https://doi.org/10.36007/5383.2026.65

KÉPSZEGMENTÁLÓ ALGORITMUSOK ÉRTÉKELÉSE PENÉSZKOLÓNIÁK NÖVEKEDÉSÉNEK ELEMZÉSÉRE
EVALUATION OF IMAGE SEGMENTATION ALGORITHMS FOR THE ANALYSIS OF MOLD COLONY GROWTH

Balázs VÍGH[15]

ABSTRACT
The aim of this thesis is to present an image processing pipeline for analyzing the growth of mold colonies on agar plates. The developed program has a modular structure and includes preprocessing steps, different segmentation algorithms, and an evaluation module. The segmentation masks were compared with annotations made using the Ilastik software, and the evaluation was carried out with standard metrics. The methodology was tested on a limited number of images, with the main purpose of demonstrating the processing steps and presenting an expandable, modular program design. The proposed pipeline can be used to compare the performance of different segmentation algorithms and may serve as a basis for later statistical studies on larger datasets.

KEYWORDS
Algorithm evaluation, Image processing pipeline, Image segmentation, Modular software, Mold colony growth

BEVEZETŐ

A penészgombák terjedésének vizsgálata elengedhetetlen az élelmiszerbiztonság, a gyógyszeripar és az egészségügy szempontjából. Ezek a mikroorganizmusok nemcsak esztétikai romlást (elszíneződés, kellemetlen szagok, állagváltozás) idézhetnek elő, hanem veszélyes mikotoxinokat is termelhetnek, amelyek fogyasztása komoly egészségkárosodást okozhat. (Dagnas & Membré, 2013) A kolóniák számának és méretének pontos nyomon követése lehetővé teszi a növekedési dinamika kvantitatív modellezését. Ehhez azonban digitális adatokra van szükség, amelyek vagy rendelkezésre állnak, vagy saját felvételekkel kell azokat előállítani.

Mivel számunkra nem álltak rendelkezésre megfelelő képanyagok, egy korábbi munkában időzített felvételeket készítettünk a Petri-csészében növekvő penészkolóniákról. Az oltást biológus szakember végezte, a felvételekhez pedig ipari minőségű, 5 megapixeles JAI BB-500GE kamerát használtunk. A kamera fixen, a Petri-csészére merőlegesen került elhelyezésre, így biztosítva a felvételek ismételhetőségét és a geometriai torzítás minimalizálását. A megvilágítást szabályozható fényerejű LED lámpa biztosította, amely a szekrény belső felületeiről visszaverődve egyenletes, szórt fényt adott. Ez csökkentette az árnyékok és a lokális kontrasztkülönbségek hatását, bár az üvegfedél csillogása továbbra is megjelent a képeken. A felvételek során tapasztalt véletlenszerű fényváltozások lehetőséget adtak arra, hogy az alkalmazott szegmentáló algoritmusok robusztusságát eltérő képminőségi

[15] Mgr. Vígh Balázs, Selye János Egyetem, GIK, Informatikai Tanszék, vighb@ujs.sk

körülmények között is vizsgáljuk. Ugyanakkor a nagy mennyiségű képanyag feldolgozása, valamint a kolóniák számának és területének meghatározása egy hatékony, összehasonlítható és moduláris módszertan kidolgozását tette szükségessé.

Jelen cikk ennek a módszertannak a bemutatására fókuszál. Az első fejezetben áttekintjük a releváns szakirodalmat, különösen a képfeldolgozási technikák fejlődését és az algoritmusok teljesítményének értékelésére alkalmazott mérőszámokat. Ezt követően részletesen ismertetjük a kidolgozott program felépítését, a választott szegmentáló algoritmusokat és az annotált képanyag előállításához használt eszközöket. Végül bemutatjuk az eredményeket, majd a következtetéseket és lehetséges jövőbeli irányokat.

SZAKIRODALMI ÁTTEKINTÉS

A képszegmentálás célja a képen található objektumok elkülönítése a háttértől. (Li et al., 2024) Az évek során folyamatosan fejlődött. A klasszikus megközelítések viszonylag egyszerű, de hatékony algoritmusokra épültek. Ilyenek például a globális és lokális küszöbölés, ahol a pixeleket intenzitásuk alapján választják szét. (Wang et al., 2025) Továbbá a régióalapú módszerek, amelyek hasonló tulajdonságú pixeleket csoportosítanak (region growing, region merging), vagy a watershed transzformációk. Széles körben alkalmazták az élalapú szegmentálást is (Sobel, Prewitt, Canny), amely az intenzitásváltozások mentén határolta el az objektumokat. Ezek mellett megjelentek a klaszterezésalapú technikák (K-means, fuzzy C-means, Gaussian Mixture Models), valamint a gráfalapú módszerek (Graph Cuts, Normalized Cuts, Felzenszwalb–Huttenlocher), amelyek pontosabb eredményt adtak összetettebb képeken is.

A gépi tanulás megjelenésével a szegmentáló algoritmusok egyre inkább képesek lettek alkalmazkodni a bonyolult képi struktúrákhoz, miközben csökkent a kézi paraméterezés igénye. Ez a fejlődés a pontosság, illetve a robusztusság javulását eredményezte.

Az igazi áttörést a mélytanulás hozta el. A konvolúciós neurális hálózatok lehetővé tették a képek pixelenkénti feldolgozását. Jelentős mérföldkőnek számítanak az olyan architektúrák, mint a Fully Convolutional Network, amely képes mély és felszíni jellemzőket kombinálni, illetve az U-Net, amely kifejezetten az orvosi képfeldolgozásban vált meghatározóvá. (Ronneberger et al., 2015) Később megjelentek a többskálás megközelítések, például a DeepLab, amely ritkított konvolúciók és többléptékű jellemzők kombinálásával növelte a szegmentálás pontosságát. Az utóbbi években pedig a többfeladatos és transzformátor alapú architektúrák kerültek előtérbe. (Chen et al., 2021)

A szegmentálási algoritmusok teljesítményét különböző mérőszámokkal értékelik. A képminőség jellemzésére gyakran használnak olyan mutatókat, mint az MSE, PSNR vagy SSIM, amelyek a rekonstrukció vagy zajszűrés hatékonyságát mérik. A szegmentálás pontosságát az objektumszintű osztályozásra használható mutatók adják vissza, például az Accuracy, Precision, Recall, F1-score, Jaccard-index (IoU), Dice. (Müller et al., 2022; Reinke et al., 2024)

MÓDSZEREK ÉS MÓDSZERTAN

A dolgozatban egy Python nyelven fejlesztett, moduláris felépítésű programot mutatunk be, amely penészkolóniák képi feldolgozására és szegmentálására készült. A képfeldolgozási feladatokhoz az OpenCV, a NumPy és a scikit-image, az eredmények rendszerezéséhez és exportálásához pedig a pandas könyvtárat használtuk. A program felépítése lehetővé teszi új algoritmusok és előfeldolgozási módszerek egyszerű beépítését, valamint azok összehasonlító értékelését. A jelenlegi munkában a klasszikus szegmentáló algoritmusok kerültek

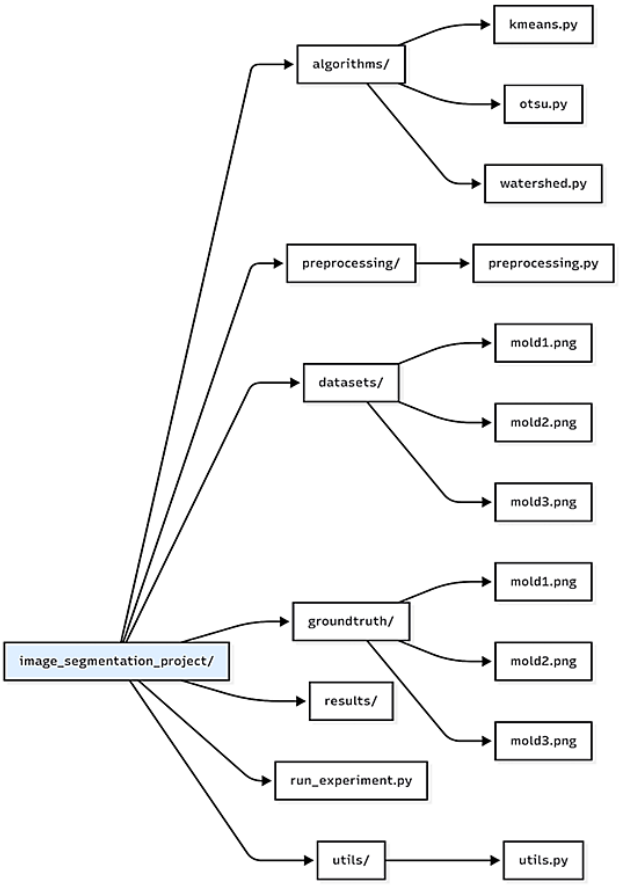
kiválasztásra, mivel ezek könnyen implementálhatók, jól átláthatók, és gyorsan tesztelhetők a meglévő képanyagon. Bár nem a legmodernebb, mélytanulás alapú módszerek, a választás lehetővé teszi a pipeline működésének demonstrálását és a különböző előfeldolgozási lépések hatásának vizsgálatát. A program struktúrája azonban biztosítja, hogy később modern algoritmusok is egyszerűen integrálhatók legyenek.

Program felépítése és könyvtárszerkezet

A projekt mappastruktúrája modulárisan épül fel (1.ábra), amely lehetővé teszi az egyes feldolgozási lépések elkülönített fejlesztését és tesztelését. A főbb elemek a következők:

- preprocessing/: az előfeldolgozási modulokat tartalmazza, például CLAHE (Contrast Limited Adaptive Histogram Equalization), gamma korrekció és fehérkiegyenlítés.
- algorithms/: a különböző szegmentáló algoritmusokat foglalja magában, mint az Otsu, Watershed, K-means.
- utils/: segédfüggvényeket tartalmaz, többek között a mérőszámok számítására (IoU, Dice, Precision, Recall), valamint a kolóniák statisztikai jellemzőinek meghatározására (darabszám, terület).
- main.py: a főprogram, amely betölti a képeket, alkalmazza az előfeldolgozást és a szegmentáló algoritmusokat, majd összegyűjti és menti az eredményeket.

Ez a moduláris felépítés biztosítja, hogy az egyes algoritmusok és előfeldolgozási lépések egymástól függetlenül fejleszthetők legyenek.



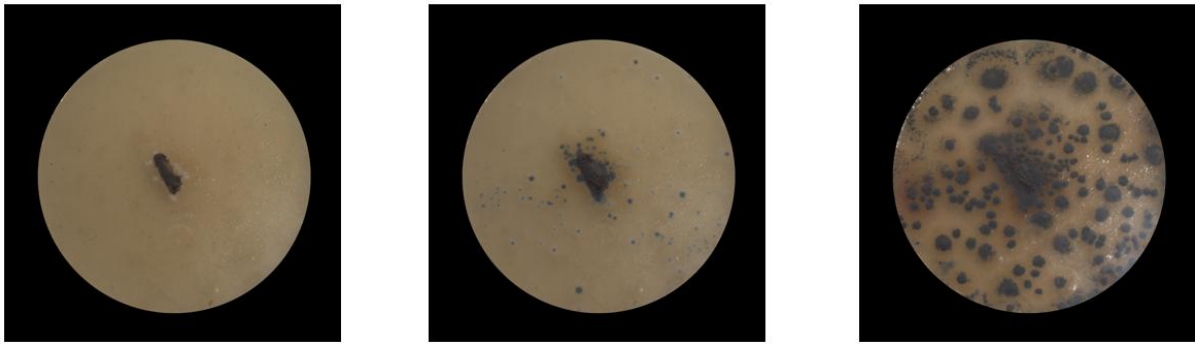
1. ábra: A program mappaszerkezete és fő moduljai.
Forrás: Saját szerkesztés

Képfeldolgozási folyamat

A teljes feldolgozási folyamatot a run_experiment.py szkript valósítja meg. A program a datasets/ könyvtárból betölti a bemeneti képeket, majd a groundtruth/ könyvtárból a hozzájuk tartozó annotált maszkokat. Ezt követően lefuttatja az összes elérhető szegmentáló algoritmust, és az így kapott kimeneti maszkokat a results/ könyvtárban tárolja. A feldolgozás után a program kiszámolja a kiértékeléshez használt mérőszámokat (IoU, Dice, Precision, Recall), valamint meghatározza a kolóniák számát és területét. Az összes eredményt a szkript összegyűjti, majd egy results_summary.xlsx fájlba exportálja, amely a későbbi összehasonlítást és elemzést teszi lehetővé.

Képkiválasztás

A feldolgozáshoz nem az összes rögzített timelapse kép került bevonásra, hanem célzottan olyan felvételek, amelyek különböző időpontokban készültek, és eltérő fényviszonyokat mutattak. A képek kiválasztása során figyelembe vettük a fényességeloszlást, hogy a mintában szerepeljenek világosabb, sötétebb és közepes megvilágítású felvételek is. A mintaválasztást tovább korlátozta, hogy csak korlátozott számú, manuálisan annotált ground truth kép állt rendelkezésre, amelyeket az Ilastik program segítségével készítettünk. Ezért a dolgozat nem törekszik a statisztikailag reprezentatív összehasonlításra. A 2. ábra szemlélteti a kiválasztott képeket.

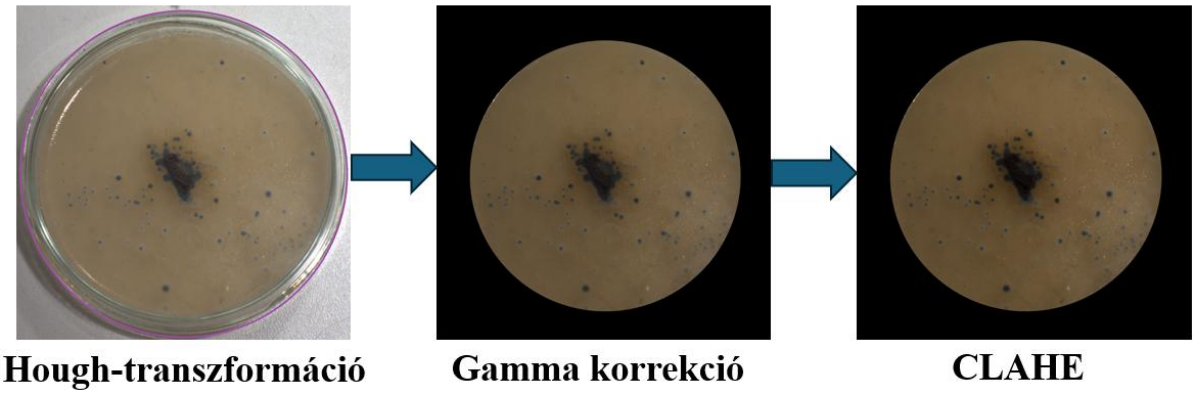


1. 2. 3.

2. ábra: A kiválasztott képek.
Forrás: Saját szerkesztés

Előfeldolgozás

A nyers képek feldolgozása előtt több előfeldolgozási lépést alkalmaztunk, hogy javítsuk a szegmentálás pontosságát és csökkentsük a fényviszonyokból adódó eltéréseket. A Hough-transzformáció kördetektáló változatát használtuk a Petri-csésze peremének azonosítására. A detektált kör sugarát szándékosan csökkentettük 200 pixellel, majd a körön kívüli részeket eltávolítottuk, így a csésze fényes széle nem került bele a feldolgozásba. A kördetektálást követően a képek fényességét gamma-korrekcióval kiegyenlítettük, majd CLAHE-t alkalmaztunk a lokális kontraszt javítására.



3. ábra: Előfeldolgozás alkalmazása.
Forrás: Saját szerkesztés

K-means szegmentálás

A k-means alapú szegmentáló eljárás a képet először az RGB színtérből a CIE LAB színtérbe alakítja át, amely jobban illeszkedik az emberi színérzékeléshez. Az algoritmus a LAB komponensek közül az a és b csatornákat használja, amelyek a színinformációt hordozzák, míg az L csatorna (világosság) nem kerül figyelembevételre. Az a és b értékekből kétdimenziós vektort képezünk minden pixelhez, majd ezeket a vektorokat a k-means klaszterező algoritmus bemeneteként használjuk. A k-means célja, hogy a pixeleket k darab színklaszterbe ossza, ahol a jelen megvalósításban a klaszterek száma alapértelmezetten három. A klaszterezés után minden pixelhez hozzárendelésre kerül egy klasztercímke, valamint megkapjuk az egyes klaszterek középpontjait a színtérben. A kiválasztott klaszterhez tartozó pixelekből bináris maszk készül. A célklaszterhez tartozó pixelek értéke 255 (fehér), míg minden más pixel 0 (fekete). A nyers maszkban előfordulhatnak zajok és szigetszerű hibák, ezért a képet további morfológiai műveleteknek vetjük alá. Először egy nyitás (open) művelet távolítja el a kisebb zajfoltokat, majd zárás (close) művelet tölti be a kisebb lyukakat a kolóniák belsejében. Végül a maszk bitenkénti invertálása történik meg, hogy a kolóniák fekete területként, a háttér pedig fehérként jelenjen meg, illeszkedve a későbbi feldolgozási lépésekhez.

Otsu-küszöbölés

A szegmentálás első lépéseként a bemeneti színes képet szürkeárnyalatos képpé alakítjuk. Ezt követően az Otsu eljárás automatikusan meghatározza azt a határértéket, amely a kép pixeleinek intenzitáseloszlását két osztályra bontja. Eredményeként bináris maszk jön létre. A küszöb alatti pixelek 0 (fekete), a küszöb felettiek pedig 255 (fehér) értéket kapnak. Szintén itt is morfológiai szűréseket végzünk.

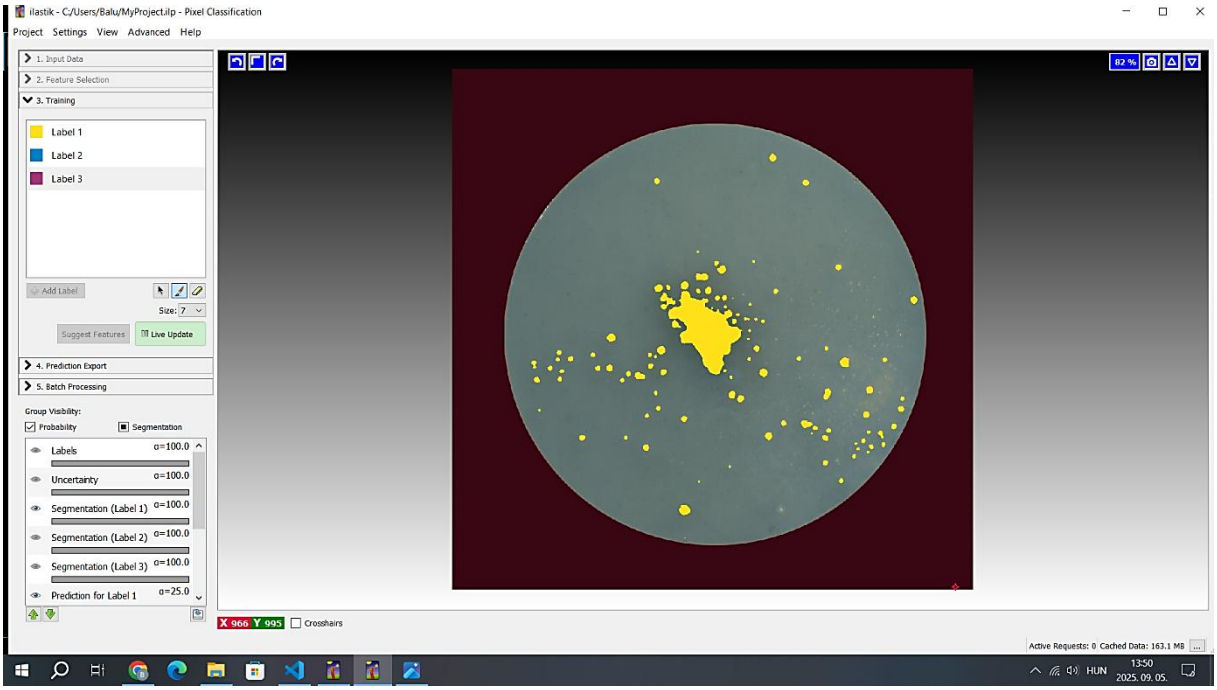
Watershed transzformáció

A watershed szegmentáció lényege, hogy a képet topográfiai felszínként értelmezi, ahol a világos pixelek magaslatokat, a sötétek pedig völgyeket jelentenek. Az algoritmus a kép intenzitáseloszlását "domborzati térképként" kezeli, és az úgynevezett vízválasztó vonalakat keresi meg, amelyek elválasztják az egyes régiókat. Az előtér azoknak a pixeleknek a halmaza, amelyek nagy valószínűséggel a penészkolóniákhoz tartoznak, míg a háttér a Petri-csésze agaros része. Ezek elkülönítéséhez a képet először binarizáltuk, majd morfológiai műveletekkel kiszűrtük a zajokat. A nagy, homogén régiókat háttérnek, a kontrasztos, tömör területeket előtérnek tekintjük. Ezzel párhuzamosan a biztos előtér pontos kijelölésére távolságtranszformációt alkalmaztunk, ahol a pixelek értékét a legközelebbi háttérponttól mért

távolság adja meg. A háttér és előtér közötti, bizonytalan zóna a két maszk kivonásával keletkezett, majd a biztos előtér pixeleinek címkézése után a watershed algoritmus ezeket a címkéket használta a régiók szétválasztásához. Az eljárás eredményeként a vízválasztó vonalak a kolóniák határainál alakultak ki, lehetővé téve a szomszédos telepek megbízható elkülönítését.

Annotált képek létrehozása

A szegmentáló algoritmusok kvantitatív értékeléséhez szükség volt ground truth maszkokra, amelyek pontosan meghatározzák a penészkolóniák, az agar és a háttér pixeleit. Ehhez az Ilastik programot használtuk, amely lehetővé teszi a pixel classification alapú annotációt. A felhasználó által kijelölt címkék (1 – penész, 2 – agar, 3 – háttér) alapján a program automatikusan tanít egy osztályozót, amely a képen minden pixelhez hozzárendeli a megfelelő osztályt. Az Ilastik belső működése során Random Forest alapú döntési fákból álló osztályozót alkalmaz, amely a pixel intenzitás és textúrajellemzők alapján végzi a besorolást.



4. ábra: Ilastik annotációs program.
Forrás: Saját szerkesztés

Teljesítménymérés

Az Ilastik szoftverrel készült annotációs maszkokat PNG formátumban mentettük el. A képek több osztályt (penész, agar, háttér) tartalmaztak, azonban a fájl szürkeárnyalatos értékei nem 0–255 intenzitást, hanem osztályazonosítókat tartalmaztak (1, 2, 3). Emiatt a fájlok szabad szemmel feketének tűntek. A szegmentációs algoritmusok által generált maszkok ezzel szemben 0 és 255 intenzitásértékekkel rendelkeztek, így közvetlen összehasonlításra alkalmatlanok voltak. A konzisztens összehasonlítás érdekében mindkét maszkot bináris formátumra alakítottuk. A ground truth esetében a vizsgált osztály (penész) pixelei 1 értéket kaptak, minden más osztályt 0-ra állítottunk. A predikciós maszkokat hasonló módon binarizáltuk, ahol minden nem nullás értéket 1-re képeztünk le. Ezt követően a két bináris mátrix pixelről pixelre történő összehasonlításával számítottuk ki a teljesítménymutatókat.

Azok a pixelek, amelyek mindkét maszkon 1-es értéket vettek fel, az igaz pozitív metszetbe kerültek, míg az eltérések hamis pozitívként vagy hamis negatívként jelentek meg.

A szegmentációs algoritmusok teljesítményének objektív értékeléséhez több, a szakirodalomban széles körben alkalmazott mérőszámot használtunk. A vizsgálat során a predikciós maszkokat a ground truth annotációval hasonlítottuk össze pixelről pixelre, és az átfedések alapján számoltuk ki a pontossági mutatókat.

Az Intersection over Union (IoU) a két maszk közös pixeleinek arányát méri az egyesített területhez viszonyítva:

$$IoU = \frac{|gt \cap pred|}{|gt \cup pred|}$$

Ez a mutató 0 és 1 közötti értéket vehet fel, ahol az 1 tökéletes egyezést jelent.

A Dice-koefficiens szintén a közös pixelek arányát vizsgálja, azonban nagyobb súlyt ad az átfedésnek, mivel kétszeresen szerepelteti a metszetet a számlálóban:

$$Dice = \frac{2 \cdot |gt \cap pred|}{|gt| + |pred|}$$

A Precízió (Precision) azt mutatja meg, hogy a predikció során jelölt pixelek közül mekkora hányad felel meg ténylegesen a ground truth maszkban jelölt osztálynak:

$$Precision = \frac{|gt \cap pred|}{|pred|}$$

Ezzel szemben az Érzékenység (Recall) azt fejezi ki, hogy a ground truth maszkban ténylegesen megjelölt pixelek közül mennyit sikerült helyesen megtalálni a predikcióban:

$$Recall = \frac{|gt \cap pred|}{|gt|}$$

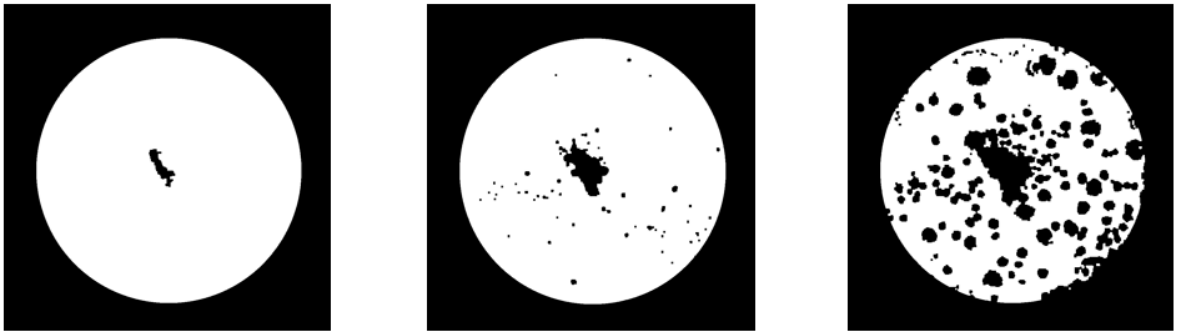
EREDMÉNYEK

A kész program lehetőséget adott arra, hogy a kiválasztott algoritmusok által generált maszkokat összehasonlítsuk előfeldolgozás nélküli és előfeldolgozással ellátott formában.

A k-means algoritmus esetében jól látható, hogy a szegmentálás eredménye nagyjából megegyezik előfeldolgozás nélkül és előfeldolgozással is, jelezve a módszer viszonylagos robusztusságát a fényviszonyok változásával szemben.



5. ábra: k-means algoritmus előfeldolgozás nélkül.
Forrás: Saját szerkesztés

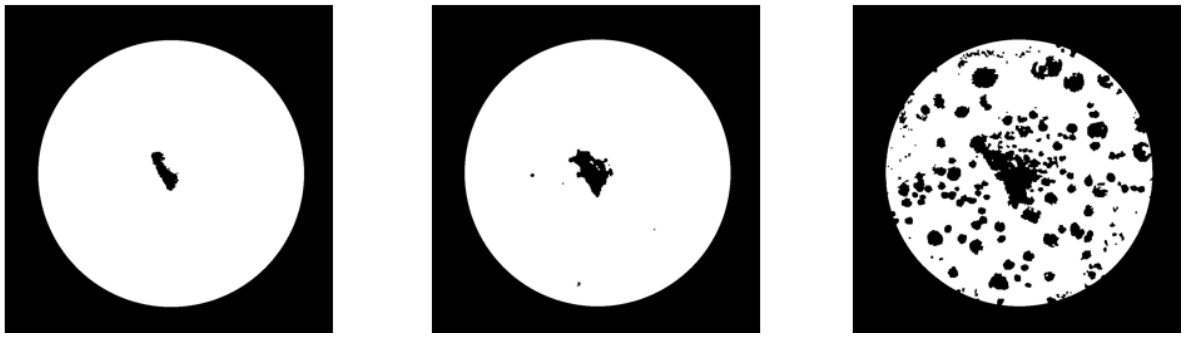


6. ábra: k-means algoritmus előfeldolgozással.
Forrás: Saját szerkesztés

Az Otsu algoritmus teljesítményét ezzel szemben jelentősen befolyásolta az előfeldolgozás. A fényesebb területeket előfeldolgozás nélkül nem találta meg, és egyes képeken még az előfeldolgozás sem nyújtott megfelelő javulást.

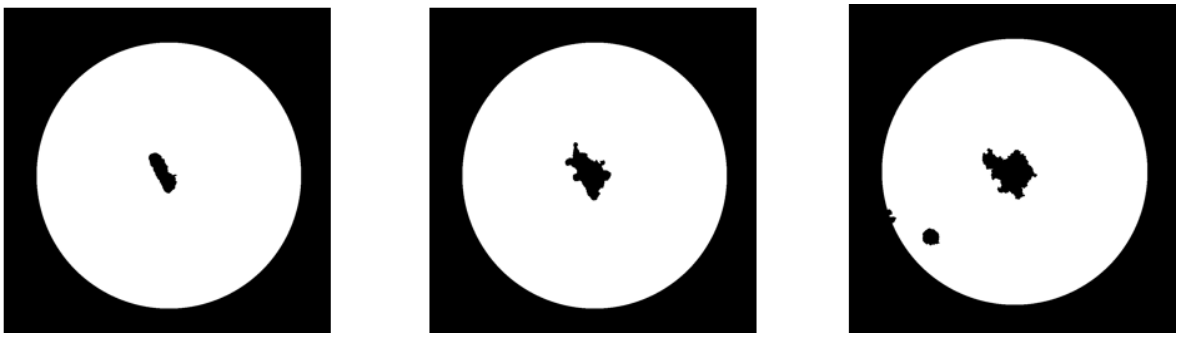


7. ábra: Otsu algoritmus előfeldolgozás nélkül.
Forrás: Saját szerkesztés



8. ábra: Otsu algoritmus előfeldolgozással.
Forrás: Saját szerkesztés

A Watershed módszernél szintén megfigyelhető, hogy az előfeldolgozás alkalmazása több kolóniát tett azonosíthatóvá, így a telepek száma és a feldolgozás pontossága jelentősen javult.



9. ábra: Watershed algoritmus előfeldolgozás nélkül.
Forrás: Saját szerkesztés



10. ábra: Watershed algoritmus előfeldolgozással.
Forrás: Saját szerkesztés

Az eredmények alapján megállapítható, hogy míg a k-means algoritmus robusztusan működött, addig az Otsu és a Watershed módszerek teljesítménye erősen függött az előfeldolgozási lépések típusától és minőségétől.

A következő két táblázat szemlélteti a mintaeredményeket kis adathalmazú képeken. Az egyik az előfeldolgozás nélküli, a másik az előfeldolgozást alkalmazó eredményeket tartalmazza. A táblázatokból jól látszik, hogy a k-means stabil teljesítményt nyújtott, míg az

Otsu és a Watershed eredményei nagyobb szórást mutattak, és számos esetben az előfeldolgozás meglététől függtek.

1. táblázat: Algoritmusok teljesítménye előfeldolgozás nélkül.

Kép	Algoritmus	IoU	Dice	Precision	Recall	Telepszám (db)	Terület (px)
mold1.png	Otsu's	0,561	0,719	0,868	0,614	1	4036
mold1.png	Watershed	0,582	0,736	1,000	0,512	1	6252
mold1.png	k-means	0,553	0,712	0,994	0,554	1	3463
mold2.png	Otsu's	0,521	0,686	0,981	0,527	8	7822
mold2.png	Watershed	0,600	0,750	0,987	0,605	1	8730
mold2.png	k-means	0,884	0,938	0,995	0,886	54	11849
mold3.png	Otsu's	0,739	0,850	0,761	0,962	88	12740
mold3.png	Watershed	0,751	0,858	0,778	0,957	3	14029
mold3.png	k-means	0,894	0,944	0,961	0,928	142	13825

2. táblázat: Algoritmusok teljesítménye előfeldolgozással.

Kép	Algoritmus	IoU	Dice	Precision	Recall	Telepszám (db)	Terület (px)
mold1.png	Otsu's	0,511	0,676	0,985	0,514	1	4041
mold1.png	Watershed	0,532	0,694	1,000	0,522	1	6761
mold1.png	k-means	0,553	0,712	0,994	0,554	1	3463
mold2.png	Otsu's	0,501	0,668	0,980	0,507	11	7782
mold2.png	Watershed	0,570	0,726	1,000	0,555	6	8245
mold2.png	k-means	0,884	0,938	0,995	0,886	54	11849
mold3.png	Otsu's	0,679	0,809	0,805	0,812	102	10473
mold3.png	Watershed	0,501	0,668	0,980	0,507	84	8462
mold3.png	k-means	0,894	0,944	0,961	0,928	142	13825

Az eredmények azt mutatják, hogy a program alkalmas a különböző szegmentáló algoritmusok összehasonlító értékelésére, valamint az előfeldolgozás hatásának vizsgálatára. A további munka során célszerű lenne nagyobb és változatosabb adathalmaz bevonása, pontosabb ground truth annotációk előállítása, valamint fejlettebb gépi tanuláson alapuló módszerek kipróbálása. A program skálázhatósága GPU alapú futtatással tovább javítható, ami a számítási hatékonyság növekedését is eredményezné.

FELHASZNÁLT IRODALOM

Folyóiratcikk

Dagnas, S., & Membré, J.-M. (2013). Predicting and Preventing Mold Spoilage of Food Products. Journal of Food Protection, 76(3), 538–551. https://doi.org/10.4315/0362-028X.JFP-12-349

Reinke, A., Tizabi, M. D., Baumgartner, M., Eisenmann, M., Heckmann-Nötzel, D., Kavur, A. E., Rädsch, T., Sudre, C. H., Acion, L., Antonelli, M., Arbel, T., Bakas, S., Benis, A., Buettner, F., Cardoso, M. J., Cheplygina, V., Chen, J., Christodoulou, E., Cimini, B. A., … Maier-Hein, L. (2024). Understanding metric-related pitfalls in image analysis validation. Nature Methods, 21(2), 182–194. https://doi.org/10.1038/s41592-023-02150-0

Wang, G., Li, Z., Weng, G., & Chen, Y. (2025). An overview of industrial image segmentation using deep learning models. Intelligence & Robotics, 5(1), 143–180. https://doi.org/10.20517/ir.2025.09

Konferenciacikk

Li, M., Luo, J., Li, X., & Zhao, Y. (2024). An overview of traditional and deep-learning-based image segmentation methods. In C. Qin & H. Zhou (Eds.), International Conference on Image Processing and Artificial Intelligence (ICIPAl 2024) (p. 59). SPIE. https://doi.org/10.1117/12.3035203

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation (No. arXiv:1505.04597). arXiv. https://doi.org/10.48550/arXiv.1505.04597

Preprint

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A. L., & Zhou, Y. (2021). TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2102.04306

Müller, D., Soto-Rey, I., & Kramer, F. (2022). *Towards a Guideline for Evaluation Metrics in Medical Image Segmentation* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2202.05273