

ПІЦУН ОЛЕГ

Західноукраїнський національний університет

<https://orcid.org/0000-0003-0280-8786>e-mail: o.pitsun@wunu.edu.ua**БЕРЕЗЬКИЙ МИКОЛА**

Західноукраїнський національний університет

<https://orcid.org/0000-0001-6507-9117>e-mail: mykolaberezkyi@gmail.com**ШИМЧУК МИРОСЛАВ**

Західноукраїнський національний університет

MLOPS ПІДХІД ДЛЯ АВТОМАТИЧНОЇ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ

Розробка штучного інтелекту та алгоритмів класифікації та сегментації зображень значною мірою сприяла розвитку технологій у сфері автоматичної діагностики з мінімальною участю людини. Ключовою особливістю цього типу завдань є необхідність використання великої кількості даних та потреба у значних обчислювальних ресурсах. Також існує потреба у використанні хмарних обчислень для спільної роботи над проектами. При аналізі таких специфічних та складних зображень, як імуногістохімічні, гістологічні та цитологічні, використання лише алгоритмів підходу є недостатнім. Тому поширення набуло використання згорткових нейронних мереж з архітектурою U-net для автоматичної сегментації. У цій статті запропоновано структуру життєвого циклу сегментації біомедичних зображень на основі практики MLOps. Особливістю цього підходу є розробка механізму обробки зображень за допомогою масок зображень.

Ключові слова: Глибоке навчання, мікросервіси, архітектура програмного забезпечення, біомедичні зображення.

PITSUN OLEP**BEREZKY MYKOLA****SHYMCHUK MYROSLAV**

West Ukrainian National University

MLOPS APPROACH FOR AUTOMATIC IMAGE SEGMENTATION

The development of artificial intelligence and algorithms for classification and image segmentation has significantly contributed to the development of technologies in the field of automatic diagnosis with minimal human participation. The key feature of this type of task is the need to use a large amount of data and the need for significant computing resources. There is also a need to use cloud computing for collaborative work on projects. When analyzing such specific and complex images as immunohistochemical, histological, and cytological, only the use of approach algorithms is insufficient. Therefore, the use of convolutional neural networks with the U-net architecture for automatic segmentation has become widespread. The choice of tools for implementing the pipeline plays an important role in ensuring a clear and efficient team process. Key tools include containerization systems, infrastructure as code delivery tools, and CI/CD platforms such as GitHub Actions. When designing a pipeline for machine learning applications, you should consider the available resources (hardware or cloud). The selection of frameworks for working with data and implementing U-nets should be considered during the automatic segmentation stage. The advantage, and at the same time the limitation, of this system is that it specializes only in automatic segmentation using U-net. However, the image processing stage and the generation of a dataset for segmentation tasks is a distinct feature that distinguishes this system from others. The main goal of this article is to develop a mechanism for creating, training, and managing neural network models designed for automatic segmentation of immunohistochemical images.

This paper proposes a life cycle structure for biomedical image segmentation based on MLOps practices. A feature of this approach is the development of a mechanism for image processing using image masks.

Keywords: Deep learning, microservices, software architecture, biomedical images.

Стаття надійшла до редакції / Received 03.06.2025

Прийнята до друку / Accepted 26.06.2025

Постановка проблеми

Інфраструктура машинного навчання вимагає ефективних механізмів управління та контролю даних. Це пов'язано з тим, що сучасні технології штучного інтелекту демонструють високий рівень продуктивності, але водночас вимагають значних обчислювальних ресурсів. Зокрема, це стосується апаратного, програмного забезпечення, хмарних систем, а також механізмів інтеграції та налаштування відповідного програмного забезпечення в хмарних середовищах. Хоча використання стандартних персональних комп'ютерів є прийнятним для виконання завдань класифікації або сегментації даних, їх застосування обмежується лише локальними завданнями. З огляду на зростаючу складність розробки систем штучного інтелекту, існує потреба у спеціалістах MLOps. Інженери MLOps відіграють ключову роль, виступаючи посередниками між командами розробників програмного забезпечення та інженерами. Формування набору даних є ключовим етапом у розробці будь-якої системи класифікації або автоматичної сегментації даних. У зв'язку з цим актуальним завданням є розробка єдиного підходу до структурування файлів зображень, що сприятиме збільшенню обсягу оброблюваних даних. Для забезпечення уніфікації процесу генерації наборів даних у рамках цього дослідження розроблено спеціалізований програмний модуль. Цей модуль дозволяє встановлювати правила генерації набору даних та адаптувати існуючі набори даних до заданого шаблону. Головною метою цієї статті є розробка механізму створення, навчання та управління моделями нейронних

мереж, призначеними для автоматичної сегментації імуногістохімічних зображень.

Для вирішення проблеми необхідно вирішити такі завдання:

- розробити програмний модуль для формування вимог до набору даних для подальшої автоматичної сегментації на основі підходу U-net;
- реалізувати механізми безперервної інтеграції та доставки коду (CI/CD) з функцією генерації версій програмного забезпечення, що допоможе автоматизувати процес сегментації біомедичних зображень.

Аналіз досліджень та публікацій

AWS Recognition Custom Labels надає інструменти для застосування штучного інтелекту до різноманітних завдань, таких як виявлення об'єктів на зображеннях, а також пропонує широкий спектр додаткових ресурсів, включаючи відеопрезентації. У [2] автори розробили робочий процес для класифікації зображень за допомогою згорткових нейронних мереж, надаючи стратегії та понад 872 рекомендації щодо вибору оптимальних гіперпараметрів CNN.

У [3] запропоновано підхід до отримання, аналізу, зберігання, обробки та автоматичної діагностики біомедичних зображень. У [4] представлено робочий процес для класифікації п'яти типів лейкоцитів за допомогою згорткової нейронної мережі для сортування зображень у реальному часі.

У [5] автори представили інструмент для обробки біомедичних зображень з використанням елементів машинного навчання. Функції інструменту включають сегментацію та виявлення об'єктів. Однак недоліком є відсутність детального конвеєра та компонентів для безперервної інтеграції коду. Класифікація гістологічних зображень, поділена на чотири класи, описана в [6]. Автори представляють перелік ключових кроків обробки зображень та інструменти для створення архітектур для їх класифікації. Методи класифікації зображень раку молочної залози обговорюються в [7–9]. У цих дослідженнях також аналізується використання фреймворку SSSpark для роботи з великими даними та прискорення обробки зображень. У [10] запропоновано гібридну модель автоматизованої діагностики (CAD) на основі цифрових рентгенівських мамограм. Особливості розробки та використання моделей згорткових нейронних мереж для обробки імуногістохімічних зображень наведено в [11–13].

Аналіз основних функцій MLOps розглянуто в [14,15]. Наведено основну мету, поставлену перед інженером MLOps. Комплексна архітектура та робочий процес MLOps наведено в [16]. Основні інструменти, що використовуються інженером MLOps, розглянуто в [17,18]. Аналіз підходів MLOps до обробки біомедичних зображень наведено в [19].

Особливості розробки та використання моделей згорткових нейронних мереж для обробки імуногістохімічних зображень наведено в роботах [11–13].

Аналіз основних функцій MLOps розглянуто в роботах [14,15]. Наведено основну мету, поставлену перед інженером MLOps. Наскрізна архітектура та робочий процес MLOps наведено в роботі [16]. Основні інструменти, що використовуються інженером MLOps, розглянуті в роботах [17,18]. Аналіз підходів MLOps до обробки біомедичних зображень наведено в [19].

Формулювання цілей статті

Метою цієї статті є розробка механізму створення середовища для запуску проєктів на основі підходу U-net, який охоплюватиме такі аспекти:

- налаштування параметрів обладнання за допомогою скриптів з використанням методу IaasC;
- розробка правил створення необхідної структури набору даних, придатної для роботи з мережами U-net;
- створення конвеєра для автоматичної сегментації за допомогою бібліотек машинного навчання, моделей та необхідних компонентів;
- реалізація безперервної інтеграції та доставки коду за допомогою інструментів CI/CD, а також організація контролю версій коду проєкту.

Виклад основного матеріалу

Під час проєктування конвеєра для програм машинного навчання слід враховувати доступні ресурси (апаратні чи хмарні). У деяких випадках велика кількість ресурсів, таких як оперативна пам'ять, місце на жорсткому диску, потужність процесора або графічного процесора, може бути не потрібна. Однак для таких завдань, як навчання моделей для автоматичної сегментації зображень, потрібні значні обчислювальні ресурси, оскільки процес навчання моделі набагато довший та ресурсоємніший, ніж її подальше застосування на тестових даних або в реальних умовах.

Ключовим кроком у створенні конвеєра є чітке розуміння всіх задіяних процесів. Наприклад, для автоматичної сегментації важливо передбачити підготовку даних, створення оригінальних зображень з відповідними масками, налаштування та запуск мережі U-net, а також вибір гіперпараметрів, таких як кількість епох та розмір пакета. Візуалізація отриманих результатів для подальшого аналізу також є важливим аспектом.

Вибір інструментів для реалізації конвеєра відіграє важливу роль у забезпеченні чіткого та ефективного командного процесу. До основних інструментів належать системи контейнеризації, інфраструктура як інструменти реалізації коду та платформи CI/CD, такі як GitHub Actions. На етапі автоматичної сегментації слід враховувати вибір фреймворків для роботи з даними та реалізації мереж U-net.

Генерація набору даних є ключовим етапом роботи з великими даними. Головною метою створення набору даних для автоматичної сегментації за допомогою U-net є правильний поділ вибірки на навчальну та тестову частини з урахуванням правил організації каталогів.

Процес «маркування даних» передбачає маркування файлів певними ключовими словами, наприклад, «_mask». Особливістю автоматичної сегментації є необхідність поділу вибірки на вихідні зображення та маски, що відрізняється від задач класифікації, де зображення сортуються за класами шляхом розміщення їх у відповідних каталогах.

Вирішення задач класифікації та сегментації зображень у великих масштабах зазвичай вимагає спеціалізованого обладнання з високою продуктивністю процесора, великим обсягом оперативної пам'яті та потужним графічним процесором. Для запуску програмного забезпечення також необхідно додатково встановлювати фреймворки, бібліотеки тощо. Тому використання звичайного ПК для таких завдань є неефективним.

Ручне налаштування апаратного середовища або хмарних систем є трудомістким та вимагає спеціальних знань. Це займає багато часу, який можна було б витратити на безпосередню роботу з нейронними мережами. Крім того, процес встановлення додаткових бібліотек та їх інтеграції також вимагає значних зусиль.

Датасети

Машинне навчання вимагає не лише програмних засобів для виконання певного завдання, але й підготовленого набору даних, на основі якого виконуються обчислення. Формат підготовки набору даних залежить від поставленого завдання. Наприклад, для завдань класифікації вибірка зазвичай поділяється на навчальну та тестову частини у вигляді окремих каталогів у файловій системі.

У випадку автоматичної сегментації зображень за допомогою мереж U-net додатково враховується компонент "маска зображення". Хоча у відкритому доступі доступна велика кількість наборів даних, їхні файли часто організовані по-різному, що вимагає додаткової обробки перед використанням.

Критерії формування набору даних для завдань автоматичної сегментації включають:

- розмір зображення;
- колірну модель;
- структуру каталогів;
- ключові слова в іменах файлів.

Для уніфікації підходів до створення наборів даних у цій роботі розроблено окремий програмний модуль, який дозволяє адаптувати існуючі набори даних до заданого шаблону. Використання уніфікованого шаблону пришвидшує процес навчання, збільшує розмір навчальної вибірки, а також покращує точність і якість програмного модуля для автоматичної сегментації.

Однією з ключових функцій цього програмного модуля є можливість задати колірну модель для зображень навчальної та тестової вибірок. Колірна модель дозволяє зберігати інформацію про зображення та чітко розрізняти досліджувані об'єкти та фон. Завдяки детальній інформації про колір, алгоритми машинного навчання точніше ідентифікують та розрізняють різні об'єкти на зображеннях. Сучасні нейронні мережі та методи глибокого навчання значною мірою покладаються на дані про колір для сегментації та класифікації зображень.

Ще одним важливим аспектом є співвідношення сторін та розмір вхідних зображень, що суттєво впливає на якість їх обробки, класифікації та сегментації. Під час роботи з даними з різних джерел часто необхідно масштабувати зображення до єдиного розміру, щоб забезпечити їх сумісність та підвищити ефективність навчання. Рівномірні розміри зображень спрощують та покращують процес навчання. Занадто малі зображення можуть втрачати важливі деталі, а занадто великі зображення можуть вимагати значних обчислювальних ресурсів.

Крім того, різні архітектури нейронних мереж, такі як згорткові нейронні мережі, працюють із заздалегідь визначеними розмірами зображень. Тому можливість налаштування параметрів набору даних значно полегшує процес навчання. Розмір зображення також впливає на пропорції об'єктів та розподіл класів на зображенні. Наприклад, при зменшенні розміру можуть змінюватися пропорції об'єктів, що, у свою чергу, може вплинути на якість сегментації. На рисунку 1 наведено приклади імуногістохімічних зображень.

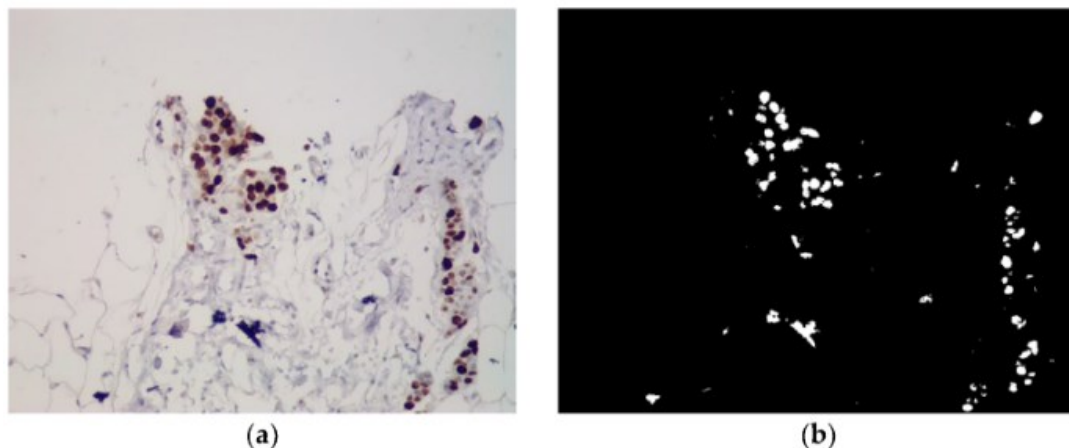


Рис. 1. Приклади імуногістохімічних зображень а) – оригінальне зображення; б) – маска

На рисунку 2 наведено графічний інтерфейс програмного модулю.

Рис. 2. Графічний інтерфейс програмного модулю

Результатом формування параметрів набору даних є JSON-файл (рис. 3), який надалі можна використовувати в програмному модулі, що дозволить сформувати необхідну структуру каталогів.

```
{
  "_token": "rglJdnCHqUur0ffm7patT4JyT37LKIdC8QWMLHI4",
  "height": "512",
  "width": "512",
  "image_type": "2",
  "color_model": "1",
  "separate_file_format": "on",
  "description": "_mask"
}
```

Рис. 3. JSON-файл із параметрами набору даних

Інфраструктура проекту

Сучасним рішенням є використання механізму «Інфраструктура як код» (IaC), який передбачає опис необхідних параметрів та залежностей у вигляді коду для подальшого автоматичного розгортання на хмарних платформах. В рамках нашої розробки цей підхід також був реалізований за допомогою інструменту Terraform.

У цьому випадку набір біомедичних зображень, зокрема типу KI-67, розміщено на платформі Zenodo для зручності обміну ним між вченими, що дозволяє порівнювати результати досліджень. Однак використання саме цього ресурсу не є обов'язковим, і можна використовувати інші платформи або ресурси.

Розділ «provider» визначає постачальника послуг, у нашому випадку це DigitalOcean. Провайдери в Terraform – це абстракції для різних хмарних платформ (наприклад, AWS, Google Cloud, Azure) та інших сервісів (наприклад, GitHub, Kubernetes). Ви можете працювати з кількома провайдерами або використовувати різні облікові записи одного провайдера.

Розділ «token, keys» відповідає за зберігання токенів та SSH-ключів, необхідних для підключення до хмари. Зазвичай цей розділ використовується для надання облікових даних та автентифікації, необхідних для доступу до різних хмарних сервісів та провайдерів.

Розділ «resource» містить інформацію про операційну систему, на якій буде розміщено наш проект, назву дроплету та інші параметри. Ресурси можуть включати віртуальні машини, бази даних, мережеві компоненти, облікові записи користувачів. Опис інфраструктури як коду (IaC) спрощує керування конфігураціями та їх версійне керування. Terraform автоматично розраховує граф залежностей та забезпечує правильний порядок створення та видалення ресурсів.

Розділ «remote-exec» містить набір команд, які мають виконуватися автоматично під час створення дроплету. Це може бути, наприклад, завантаження набору даних, програмного коду або окремих програм. Розділ «remote-exec» у конфігураціях Terraform використовується для виконання команд на віддалених машинах після створення ресурсу. Він є частиною блоку provisioner, який дозволяє виконувати дії з ресурсами під час їх створення, оновлення або видалення. Процес розгортання проекту для автоматичної сегментації показано на рисунку 4.

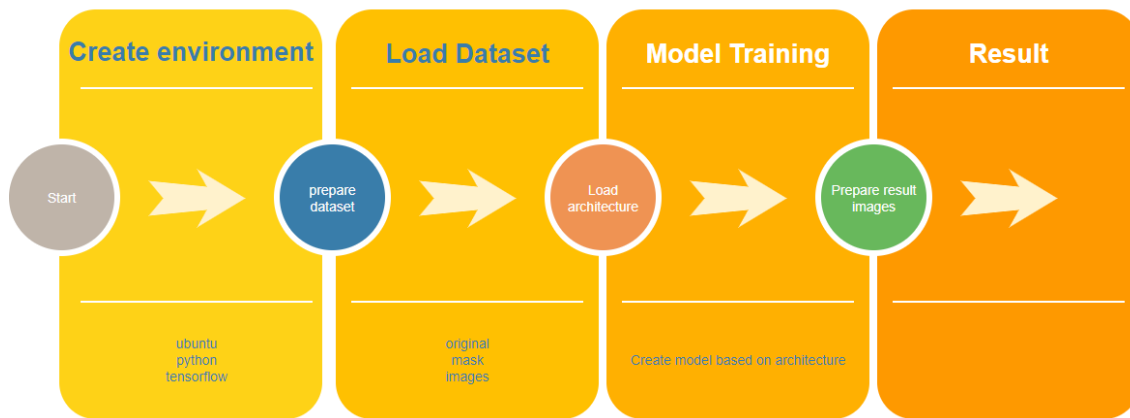


Рис. 4. Розподіл компонентів файлу main.tf

Результати сегментації зображення показано на рисунку 5.

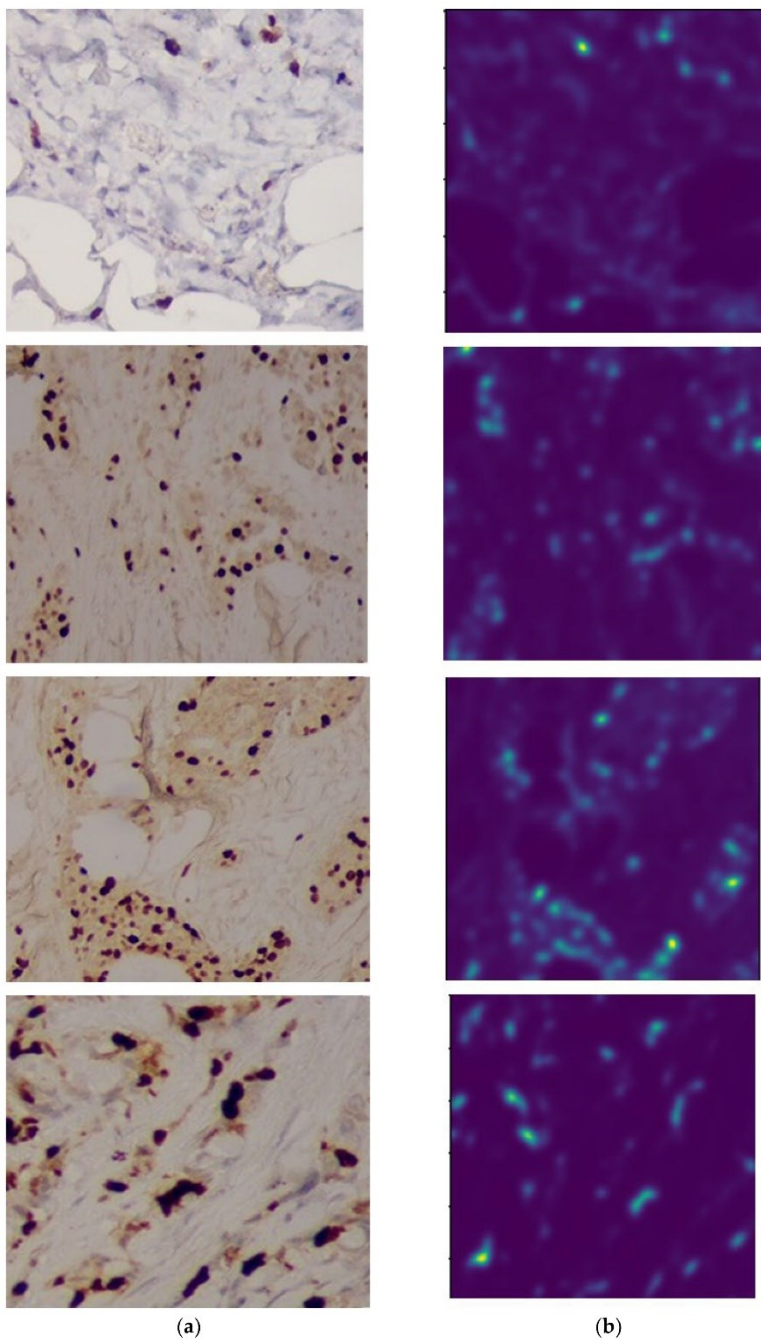


Рис. 5. Результати сегментації зображення. (a) Оригінал; (b) Результат сегментації U-net.

Аналіз рішень з використанням штучного інтелекту наведено в таблиці 1.

Таблиця 1

Аналіз рішень для використання штучного інтелекту

Програмна система	Класифікація зображень	тегування	семантичний пошук	виявлення об'єктів	регресія зображення	сегментація
Nyckel	+	+	+	+	+	-
Ximilar	+	+	+	+	+	-
Roboflow	+	+	-	+	-	+
Hasty	-	+	-	+	-	+
Levity	+	+	-	-	-	-
Google Vertex AI	+	+	+	+	-	-
AWS Rekognition Custom Labels	+	+	-	+	-	+

Висновки з даного дослідження

і перспективи подальших розвідок у даному напрямі

У цій статті представлено життєвий цикл процесу автоматичної сегментації біомедичних зображень, який базується на підході «Інфраструктура як код». Ключовою особливістю розробленого підходу є блок, що відповідає за генерацію набору даних з оригінальних зображень та масок. Цей життєвий цикл поєднує принципи DevOps та методи машинного навчання для досягнення ефективної автоматичної сегментації.

Перевагою, і водночас обмеженням, цієї системи є те, що вона спеціалізується лише на автоматичній сегментації за допомогою Unet. Однак, етап обробки зображень та генерації набору даних для завдань сегментації є чітко вираженою особливістю, яка відрізняє цю систему від інших.

Загалом, розроблені механізми дозволяють значно прискорити, стандартизувати та оптимізувати процес запуску програмного забезпечення для автоматичної сегментації за допомогою машинного навчання. Використання такого уніфікованого підходу не тільки покращує якість розробки, але й дозволяє відтворювати експерименти на різних хмарних платформах..

Література

1. Kruisz, M. (2020). *Evaluation of microservice implementation approaches for image processing* (Doctoral dissertation, Wien).
2. Brogi, A., Neri, D., & Soldani, J. (2018). A microservice-based architecture for (customisable) analyses of Docker images. *Software: Practice and Experience*, 48(8), 1461–1474. <https://doi.org/10.1002/spe.2583>
3. Lotz, J., Vogelsang, A., Benderius, O., & Berger, C. (2019). Microservice architectures for advanced driver assistance systems: A case study. In *2019 IEEE International Conference on Software Architecture Companion (ICSAC-C)* (pp. 45–52). IEEE. <https://doi.org/10.1109/ICSAC-C.2019.00016>
4. Liu, X., Liu, W., Yi, S., & Li, J. (2020). Research on software development automation based on microservice architecture. In *Proceedings of the 2020 International Conference on Aviation Safety and Information Technology* (pp. 670–677). ACM. <https://doi.org/10.1145/3434581.3434711>
5. Singh, V., & Peddoju, S. K. (2017). Container-based microservice architecture for cloud applications. In *2017 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 847–852). IEEE. <https://doi.org/10.1109/CCAA.2017.8229914>
6. Sergi, I., Leo, M., Carcagni, P., La Franca, M., Distant, C., & Patrono, L. (2022). A microservices architecture based on a deep-learning approach for an innovative fruition of art and cultural heritage. *Journal of Communications Software and Systems*, 18(2), 184–194. <https://doi.org/10.24138/jcomss-2022-0001>
7. Cerny, T., Abdelfattah, A. S., Bushong, V., Al Maruf, A., & Taibi, D. (2022). Microservice architecture reconstruction and visualization techniques: A review. In *2022 IEEE International Conference on Service-Oriented System Engineering (SOSE)* (pp. 39–48). IEEE. <https://doi.org/10.1109/SOSE55356.2022.00011>
8. Nagothu, D., Xu, R., Nikouei, S. Y., & Chen, Y. (2018). A microservice-enabled architecture for smart surveillance using blockchain technology. In *2018 IEEE International Smart Cities Conference (ISC2)* (pp. 1–4). IEEE. <https://doi.org/10.1109/ISC2.2018.8656968>
9. Berezsky, O., Pitsun, O., Datsko, T., Tsmots, I., & Teslyuk, V. (2020). Specified diagnosis of breast cancer on the basis of immunohistochemical images analysis. *CEUR Workshop Proceedings*, 2753, 129–135. <https://ceur-ws.org/Vol-2753/short5.pdf>
10. Berezsky, O., Pitsun, O., Melnyk, G., Datsko, T., Izonin, I., & Derysh, B. (2023). An approach toward automatic specific diagnosis of breast cancer based on an immunohistochemical image. *Journal of Imaging*, 9(1), 12. <https://doi.org/10.3390/jimaging9010012>

11. Berezsky, O., Pitsun, O., Melnyk, G., Derysh, B., & Liashchynskiy, P. (2022). Application of MLOps practices for biomedical image classification. *CEUR Workshop Proceedings*, 3302, 69–77. <https://ceur-ws.org/Vol-3302/short3.pdf>
12. González, G., & Evans, C. L. (2019). Biomedical image processing with containers and deep learning: An automated analysis pipeline. *BioEssays*, 41(6), 1900004.
13. Rajput, A. S., Tanwar, V. K., & Raman, B. (2021). -Score-based secure biomedical model for effective skin lesion segmentation over eHealth cloud. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(2s), Article 1–19. <https://doi.org/10.1145/3430806>
14. Navale, V., & Bourne, P. E. (2018). Cloud computing applications for biomedical science: A perspective. *PLoS Computational Biology*, 14(6), e1006144. <https://doi.org/10.1371/journal.pcbi.1006144>
15. Berezsky, O., Pitsun, O., Derysh, B., Berezska, K., Melnyk, G., & Batko, Y. (2020). Adaptive immunohistochemical image pre-processing method. In *2020 10th International Conference on Advanced Computer Information Technologies (ACIT)* (pp. 820–823). IEEE. <https://doi.org/10.1109/ACIT49673.2020.9208920>
16. Berezsky, O., Pitsun, O., Batryn, N., Berezska, K., & Dubchak, L. (2018). Modern automated microscopy systems in oncology. *CEUR Workshop Proceedings*, 2255, 311–325. <https://ceur-ws.org/Vol-2255/paper28.pdf>
17. Kumar, A., Singh, S. K., Lakshmanan, K., Saxena, S., & Shrivastava, S. (2021). A novel cloud-assisted secure deep feature classification framework for cancer histopathology images. *ACM Transactions on Internet Technology*, 21(2), Article 52 <https://doi.org/10.1145/3424221>

References

1. Krusz, M. (2020). *Evaluation of microservice implementation approaches for image processing* (Doctoral dissertation, Wien).
2. Brogi, A., Neri, D., & Soldani, J. (2018). A microservice-based architecture for (customisable) analyses of Docker images. *Software: Practice and Experience*, 48(8), 1461–1474. <https://doi.org/10.1002/spe.2583>
3. Lotz, J., Vogelsang, A., Benderius, O., & Berger, C. (2019). Microservice architectures for advanced driver assistance systems: A case study. In *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)* (pp. 45–52). IEEE. <https://doi.org/10.1109/ICSA-C.2019.00016>
4. Liu, X., Liu, W., Yi, S., & Li, J. (2020). Research on software development automation based on microservice architecture. In *Proceedings of the 2020 International Conference on Aviation Safety and Information Technology* (pp. 670–677). ACM. <https://doi.org/10.1145/3434581.3434711>
5. Singh, V., & Peddoju, S. K. (2017). Container-based microservice architecture for cloud applications. In *2017 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 847–852). IEEE. <https://doi.org/10.1109/CCAA.2017.8229914>
6. Sergi, I., Leo, M., Carcagni, P., La Franca, M., Distant, C., & Patrono, L. (2022). A microservices architecture based on a deep-learning approach for an innovative fruition of art and cultural heritage. *Journal of Communications Software and Systems*, 18(2), 184–194. <https://doi.org/10.24138/jcomss-2022-0001>
7. Cerny, T., Abdelfattah, A. S., Bushong, V., Al Maruf, A., & Taibi, D. (2022). Microservice architecture reconstruction and visualization techniques: A review. In *2022 IEEE International Conference on Service-Oriented System Engineering (SOSE)* (pp. 39–48). IEEE. <https://doi.org/10.1109/SOSE5356.2022.00011>
8. Nagothu, D., Xu, R., Nikouei, S. Y., & Chen, Y. (2018). A microservice-enabled architecture for smart surveillance using blockchain technology. In *2018 IEEE International Smart Cities Conference (ISC2)* (pp. 1–4). IEEE. <https://doi.org/10.1109/ISC2.2018.8656968>
9. Berezsky, O., Pitsun, O., Datsko, T., Tsmots, I., & Teslyuk, V. (2020). Specified diagnosis of breast cancer on the basis of immunohistochemical images analysis. *CEUR Workshop Proceedings*, 2753, 129–135. <https://ceur-ws.org/Vol-2753/short5.pdf>
10. Berezsky, O., Pitsun, O., Melnyk, G., Datsko, T., Izonin, I., & Derysh, B. (2023). An approach toward automatic specifics diagnosis of breast cancer based on an immunohistochemical image. *Journal of Imaging*, 9(1), 12. <https://doi.org/10.3390/jimaging9010012>
11. Berezsky, O., Pitsun, O., Melnyk, G., Derysh, B., & Liashchynskiy, P. (2022). Application of MLOps practices for biomedical image classification. *CEUR Workshop Proceedings*, 3302, 69–77. <https://ceur-ws.org/Vol-3302/short3.pdf>
12. González, G., & Evans, C. L. (2019). Biomedical image processing with containers and deep learning: An automated analysis pipeline. *BioEssays*, 41(6), 1900004.
13. Rajput, A. S., Tanwar, V. K., & Raman, B. (2021). -Score-based secure biomedical model for effective skin lesion segmentation over eHealth cloud. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(2s), Article 1–19. <https://doi.org/10.1145/3430806>
14. Navale, V., & Bourne, P. E. (2018). Cloud computing applications for biomedical science: A perspective. *PLoS Computational Biology*, 14(6), e1006144. <https://doi.org/10.1371/journal.pcbi.1006144>
15. Berezsky, O., Pitsun, O., Derysh, B., Berezska, K., Melnyk, G., & Batko, Y. (2020). Adaptive immunohistochemical image pre-processing method. In *2020 10th International Conference on Advanced Computer Information Technologies (ACIT)* (pp. 820–823). IEEE. <https://doi.org/10.1109/ACIT49673.2020.9208920>
16. Berezsky, O., Pitsun, O., Batryn, N., Berezska, K., & Dubchak, L. (2018). Modern automated microscopy systems in oncology. *CEUR Workshop Proceedings*, 2255, 311–325. <https://ceur-ws.org/Vol-2255/paper28.pdf>
17. Kumar, A., Singh, S. K., Lakshmanan, K., Saxena, S., & Shrivastava, S. (2021). A novel cloud-assisted secure deep feature classification framework for cancer histopathology images. *ACM Transactions on Internet Technology*, 21(2), Article 52 <https://doi.org/10.1145/3424221>