УДК 519.6

DOI: 10.20535/2523-4455.mea.262933

Метод трансформації класифікаційних міток зображення в сегментаційні маски

Сидорський В. С., ORCID 0000-0001-9697-7403
Respeecher respeecher.com
Київ, Україна

***Анотація*—Задача бінарної або багато класової сегментації зображення постає в багатьох областях промисловості, медицини, сільського господарства та інших прикладних областях діяльності людини. На даний момент існує велика кількість алгоритмів машинного навчання, які можуть бути використані для цього, проте найбільш ефективним підходом на сьогодні є згорткові нейронні мережі. Водночас нейронні мережі потребують більших тренувальних вибірок в порівнянні з класичними алгоритми машинного навчання. Водночас накопичення тренувальної вибірки потребує великої кількості людських і фінансових ресурсів, а також часу. Отже постає задача дослідити методи зменшення кількості ресурсів для накопичення тренувального набору даних.**

Попередні дослідження в цій сфері були присвячені методам часткового навчання або ж навчання без вчителя. Проте всі вони потребують накопичення певної тренувальної вибірки - масок для зображень. В даному дослідженні буде розглянуто інший підхід - трансформація класифікаційної розмітки (міток класів) в сегментаційну (маски зображень). Важливо зазначити, що подібні підходи достатньо нові та малодосліджені. Запропонований метод не потребує накопичення масок зображень, а значить і великої кількості ресурсів для їх збору. Розглянутий метод грунтується на алгоритмі GradCam, який дає можливість отримати активаційну маску зображення, маючи лише мітку класу. Проте для подальшого використання отриманої маски, необхідно застосувати ряд перетворень для покращення якості сегментації. Для підтвердження ефективності запропонованого методу були проведені експерименти на задачі сегментації дефектів на листах сталі — Kaggle-Severstal: Steel Defect Detection. Експериментальні результати показали адекватність запропонованого підходу - було отримано маски, якість яких достатня для локалізації дефектів. Результати були оцінені за метрикою Dice: класична схема тренування – 0.621, запропонований підхід – 0.465. Проте запропонований метод потребує значно менше ресурсів в порівнянні з підходам класичного навчання та багатьма підходами часткового навчання.

Ключові слова — сегментація; нейронні мережі; часткове навчання.

I. Вступ

Задача бінарної або багато класової сегментації зображення постає в багатьох областях промисловості [1], медицини [2], сільського господарства [3] та інших прикладних областях діяльності людини. На даний момент існує велика кількість алгоритмів машинного навчання, які можуть бути використані для цього, проте найбільш ефективним підходом на сьогодні є згорткові нейронні мережі, наприклад мережі типу Unet, Unet++, FPN [4]–[7]. Водночас нейронні мережі потребують більших тренувальних вибірок в порівнянні з класичними алгоритми машинного навчання [8] — мінімальна величина вибірки може сягати кількох тисяч зображень . Більш того за допомогою збільшення тренувального набору даних можливо значно покращити якість моделі (навіть використовуючи менш якісну цільову змінну або ж не використовуючи її взагалі - підходи навчання без вчителя [9]). Отже постає задача дослідити методи збільшення тренувального набору даних для задачі сегментації.

Існують кілька підходів часткового навчання або навчання без вчителя, які є перспективними для вирішення даного класу задач. Двоетапне навчання [10], при якому на першому етапі відбувається навчання без вчителя, наприклад відновлення замаскованого тексту. Другий етап тренування проводиться вже на цільовій задачі. Під час псевдо лейблінгу [11] навчання алгоритма відбувається одразу на цільовій задачі, а потім його використовують для створення додаткових масок (або міток класів) для збільшення навчального датасету. Таку операцію можна повторювати ітеративно, аж поки не буде досягнуто плато по якості. Притренування на більшому наборі даних [12] — класичний метод для задач комп'ютерного зору. Спочатку оптимізують мережу на великому наборі даних, а потім вже на цільовій задачі.

Всі вищевказані підходи допомагають підвищити якість за рахунок використання додаткових даних або ж отримати задовільну якість при використанні малого початкового набору даних. Проте для певних задач, наприклад сегментації, собівартість навіть однієї розміченої картинки може бути дуже високою. Це зумовлено наступними факторами:

- Сегментація потребує залучення експертів в певній області для виконання достатньо монотонної роботи - розмітки.
- Сегментація потребує попіксельної розмітки, а в задачах медичного або промислового домену використовують зображення дуже високої роздільної здатності (більше

DOI: 10.20535/2523-4455.mea.262933

10 тисяч). Отже ручна анотація навіть одного зображення може забрати багато часу.

Отже важливо також розвивати методи, які дозволяють отримати маски для задачі сегментації взагалі без залучення ручної розмітки або ж трансформувати достатньо дешеву розмітку, наприклад класифікаційну, в сегментаційну. Метою даної роботи є створення та застосування другого підходу (трансформація міток класів у маски зображення) для вирішення задачі накопичення тренувального набору даних для задачі сегментації.

II. Загальний опис алгоритму GradCam

Для досягнення мети дослідження пропонується використати алгоритм GradCam [13] для отримання мап активацій класифікаційної мережі. Після обробки цих мап, їх можна використати як сегментаційні маски зображень. Далі пропонується натренувати сегментаційну мережу на отриманих масках або ж використовувати отримані маски, як результат сегментації.

Алгоритм GradCam зазвичай використовують для інтерпретації результатів роботи мережі для задачі класифікації. Розглянемо детально алгоритм GradCam на прикладі згорткової класифікаційної мережі.

Більшість згорткових мереж мають енкодер, а потім повнозв'язний шар. Розглянемо енкодер: зазвичай згорткові шари знижують роздільну здатність зображення (так в мережі VGG [14] або AlexNet [15] після останнього шару маємо зображення 14×14), проте збільшують кількість каналів (зазвичай після останнього шару кількість каналів варіюється від 768 до кількох тисяч). Після останньої згортки отримуємо мапу активацій A_{ij}^k, де i, j – висота та ширина зображення, а k – кількість каналів. Після чого застосовують усереднення активацій по висоті та ширині та отримують один вектор розмірності k. Після чого цей вектор класифікують повнозв'язним шаром на N класів. Отже фінальний вихід мережі — це вектор розмірності N, який подається в нормовану експоненційну функцію [16] (шар "Softmax"), щоб отримати розподіл ймовірностей. Розглянемо вектор y – вектор перед шаром "Softmax", кожну координату вектора y^c, де c – відповідний клас, можна розглянути, як диференційне перетворення $y^c\left(A_{ij}^k\right)$, тоді розглянемо

$$a_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k},$$

де Z – площа мапи активації.

Отже можемо розглядати a_k^c, як коефіцієнт впливу кожного каналу активації для прийняття рішення для класу c. Тоді

$$L_{GradCam}^C = \mathrm{Re}\,LU\left(a_k^c A^k\right), \qquad (1)$$

де $\mathrm{Re}\,LU$ – випрямлений лінійний вузол [17].

Таким чином використовуючи $\mathrm{Re}\,LU$ ми нехтуємо тими канали, які зменшують значення активації цільового класу (що означає, що вони відповідають за інші класи). Тоді отримавши $L_{GradCam}^c$ ми збільшуємо його роздільну здатність до роздільної здатності вхідного зображення і отримуємо мапу активацій від оригінального зображення, яка вказує на регіони, які найбільш впливають на прогноз даного класу.

Покращенням даного алгоритму є алгоритм виділення не просто області, а конкретних пікселів, які впливають на прийняття рішення (Guided GradCam). Розглянемо механізм Guided Back Propogation [18]: Нехай $x=\left\{x_{ij}^k\right\}$, i, j – відповідають за ширину та висоту, а k – канали зображення. Тоді позначимо $f(x)=A=\left\{A_{ij}^k\right\}$, частина мережі, яка повертає мапу активацій. Можемо розглянути такі градієнти

$$R_{ij}^k = \frac{\partial f}{\partial x_{ij}^k}, \quad R_{ij} = \frac{1}{K}\sum_k R_{ij}^k,$$

де K – загальна кількість каналів вхідного зображення.

Тоді розглянемо

$$RG_{ij}^k = \mathrm{Re}\,LU\left(R_{ij}^K\right)\cdot I\left(f(x)>0\right),$$

$$RG = \left\{RG_{ij}\right\}, \quad RG_{ij} = \frac{1}{K}\sum_k RG_{ij}^k.$$

Таким чином отримуємо мапу активацій, такої самої роздільної здатності, що й вхідне зображення. Застосувавши $\mathrm{Re}\,LU$ на фінальній активації $I\left(f(x)>0\right)$, відкинемо регіони, які мають від'ємну інтенсивність шару активацій, що вказує на те що мережа має їх більше ігнорувати. А застосувавши $\mathrm{Re}\,LU$ до градієнтів ($\mathrm{Re}\,LU\left(R_{ij}^K\right)$), відкинемо регіони, які мають зменшувати вплив на фінальний прогноз мережі. Тоді застосувавши формулу (1), можемо отримати модифікацію: нехай $Lupscaled_{GradCam}^c - L_{GradCam}^c$ збільшений до роздільної здатності початкового зображення, тоді

$$L_{Guided-GradCam}^c = Lupscaled_{GradCam}^c \cdot RG$$

На Рис. 1 зображено візуальні результати роботи алгоритму GradCam.

DOI: 10.20535/2523-4455.mea.262933

Рис. 1 Результати застосування GradCam [7]

Далі розглянемо застосування алгоритму GradCam для безпосередньої генерації сегментаційних масок.

III. Опис експериментальних даних і метрик

Для демонстрації запропонованого в роботі метода було обрано набір даних з ресурсу Kaggle — Severstal: Steel Defect Detection [19]. Основна задача цього змагання — віднайти листи сталі з дефектами та сегментувати зону дефекту.

Вибірка містить 7095 зображень з дефектами та 5473 зображення без дефектів, загалом — 12568 зображень. Усі зображення є чорно-білими знімками, роздільної здатності 1600 на 256. Всього розглядають чотири види дефектів (рис. 2-5). Розподіл по дефектам у вибірці: 897 дефектів першого класу; 247 дефектів другого класу; 5150 дефектів третього класу; 801 дефектів четвертого класу. Також важливо зазначити, що на одному зображенні можуть бути представлені дефекти декількох классів.

Результат прогнозу оцінюють за допомогою Індексу Соренса (Dice) — який підраховують для кожного зображення і для кожного класу дефекта окремо, а потім усереднюють по всім класам та всій тестовій вибірці. В разі порожнього прогнозу та порожньої маски метрика вважається одиницею. Отже фінальну метрику можемо записати так

$$M = \frac{\sum_{i}^{n}\sum_{j}^{4} Dice\left(x_{ij}, \hat{x}_{ij}\right)}{4n},$$

де $Dice$ – Індекс Соренса з урахуванням випадку з порожніми масками; n – кількість зображень; x_{ij} – маска i-го зображення j-го класу; $\hat{x}_{ij}$ – прогноз маски i-го зображення j-го класу.

Для оцінки результату пропонується використовувати тільки дефектні зображення, а для розбиття на тренувальні та тестові вибірки використовувати механізм перехресного затвердження [20] при розбитті на п’ять тренувальних підвибірок. Фінальна метрика обраховується на об'єднані всіх п’яти тренувальних підвибірок.

Розглянемо вибірку візуально (Рис. 2 – Рис. 5)

IV. Побудова класифікаційної мережі

Побудуємо згорткову мережу для класифікації дефектів зображень. Задача класифікації полягає у наступному: визначити, які дефекти присутні на кожному з зображень. Отже прогноз моделі є вектор ймовірностей $y_i = \left[y_{i1}, y_{i2}, y_{i3}, y_{i4}\right]$, де кожен $y_{i1/2/3/4}$ – відповідає ймовірності дефекта 1/2/3/4. Розглянемо процес оптимізації нейронної мережі більш детально.

Для оцінки класифікації будемо використовувати метрику ROC-AUC [21].

Модель складається з згорткового енкодера та шару класифікації, який в свою чергу складається з шару виключення (Dropout) [22], лінійного шару (Linear), шару активації “Elu” (Elu) [23], шару виключення (Dropout), лінійного шару (Linear) та шару активації “Сигмоїд” (Sigmoid) (рис. 6).

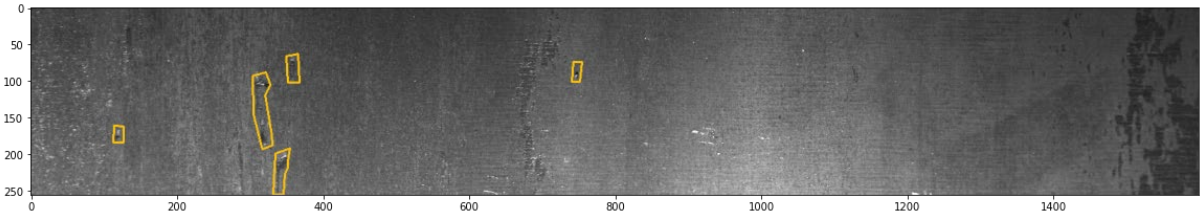

Рис. 2 Приклад дефекту 1-го типу [19]

DOI: 10.20535/2523-4455.mea.262933

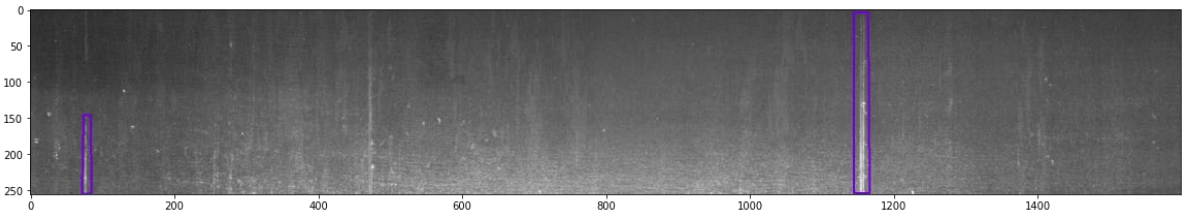

Рис. 3 Приклад дефекту 3-го типу [19]

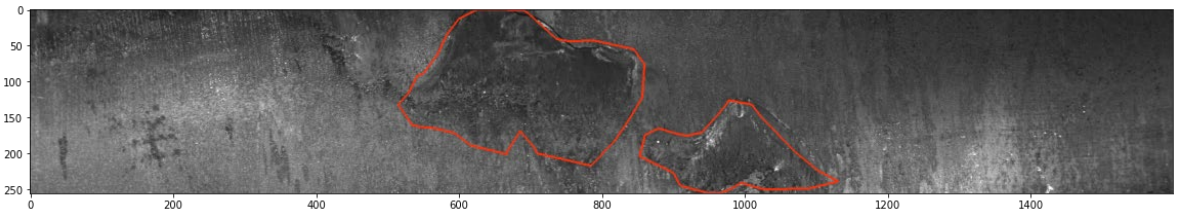

Рис. 4 Приклад дефекту 4-го типу [19]

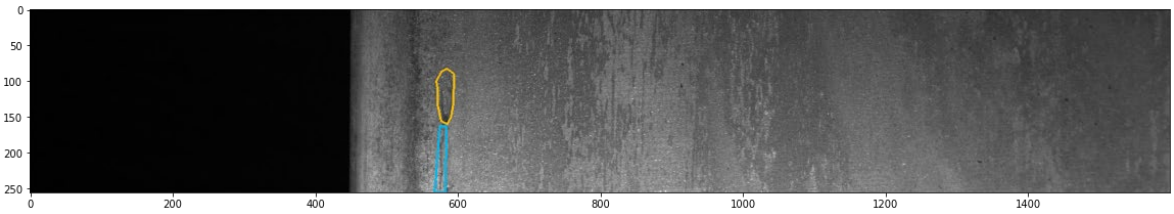

Рис. 5 Приклад дефекту 1-го типу (жовтий) й 2-го типу (блакитний) [19]

Цільова функція врат — бінарна перехресна ентропія:

$$Loos = \frac{\sum_{i}^{n}\sum_{j}^{4} BCE\left(y_{ij}, \hat{y}_{ij}\right)}{4n},$$

де y_{ij} – значення класу j для зображення i (1 — в разі присутності дефекту, 0 — в разі відсутності дефекту); $\hat{y}_{ij}$ – прогнозована ймовірність класу j для зображення i; n – величина підвибірки, яка мала значення 64.

Для оптимізації параметрів мережі було обрано алгоритм Adam [24] з коефіцієнтом швидкості навчання 0.0001 для енкодер та 0.001 для шару класифікації. Різні коефіцієнти швидкості навчання були обрані, бо енкодер був ініціалізований вагами з ImageNet, а отже навіть без тренування міг створювати адекватне стиснуте представлення зображення, в той час як шар класифікації був ініціалізований з нуля та потребував більш швидкої оптимізації. Також використання меншого коефіцієнта швидкості навчання в разі використання механізму transfer learning дозволяє зменшити ефект ‘забування’ - дуже сильної адаптації під нові дані, що зменшує узагальнюючу здатність моделі.

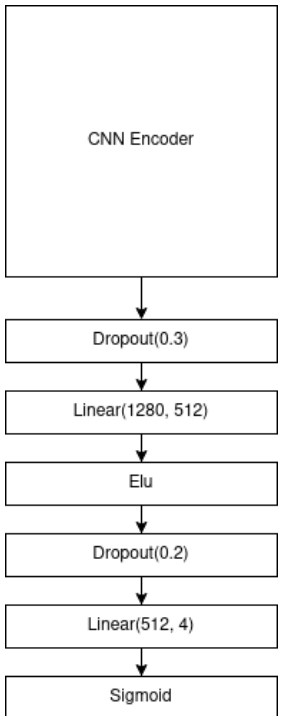

Рис. 6 Архітектура класифікаційної мережі

DOI: 10.20535/2523-4455.mea.262933

ТАБЛИЦЯ 1 РЕЗУЛЬТАТИ КВАЛІФІКАЦІЇ НА ОБ'ЄДНАНІЙ ВИБІРЦІ ПЕРЕХРЕСНОГО ЗАТВЕРДЖЕННЯ ЗА МЕТРИКОЮ ROC-AUC

Модель	Метрика першого дефекта	Метрика другого дефекта	Метрика третього дефекта	Метрика четвертого дефекта
EfficentNet-b0 + Horizontal Flip	0.9938	0.9917	0.9886	0.9974
EfficentNet-b3 + Horizontal Flip + Vertical Flip	0.9970	0.9932	0.9932	0.9962
EfficentNet-b3 + Horizontal Flip	0.9970	0.995	0.9921	0.9960

Для зменшення коефіцієнта швидкості навчання був обраний механізм зменшення коефіцієнта швидкості навчання при досягненні плато функції втрат на валідаційний вибірці [25]. В разі якщо функція втрат не зменшується на валідаційний вибірці більше 3 епох, тоді коефіцієнт швидкості навчання зменшується в 2 рази.

В якості енкодера було обрано мережі сімейства EfficentNet [26].

Експериментальні результати для кількох моделей з різними типами аугментації наведені в Таблиця 1.

Як видно з таблиці, експеримент з застосуванням енкодера EfficentNet-b3 та аугментації Horizontal Flip показує найкращі результати.

V. ЗАСТОСУВАННЯ GRADCAM ДЛЯ ОТРИМАННЯ МАСОК КЛАСІВ ДЕФЕКТІВ

Наступним кроком є отримання сегментаційних масок за допомогою алгоритму GradCam [13] з класифікаційних моделей, які були описані в попередньому розділі. Перш за все необхідно визначити шар моделі, який буде використовуватися для отримання мапи активацій. Розглянемо останні шари (Рис. 7) моделі EfficentNet-b0 (для моделі EfficentNet-b3 вони такі самі, проте з більшою кількістю каналів)

Отже останній шар перед пулінгом — шар активації “SiLU” [27], саме його виходи пропонується використати, як мапу активацій для алгоритму GradCam. Приклад мапи активацій для дефекту 4-го класу наведено на Рис. 10.

```
(conv_head): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=False)
(bn2): BatchNorm2d(1280, eps=0.001, momentum=0.1, affine=True, track_running_stats=True)
(act2): SiLU(inplace=True)
(global_pool): SelectAdaptivePool2d (pool_type=avg, flatten=Flatten(start_dim=1, end_dim=-1))
```

Рис. 7 Архітектура останніх шарів EfficentNet-b0

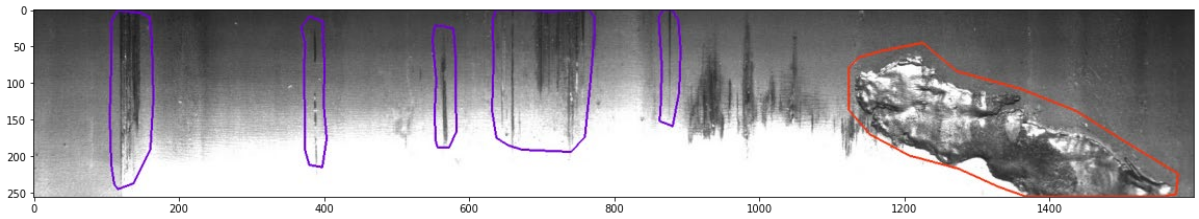

Рис. 8 Приклад дефекту 4-го класу (червоний) [19]

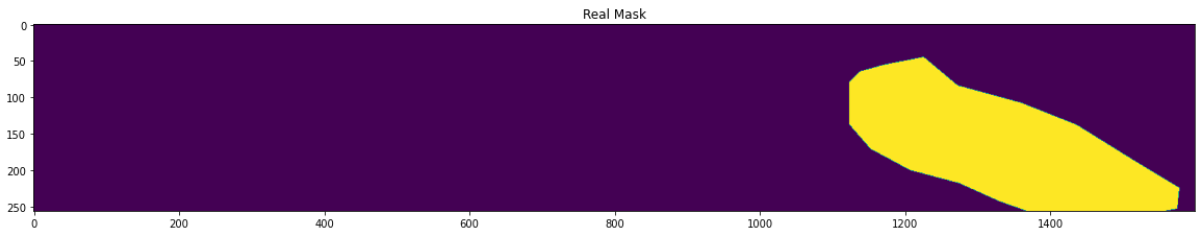

Рис. 9 Бінарна маска дефекту 4-го класу [19]

DOI: 10.20535/2523-4455.mea.262933

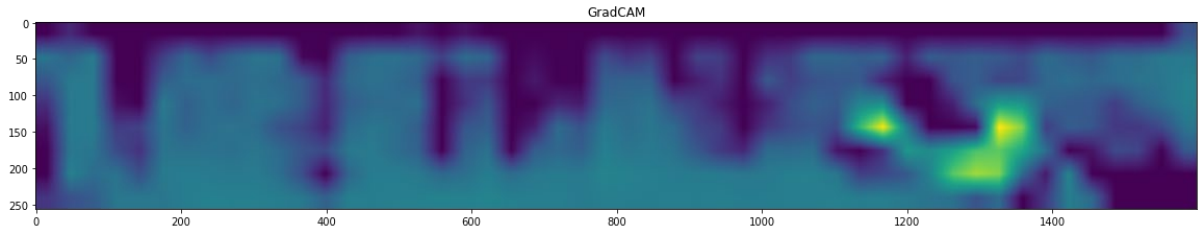

Рис. 10 GradCam мапа дефекту 4-го класу

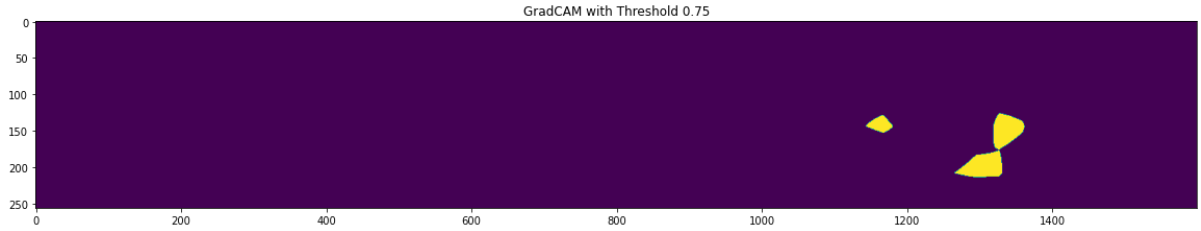

Рис. 11 GradCam бінаризована (поріг 0.75) мапа дефекту 4-го класу

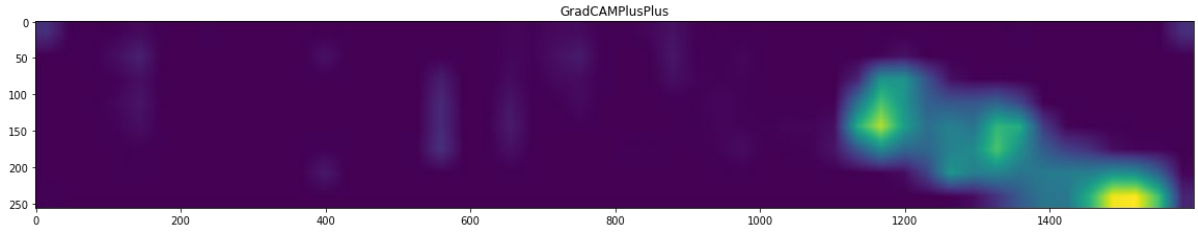

Рис. 12 GradCam мапа дефекту 4-го класу

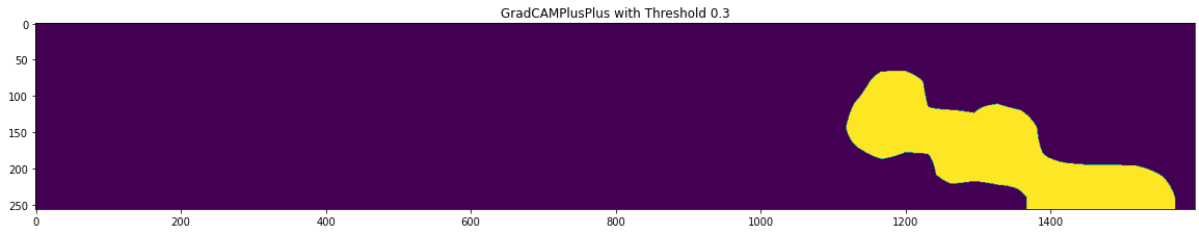

Рис. 13 GradCam бінаризована (поріг 0.3) мапа дефекту 4-го класу

Важливо зазначити, що дана реалізація алгоритму GradCam дає нормовані значення в межах [0,1]. Якщо проаналізувати мапу 4-го класу (Рис. 10), то бачимо, що найяскравіші області відповідають саме зоні дефекту, проте локалізація всього дефекту достатньо погана, також важливо помітити, що вся мапа достатньо яскрава. Це може бути спричинено аномальною яскравістю знизу зображення, а також наявністю інших дефектів.

Для підвищення якості масок, пропонується розглянути ці ж самі зображення проте використавши наступні модифікації:

- замість звичайного алгоритму GradCam, використаємо покращення цього алгоритму, яке використовує похідні другого порядку GradCamPlusPlus [28];
- використаємо усереднення вагів з трьох кращих епох за валадіційною функцією втрат [29];
- застосуємо згладжування за допомогою аугментацій [30] — отримуємо результати алгоритму GradCam на оригінальному зображенні, горизонтальному відзеркаленні, а також на зображенні домноженому на [1.0, 1.1, 0.9] (набір коефіцієнтів в околі одиниці), після чого усереднемо ці результати.

DOI: 10.20535/2523-4455.mea.262933

ТАБЛИЦЯ 2 РЕЗУЛЬТАТИ АЛГОРИТМУ GRADCAM НА ОБ'ЄДНАНІЙ ВИБІРЦІ ПЕРЕХРЕСНОГО ЗАТВЕРДЖЕННЯ

Назва експеременту	**Dice**
GradCaM + EfficentNet-b0	0.382956
GradCamPlusPlus + EfficentNet-b0	0.440810
GradCamPlusPlus + SWA + Aug Smooth + EfficentNet-b0	0.452208
GradCamPlusPlus + SWA + Aug Smooth + EfficentNet-b3 + Horizontal Flip + Vertical Flip	0.379971
GradCamPlusPlus + SWA + Aug Smooth + EfficentNet-b3 + Horizontal Flip	0.386530
GradCamPlusPlus + SWA + Aug Smooth + EfficentNet-b0 + confidence=0.5; min_seg_len=0; threshold=0.3	0.465

ТАБЛИЦЯ 3 ПОРІВНЯННЯ РЕЗУЛЬТАТІВ АЛГОРИТМУ GRADCAM З КЛАСИЧНОЮ СЕГМЕНТАЦІЄЮ НА ОБ'ЄДНАНІЙ ВИБІРЦІ ПЕРЕХРЕСНОГО ЗАТВЕРДЖЕННЯ

Назва експерименту	Dice
GradCam + покращення	0.465
TernausNet + справжні маски	0.621
TernausNet + маски алгоритму GradCam	0.418

Як бачимо з Рис. 12 – Рис. 13, запропоновані модифікації значно покращили якість сегментації, особливо в розрізі зменшення помилки другого роду. Таким чином дефекти стали краще локалізовані.

Для покращення середнього значення метрики по всьому датасету були експериментально підібрані значення таких параметрів:

- confidence - поріг прийняття зображення за ймовірністю класифікаційної мережі;
- min_seg_len - мінімальна площа маски сегментації;
- threshold - поріг прийняття пікселя до маски.

Проведемо такі експерименти з більшим енкодером (EfficentNet-b3), результати наведені в Таблиця 2.

VI. ПОРІВНЯННЯ З МЕРЕЖЕЮ НАТРЕНОВАНОЮ НА СПРАВЖНІХ МАСКАХ

Для отримання остаточних результатів розглянемо три варіанти сегментаційної мережі (Табл. 3):

- сегментація за допомогою алгоритму GradCam [13] з використанням усіх запропонованих покращень;
- сегментаційна мережа TernausNet [5] на базі енкодера EfficentNet-b0 натренована на справжніх масках з використанням наступних евристик: min_seg_len=0; threshold=0.2;
- сегментаційна мережа TernausNet [5] на базі енкодера EfficentNet-b0 натренована на масках з алгоритму GradCam з використанням наступних евристик: min_seg_len=0; threshold=0.2

ВИСНОВКИ

В роботі запропонований новий підхід до отримання масок для тренування сегментаційних нейронних мереж на основі використання класифікаційних нейромереж**.** Порівнявши результати (табл. 2-3) можемо зробити висновок, що маски, отримані за допомогою алгоритму GradCam, наразі очікувано, поступаються в якості маскам, отриманим з класичного методу тренування сегментаційних мереж. Проте запропонований в роботі метод є унікальним в тому, що взагалі не потребує ручної розмітки маски, а лише мітку дефекту для тренування, що є набагато зручнішим при практичному застосуванні для вирішення подібних задач.

Аналізуючи метрики й візуальні результати запропонованого алгоритму для використаного набору даних, можна зробити висновок, що його якість є достатньою для такої задачі - запропонований алгоритм в змозі локалізувати дефект, на зображеннях листах сталі, а цього більш ніж достатньо для подальшого вилучення бракованого виробу (або вилучення саме дефектної частини). Водночас він не потребує великої кількості людських ресурсів і часу для отримання тренувальної сегментаційної розмітки.

Напрямками подальших досліджень є розвиток застосування запропонованого методу в задачах сегментації, які не потребують високоточного виділення об'єкту, а лише його локалізації на зображенні, а також підвищення значень метрик, що уможливить використання даного методу для високоточної сегментації.

ПЕРЕЛІК ПОСИЛАНЬ

[1] C. Gh.; Amza, G. Amza, and D. Popescu, “Image Segmentation for Industrial Quality Inspection,” *Fiabilitate şi Durabilitate*, no. 01.Supliment, pp. 126–132, 2012, **URL:** https://www.utgjiu.ro/rev_mec/mecanica/pdf/2012-01.Supliment/21_Catalin%20Amza,%20Gheorghe%20Amza,%20Diana%20Popescu.pdf.

[2] R. Azad, N. Khosravi, M. Dehghanmanshadi, J. Cohen-Adad, and D. Merhof, “Medical Image Segmentation on MRI Images with Missing Modalities: A Review,” Mar. 2022, **DOI:** 10.48550/arxiv.2203.06217.

[3] K. Prakash, P. Saravanamoorthi, R. Sathishkumar, and M. Parimala, “A Study of Image Processing in Agriculture,” *International Journal of Advanced Networking and Applications - IJANA*, vol. 9, no. 1, pp. 3311–3315, 2017, **URL:** https://www.ijana.in/v9-1.php#.

[4] W. Weng and X. Zhu, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *IEEE Access*, vol. 9, pp. 16591–16603, May 2015, **DOI:** 10.48550/arxiv.1505.04597. **DOI:** 10.1109/ACCESS.2021.3053408

DOI: 10.20535/2523-4455.mea.262933

[5] V. Iglovikov and A. Shvets, “TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation,” Jan. 2018, **DOI:** 10.48550/arxiv.1801.05746.

[6] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: A Nested U-Net Architecture for Medical Image Segmentation,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11045 LNCS, pp. 3–11, Jul. 2018, **DOI:** 10.48550/arxiv.1807.10165.

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” Dec. 2016, **DOI:** 10.48550/arxiv.1612.03144.

[8] L. E. Aik, T. W. Hong, and A. K. Junoh, “A New Formula to Determine the Optimal Dataset Size for Training Neural Networks,” *ARPN Journal of Engineering and Applied Sciences*, vol. 14, no. 1, pp. 52–61, Jan. 2019, **URL:** http://www.arpnjournals.org/jeas/research_papers/rp_2019/jeas_0119_7525.pdf.

[9] S. Dridi, *Unsupervised Learning - A Systematic Literature Review*. 2021, **URL:** https://www.researchgate.net/publication/357380639_Unsupervised_Learning_-_A_Systematic_Literature_Review.

[10] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha, “AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing,” Aug. 2021, **DOI:** 10.48550/arxiv.2108.05542.

[11] E. Arazo, Di. Ortego, P. Albert, N. E. O’Connor, and K. McGuinness, “Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning,” *Proceedings of the International Joint Conference on Neural Networks*, Aug. 2019, **DOI:** 10.48550/arxiv.1908.02983. **DOI:** 10.1109/IJCNN48605.2020.9207304

[12] Y. Tay *et al.*, “Are Pre-trained Convolutions Better than Pre-trained Transformers?,” *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 4349–4359, May 2021, **DOI:** 10.48550/arxiv.2105.03322.

[13] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization,” *Int J Comput Vis*, vol. 128, no. 2, pp. 336–359, Oct. 2016, **DOI:** 10.1007/s11263-019-01228-7.

[14] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Sep. 2014, **DOI:** 10.48550/arxiv.1409.1556.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, 2012, vol. 25, **URL:** https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[16] T. Pearce, A. Brintrup, and J. Zhu, “Understanding Softmax Confidence and Uncertainty,” Jun. 2021, **DOI:** 10.48550/arxiv.2106.04972.

[17] YU. P. Zaychenko, *Osnovy proektuvannya intelektual'nykh system. Navch. posibnyk. [Fundamentals of designing intelligent systems. Education manual.]*. Kyiv: Vydavnychyy dim «Slovo», 2004.

[18] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for Simplicity: The All Convolutional Net,” *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*, Dec. 2014, **DOI:** 10.48550/arxiv.1412.6806.

[19] “Severstal: Steel Defect Detection | Kaggle.” [Online]. Available: https://www.kaggle.com/c/severstal-steel-defect-detection.

[20] D. Berrar, “Cross-Validation,” in *Encyclopedia of Bioinformatics and Computational Biology*, Elsevier, 2019, pp. 542–545. **DOI:** 10.1016/B978-0-12-809633-8.20349-X.

[21] K. Namdar, M. A. Haider, and F. Khalvati, “A Modified AUC for Training Convolutional Neural Networks: Taking Confidence into Account,” *Front Artif Intell*, vol. 4, Jun. 2020, **DOI:** 10.3389/frai.2021.582928.

[22] A. Labach, H. Salehinejad, and S. Valaee, “Survey of Dropout Methods for Deep Neural Networks,” Apr. 2019, **DOI:** 10.48550/arxiv.1904.13310.

[23] D. A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),” *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, Nov. 2015, **DOI:** 10.48550/arxiv.1511.07289.

[24] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec. 2014, **DOI:** 10.48550/arxiv.1412.6980.

[25] K. Mukherjee, A. Khare, and A. Verma, “A Simple Dynamic Learning Rate Tuning Algorithm For Automated Training of DNNs,” Oct. 2019, **DOI:** 10.48550/arxiv.1910.11605.

[26] M. Tan and Q. v. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10691–10700, May 2019, **DOI:** 10.48550/arxiv.1905.11946.

[27] S. Elfwing, E. Uchibe, and K. Doya, “Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning,” *Neural Networks*, vol. 107, pp. 3–11, Feb. 2017, **PMID:** 29395652. **DOI:** 10.1016/j.neunet.2017.12.012.

[28] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks,” *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, vol. 2018-January, pp. 839–847, May 2018, **DOI:** 10.1109/WACV.2018.00097.

[29] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging Weights Leads to Wider Optima and Better Generalization," *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, vol. 2, pp. 876–885, Mar. 2018, **DOI:** 10.48550/arxiv.1803.05407.

[30] D. Shanmugam, D. Blalock, G. Balakrishnan, and J. Guttag, "Better Aggregation in Test-Time Augmentation," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1194–1203, Nov. 2020, **DOI:** 10.48550/arxiv.2011.11156. **DOI:** 10.1109/ICCV48922.2021.00125.

Надійшла до редакції 23 травня 2022 року
Прийнята до друку 25 серпня 2022 року

DOI: 10.20535/2523-4455.mea.262933

UDC 519.6

The Method of Transformation of Image Classification Labels into Segmentation Masks

V. S. Sydorskyi, ORCID 0000-0001-9697-7403
Respeecher respeecher.com
Kyiv, Ukraine

***Abstract*—Semantic image segmentation plays a crucial role in a wide range of industrial applications and has been receiving significant attention. Unfortunately, image segmentation tasks are notoriously difficult and different industries often require human experts. Convolutional neural networks (CNNs) have been successfully applied in many fields of image segmentation. But all of them still require a huge amount of hand-labeled data for training. A lot of research was conducted in the field of unsupervised and semi-supervised learning, which studies how to shrink the amount of training data at the same time preserving the quality of the model. But still another field of research - transformation of "cheap" (in terms of time, money and human resources) markup into "expensive" is novel. In this work a new approach of generating semantic segmentation masks, using only classification labels of the image, was proposed. Proposed method is based on the GradCam algorithm, which can produce image activation heatmap, using only class label. But GradCams' heatmaps are raw for final use, so additional techniques and transforms should be applied in order to get final usable masks. Experiments were conducted on the task of detecting defects on steel plates — Kaggle- Severstal: Steel Defect Detection. After that Dice metric was computed using a classical training approach and proposed method: classical approach - 0.621, proposed method - 0.465. Proposed approach requires much less human resources compared to the classical approach. Moreover, after visual inspection of results it is obvious that the proposed approach has successfully completed the task of defect localization.**

Keywords — segmentation; neural networks; semi-supervised.