

КЛАСИФІКАЦІЯ ПШЕНИЦІ ЗА ДОПОМОГОЮ ВДОСКОНАЛЕНОЇ НЕЙРОННОЇ МЕРЕЖІ CNN-CAPSNET

¹Ярослав КУЛИК, ¹Богдан КНИШ,
¹Вінницький національний технічний університет
Yaroslav_Kulik@i.ua

Вступ. Моделі розпізнавання пшениці важливі для сільського господарства, зокрема в точному землеробстві [1]. Ці моделі забезпечують можливість автоматизації та оптимізації процесів вирощування цієї культури. Це особливо актуально при постійному підвищенні вимог до продуктивності та якості сільськогосподарської продукції. Застосування моделей розпізнавання для класифікації пшениці дозволяє підвищити врожайність, оптимізувати використання ресурсів, виявляти хвороби та шкідників, проводити моніторинг стану посівів, покращувати якість продукції тощо

Значення моделей розпізнавання пшениці в сільському господарстві полягає в їхній здатності підвищувати точність оцінки якості зерна, виконувати ідентифікацію сорту та класу насіння та покращувати процеси керування вирощуванням урожаю. Використання моделей значно полегшує та прискорює контроль за дотриманням стандартів вирощування урожаю. За допомогою моделей розпізнавання можна вчасно виявляти ознаки хвороб та шкідників, що дозволяє приймати оперативні заходи для їх контролю та запобігання поширенню, а також моделювати та оцінювати вплив різних факторів на ефективність вирощування пшениці.

Застосування моделей допоможе зменшити витрати, покращити якість продукції, а також відкрити нові можливості для сільськогосподарських підприємств.

Метою роботи є вдосконалення архітектури нейронної мережі CNN-CapsNet для класифікації пшениці. Це дасть можливість отримати нову нейромережу із підвищеною точністю для класифікації зображень пшениці, яка може бути використана для прогнозування якості урожаю пшениці та незалежної оцінки класу пшениці.

Основні результати. Архітектура CNN-CapsNet формується на основі двох частин, а саме CNN і CapsNet. Згорткові шари використовуються для отримання початкових карт функцій. Потім карти функцій завантажуються в модель CapsNet для виконання остаточної класифікації. Модель CNN використовує архітектуру CNN-CapsNet, сформовану на основі чотирьох згорткових шарів. Кожен згортковий рівень використовує 64, 64, 128 і 128 ядер відповідно. Це «3×3».

Для всіх шарів використовується функція активації «ReLU». Після другого шару згортки розташовується шар проміжного з'єднання. На даний момент розроблено дві окремі моделі. Одна використовується для виявлення випадків на основі двійкового таксономічного призначення «дозріла пшениця» або «недозріла пшениця». Друга модель виконує класифікацію на основі трьох категорій «дозріла пшениця», «недозріла пшениця» та «хвора/пошкоджена пшениця». Приклад класифікації показаний на рисунку 1.

При початковій швидкості навчання 0,001 модель узгоджується з оптимізатором Адама. Функція зменшення ваги використовується для чисельної стабільності. Рекомендується встановити початкове значення 0,5. Обидві моделі навчалися протягом 100 епох. Пакет запуску містить 32 значення, а тестовий пакет містить одне значення. Зображення «64×64» використовуються для навчання та тестування мережі. Навчання завершується методом ранньої зупинки, щоб уникнути наслідків перенавчання.



Рисунок 1 - Приклад 3-класового розпізнавання пшениці

Для перевірки ефективності та адекватності моделі було виконано порівняння даної моделі з іншими по критерію ефективності. Для порівняння використовувались моделі з подібною архітектурою, які поєднують архітектури CNN та CapsNet: FFCDNN [2], grain-CNN [3], CapsNet [4], Faster-RCNN [5]. Для перевірки моделі були використані зображення з бази Wheat-Ears-Detection-Dataset [6]. Запропонована модель показала хороші результати точності, проте дані результати потребують більш ретельної перевірки на більшій кількості зображень з різних наборів даних. Також результати не є стійкими, оскільки в деяких випадках запропонована модель показує кращі результати, в деяких гірші. При цьому середні показники швидкодії моделі є гіршими за більшість моделей. Практичне використання даної моделі потребує комплексної оцінки по інтегральному критерію та врахування більшої кількості факторів.

Висновки. У роботі запропоновано модель нейронної мережі CNN-CapsNet із вдосконаленою архітектурою, яка поєднує архітектури нейромереж CNN та CapsNet, відповідно, і дозволяє використати переваги обох архітектур.

Дану модель можна використовувати для прогнозування якості урожаю пшениці та незалежної оцінки класу пшениці, що є важливим зокрема для планування бізнес-процесів сільськогосподарських підприємств та галузі точного землеробства в цілому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] B. Knysh, Y. Kulyk. Development of an image segmentation model based on a convolutional neural network. *Eastern-European Journal of Enterprise Technologies*, 2 (2 (110)), (2021), pp. 6–15. doi: <https://doi.org/10.15587/1729-4061.2021.228644>.
- [2] Y. Shi, L. Han, P. González-Moreno, D. Dancey, W. Huang, Z. et al. A fast Fourier convolutional deep neural network for accurate and explainable discrimination of wheat yellow rust and nitrogen deficiency from Sentinel-2 time series data. *Front Plant Sci.*, 2023 Oct 4;14:1250844. doi: 10.3389/fpls.2023.1250844. URL: <https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2023.1250844/full>.
- [3] S. Lingwal, K. K. Bhatia, M. S. Tomer. Image-based wheat grain classification using convolutional neural network. *Multimedia Tools and Applications*, 80, 2021, pp. 35441 - 35465.
- [4] Hinton G.E., Krizhevsky A., Wang S.D. Transforming autoencoders. *International Conference on Artificial Neural Networks*. Berlin, Heidelberg: Springer; (2011): 44-51.
- [5] S. Madec, X. Jin, H. Lu, B. De Solan, S. Liu, F. Duyme et al. Ear density estimation from high resolution RGB imagery using deep learning technique. *Agric. For. Meteorol.* 264, 2019, pp. 225–234, doi:10.1016/j.agrformet.2018.10.013.
- [6] Simon Madec, Frederic Baret, Benoit de Solan, Shouyang Liu. The Wheat-Ears-Detection-Dataset (WEDD) Wheat-Ears-Detection-Dataset Dataset from the Ear density estimation from high resolution RGB imagery, 2024. URL: <https://github.com/simonMadec/Wheat-Ears-Detection-Dataset>.