



ПРИКЛАДНА МАТЕМАТИКА, КОМП'ЮТЕРНІ НАУКИ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

УДК 004.891

<https://doi.org/10.31713/vt3202513>

Климюк Ю. Є. [1; ORCID ID: 0000-0003-3672-8469],

к.т.н., доцент

¹Національний університет водного господарства та природокористування, м. Рівне

ГЕНЕРУВАННЯ ПОКРАЩЕНИХ ТРЕНУВАЛЬНИХ, ПЕРЕВІРОЧНИХ І ТЕСТОВИХ НАБОРІВ ДАНИХ ДЛЯ КЛАСИФІКАТОРА ЗОБРАЖЕНЬ ОЦІНОК

Генерування покращених тренувальних, перевірочних та тестових наборів даних для класифікатора зображень оцінок, призначеного для розпізнавання оцінок із скан-копій додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів, реалізовано шляхом розробки нової функції `generation_attachments()` та допоміжної функції `draw_image_text()` з використання бібліотеки `Pillow` замість `OpenCV` для формування текстових написів на зображеннях шрифтом «Times New Roman» українською мовою і курсивним накресленням тексту, та оновлення функції `generation_dataset_grade()`. У новій версії функції `generation_attachments()`, призначеної для генерування набору зображень додатків до свідоцтв із різними допустимими оцінками, на основі стандартного бланку додатку до свідоцтва про здобуття повної загальної середньої освіти розміру A5 `attachment.jpg` реалізовано формування набору додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів із різними допустимими оцінками у форматі `png`-зображень. Допоміжна функція `draw_image_text()` служить для формування текстових написів на заданому зображенні у заданій позиції шрифтом «Times New Roman» українською мовою курсивним накресленням тексту. В оновленій версії функції `generation_dataset_grade()`, призначеній для генерування тренувальних, перевірочних та тестових наборів даних зображень оцінок, на основі згенерованих `png`-зображень додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів із різними допустимими оцінками генеруються окремі `png`-зображення оцінок та випадковим чином розподіляються по папках `test`, `train`, `valid`. У результаті числового експерименту отримано покращені тренувальний, перевірочний та тестовий набори даних, які відповідно містять 6892, 3795 і 2063 зображень оцінок. Запропоновані нова версія функції `generation_attachments()`, допоміжна функція `draw_image_text()` та

оновлена версія функції `generation_dataset_grade()` дозволили реалізувати розробку покращеного класифікатора зображень оцінок для розпізнавання оцінок із скан-копій додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів. Точність покращеного класифікатора стосовно розпізнавання зображень оцінок на тестовому наборі даних склала 97,19%, що дозволяє на його основі розробити додаток для здійснення розрахунку середнього бала свідоцтва про здобуття повної загальної середньої освіти абітурієнта.

Ключові слова: генерування; набір даних; зображення; оцінка; класифікатор; свідоцтво; додаток; абітурієнт.

Вступ. Кожного року процес подання електронних заяв ускладнюється необхідністю обчислення середнього бала свідоцтва (атестата) про здобуття повної загальної середньої освіти абітурієнтами та його подальшою перевіркою працівниками приймальними комісіями [1; 2]. У [3] реалізовано розробку класифікатора зображень оцінок для розпізнавання оцінок з додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів реалізовано мовою програмування Python з використанням редактора VS Code та розширення Jupyter Notebook шляхом донавчання моделі ResNet50, попередньо навченої на наборі даних ImageNet, за допомогою бібліотеки PyTorch [4–7] на основі згенерованого набору даних, що містить 1881 тренувальних, 331 перевіірочних та 338 тестових зображень оцінок, а для обробки зображень використано бібліотеку OpenCV [8–11]. Доновчання моделі проведено для 50 епох, найкращою виявилась 33. Точність отриманої моделі стосовно розпізнавання зображень оцінок на тестовому наборі даних склала 93,79%.

Базова версія функції `generation_attachments()` реалізована для генерування на основі стандартного бланку додатку до свідоцтва про здобуття повної загальної середньої освіти розміру A5 `attachment.jpg` набору png-зображень додатків до свідоцтв із різними допустимими оцінками з використанням функції `putText()` та шрифту «Hershey Complex» бібліотеки OpenCV. Базова версія функції `generation_dataset_grade()` реалізована для генерування на основі отриманих png-зображень додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів із різними допустимими оцінками окремих png-зображень оцінок та їх випадкового розподілу по папках `test`, `train`, `valid`. Базовий код для генерування тренувальних, перевіірочних і тестових наборів даних зображень оцінок забезпечує формування списку можливих оцінок у додатках



до свідоцтв `list_grades`, створення папки `attachment` зі згенерованими зображеннями додатків до свідоцтв із різними допустимими оцінками та створення папки `dataset_grades` зі згенерованими наборами зображень можливих оцінок додатків до свідоцтв, розподілених по вкладених папках `test`, `train`, `valid`. Недоліком такого підходу є підтримка функцією `putText()` бібліотеки `OpenCV` лише одного шрифту «Hershey Complex» з підтримкою української мови зі звичайним накреслення тексту. Тому досить актуальною задачею є формування покращених тренувальних, перевірочних та тестових наборів даних для класифікатора зображень оцінок, призначеного для розпізнавання оцінок із скан-копій додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів, за рахунок використання більш поширених на практиці для підготовки додатків до свідоцтв шрифтів з підтримкою української мови, курсивного накреслення тексту тощо, наприклад «Times New Roman».

Постановка задачі. Формування покращених тренувальних, перевірочних та тестових наборів даних класифікатора зображень оцінок, призначеного для розпізнавання оцінок із скан-копій додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів, потребує:

- розробки нової функції `generation_attachments()`, призначеної для формування на основі стандартного бланку додатку до свідоцтва про здобуття повної загальної середньої освіти розміру A5 `attachment.jpg` набору додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів із різними допустимими оцінками у форматі `png`-зображень за допомогою бібліотеки `Pillow`;

- розробки допоміжної функції `draw_image_text()` для формування текстових написів на заданому зображенні у заданій позиції з використання шрифту «Times New Roman» з підтримкою української мови і курсивного накреслення тексту;

- оновлення функції `generation_dataset_grade()`, призначеної для формування на основі згенерованих `png`-зображень додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів із різними допустимими оцінками набору окремих зображень оцінок та їх розподілу випадковим чином по папках `test`, `train`, `valid`.

Алгоритм розв'язання. Для формування покращених тренувальних, перевірочних та тестових наборів даних класифікатора зображень оцінок, призначеного для розпізнавання оцінок з додатків до свідоцтв про здобуття повної загальної

середньої освіти абітурієнтів, реалізовано розробку нової функції `generation_attachments()` (лістинг 1) для формування на основі стандартного бланку додатку до свідоцтва про здобуття повної загальної середньої освіти розміру A5 `attachment.jpg` набору зображень додатків до свідоцтв із різними допустимими оцінками, допоміжної функції `draw_image_text()` для формування текстових написів на заданому зображенні у заданій позиції шрифтом «Times New Roman» українською мовою курсивним накресленням тексту (лістинг 2) за допомогою бібліотеки Pillow, оновлення функції `generation_dataset_grade()` (лістинг 3) для формування тренувальних, перевірочних та тестових наборів даних зображень оцінок.

Лістинг 1. Функція для формування набору зображень додатків

```
def generation_attachments(file_attachment="attachment.jpg", grades=["",  
"один",  
"два", "три", "чотири", "п'ять", "шість", "сім", "вісім", "дев'ять", "десять",  
"одинадцять", "дванадцять"]):  
    ...
```

Function of generating attachments to the certificate

Parameters

attachment: An image of the attachment form for the certificate
(default="attachment.jpg")

grades: The list of possible grades in the attachments to the certificates
(default=["", "один", "два", "три", "чотири", "п'ять", "шість", "сім",
"вісім", "дев'ять", "десять", "одинадцять", "дванадцять"])

Returns

folder: A folder with generated attachments to certificates

...

```
folder_files_attachments = file_attachment.split(".")[0]
```

```
if os.path.exists(folder_files_attachments):
```

```
    shutil.rmtree(folder_files_attachments)
```

```
os.makedirs(folder_files_attachments)
```

```
for shift_y in range(5):
```

```
    for shift_x in range(5):
```

```
        for index in range(len(grades)):
```

```
            image_file_attachment= Image.open(file_attachment)
```

```
            draw_image_file_attachment
```

=

```
ImageDraw.Draw(image_file_attachment)
```

```
    # Let's write the text you want to put on the image.
```

```
    height = 43
```

```
    line_count = [12, 12, 6]
```

```
    x = [568, 1363, 1363]
```

```
    y = [553, 78, 808]
```

```
    for k in range(3):
```

```
        for line in range(line_count[k]):
```

```
            draw_image_text(draw_image_file_attachment,
```

```
                            x[k] + shift_x, y[k] + shift_y + line * height, grades[index])
```



```
file_attachment_grades = "{}_{}_{}.png".format(
    folder_files_attachments, str(shift_y), str(shift_x), str(index))
image_file_attachment.save(os.path.join(folder_files_attachments,
    file_attachment_grades))
return folder_files_attachments
```

Лістинг 2. Допоміжна функція для формування текстових написів

```
def draw_image_text(draw_image_file_attachment, x, y, text):
    result = draw_image_file_attachment.text(
        (x, y), text, fill="black", font=ImageFont.truetype("timesi.ttf", 34)
    )
    return result
```

Лістинг 3. Функція для формування тренувальних, перевірочних та тестових наборів даних зображень оцінок

```
def generation_dataset_grades(folder_attachments="attachment", grades=["",
"один",
"два", "три", "чотири", "п'ять", "шість", "сім", "вісім", "дев'ять", "десять",
"одинадцять", "дванадцять"]):
    ...
```

The function of forming a set of attachments to the certificates of applicants for obtaining a complete general secondary education of images with different permissible grades

Parameters

folder_attachments: Folder with generated attachments to certificates (default="attachment")

grades: The list of possible grades in the attachments to the certificates (default=["", "один", "два", "три", "чотири", "п'ять", "шість", "сім", "вісім", "дев'ять", "десять", "одинадцять", "дванадцять"])

Returns

folder_dataset: Folder with generated certificate attachments

A folder with a generated set of certificate attachments with different valid ratings

...

```
folder_dataset_grades="dataset_grades"
if os.path.exists(folder_dataset_grades):
    shutil.rmtree(folder_dataset_grades)
os.makedirs(folder_dataset_grades)
folders = []
folders.append(os.path.join(folder_dataset_grades, 'train'))
folders.append(os.path.join(folder_dataset_grades, 'valid'))
folders.append(os.path.join(folder_dataset_grades, 'test'))
for folder in folders:
    os.makedirs(folder)
    for index in range(len(grades)):
        os.makedirs(os.path.join(folder, str(index)))
```

```
for shift_y in range(5):
    for shift_x in range(5):
        for index in range(len(grades)):
            name_attachment = "{}_{}_{}_{}".format(folder_attachments,
                str(shift_y), str(shift_x), str(index))
            file_attachment = "{}.png".format(name_attachment)
            image = cv2.imread(os.path.join(folder_attachments, file_attachment))
            height = 43
            length = 240
            x = [540, 1335, 1335]
            y = [555, 80, 810]
            line_count = [12, 12, 6]
            shift_line = 0
            for k in range(3):
                for line in range(line_count[k]):
                    tiles = image[y[k] + line * height : y[k] + (line + 1) * height,
                        x[k] : x[k] + length]
                    scale_down = 0.935
                    tiles = cv2.resize(tiles, None, fx=scale_down, fy=scale_down,
                        interpolation=cv2.INTER_LINEAR)
                    if random.randint(0, 1) == 0:
                        cv2.imwrite(os.path.join(random.choice(folders), str(index),
                            "grade_{}_{}.png".format(name_attachment,
                                str(shift_line + line))), tiles)
                    elif random.randint(0, 1) == 0:
                        cv2.imwrite(os.path.join(random.choice(folders[:2]), str(index),
                            "grade_{}_{}.png".format(name_attachment,
                                str(shift_line + line))), tiles)
                    else:
                        cv2.imwrite(os.path.join(folders[0], str(index),
                            "grade_{}_{}.png".format(name_attachment,
                                str(shift_line + line))), tiles)
                shift_line += line_count[k]
            cv2.waitKey(0)
            cv2.destroyAllWindows()
        return folder_dataset_grades
```

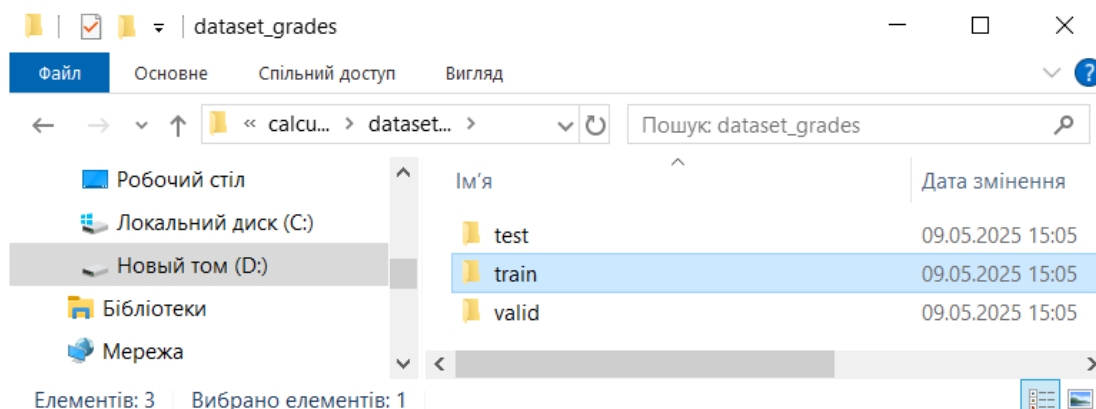
Результати числових розрахунків. Для тестування розробленої нової функції `generation_attachments()`, допоміжної функції `draw_image_text()`, оновленої функції `generation_dataset_grade()` та коду для генерування тренувальних, перевірочних і тестових наборів даних зображень оцінок проведено експеримент. У результаті отримано папку `attachment` зі згенерованими 425 зображеннями додатків до свідоцтв із різними допустимими оцінками та папку `dataset_grades` зі згенерованими тренувальним, перевірочним та тестовим наборами зображень можливих оцінок додатків до свідоцтв, випадково розподілених по вкладених папках `train`, `valid` і



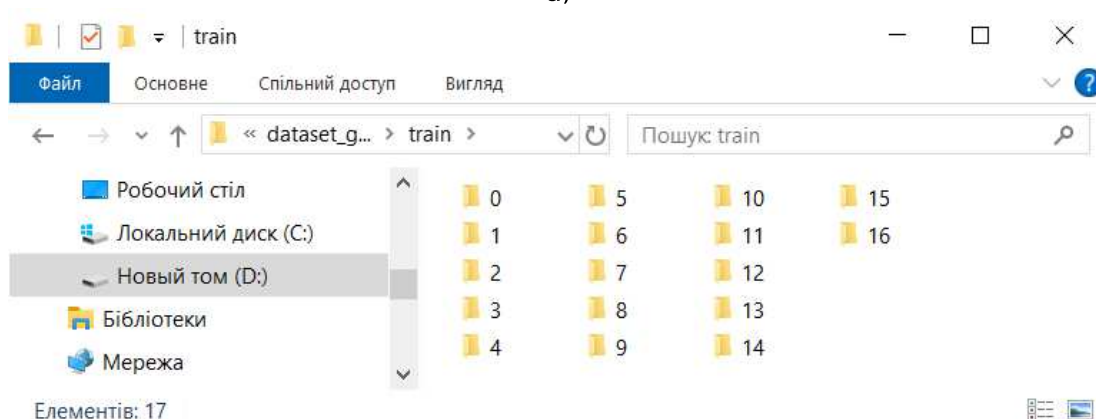
test (рис. 1, 2), які відповідно містять 6892, 3795 і 2063 зображень оцінок. Точність покращеного класифікатора стосовно розпізнавання зображень оцінок на тестовому наборі даних склала 97,19% (рис. 3).

```
calculation_set_values_pil.ipynb •
calculation_set_values_pil.ipynb > # The list of possible grades in the attachments to the certificates.
Generate + Code + Markdown | Run All | Restart | Clear All Outputs ... Python 3.11.5
...
A folder "attachment" was created containing the following files:
['attachment_0_0_0.png', 'attachment_0_0_1.png', 'attachment_0_0_10.png', 'attachment_0_0_11.png',
A total of 425 attachments to certificates were generated.
A folder "dataset_grades" with the following subfolders has been created:
['test', 'train', 'valid']
The folder "test" contains the following subfolders:
['0', '1', '10', '11', '12', '13', '14', '15', '16', '2', '3', '4', '5', '6', '7', '8', '9']
A total of 2063 grades images were generated.
The folder "train" contains the following subfolders:
['0', '1', '10', '11', '12', '13', '14', '15', '16', '2', '3', '4', '5', '6', '7', '8', '9']
A total of 6892 grades images were generated.
The folder "valid" contains the following subfolders:
['0', '1', '10', '11', '12', '13', '14', '15', '16', '2', '3', '4', '5', '6', '7', '8', '9']
A total of 3795 grades images were generated.
```

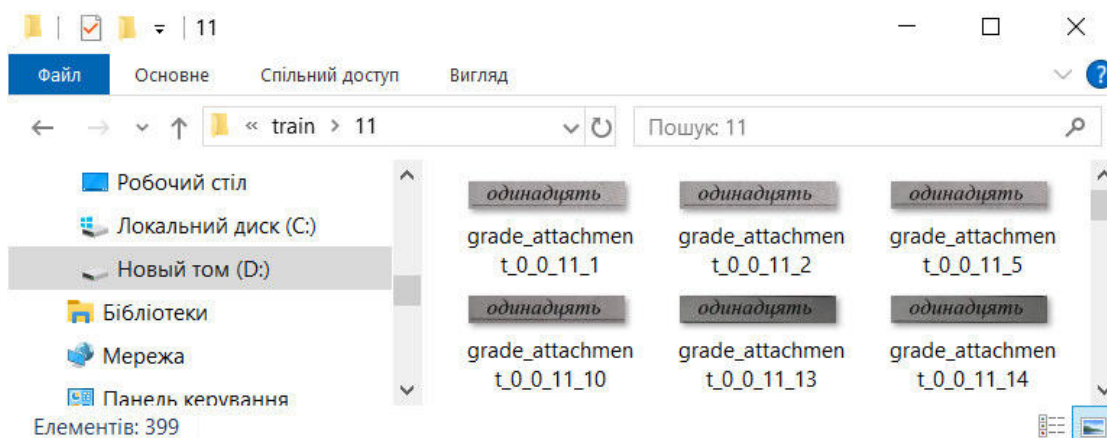
Рис. 1. Результат виконання коду для генерування зображень додатків до свідоцтв і формування тренувальних, перевірочних та тестових наборів даних зображень оцінок



а)



б)



в)

Рис. 2. Інформація щодо підготовки тренувальних, перевірочних та тестових наборів даних зображень оцінок: а) вміст папки dataset_grades; б) вміст папки train, вкладеної у папку dataset_grades; в) вміст папки 11, вкладеної у папку train

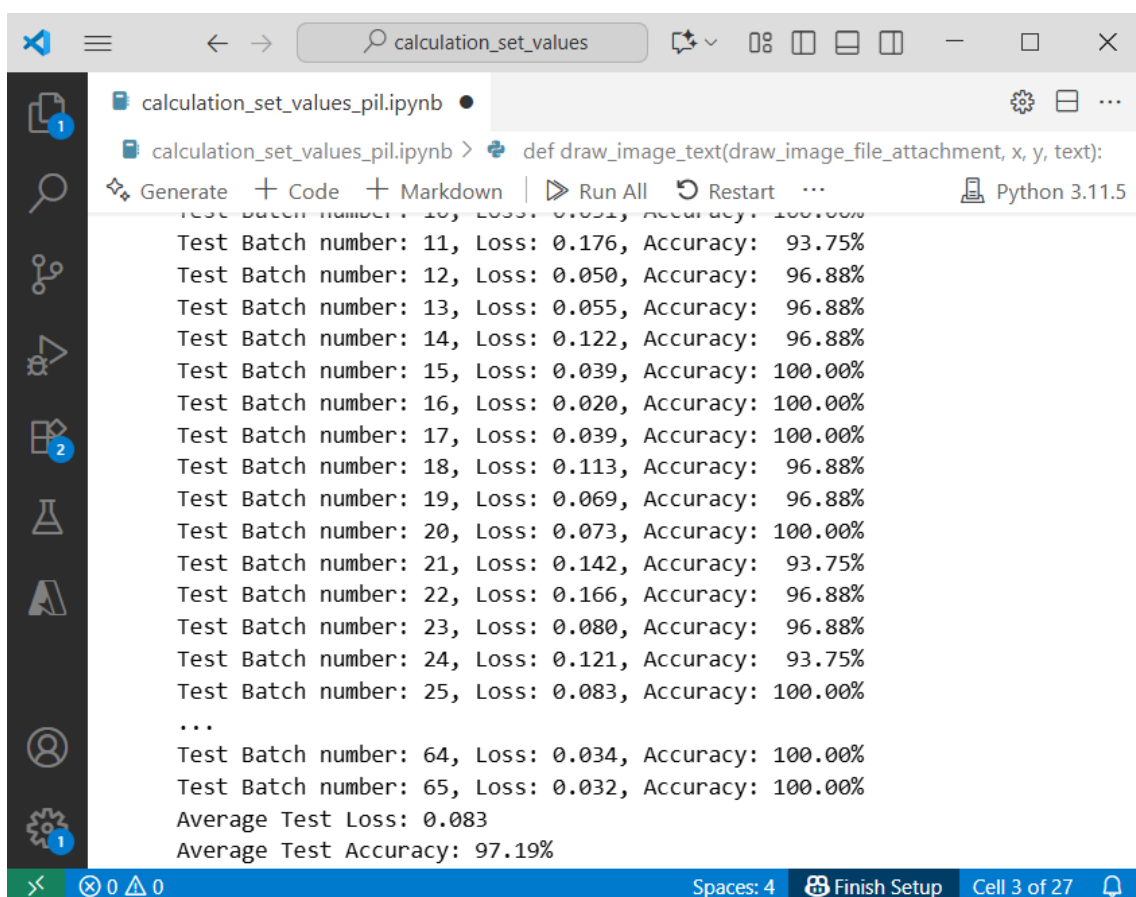


Рис. 3. Точність покращеного класифікатора стосовно розпізнавання зображень оцінок на тестовому наборі даних



Висновки. Генерування покращених тренувальних, перевірочних та тестових наборів даних для класифікатора зображень оцінок, призначеного для розпізнавання оцінок із скан-копій додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів, реалізовано мовою програмування Python з використанням редактора VS Code та розширення Jupyter Notebook шляхом розробки нової функції `generation_attachments()` і допоміжної функції `draw_image_text()` з використання бібліотеки Pillow замість OpenCV для формування текстових написів на зображенні шрифтом «Times New Roman» українською мовою та курсивним накресленням тексту та оновлення функції `generation_dataset_grade()`. У результаті числового експерименту отримано покращені тренувальний, перевірочний та тестовий набори даних, які відповідно містять 6892, 3795 і 2063 зображень оцінок.

Запропоновані нова версія функції `generation_attachments()`, допоміжна функція `draw_image_text()` та оновлена версія функції `generation_dataset_grade()` дозволили реалізувати розробку покращеного класифікатора зображень оцінок для розпізнавання оцінок із скан-копій додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів. Точність покращеного класифікатора стосовно розпізнавання зображень оцінок на тестовому наборі даних склала 97,19%, що дозволяє на його основі розробити додаток для здійснення розрахунку середнього бала свідоцтва про здобуття повної загальної середньої освіти абітурієнта.

1. Вступ на бакалавра. Розрахунок середнього бала атестата у 2022 році. *Освіта*. 2022. URL: <https://osvita.ua/consultations/bachelor/7132/>. (дата звернення: 02.11.2025).
2. Інструкція щодо роботи з системою подання заяв в електронній формі. Вступ для здобуття вищої, фахової передвищої освіти на основі свідоцтва про повну загальну середню освіту (11 класів), диплома фахового молодшого бакалавра, молодшого бакалавра, молодшого спеціаліста (НРК5). 2024. URL: https://vstup.edbo.gov.ua/files/Instrukciya_EK_VO_PZSO_NRK5_2024.pdf. (дата звернення: 02.11.2025).
3. Климчук Ю. Є. Розробка класифікатора зображень оцінок для розпізнавання оцінок із сканкопій додатків до свідоцтв про здобуття повної загальної середньої освіти абітурієнтів. *Вісник Національного університету водного господарства та природокористування. Технічні науки*. 2024. Вип. 2(106). С. 150–167.
4. James Samuel. Building an Image Classifier With Pytorch. *Samuel James*. Dev. 2020. URL: https://dev.to/abiodunjames/building-an-image-classifier-using-pytorch-46dk?comments_sort=latest. (дата звернення: 02.11.2025).
5. Neo Benedict. Building an Image Classification Model From Scratch Using PyTorch. *Benedict Neo / Medium*. 2021. URL: <https://medium.com/bitgrit-data-science-publication/building-an-image-classification-model-with-pytorch-from-scratch-f10452073212>. (дата звернення: 02.11.2025).
6. Pointer Ian. Programming PyTorch for Deep Learning: Creating and

Deploying Deep Learning Applications. *Ian Pointer*. O'Reilly, 2019. 226 p. **7.** Papa Joe. PyTorch Pocket Reference: Building and Deploying Deep Learning Models. *Joe Papa*. O'Reilly Media, 2021. 310 p. **8.** Chung Bryan WC. Pro Processing for Images and Computer Vision with OpenCV. *Bryan Chung WC*. Apress, 2017. 301 p. **9.** Gabriel Garrido. OpenCV 3.x with Python By Example Second Edition: Make the most of OpenCV and Python to build applications for object recognition and augmented reality. *Garrido Gabriel, Joshi Prateek*. Packt Publishing, 2018. 255 p. **10.** Rosebrock Adrian. Practical Python and OpenCV: An Introductory, Example Driven Guide to Image Processing and Computer Vision. PyImageSearch, 2016. 166 p. **11.** Fernández Villán Alberto. Mastering OpenCV 4 with Python : A practical guide covering topics from image processing, augmented reality to deep learning with OpenCV 4 and Python 3.7. Packt Publishing : Birmingham – Mumbai, 2019. 650 p.

REFERENCES:

1. Vstup na bakalavra. Rozrakhunok serednoho bala atestata u 2022 rotsi. *Osvita*. 2022. URL: <https://osvita.ua/consultations/bachelor/7132/>. (data zvernennia: 02.11.2025).
2. Instruktsiia shchodo roboty z systemoiu podannia zaiav v elektronni formi. Vstup dlia zdobuttia vyshchoi, fakhovoi peredvyshchoi osvity na osnovi svidotstva pro povnu zahalnu seredniu osvitu (11 klasiv), dyploma fakhovoho molodshoho bakalavra, molodshoho bakalavra, molodshoho spetsialista (NRK5). 2024. URL: https://vstup.edbo.gov.ua/files/Instrukciya_EK_VO_PZSO_NRK5_2024.pdf. (data zvernennia: 02.11.2025).
3. Klymiuk Yu. Ye. Rozrobka klasyfikatora zobrazhen otsinok dlia rozpoznavannia otsinok iz skankopii dodatkov do svidotstv pro zdobuttia povnoi zahalnoi serednoi osvity abiturientiv. *Visnyk Natsionalnoho universytetu vodnoho hospodarstva ta pryrodokorystuvannia. Tekhnichni nauky*. 2024. Vyp. 2(106). S. 150–167.
4. James Samuel. Building an Image Classifier With Pytorch. *Samuel James*. Dev. 2020. URL: https://dev.to/abiodunjames/building-an-image-classifier-using-pytorch-46dk?comments_sort=latest. (data zvernennia: 02.11.2025).
5. Neo Benedict. Building an Image Classification Model From Scratch Using PyTorch. *Benedict Neo / Medium*. 2021. URL: <https://medium.com/bitgrit-data-science-publication/building-an-image-classification-model-with-pytorch-from-scratch-f10452073212>. (data zvernennia: 02.11.2025).
6. Pointer Ian. Programming PyTorch for Deep Learning: Creating and Deploying Deep Learning Applications. *Ian Pointer*. O'Reilly, 2019. 226 p.
7. Papa Joe. PyTorch Pocket Reference: Building and Deploying Deep Learning Models. *Joe Papa*. O'Reilly Media, 2021. 310 p.
8. Chung Bryan WC. Pro Processing for Images and Computer Vision with OpenCV. *Bryan Chung WC*. Apress, 2017. 301 p.
9. Gabriel Garrido. OpenCV 3.x with Python By Example Second Edition: Make the most of OpenCV and Python to build applications for object recognition and augmented reality. *Garrido Gabriel, Joshi Prateek*. Packt Publishing, 2018. 255 p.
10. Rosebrock Adrian. Practical Python and OpenCV: An Introductory, Example Driven Guide to Image Processing and Computer Vision. PyImageSearch, 2016. 166 p.
11. Fernández Villán Alberto. Mastering OpenCV 4 with Python : A practical guide covering topics from image processing, augmented reality to deep learning with OpenCV 4 and Python 3.7. Packt Publishing : Birmingham–Mumbai, 2019. 650 p.



Klymiuk Yu. Ye. [1; ORCID ID: 0000-0003-3672-8469],

Candidate of Engineering (Ph.D.), Associate Professor

¹National University of Water and Environmental Engineering, Rivne

GENERATING IMPROVED TRAINING, VALIDATION, AND TEST DATASETS FOR THE GRADES IMAGE CLASSIFIER

Generation of improved training, validation, and test datasets for the grade image classifier designed to recognize grades from scanned copies of attachments to applicants' certificates of completion of secondary education, caused by the OpenCV library's `putText()` function supporting only one font, "Hershey Complex," with support for the Ukrainian language with regular text layout, and was implemented by developing a new function `generation_attachments()` using the Pillow library instead of OpenCV to generate text captions on images in the "Times New Roman" font in Ukrainian and italic text, and updating the function `generation_dataset_grade()`. In the new version of the `generation_attachments()` function, designed to generate a set of images of attachments to certificates with different acceptable grades, based on the standard form of the attachment to the certificate of completion of general secondary education of A5 size `attachment.jpg`, the formation of a set of attachments to the certificates of completion of general secondary education of applicants with different acceptable grades in the format of png images is implemented. The auxiliary function `draw_image_text()` is used to form text inscriptions on a given image in a given position in the font "Times New Roman" in Ukrainian with italic text. In the updated version of the `generation_dataset_grade()` function, designed to generate training, validation, and test datasets of grade images, based on the generated png images of the supplements to the certificates of completion of general secondary education of applicants with different acceptable grades, separate png images of grades are generated and randomly distributed into the test, train, and valid folders. As a result of the numerical experiment, improved training, validation, and test datasets were obtained, which contain 6892, 3795, and 2063 assessment images, respectively. The proposed new version of the `generation_attachments()` function, the auxiliary `draw_image_text()` function, and the updated version of the `generation_dataset_grade()` function made it possible to develop an improved grade image classifier for recognizing grades from scanned copies of attachments to applicants' certificates of completion of secondary education. The accuracy of the improved classifier in recognizing images of grades on the test dataset was 97.19%, which allows us to develop an

application based on it to calculate the average score of the applicant's certificate of completion of secondary education.

Keywords: generation; dataset; image; assessment; classifier; certificate; attachment; applicant.

Отримано: 17 червня 2025 року
Прорецензовано: 02 вересня 2025 року
Прийнято до друку: 25 вересня 2025 року