

УДК 004.93

## МЕТОДИКА ГЕНЕРАЦІЇ ЗОБРАЖЕНЬ ХВОРИХ ЛИСТКІВ РОСЛИН НА ОСНОВІ ГЕНЕРАТИВНО-ЗМАГАЛЬНОЇ НЕЙРОННОЇ МЕРЕЖІ

Усатенко Д.С.

доцент каф. ІС, к.т.н. Галчонков О.М.

Державний університет «Одеська політехніка», УКРАЇНА

**АНОТАЦІЯ.** Virшена задача генерації зображень хворих листків рослин для збільшення набору даних в задачах класифікації захворювань рослин. В роботі запропонована методика генерації зображень з використанням генеративно-змагальної нейронної мережі *DCGAN*. Комп'ютерна реалізація виконана засобами пакетів *Tensorflow*, *Keras*.

**Вступ.** Однією з розповсюджених задач, які вирішуються за допомогою методів машинного навчання є генерація даних (зображень, звуків і т.д.) з певним розподіленням. Для навчання класифікаторів потрібно мати велику різноманітну вибірку вхідних зображень. Розробка методики на основі генеративно-змагальних нейронних мереж для генерації зображень хворих листків рослин є актуальною, тому що розробивши оптимальну методологію для генерації ми зможемо генерувати унікальні набори зображень для їх використання в задачах виявлення хвороб рослин. Це дозволить зменшити час на збір набору даних, так як для навчання генератора потрібно набагато менший набір. Також це дозволить різноманітнити вже готовий набір.

**Мета роботи** – розробити та дослідити методику для генерації зображень хворих листків рослин, з використанням генеративно-змагальних мереж з архітектурою *DCGAN*.

**Основна частина роботи.** Сучасні методи машинного навчання дозволяють вирішувати задачу генерації зображень хворих листків рослин з використанням різних архітектур нейронних мереж. В даній роботі для генерації зображень була використана генеративно-змагальна мережа з архітектурою *DCGAN*. Архітектура мережі *DCGAN* відрізняється від класичних генеративно-змагальних мереж тим, що вона використовує згорткові шари та згорткові транспонуючі шари.

Методика генерації зображень хворих листків рослин складається з наступних етапів:

1. Завантажити вхідні зображення в список.
2. Виконати попередню обробку зображень.
3. Спроекувати топологію моделі генератора (табл. 1).
4. В якості функції активації для прихованих шарів генератора використовується функція *LeakyReLU* (*ReLU* с витоком). В якості функції активації вихідного шару використовується *tanh*.
5. Спроекувати топології моделі дискримінатора (табл. 1).
6. Навчати моделі впродовж 1000 епох.
7. Оцінити функції втрат генератора ( $L_G$ ) и дискримінатора ( $L_D$ ) [1].
8. Порівняння втрат здійснюється методом бінарної крос ентропії (*BinaryCrossEntropy* в пакеті *Keras*).
9. Згенерувати зображення (рис. 1).

Для навчання генеративно-змагальної мережі *DCGAN* був використаний набір даних *New Plant Diseases Dataset* [2]. В ньому представлені зображення хворих листків рослин, які розподілені по видам рослин. Для роботи були вибрані 7000 зображень хворих лисків полуниці.

В результаті навчання генеративно-змагальної мережі *DCGAN* з запропонованою архітектурою, були отримані наступні метрики якості:  $L_G = 5, 39$ ,  $L_D = 0, 324$ .

При генерації зображень хворих листків полуниці з застосуванням запропонованої в роботі методики, дискримінатор менше пригнічується генератором, однак згенеровані зображення схожі на вихідні. Подальші дослідження будуть направлені на покращення архітектури генератора.

Таблиця 1 – Опис архітектури генератора та дискримінатора

Вихідний розмір	Генератор (шари)	Вихідний розмір	Дискримінатор (шари)
4096	<i>Dense</i>	128×128×64	<i>Conv2D</i>
4096	<i>LeakyReLU</i>	128×128×64	<i>LeakyReLU</i>
4×4×256	<i>Reshape</i>	64×64×128	<i>Conv2D</i>
8×8×128	<i>Conv2DTranspose</i>	64×64×128	<i>LeakyReLU</i>
8×8×128	<i>LeakyReLU</i>	64×64×128	<i>Dropout</i>
8×8×128	<i>BatchNormalization</i>	32×32×128	<i>Conv2D</i>
16×16×128	<i>Conv2DTranspose</i>	32×32×128	<i>LeakyReLU</i>
16×16×128	<i>LeakyReLU</i>	32×32×128	<i>Dropout</i>
16×16×128	<i>BatchNormalization</i>	16×16×256	<i>Conv2D</i>
32×32×128	<i>Conv2DTranspose</i>	16×16×256	<i>LeakyReLU</i>
32×32×128	<i>LeakyReLU</i>	16×16×256	<i>Dropout</i>
32×32×128	<i>BatchNormalization</i>	8×8×256	<i>Conv2D</i>
64×64×128	<i>Conv2DTranspose</i>	8×8×256	<i>LeakyReLU</i>
64×64×128	<i>LeakyReLU</i>	8×8×256	<i>Dropout</i>
64×64×128	<i>BatchNormalization</i>	4×4×512	<i>Conv2D</i>
128×128×128	<i>Conv2DTranspose</i>	4×4×512	<i>LeakyReLU</i>
128×128×128	<i>LeakyReLU</i>	4×4×512	<i>Dropout</i>
128×128×128	<i>BatchNormalization</i>	8192	<i>Flatten</i>
128×128×3	<i>Conv2D</i>	1	<i>Dense</i>

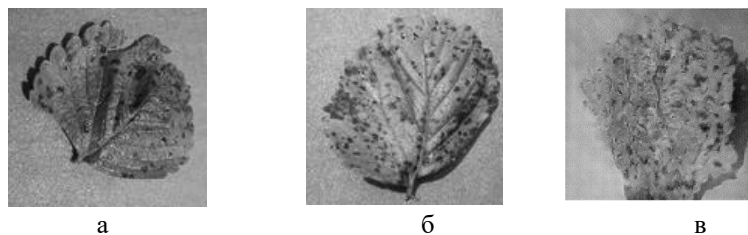


Рис. 1 – Приклади хворих листків полуниці: а, б – вихідні зображення з набору даних *New Plant Diseases Dataset*; в – згенероване зображення по запропонованій методиці

**Висновки.** Результатом роботи є методика генерації зображень хворих листків полуниці, яка реалізована з використанням генеративно-змагальної мережі з архітектурою *DCGAN*. Проведено комп’ютерне дослідження працездатності запропонованої методики на базі пакетів *Keras*, *Tensorflow*. Запропоновану методику генерації зображень хворих листків рослин можна застосовувати для генерації унікальних наборів зображень для їх використання в задачах виявлення хвороб рослин.

#### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. GANs in Action: Deep Learning with Generative Adversarial Networks / [Vladimir Bok, Jakub Langr] – книга. – [Печатное издание]. – 120 с.
2. New Plant Diseases Dataset | Kaggle – стаття. – [Електронний ресурс]. – Режим доступу: <https://www.kaggle.com/vipooool/new-plant-diseases-dataset>