűek-e.



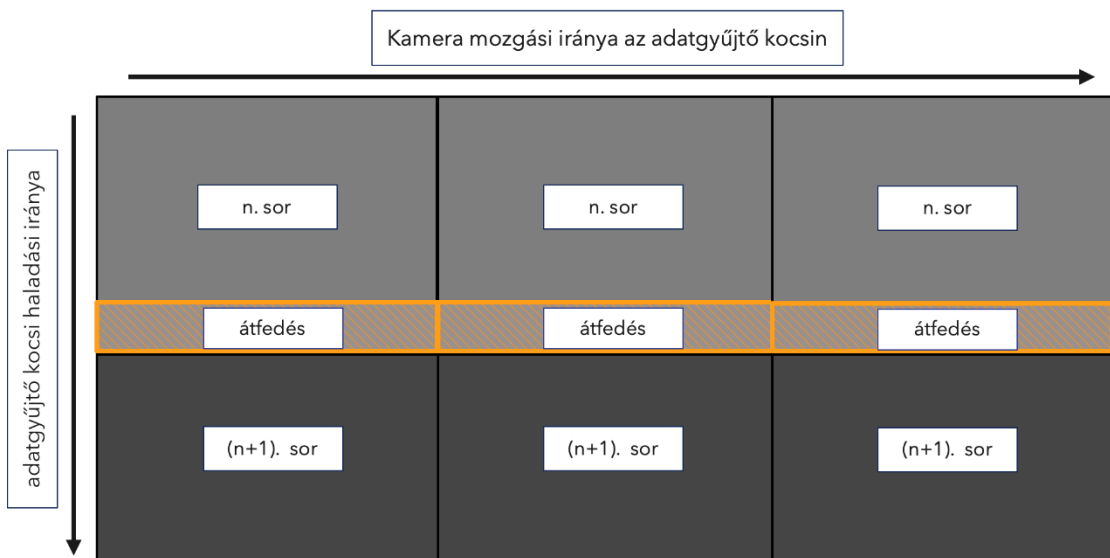
**22. ábra: A kép szélét érintő gombák helyzetének lehetséges előfordulási formái, a leszedendő méretű gombák hibás pozíció jelölésével (Saját szerkesztésű ábra)**

A fényképek készítésekor olyan mértékű átfedést kell alkalmazni, (lásd a 24. ábra) hogy ne legyen olyan gomba, amit mindkét sor képén, a kép szélén található „keretpixelek” metszenek. Amennyiben az  $n+1$ . sor fényképének tetején van egy keret által metszett gomba, akkor azt biztosan nem metszi az  $n$ -edik soron lévő képen. Az átfedés mértéke tapasztalati úton állítható be, a természetett gombakalap átmérőjének 120%-ra választanám meg. Azok a gombák melyek a képek átfedésen helyezkednek el, de nem érintik egyik kép szélét sem, és méretük alapján bekerültek a leszedendő gombák koordinátáinak listájába, kétszer fognak szerepelni. Ezek kiszűrését a szedés előtt kell elvégezni, ellenőrizve az azonos, vagy rendkívül közel eső koordinátákat.

Figyelembe kell venni, hogy szükségesnél nagyobb átfedés választása esetén, nagyméretű termőterület adatgyűjtése esetén jelentősen több időbe telik a képfeldolgozás, így célszerű az átfedés pontos megválasztása.



23. ábra A kép szélén lévő gombák a nyers képen (felső), majd a binarizálás után (az alsó képen). Piros X-szel a listából törendő objektumok jelölve (Saját szerkesztésű ábra)



24. ábra: Adatgyűjtő kocsi haladása és a kép szélén lévő gombák problémájának megoldása (Saját ábra)

## **9.7 Adatgyűjtő egység tápellátásának és a kommunikációs vezetékek problémája**

Meg kell oldani az adatgyűjtőkocsi tápellátását, a kocsi vezérléshez szükséges kommunikáció megvalósítását, és a képek továbbítását a képfeldolgozó egység felé. A kommunikáció legmegbízhatóbb módja a kábeles adattovábbítás lenne, mely megvalósítása kihívást jelent, hiszen több 10 méter hosszú kábel vezetését kell úgy megoldani, hogy az ne zavarja a gombatermesztés technológiai folyamatait.

A legmegbízhatóbb megoldás a vezetékes adattovábbítás megvalósítására az energialáncos kábelvezetés, melyhez ki kell alakítani egy tartószerkezetet, melybe belesüllyed a műanyag energialánc. A megoldás hátránya, hogy az állványzat összes polcán egyetlen adatgyűjtő kocsit dolgozik, így az állvány egy másik polcán való adatgyűjtéshez más kábeleket kell csatlakoztatni, amely megnehezíti az automatizálhatóságot. Az energialáncos kialakítása rendkívül költséges lenne, hiszen az állványzat minden egyes polcára létre kellene hozni a kábelvezető csatornát. Erre a feladatra alkalmas műanyag energialánc métere körülbelül 6500 Ft-ba kerül [13], így egy 60 méter hosszú, 8 polcos rendszer esetén 3.120.000 összegű lenne, ezzel nagymértékben megrágitaná a projekt árát, ezért vezeték nélküli módon kellene megvalósítani a kommunikációt, melynek lehetőségét a későbbiekben ismertetem, melynél a tápellátást akkumulátorról lehetne megoldani.

Így, hogy összegyűjtöttem a tervezés fontosabb nehézségeit, egy javaslatot teszek az adatgyűjtő kocsi felépítési tervének kialakításához.



# 10 Az adatgyűjtő egység prototípusának megtervezése

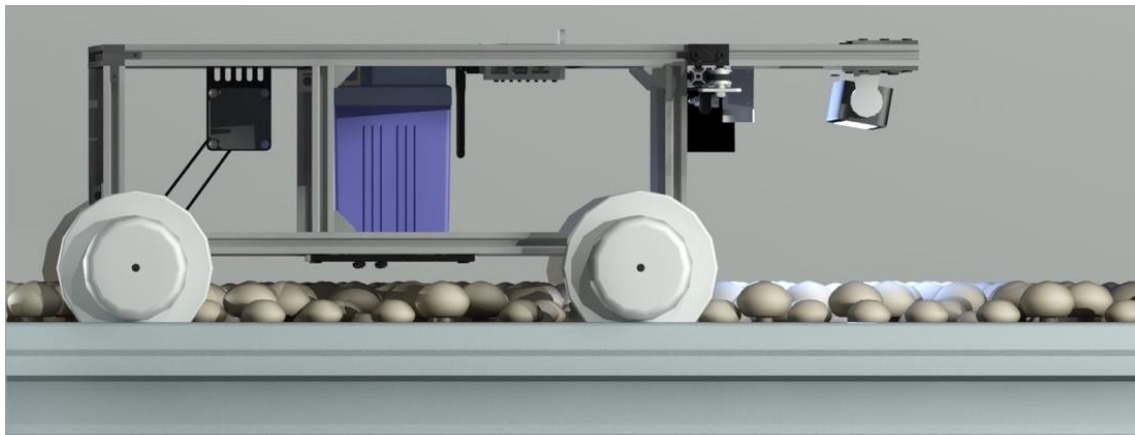
## 10.1 Célkitűzés

Cél egy adatgyűjtő kocsí prototípusának megtervezése, mely alkalmas a bemutatott polcrendszeren való adatgyűjtésre és továbbításra. A felépítésének és tervezésének alapelve, hogy a könnyű mozgathatóság érdekében könnyű szerkezete legyen, és lehetőség szerint minél olcsóbban lehessen megvalósítani. Célom egy méretpontos 3 dimenziós tervet készíteni az adatgyűjtőkocsiról, ahogy a polcrendszerbe illeszkedik. Javaslatot teszek az adatgyűjtőkocsi vezérlésének, hajtásának megvalósítására, a képanyag elkészítési és továbbítási módjára.

## 10.2 Az adatgyűjtő kocsí felépítése és a rajta lévő egységek

Az adatgyűjtő kocsí szerkezetéhez extrudált alumínium építőprofil (20 x 20mm) választottam, mert számos alkatrész és kiegészítő kapható hozzá és könnyű szerelhetősége miatt egyszerűen készíthető vele erős szerkezet. Sokszor használják ezeket prototípusok, illetve modellek készítésére.

Az adatgyűjtő kocsinak 4 fém kereke van, melyek páranként közös tengelyen vannak. A kerekek futó felülete felfekszik a polcok oldalára, ami meg is vezeti azokat. A két hátsó kerék tengelyét bordásszív segítségével 2 fázisú léptetőmotor hajtja, melynek a vezérlőkártyája szintén a kocsin helyezkedik el. Az erre a célra kiválasztott *Nema34* típusú léptető motor elegendő forgatónyomatékkal rendelkezik ahhoz, hogy az adatgyűjtő kocsí hajtani tudja. [14] (25. ábra)

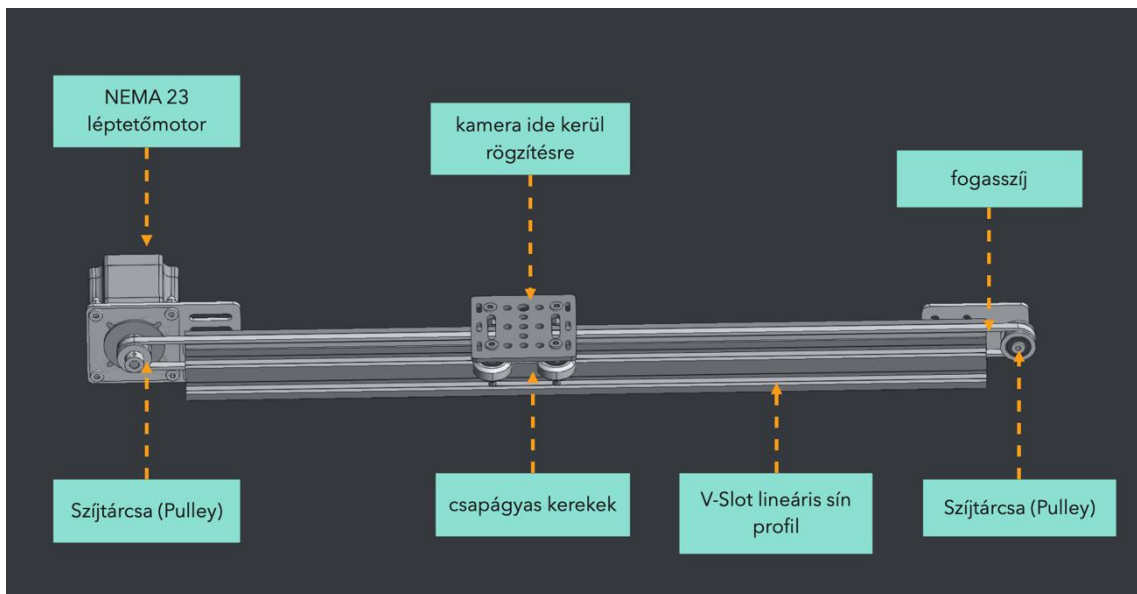


25. ábra: Az adatgyűjtő kocsí vázlata (Saját ábra)



## 10.2.1 Kamera mozgatása és elhelyezése az adatgyűjtő kocsin

A kamera mozgatásának megtervezésekor az *Openbuilds* webshopjában található alkatrészek közül választottam, ahol kész mechanikák is elérhetőek. A kamera mozgató mechanika a *V-Slot NEMA23 Linear Actuator*. [15]. (26. ábra) Alumínium profilon „kapaszkodó” görgőkön guruló kocsi, mely a léptetőmotorral hajtott szíjtárcsával és egy hosszú bordásszíjjal van összekötve. A konstrukció előnye, hogy egyedi méretre vágható alumínium profil hosszával megegyező méretre is lehet készíteni. A kamerát a kocsi elején, a szerkezeten túllógva helyeztem el, hogy a kocsi szerkezeti elemei ne lógnanak bele a kamera által elkészített képbe. A kamera mozgató motor és mechanika tömegét a kocsi hajtását végző *Nema34* típusú léptetőmotor a hátsó tengelyen ellensúlyozza.



26. ábra: NEMA23 léptetőmotorral hajtott mechanika a kamera mozgatására [15] (Saját szerkesztésű ábra)

## 10.2.2 Vision kamera kiválasztása

Olyan viszonylag olcsó, de a feladatra alkalmas kamerát kerestem, amely képes Gigabit Ethernet sebességű adatátvitelre. Ezért a *Basler Ace 2* típusú vision kamerát választottam, 350 eurós ára [16] igen kedvezőnek mondható a *vision* kamerák között. (6. táblázat) Az RJ-45 kimenettel rendelkező modellt *CAT5* vagy *CAT6* szabványú kábellel csatlakoztatni lehet az adatgyűjtő eszközhöz. A kamera tápellátása 2 féle módon valósítható meg, DC tápcsatlakozón keresztül, vagy PoE (Power over Ethernet) módon, mely lehetővé teszi, hogy csak egyetlen vezeték kelljen a kamerához vezetni. A kiválasztott kamera 2,3 megapixeles színes CMOS szenzorral rendelkezik, ami 1920 x 1200 pixel felbontású képet képes készíteni, mely felbontás elegendő a gombok detektálására, hiszen a

képfeldolgozást a 3. fejezetben is 1262 x 842 pixel felbontású képeken végeztem. Figyelembe kell venni, hogy nagyobb felbontás esetén a képfelismerő szoftver futása hosszabb futásidőt eredményezne. [16] [17]

	Szenzor típus	Ár	Elkészült kép	Felbontás	Mérete
Basler ace 2 (51gcBAS)	CMOS 1/2.3"	349 €	Színes kép 1 pixel: 8-12 bit széles	2.3 MP	55.5 x 29 x 29 [mm]
	Érzékelő	Tápellátás	Érzékelő mérete	Felbontás	Kommunikáció
	Sony IMX392	PoE vagy 12-24 VDC	6.6 x 4.1 [mm]	1920 x 1200 pixel	GigE

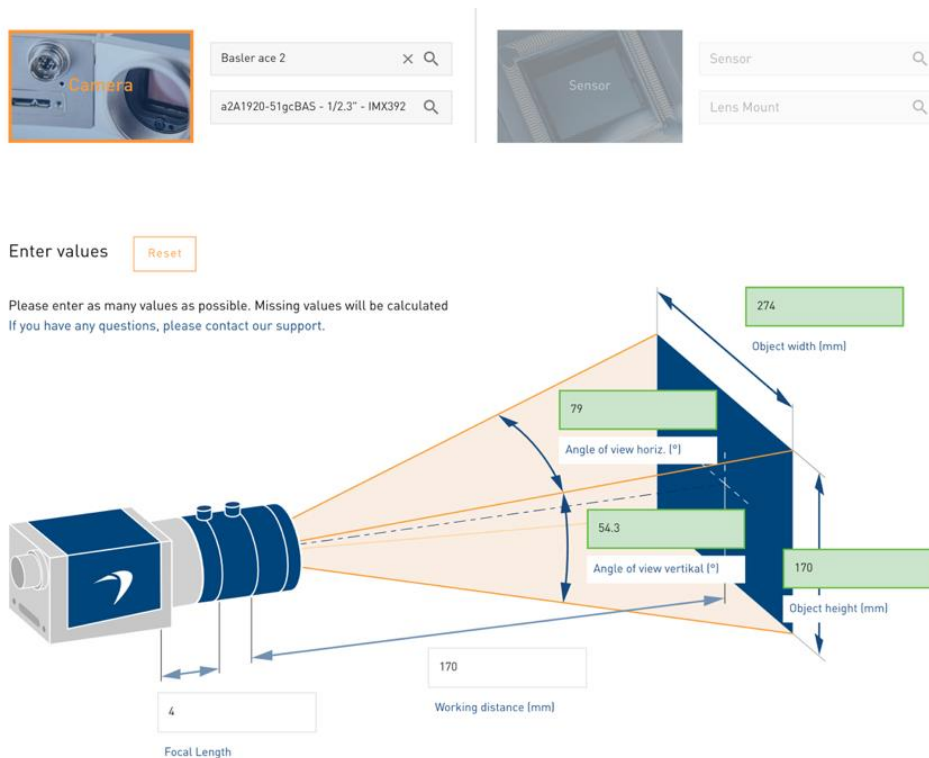
6. táblázat: Basler ace 2 (51gcBAS) típusú ipari vision kamera főbb tulajdonságai [16]

A kamera kiválasztásánál fontos szempont volt, hogy minél nagyobb legyen a látótere a rendelkezésre álló szűk hely miatt, ezért a kamerához az elérhető legkisebb, 4 mm-es gyújtótávolságú *Basler Lens C125-0418-5M-P* típusú lencsét választottam. A katalógus adatok szerint az optika 100mm-es minimális munkatávolsággal rendelkezik. [18] A 9.2. fejezetben ismertetett méretek szerint 26 cm áll rendelkezésre az adatgyűjtőkocsi megvalósítására, ebből 5,5 cm a kamera teljes mérete kábelekkal, és 3 cm az objektív, így a gombák távolsága az objektívtól legfeljebb 17.5 cm lehet. A kamera látótere 6.6 x 4.1 méretű szenzor esetén 27.39 x 17.01 centiméter. Ennek számítása során a tárgy és a lencse távolságát 17,0 centiméterre választottam, hogy véletlenül se akadhasson be a kamera a mozgás során a felette lévő polcra. (29. ábra) Ugyanez az objektív kapható nagyobb méretű, 5, 5.6, 8 megapixeles érzékelőkkel is, ezeket az optikákat a 7. táblázatban hasonlítottam össze. A látótér méretének kiszámításához a *Basler FoV* kalkulátorát használtam [19]. (27. ábra) A 120 cm széles termőterület befotózásához a 2,3 megapixel felbontású kamerával egymás mellett legalább 5 db képet kell készíteni, míg az 5 megapixelessel 7 db kép készítése szükséges, a kisebb a látótér

miatt. Mivel a nagyobb méretű szenzorhoz elérhető, legkisebb gyújtótávolságú lencse 6 mm-es, ezért annak kisebb a látótere, ami az adatgyűjtési idő növekedését jelentené.

Kamera típusa	Szenzor méret	Látótér mérete	Ár	Felbontás	Kamerához elérhető legkisebb gyújtótávolságú lencse
Basler ace 2 (51gcBAS)	6.6 x 4.1 mm	273.9 x 170.1 mm	349 €	2.3 MP	f4 mm
Basler ace 2 (23gcBAS)	6.7 x 5.6 mm	183 x 153 mm	619 €	5 MP	f6 mm
Basler ace 2 (14gcBAS)	7.8 x 7.8 mm	158 x 158 mm	750 €	8 MP	8 mm

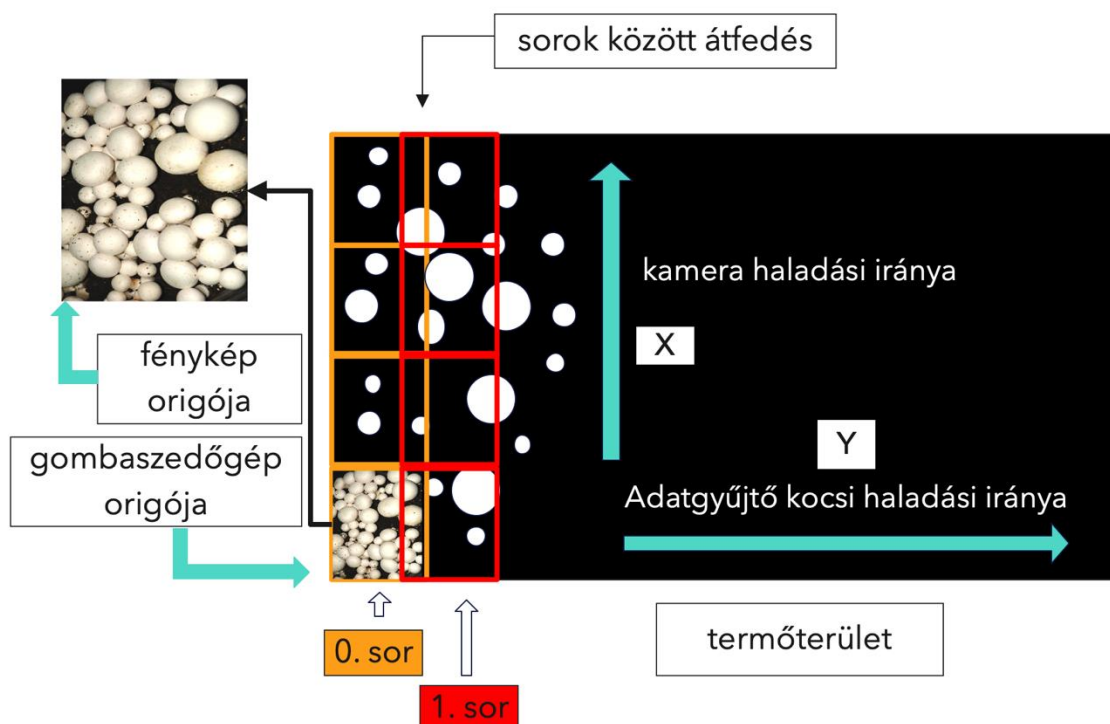
7. táblázat Basler ace 2 típusú vision kamerák árának és látóterének összehasonlítása [16] [33] [34]



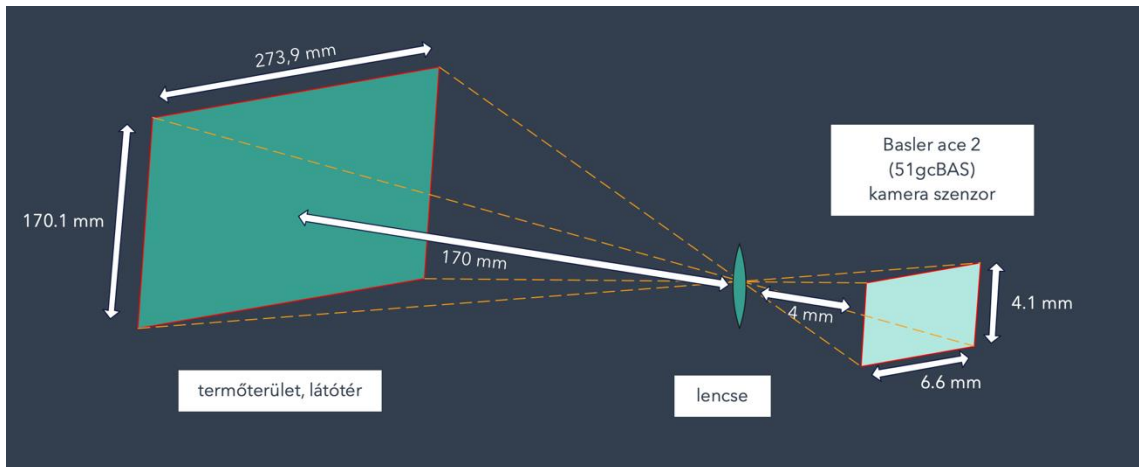
27. ábra: Basler szoftverével végzett pontos látótérkalkuláció (Saját szerkesztésű ábra)

## 10.3 Gomba pozíciók adatformátumának kidolgozása és továbbítása

Ahhoz, hogy a felismert gombák pozícióit a fénykép koordináta rendszeréből a termő terület vonatkoztatási rendszerébe lehessen átvinni, meg kell határozni, hogy az elkészített kép egy pixele a valóságban hány mikrométernek felel meg. A gyártó katalógusában a 2 megapixeles modell szenzorán (51gcBAS) egy pixel mérete  $3.45 \mu\text{m} \times 3.45 \mu\text{m}$ . Az előzőekben kiszámolt látótér mérete  $273.9 \times 170.1 \text{ mm}$ , a kamera felbontása pedig  $1920 \times 1200 \text{ pixel}$ , így az elkészített képen 1 pixel mérete  $142.65625 \times 141.75 \text{ mikrométernek}$  felel meg a termőterületen, így ez jelenti az összefüggést a két vonatkoztatási rendszer között. A gombafejek pozícióinak kiszámolásánál a kocsi haladási irányához képest a polc jobb oldali kezdő sarkát vettem origónak. (28. ábra) Ehhez a ponthoz viszonyítva, milliméterben adtam meg a gombák koordinátáit, felhasználva az előbbieken kiszámolt pixelméret értékét. Mivel a termőterület teljes feltérképezéséhez több kép elkészítése szükséges, ezért a koordináták kiszámításánál figyelembe kell venni, hogy a kép milyen messze helyezkedik az origótól, így ehhez kell viszonyítani a képen kijelölt gombák középpontját, melyek a kép bal felső sarkához vannak viszonyítva.



28. ábra: Az adatgyűjtő rendszer vonatkoztatási rendszerei (Saját ábra)



29. ábra: Kamera látóterének szemléltetése (Saját ábra)

A polc origójához képesti pozíciók kiszámolásához az alábbi képletet vezettem be:

$$\text{Pozíció}_{\text{szedőfej}}(X, Y) = (S_x \cdot P_x; n \cdot (F_{sz} - A) + S_y \cdot P_y)$$

$S_x$ : Szenzor 1 pixel hosszának megfelelő távolság a készített képen (142.65  $\mu\text{m}$ )

$S_y$ : Szenzor 1 pixel szélességének megfelelő távolság a készített képen (141.75  $\mu\text{m}$ )

$P_x$ : képfeldolgozás során meghatározott pozíció pixel X koordinátája

$P_y$ : képfeldolgozás során meghatározott pozíció pixelének Y koordinátája

$n$ : fénykép készítési sorának száma

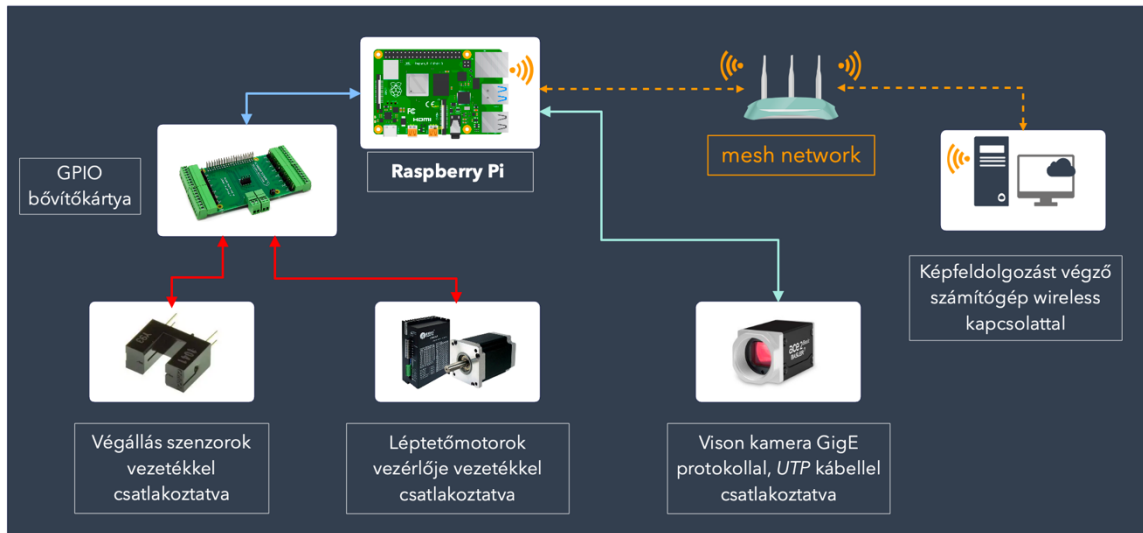
$F_{sz}$ : fénykép szélessége

$A$ : sorok átfedésének szélessége

További kutatási és fejlesztési feladatomban lesz egy gombaszedő robot megtervezése, ami a polcon kijelölt origóhoz képest kapja meg a koordinátákat, melyet a képfeldolgozó számítógép számol ki.

## 10.4 Eszközök közötti kommunikáció megtervezése

A vezetékes kommunikáció problémáit a 9.7. fejezetben ismertettem. A kábeles adattovábbítás megvalósítása nehézkes és költséges lenne, így vezeték nélküli megoldást kellett keresni, mellyel gazdaságosabban, de mégis megbízhatóan lehet az adatok továbbítását az eszközök között megvalósítani.

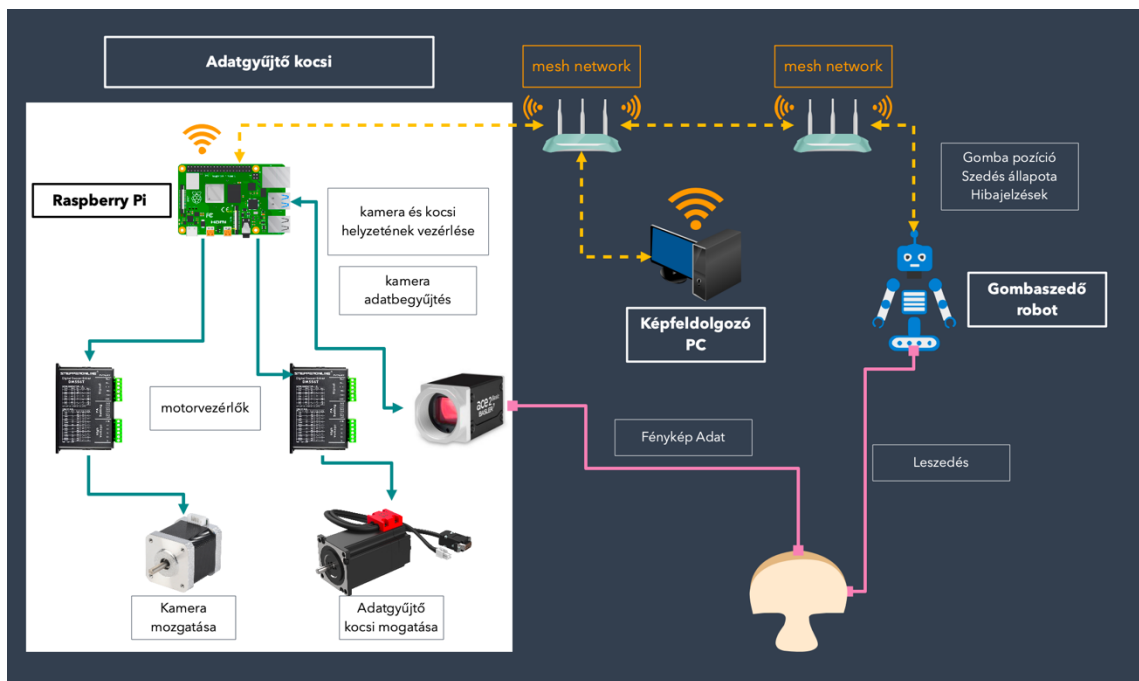


30. ábra Raspberry Pi-hez csatlakoztatott eszközök kommunikációs blokkvázlata

Az adatgyűjtési művelet alatt több eszköz együttes működését kell lehetővé tenni, az egyes folyamatok szekvenciájának betartásával. A kamera, illetve az adatgyűjtő kocsis csak akkor indulhat tovább a következő pozícióra, ha az elkészült képet előbb sikerült eltárolni. Amennyiben nem sikerült, vagy a gyengébb jelerősség miatt csak lassabban érkeztek át a képek, illetve a kocsis vezérlési parancsai, akkor holtidők alakulnak ki az adatgyűjtésben, mely lassítja a folyamatot. Ennek elkerülése érdekében a kocsis vezérlését a kocsin helyeztem el, valamint a képeket a kocsin ideiglenesen tárolni is kell, hogy azok készítése várakozás nélkül, gördülékenyen történjen, gyengébb jelerősségű kapcsolat esetén is. Ezt a feladatot egy *Raspberry Pi 4.* generációs kártyával oldottam meg, melynek előnye, hogy Python kódokat is tud futtatni az I/O portjaira kötött léptetőmotorok vezérléséhez, valamint a Gigabites Ethernet portjára csatlakoztatott vision kamerát is vezérelni tudja. A Gigabit/sec sebességű adattovábbításnak köszönhetően a Raspberry-hoz gyorsan érkezik meg az elkészített kép, mely annak mentése után késedelem nélkül vezérli a következő képkészítési pozícióra a kamerát, majd készülhet a következő kép. A képek továbbítását a képfeldolgozást végző PC-nek vezeték nélküli kapcsolaton keresztül, *mesh network* infrastruktúrával valósítottam meg, a 30. ábrán látható módon.

Számítógépen lehet elindítani szoftvert, mely vezérli az adatgyűjtés és feldolgozás teljes műveletét, folyamatosan nyomon követi, hogy hol helyezkedik el az adatgyűjtő kocsi, és eltárolja a Raspberry által küldött fényképeket. (31. ábra) A Raspberry-n futó Python nyelven megírt programban implementálva van a teljes polcrendszer adatgyűjtéséhez szükséges vezérlés, a kameramozgató motor, illetve kocsi pozíciók helyzetével úgy, hogy azok ne tévesszenek pozíciót, de mégis a lehetséges legnagyobb sebességgel tegyék azt, melyet a léptetőmotorok konstans sebességnövelésével valósítanak meg.

Az Raspberry az I/O lábaira kötött motorvezérlőknek továbbítja az információkat *opencollector*-os módon: a motor forgási irányát, a motor mozgását engedélyező jelet, valamint az impulzus jeleket. Ahhoz, hogy elegendő számú I/O porttal rendelkezzen az eszköz, hogy a motorvezérlőket, szenzorokat és egyéb csatlakoztatott eszközöket működtetni lehessen, egy I2C buszon kommunikáló GPIO bővítőkártyát alkalmazok, mely Python könyvtárának segítségével 16 izolált GPIO port [20] áll rendelkezésre, így megkönnyíti a kommunikáció megvalósítását.



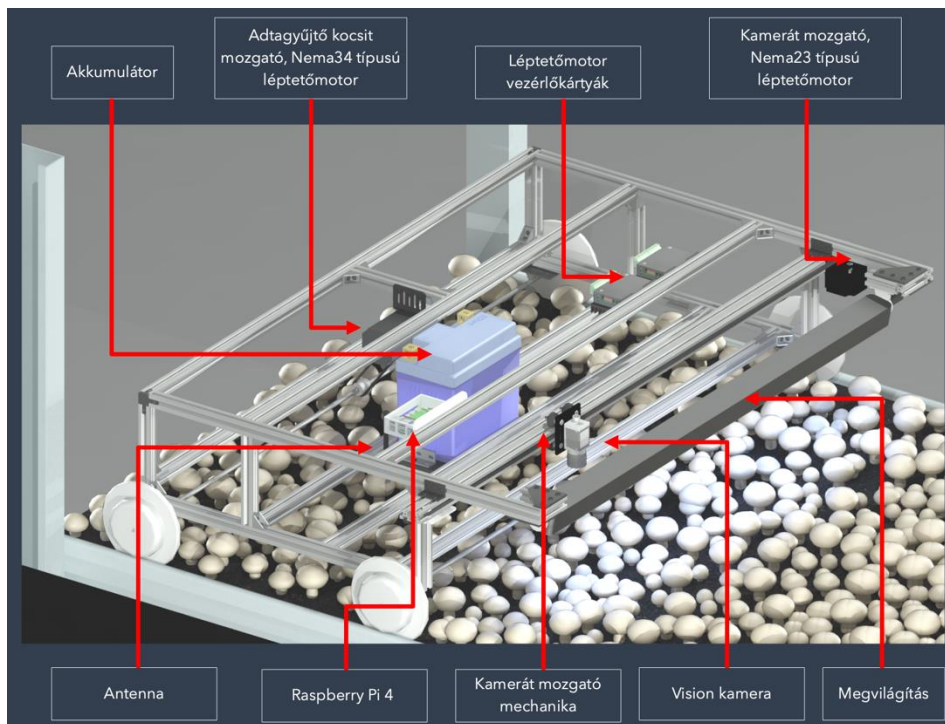
31. ábra: Adatgyűjtő kocsi helyzeti vezérlőjének alapvető elemei (Saját szerkesztésű ábra)



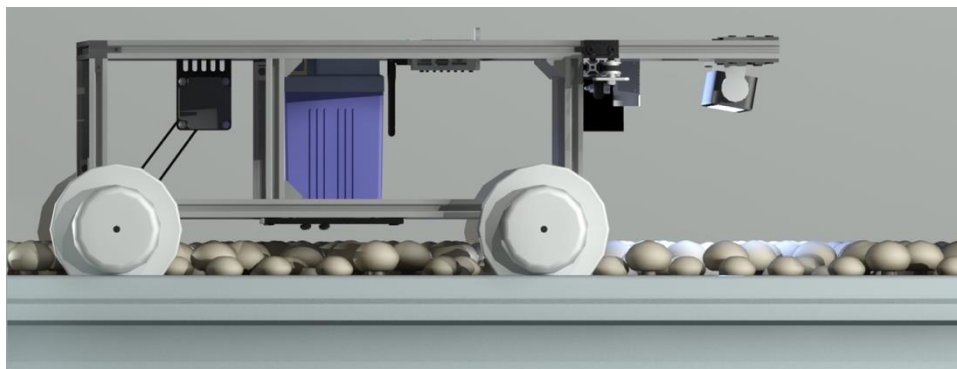
# 11 Adatgyűjtő kocsik megvalósításának megtervezése

## 11.1 Adatgyűjtő kocsik 3D-s megtervezése

Az adatgyűjtő kocsik terveit a minél valóságosabb tervezhetőség miatt az Autocad 2022 programmal, 3D modellezéssel végeztem. A polcrendszer méretezését a gombatermesztő berendezéseket gyártó cégek által közreadott méretekkel végeztem. [12] Némely, a tervezéshez felhasznált berendezés, akkumulátor, kamera, lámpatest, motor és vezérlők 3D modelljét a *grabcad.com* oldalról töltöttem le az élethűbb terv elérése céljából. A szemléletesebb képek elérését fények, megvilágítás beiktatásával, az anyagi tulajdonságok hozzáadásával, majd a kép renderelésével értem el. A terv segítséget adott arra is, hogy pontosabban lehet költségvetést készíteni a megvalósítandó berendezéshez. (32-39. ábra)

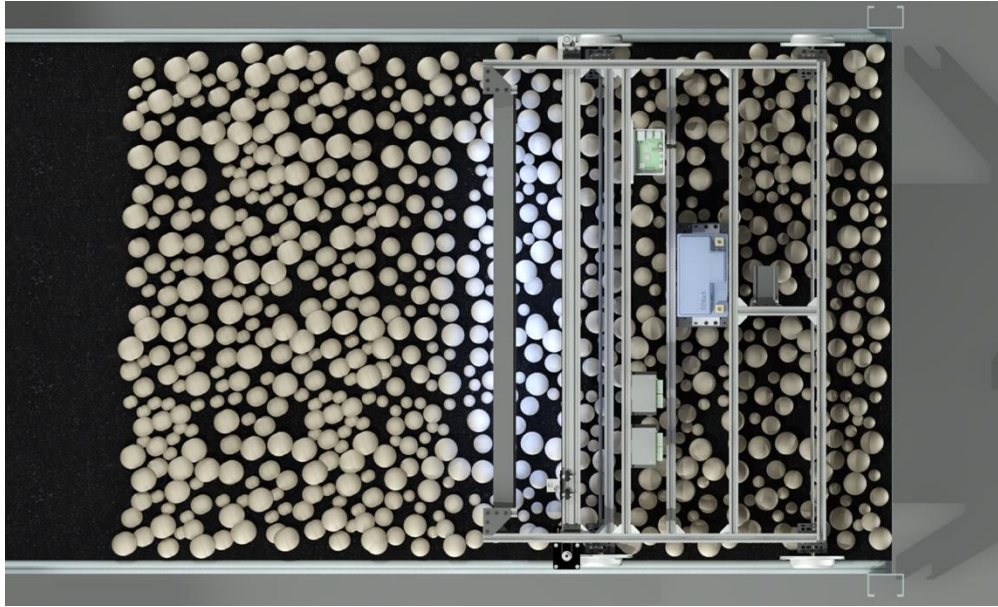


32. ábra: Az adatgyűjtő kocsik részei (saját ábra)

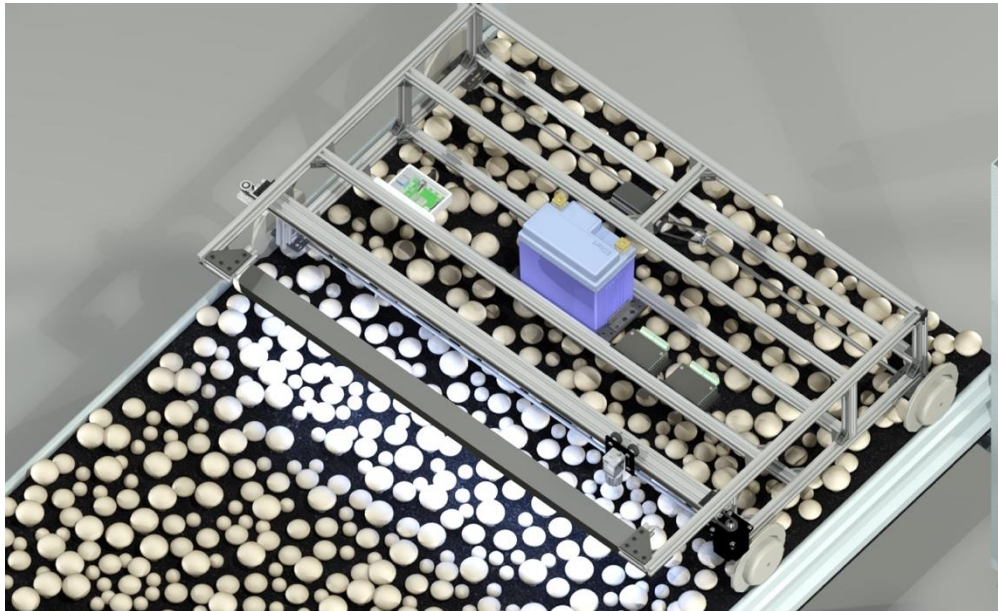


33. ábra: Adatgyűjtőkocsi jobb oldalnézetből (saját ábra)

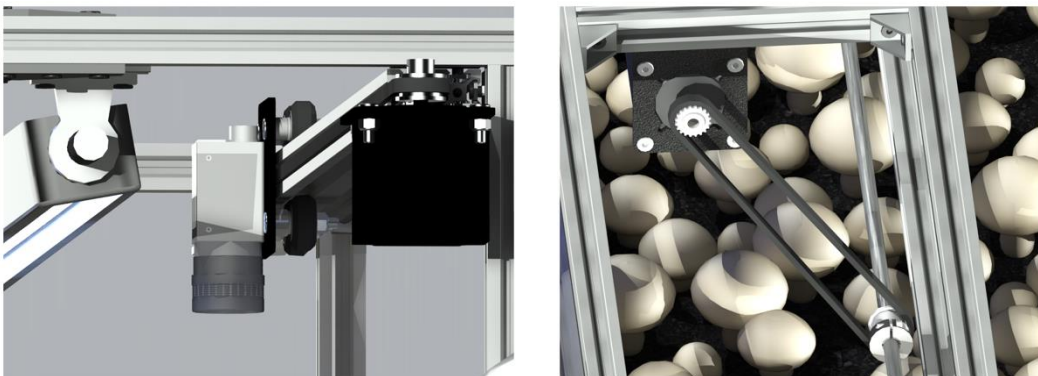




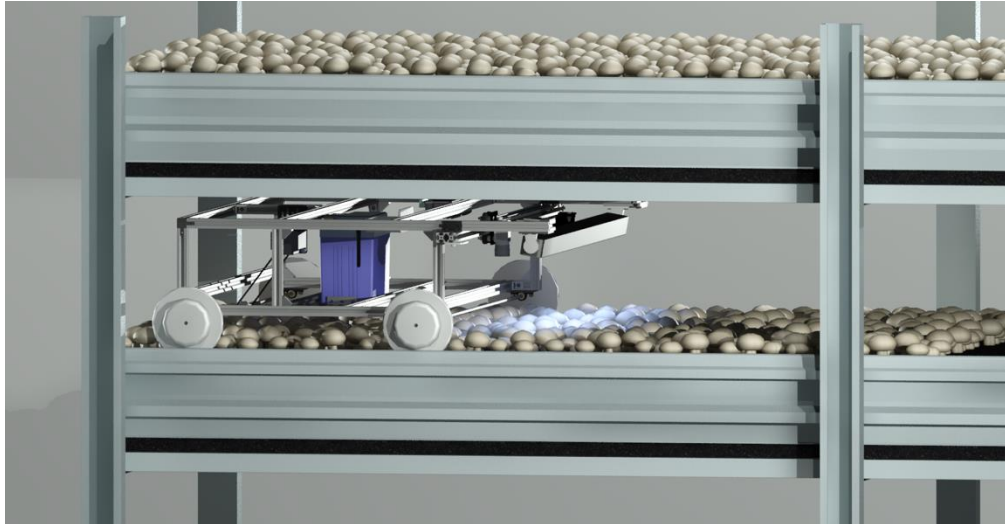
34. ábra: Adatgyűjtő kocsí prototípusának felülnézete (saját ábra)



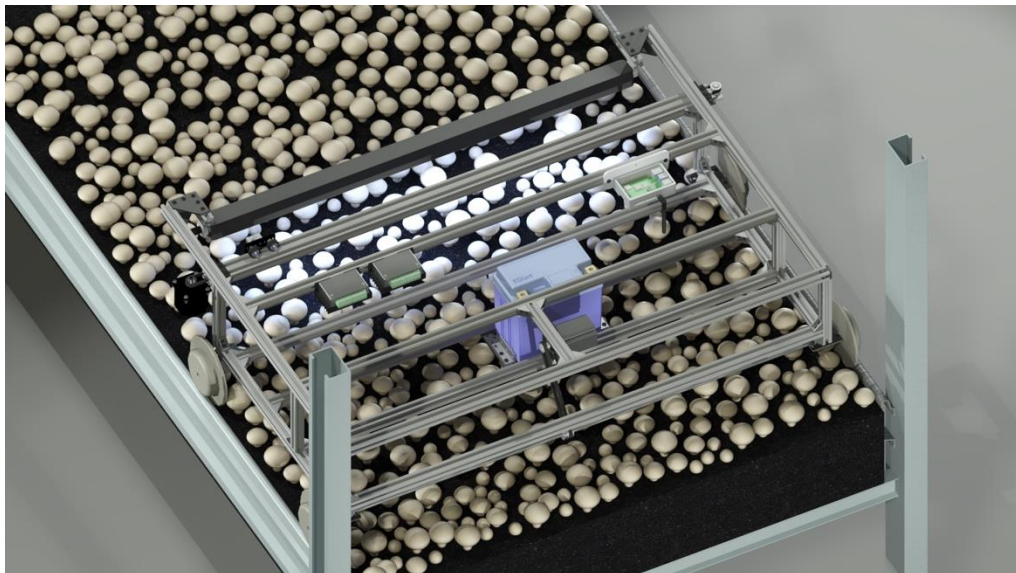
35. ábra: Adatgyűjtő kocsí helyzete a termeszto polcon (saját ábra)



36. ábra: Vision kamera mozgató mechanika, illetve a kocsí tengelyének meghajtása (saját ábra)



37. ábra: 26 cm áll rendelkezésre a felső polc alja és a gombák között (saját ábra)



38. ábra: A kép elkészítéséhez a kocsin lévő lámpa biztosítja a szükséges fényt (saját ábra)



39. ábra: Az adatkommunikáció vezeték nélkül, a tápellátás akkumulátorról történik (saját ábra)



## 11.2 Az adatgyűjtő kocsik anyagköltségeinek becslése

Elkészítettem az adatgyűjtő kocsik anyagköltségeinek becslését, hogy minél olcsóbb megvalósíthatóságát bizonyítsam (az alkatrészek aktív linkeken elérhetők). (8. táblázat).

Alkatrész megnevezése	db	Feladat	Ára	Tulajdonság	URL
Basler ace 2 (51gcBAS)	1	vision kamera, gombok fotózása, adatgyűjtés	143 000 Ft	GigE, 1/2.3" CMOS, színes, 2Mp szenzor	<a href="#">1</a>
V-Slot NEMA 23 kamera mozgató szerk.	1	kamera mozgató mechanika	70 870 Ft	1500mm, motorral, teljes kit	<a href="#">2</a>
Raspberry Pi 4B	1	adatgyűjtő kocsik vezérlése	40 929 Ft	8GB RAM 32GB ROM	<a href="#">3</a>
Műszerdoboz	1	Raspberry védelme	2 811 Ft	IP65, ABS polisztirol	<a href="#">4</a>
Antenna	1	Raspberry wireless kommunikáció	2 031 Ft	Dual band 2.4GHz/5GHz	<a href="#">5</a>
Léptetőmotor vezérlő	2	Léptető motorok vezérlése	62 800 Ft	HSS86	<a href="#">6</a>
LED-es világítás	1	Termőterület megvilágítása a fotózáshoz	54 000 Ft	1100mm, 35W, IP67	<a href="#">8</a>
Akkumulátor	1	kocsik tápellátása	29 500 Ft	48V 100Ah Li-Ion	<a href="#">9</a>
Alumínium szerelő lemez	2	Akkumulátor tartó lemez	5 777 Ft	6061 Al 155x75x6mm	<a href="#">10</a>
Nema 34 léptető motor	1	kocsik hajtása	34 290 Ft	8.7Nm, 6A, 3,6V	<a href="#">11</a>
NEMA 34 léptető motor rögzítő lemez	1	NEMA 34 motor rögzítése	3 270 Ft	6063 Al 60x90x3mm	<a href="#">12</a>
Optibelt szíjtárcsa	1	Kocsik tengely hajtása	3 556 Ft	Szíjtárcsa hajtott tengelyre	<a href="#">13</a>
Optibelt vezérműszíj	1	kocsik tengelye és Nema 23 motor közé	9 812 Ft	tengely hajtószíj	<a href="#">14</a>
Lineáris tengely 10mm	2	kerékösszekötő tengely	10 768 Ft	átm. 10mm	<a href="#">15</a>
Fém reteszhornyos nyomkarimás kerék	4	kocsik ezen gurul az állványzaton (közelítő ár)	24 000 Ft	karimás, csapágy nélk. kerék	<a href="#">16</a>
Csapágyházas csapágy	4	2 tengely felfogatása	12 548 Ft	IGUS KSTM 10mm	<a href="#">17</a>
Tengely rögzítő lemez	4	tengelycsapágyhoz	2 595 Ft	65x65x3mm	<a href="#">18</a>
Extrudált alumínium építőprofil (11 m)	1	kocsik szerkezetének felépítése	48 212 Ft	20x20mm V-slot	<a href="#">19</a>
Alumínium sarokrögzítő	34	szerkezeti elemek 90°-ban összekötéséhez	20 770 Ft	20x20x17mm ovál furattal	<a href="#">20</a>
3 utas végsarok csatlakozó	2	építőprofil végzáró, rögzítő	5 325 Ft	4 db az árban	<a href="#">21</a>
L rögzítő	4	rögzítő fül	2200 Ft	2x2 ill 2x3 ovál furattal	<a href="#">22</a>
90 fokos csatlakozólemez	4	rögzítő	2 061 Ft	M5 furatokkal	<a href="#">23</a>
M5 T anya, M5 csavar, M5 anya	150	kötőelemek a rögzítő profilokhoz és lemezekhez	20 000 Ft	20x20mm alumínium profilhoz	<a href="#">24</a>
<b>Ár összesen:</b>			<b>608 925 Ft</b>		

8. táblázat Adatgyűjtő kocsik megépítéséhez szükséges alkatrészek és áraik (saját szerkesztésű táblázat)

## 12 További kutatási lehetőségek

### Leszedendő gombák kiválasztása és a ritkításos szedési módszer megvalósítása

A termőterület napi többszöri szedése, betakarítása nagyobb termés mennyiséget eredményez, ezt *Graze picking*-nek nevezi a szakirodalom, ami magyarul ritkításos szedést jelent. A szedés lényege, hogy nem méret alapján van meghatározva, hogy mely egyedeket kell leszedni, hanem a gombák környezete dönti el azt. Mivel 24 óránként megkétszerezik a méretüket, így megfelelő helyet kell biztosítani a növekedéshez, így nem mindig a legnagyobb méretű gombákat kell leszedni, hanem azokat, melyek akadályozhatják a többi a növésben, így egy termőterületet javasolt 24 óra alatt négyszer-öttször szedni. A folyamatosan növő gombáknak (óránként 4%-os növekedést jelent) térre van szüksége, melyet a szedéssel lehet biztosítani. [23] Emiatt a legmagasabb hozam eléréséhez a szedést több műszakban, folyamatosan kell elvégezni, ami nagyszámú munkaerőt és magas költséget jelent.

A *graze picking* szedési módszert úgy tervezem megvalósítani, hogy a „felismert” gombák átmérőjének kiszámításával, és a gomba középpontjának koordinátáit el kell tárolni. Ezután meg kell vizsgálni, hogy melyik gomba van közel egymáshoz, ehhez egy gombatermesztés technológiája által meghatározott konstans értéket kell használni, ami az optimális szedési pillanatban a gomba méretét jellemzi. Amennyiben a két gomba középpontjának távolsága kisebb, mint a beállított érték és valamelyik a kettő közül nem lesz leszedve, akkor össze fognak nőni, amely hátráltatja őket a további növekedésben, alakjuk deformálódik és nehézkessé válhat a leszedésük. Ennek elkerülése érdekében le kell szedni valamelyik gombát, mielőtt azok kalapjai túl közel kerülnének egymáshoz. A magasabb terméshozam érdekében célszerű addig növeszteni a gombákat, míg az összeérés előtt azokat feltétlenül le nem kell szedni. A fehér csiperkegomba ismert növekedési sebességével megbecsülhető, hogy mekkora lenne a gomba mérete a következő ütemezett szedés idejére, így eldönthető, hogy a gombákat le kell-e szedni, vagy elegendő a következő ütemezett szedés alkalmával, mert nem fognak azok összeérni.

## **Adatgyűjtő kocsi pozíciójának pontosítása és hibadetektálás**

További kutatási és fejlesztési feladatként szenzorokkal is fel kell szerelni az adatgyűjtő kocsit, melyek segítségével pontosítani lehet a kocsi helyzetének meghatározását, illetve fel lehet ismerni a pozícióhibát, mely adódhat léptetőmotorok pontatlanságából, vagy a kocsi kerekének szorulása, elpörgése esetén. Ezek detektálását a kocsi felső keretén elhelyezett optokapuk segítségével valósítanám meg, melyek érzékelik, ha a kocsi feletti polcon lévő apró fém lapok megakadályozzák, hogy az optokapun áthaladjon a fény, így a kocsi megérkezett a megfelelő pozícióba.

## **Adatgyűjtő kocsi tápellátásának és töltésének megvalósítása**

Ahhoz, hogy az adatgyűjtő rendszer teljes mértékben vezeték nélkül tudjon működni és az adatgyűjtő kocsi is szabadon tudjon mozogni, nem csak a kommunikációs csatornákat, hanem a tápellátást is izolált módon kell megvalósítani, melyet akkumulátorral oldanék meg. A kocsi rendelkezik elegendő hellyel, az akku elhelyezésére, melynek töltése a liften valósítható meg, mikor a kocsi végzett az adatgyűjtéssel. Céлом, hogy megfelelő technológiájú és kapacitású akkumulátort válasszak, és megtervezzem az adatgyűjtő kocsi automata töltő csatlakozását.

## 13 Összefoglalás

A csiperkegomba fogyasztás világszerte egyre nő és a jövőben egyre nagyobb szerepet kap a fenntartható élelmiszertermelésben. A termesztési költségek egyik legnagyobb részét a kézi szedés bére teszi ki. Jelenleg azonban nem létezik megbízhatóan működő, teljesen automata gombaszedési megoldás, mely teljes mértékben helyettesíteni tudná a kézzel végzett gombaszedést. Kutatásom ezért a csiperkegomba automatizált szedésének technológiai fejlesztésére irányul.

Python programozási nyelven fejlesztettem egy gombafelismerő programot, amivel lehetővé vált a gombák intelligens detektálása, mellyel megoldottam a képfeldolgozás során felmerülő problémákat (összenövő gombák szegmentálása, micéliumok eltüntetése a képről). Az *OpenCV* és *NumPy* programkönyvtárak alkalmazásával sikerült lecsökkentettem a képfelismerő szoftvernek egy gyengébb hardveren való futásidejét 2.16 másodpercre. Az *OpenCV Watershed* algoritmusának felhasználásával az összenőtt gombák szétválasztása során, azok 93.67%-át sikerült helyesen szétválasztani és felismerni. Megoldást találtam a képek szélén elhelyezkedő gombák helyes felismerésére, melynek köszönhetően nem szerepel egy gombához több szedési koordináta. Kidolgoztam egy képletet, mellyel az adatgyűjtő rendszer vonatkoztatási rendszerei összekapcsolhatók, így kiszámítható a pontos szedőfej koordináta a felismert gomba pozíciója alapján.

Elkészítettem az adatgyűjtő egység prototípusának méret pontos 3D-s tervét, mely képes a modern gombatermesztő állványzat polcán végig haladni, a termőterületet végig fotózni és a képeket eljuttatni a képfeldolgozó egységnek, úgy, hogy a berendezés integrálható legyen a már elterjedt gombatermesztő rendszerekbe. A legfontosabb tervezési alapelveim a könnyű szerkezet és a lehető legolcsóbb megvalósíthatóság volt. Ajánlást tettem az adatgyűjtő kocsit hajtó, vezérlő, és a képrögzítő elemekre, melyek becsült anyagköltsége 608 ezer Ft-ba kerülne. Az eszközök közötti kommunikációt, vezeték nélküli kapcsolaton keresztül, *mesh network* infrastruktúrával, jelentősen olcsóbban valósítottam meg, mintha az vezetéken keresztül történt volna, esetleg energialáncos kábelvezetést kellett volna használni hozzá. A gombaszedés gépesítésével és technológiájának automatizálásával csökkenthető lenne a gombaszedés költsége.

## 14 Köszönetnyilvánítás

Ezúton is szeretném megköszönni témavezetőmnek, **Dr. Géczy Attilának**, a BME ETT egyetemi docensének a lelkiismeretes segítségnyújtását és hasznos konzulensi javaslatait, nagyon sokat jelentett biztatása, lelkesítése a dolgozat megírásához.

Szeretném megköszönni konzulensemnek, **Dr. Farkas Csabának**, a BME ETT adjunktusának a segítségét, tanácsait, kérdéseimre való gyors reakcióját.

Külön kiemelném **Dr. Geösel Andrásnak**, a Magyar Agrár- és Élettudományi Egyetem Kertészettudományi Intézet igazgatójának segítségét, mely nélkül a kutatás nem valósulhatott volna meg. Köszönöm a képfeldolgozás fejlesztéséhez elengedhetetlen fényképeket, amit tőle kaptam, valamint a MATE Zöldség- és Gombatermesztési Tanszék gombatermesztő laboratóriumába való bejutás lehetőségét és a gombatermesztéssel kapcsolatos rengeteg szakmai információt.

## 15 Irodalomjegyzék

- 1] „Központi Statisztikai Hivatal,” [Online]. Available: [https://www.ksh.hu/stadat\\_files/jov/hu/jov0051.html](https://www.ksh.hu/stadat_files/jov/hu/jov0051.html). [Hozzáférés dátuma: 17 09 2022].
- 2] S. István, Gombatermesztés, Budapest: Mezőgazda Kiadó, 1994.
- 3] L. Occhipinti, „Monterey Mushrooms,” 1 július 2020. [Online]. Available: <https://www.montereymushrooms.com/blog/reasons-mushrooms-are-smart-sustainable-choice>.
- 4] D. Dajewski, „new food magazine,” [Online]. Available: <https://www.newfoodmagazine.com/article/166174/coping-with-labour-shortages-for-mushroom-picking/>. [Hozzáférés dátuma: 10 09 2022].
- 5] D. Gernon, „Mushroom harvesting automation,” 22 November 2021. [Online]. Available: <https://horticultureconnected.ie/horticulture-connected-print/mushroom-harvesting-automation/>.
- 6] V. d. T. -. M. Machinery. [Online]. Available: <https://www.youtube.com/watch?v=iBENPHI4mBA>. [Hozzáférés dátuma: 17 09 2022].
- 7] V. d. T. -. M. Machinery. [Online]. Available: <https://www.youtube.com/watch?v=bscKjv0LCJ0&list=TLGGJVIIUirBDvYxNzA5MjAyMg>. [Hozzáférés dátuma: 17 09 2022].
- 8] „Mycionics Intelligent Harvesting Solution,” Mycionics Inc., [Online]. Available: <https://mycionics.com/faq/>. [Hozzáférés dátuma: 16 09 2022].
- 9] „Changing the mushroom harvesting business.,” 16 09 2022. [Online]. Available: <https://techbrew.com/mushroom-harvesting-robot/>.
- 10] „What is OpenCV?,” [Online]. Available: <https://opencv.org/about/>. [Hozzáférés dátuma: 26 09 2022].
- 11] „NumPy User Guide,” [Online]. Available: <https://numpy.org/doc/stable/user/whatisnumpy.html>. [Hozzáférés dátuma: 25 09 2022].
- 12] [Online]. Available: <https://www.mushroomshelves.com/mushroom-shelving/>. [Hozzáférés dátuma: 10 10 2022].
- 13] „Variometrum,” [Online]. Available: <https://variometrum.hu/hu/energialanc-30x15mm-r28-nyithato?language=hu->



- hu&currency=HUF&gclid=Cj0KCQjwnbmaBhD-ARIsAGTPcfX5B7Gj43QEz-XMPi7\_tEbPsb\_uRUdstGxU5dL\_bydw6sZWFwnL\_OEaAmZ7EALw\_wcB. [Hozzáférés dátuma: 18 10 2022].
- 14] „ATO,” [Online]. Available: <https://www.ato.com/Content/doc/nema-34-stepper-motor-specs.pdf>. [Hozzáférés dátuma: 15 10 2022].
- 15] „Openbuilds partstore,” [Online]. Available: <https://openbuildspartstore.com/v-slot-nema-23-linear-actuator-belt-driven/>. [Hozzáférés dátuma: 10 10 2022].
- 16] „Basler products: Basler ace 2,” [Online]. Available: <https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace2/a2a1920-51gcbas/#specs>. [Hozzáférés dátuma: 16 10 2022].
- 17] „Basler Product Documentation,” [Online]. Available: [https://docs.baslerweb.com/network-configuration-\(gige-cameras\)](https://docs.baslerweb.com/network-configuration-(gige-cameras)). [Hozzáférés dátuma: 16 10 2022].
- 18] „Basler Lens,” [Online]. Available: <https://www.baslerweb.com/en/products/lenses/fixed-focal-lenses/basler-lens-c125-0418-5m-p-f4mm/>. [Hozzáférés dátuma: 10 10 2022].
- 19] „Basler,” [Online]. Available: <https://www.baslerweb.com/en/sales-support/tools/lens-selector/>. [Hozzáférés dátuma: 10 10 2022].
- 20] „The PiHut,” [Online]. Available: <https://thepihut.com/products/16-universal-inputs-8-layer-stackable-card-for-raspberry-pi>. [Hozzáférés dátuma: 10 10 2022].
- 21] J. Oerlemans, „GRAZE PICKING BY PLANNING,” 19 November 2017. [Online]. Available: <https://www.mushroomoffice.com/graze-picking-by-planning-in-mushroom-growth/>.
- 22] M. Inc., 6 július 2021. [Online]. Available: <https://www.youtube.com/watch?v=2vv4rQj52Bc>.
- 23] F. Torres, December 2020. [Online]. Available: [https://www.researchgate.net/publication/347507086\\_Comparison\\_of\\_numerical\\_methods\\_in\\_code\\_as\\_solvers\\_for\\_simulation\\_of\\_robotic\\_systems](https://www.researchgate.net/publication/347507086_Comparison_of_numerical_methods_in_code_as_solvers_for_simulation_of_robotic_systems).
- 24] T. & I. BC, „TechBrew Robotics,” 3 március 2022. [Online]. Available: <https://www.youtube.com/watch?v=f4d0Q4u0Z9U>.
- 25] „Statista.com,” [Online]. Available: <https://www.statista.com/statistics/257314/per-capita-consumption-of-fresh-mushrooms-in-the-us/>. [Hozzáférés dátuma: 17 09 2022].

- 26] „GreenAI mushroom growing,” [Online]. Available: <https://www.green-al-mushroom.ru/eng/catalog/tovar/stellazh-6-yarusnyy/>. [Hozzáférés dátuma: 19 09 2022].
- 27] T. Danka, „Is NumPy really faster than Python?,” 2020.
- 28] „kcalKitchen,” [Online]. Available: <https://kcalKitchen.co.uk/product/white-mushrooms/>. [Hozzáférés dátuma: 19 10 2022].
- 29] „Arduino docs,” [Online]. Available: [https://store.arduino.cc/products/arduino-mega-2560-rev3?\\_gl=1\\*azb7um\\*\\_ga\\*MTEyMjMMDg3LjE2NjU3NjQyNzA.\\*\\_ga\\_NEXN8H46L5\\*MTY2NjM0NzU1Ni42LjEuMTY2NjM0NzU4MC4wLjAuMA...](https://store.arduino.cc/products/arduino-mega-2560-rev3?_gl=1*azb7um*_ga*MTEyMjMMDg3LjE2NjU3NjQyNzA.*_ga_NEXN8H46L5*MTY2NjM0NzU1Ni42LjEuMTY2NjM0NzU4MC4wLjAuMA...) [Hozzáférés dátuma: 14 10 2022].
- 30] „Központi Statisztikai Hivatal,” [Online]. Available: [https://www.ksh.hu/stadat\\_files/ara/hu/ara0004.html](https://www.ksh.hu/stadat_files/ara/hu/ara0004.html). [Hozzáférés dátuma: 17 09 2022].
- 31] V. d. Top. [Online]. Available: <https://www.youtube.com/watch?v=iBENPHI4mBA>. [Hozzáférés dátuma: 17 09 2022].
- 32] „Mushroom business magazine,” July 2017. [Online]. Available: <https://mushroombusiness.com/magazines/95-july-2019/>.
- 33] „Basler Product Documentation,” [Online]. Available: <https://docs.baslerweb.com/a2a2448-23gcbas>. [Hozzáférés dátuma: 10 10 2022].
- 34] „Basler Product Documentation,” [Online]. Available: <https://docs.baslerweb.com/a2a2840-14gcbas>. [Hozzáférés dátuma: 10 10 2022].
- 35] [Online]. Available: [https://www.mushroomshelves.com/mushroom-shelving/#iLightbox\[gallery\\_image\\_1\]/1](https://www.mushroomshelves.com/mushroom-shelving/#iLightbox[gallery_image_1]/1). [Hozzáférés dátuma: 10 10 2022].
- 36] „Why,” [Online]. Available: <https://why.org/articles/fearing-crackdown-on-workers-chester-county-mushroom-growers-urge-hybrid-status-for-immigrant-workers/>. [Hozzáférés dátuma: 10 10 2022].

## 16 Függelék

A fekete pontok eltávolításának implementálása a módosított *fillingHoleUpdate* függvénnyel:

```
def fillingHoleUpdate(preparedImage):
    inverted = cv2.bitwise_not(preparedImage)
    found_labelsBlack,objectsBlack=(objectcreator(inverted)) #found_labels
= megtalalt objectek szama
    collectedforfilling=collect_object_pixels(objectsBlack,
found_labelsBlack)
    hossz: int=np.size(collectedforfilling)
    i:int=0
    tobrange:int = 2500 #foltok mérete
    while(i<hossz):
        ez=np.size(collectedforfilling[i])
        if ez<tombrange:
            collectedforfilling=np.delete(collectedforfilling,[i])
            hossz=np.size(collectedforfilling)
        else:
            i=i+1
    filledfinalImage =createImageFromObjects(collectedforfilling)
    filledfinalImage=255-filledfinalImage*255
    #cv2.imwrite("4fillingHole.png",filledfinalImage)
    return filledfinalImage
```