

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Система контролю якості сканів у Maya EDMS з використанням
OpenCV та машинного навчання

Виконав(ла): студент(ка) VI курсу, групи СПм-61
спеціальності 121 Інженерія програмного забезпечення

	(шифр і назва спеціальності)	
	(підпис)	Попсуй В.Д. (прізвище та ініціали)
Керівник	(підпис)	Бреус В.М. (прізвище та ініціали)
Нормоконтроль	(підпис)	Стоянов Ю.М. (прізвище та ініціали)
Завідувач кафедри	(підпис)	Петрик М.Р. (прізвище та ініціали)
Рецензент	(підпис)	Гром'як Р.С. (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис) (прізвище та ініціали)
« » 20__ р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

студенту Попсуя Віталія Дмитровича
(прізвище, ім'я, по батькові)

1. Тема роботи СИСТЕМА КОНТРОЛЮ ЯКОСТІ СКАНІВ У MAYAN EDMS 3
OPENCV ТА МАШИННОГО НАВЧАННЯ

Керівник роботи Бревус Віталій Михайлович Phd

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 27 » листопада 2025 року № 4/7-1045

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи мова програмування Python; бібліотека OpenCV для аналізу
зображень; бібліотеки NumPy, Matplotlib; веб-фреймворк Flask; система електронного документообігу
Mayan EDMS; середовище розробки PyCharm

4. Зміст роботи (перелік питань, які потрібно розробити)

Аналіз предметної області, постановка завдання та цілей, пошук акторів та варіантів
використання, опис ключових варіантів використання, вибір процесу розробки, проектування
архітектури системи, побудова схем бази даних, побудова UML- діаграм класів, вибір мови та
середовища розробки, реалізація основних класів та методів, розробка інтерфейсу користувача,
тестування програмної системи, види та план тестування, розробка тестових сценаріїв,
розгортання програмної системи та системні вимоги, верифікація програмної системи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Додаток А. Тези конференції

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	28.11.2025	виконав
2	Постановка завдання та цілей	29.11.2025	виконав
3	Пошук акторів та варіантів використання	30.11.2025	виконав
4	Опис ключових варіантів використання	01.12.2025	виконав
5	Вибір процесу розробки	02.12.2025	виконав
6	Проектування архітектури системи	03.12.2025	виконав
7	Побудова схем бази даних	04.12.2025	виконав
8	Побудова UML-діаграм класів	05.12.2025	виконав
9	Вибір мови та середовища розробки	07.12.2025	виконав
10	Реалізація основних класів та методів	08.12.2025	виконав
11	Розробка інтерфейсу користувача	10.12.2025	виконав
12	Тестування програмної системи	11.12.2025	виконав
13	Види та план тестування	12.12.2025	виконав
14	Розробка тестових сценаріїв	13.12.2025	виконав
15	Розгортання програмної системи та системні вимоги	14.12.2025	виконав
16	Верифікація програмної системи	15.12.2025	виконав
17	Охорона праці	16.12.2025	виконав
18	Розробка застосунку	17.12.2025	виконав
19	Підготовка до захисту	21.12.2025	виконав

Студент

(підпис)

Попсуй В.Д.

(прізвище та ініціали)

Керівник роботи

(підпис)

Бреус В.М.

(прізвище та ініціали)

АНОТАЦІЯ

Попсуй Віталій Дмитрович. Система контролю якості сканів у Maya EDMS з використанням OpenCV та машинного навчання. Кваліфікаційна робота освітнього ступеня «магістр». Тернопільський національний технічний університет імені Івана Пулюя, кафедра програмної інженерії, група СПм-61, спеціальність 121 «Інженерія програмного забезпечення». Тернопіль, 2025. Сторінок 83, таблиць 1, рисунків 7, додатків 1, бібліографічних посилань 56.

Метою роботи є розроблення програмної системи автоматизованого контролю якості сканованих документів, інтегрованої з Maya EDMS, із застосуванням методів комп'ютерного зору та машинного навчання.

Об'єктом дослідження є процеси обробки сканованих документів у системах електронного документообігу.

Предметом дослідження є методи та програмні засоби оцінювання якості сканів документів (різкість, шум, контрастність, кут нахилу, засвітлення) з використанням OpenCV і моделей машинного навчання.

Методи дослідження включають аналіз існуючих EDMS-рішень, проєктування багаторівневої архітектури, побудову UML-діаграм, застосування алгоритмів комп'ютерного зору, інтеграцію ML-моделей, а також тестування та верифікацію програмної системи.

У роботі реалізовано модуль контролю якості сканів, який автоматично виявляє дефекти документів, обчислює метрики якості та зберігає результати аналізу в Maya EDMS із формуванням рекомендацій щодо повторного сканування. Система розроблена мовою Python та інтегрована з Maya EDMS через REST API, що забезпечує модульність і масштабованість рішення.

Ключові слова: Maya EDMS, контроль якості сканів, OpenCV, машинне навчання, комп'ютерний зір, електронний документообіг, Python.

ABSTRACT

Popsui Vitalii Scan Quality Control System in Mayan EDMS Using OpenCV and Machine Learning. Master's qualification thesis. Ternopil Ivan Puluj National Technical University, Department of Software Engineering, group SPm-61, specialty 121 "Software Engineering". Ternopil, 2025. Pages 83, tables 1, figures 7, appendices 1, references 56.

The aim of this thesis is to develop a software system for automated quality control of scanned documents integrated with Mayan EDMS using computer vision and machine learning techniques.

The object of the study is the process of processing scanned documents in electronic document management systems.

The subject of the study includes methods and software tools for evaluating scan quality parameters such as sharpness, noise level, contrast, skew angle, and illumination using OpenCV and machine learning models.

The research methods involve analysis of existing EDMS solutions, design of a multi-layer system architecture, development of UML diagrams, application of computer vision algorithms, integration of machine learning models, as well as testing and verification of the developed system.

The thesis presents an implemented scan quality control module that automatically detects document defects, calculates quality metrics, and stores analysis results in Mayan EDMS with recommendations for rescanning. The system is developed in Python and integrated with Mayan EDMS via a REST API, providing modularity and scalability.

Keywords: Mayan EDMS, scan quality control, OpenCV, machine learning, computer vision, electronic document management, Python.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API (Application Programming Interface) — програмний інтерфейс взаємодії між програмними компонентами системи та зовнішніми сервісами Mayan EDMS;

EDMS (Electronic Document Management System) — система електронного документообігу для зберігання, обробки та керування документами;

Mayan EDMS — відкрита web-орієнтована система електронного документообігу;

OpenCV (Open Source Computer Vision Library) — бібліотека комп'ютерного зору для обробки та аналізу зображень;

ML (Machine Learning) — напрям штучного інтелекту, що передбачає побудову моделей для автоматичного аналізу даних;

CNN (Convolutional Neural Network) — згортова нейронна мережа для аналізу зображень;

REST (Representational State Transfer) — архітектурний стиль побудови web-сервісів;

HTTP (HyperText Transfer Protocol) — протокол передачі даних у мережі Інтернет;

JSON (JavaScript Object Notation) — формат обміну структурованими даними між клієнтом і сервером;

OCR (Optical Character Recognition) — технологія оптичного розпізнавання тексту;

DPI (Dots Per Inch) — показник роздільної здатності зображення;

UI (User Interface) — користувацький інтерфейс системи;

UML (Unified Modeling Language) — уніфікована мова моделювання програмних систем;

ER-діаграма (Entity–Relationship Diagram) — діаграма для моделювання структури бази даних;

Docker — програмна платформа для контейнеризації застосунків;

PostgreSQL — реляційна система управління базами даних;

Flask — Python-фреймворк для створення web-застосунків та REST API;

Python — високорівнева мова програмування загального призначення;

Variance of Laplacian — метод оцінювання різкості зображення;

Deskew — процес визначення та корекції кута нахилу сканованого документа.

ЗМІСТ

ВСТУП.....	9
1 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ	11
1.1 Аналіз вимог до програмної системи.....	11
1.1.1 Аналіз предметної області.....	12
1.1.2 Визначення та постановка задачі.....	14
1.1.3 Опис ключових варіантів використання.....	19
1.2 Проєктування програмної системи	22
1.2.1 Логіка структури системи	25
1.2.2 Опис фізичної структури.....	29
2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ.....	31
2.1 Моделювання архітектури системи.....	31
2.2 Вибір системи керування базами даних	34
2.3 Побудова схеми бази даних	37
2.4 Конструювання програмної системи	30
2.4.1 Обґрунтування вибору технологій розробки	41
2.4.2 Побудова UML-діаграм класів.....	42
2.4.3 Вибір мови програмування	45
2.4.4 Обґрунтування вибору технологічного засобу реалізації веб-проєкту	46
2.4.5 Механізм опрацювання даних та доступ до бази даних	46
2.4.6 Обґрунтування вибору середовища розробки.....	48
2.4.7 Функціональні можливості графічного інтерфейсу користувача	49
3 ТЕСТУВАННЯ, ВПРОВАДЖЕННЯ ТА ПІДТРИМКА	52
3.1 Тестування програмної системи	52
3.1.1 Види та план тестування	57
3.1.2 Розробка тестових сценаріїв	60
3.2 Розгортання програмної системи та системні вимоги	63
3.3 Верифікація програмної системи	65
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	66
4.1 Охорона праці	66

4.2 Стійкість роботи системи контролю до дії електромагнітного імпульсу ядерного вибуху та заходи захисту	69
ВИСНОВКИ.....	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	79
ДОДАТКИ.....	83
Додаток А. Тези конференції	

ВСТУП

Сучасні бізнес-структури, органи державного управління та інші організації все активніше переходять до цифрової трансформації, ключовим елементом якої є впровадження автоматизованих інформаційних систем. Протягом останніх десятиліть електронний документообіг став обов'язковим атрибутом стабільної та конкурентоспроможної діяльності компаній у всьому світі. Використання таких систем забезпечує зменшення витрат, пришвидшення бізнес-процесів, підвищення точності й прозорості роботи з документами.

Однією з важливих складових сучасного електронного документообігу є електронне архівування та зберігання сканованих документів, яке вимагає високої якості цифрових копій. Низька роздільність, розмитість, тіні, неправильна орієнтація чи перекося сканів призводять до помилок в обробці, ускладнюють роботу з документами та підвищують ризики втрати цінної інформації. У великомасштабних системах документообігу, де за добу опрацьовуються тисячі сторінок, автоматичний контроль якості сканів стає критично важливим.

З огляду на зростання попиту на надійні та мобільні системи документообігу, особливо у воєнний та післявоєнний періоди відновлення України, актуальним є розроблення програмних рішень, які забезпечують автоматизацію операцій із документами без втрати якості. Саме тому вибір теми, пов'язаної із впровадженням інтелектуальних методів аналізу якості сканів, є обґрунтованим та своєчасним.

Mayan EDMS — популярна відкрита система електронного документообігу, що підтримує модульність, масштабованість і Web-орієнтованість. Однак у її базовій конфігурації відсутні засоби глибинного аналізу якості сканів. Це створює передумови для розроблення розширення, здатного автоматично визначати дефекти сканованих документів та попереджати завантаження неякісних копій у систему.

У контексті розвитку технологій комп'ютерного зору та машинного навчання стає можливим створення таких інструментів на основі бібліотеки

OpenCV та моделей аналізу зображень, які дозволяють проводити класифікацію якості, виявляти дефекти та рекомендувати повторне сканування документа.

Метою роботи є

розроблення програмного модуля для автоматичного контролю якості сканів у системі Maya EDMS на основі алгоритмів комп'ютерного зору та методів машинного навчання.

Для досягнення мети необхідно виконати такі завдання:

1. Проаналізувати предметну область, структуру бізнес-процесів електронного документообігу та вимоги до контролю якості сканованих документів.

2. Дослідити можливості Maya EDMS, її архітектуру та механізми розширення функціональності.

3. Вивчити алгоритми OpenCV, що використовуються для оцінки чіткості, контрастності, шумності та інших параметрів зображень.

4. Розробити модель машинного навчання для класифікації якісних та неякісних сканів.

5. Спроекувати архітектуру програмного модуля, що інтегрується з Maya EDMS та забезпечує автоматичну валідацію документів.

6. Реалізувати програмне забезпечення у вигляді Web-орієнтованого сервісу або вбудованого компонента.

7. Провести тестування системи та оцінити її ефективність у реальних сценаріях обробки документів.

Об'єкт дослідження:

процес автоматизації контролю якості сканованих документів у системах електронного документообігу.

Предмет дослідження:

методи, алгоритми та засоби реалізації програмного модуля контролю якості сканів у Maya EDMS із використанням OpenCV і машинного навчання.

1 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

Розділ присвячено розробці програмної системи автоматизованого контролю якості сканованих документів, інтегрованої з системою електронного документообігу Mayan EDMS. У межах розділу проведено аналіз вимог до системи, досліджено предметну область, сформовано основні завдання та визначено архітектурні принципи функціонування майбутнього рішення.

Основна увага приділяється автоматизації процесів оцінки якості сканів шляхом застосування алгоритмів комп'ютерного зору (OpenCV) та методів машинного навчання. Розділ містить детальний опис функціональних можливостей системи, її логічної та фізичної структури, а також варіантів використання, що забезпечують ефективну взаємодію користувачів із програмним комплексом.

Матеріали розділу ґрунтуються на сучасних підходах до проєктування інтелектуальних систем обробки даних і слугують основою для подальшої реалізації та впровадження запропонованого рішення.

1.1 Аналіз вимог до програмної системи

Аналіз вимог до програмної системи є необхідним етапом проєктування, оскільки дозволяє визначити функціональні можливості, обмеження та критерії якості розроблюваного програмного продукту. У межах даної магістерської роботи об'єктом дослідження є система автоматизованого контролю якості сканованих документів, інтегрована з системою електронного документообігу Mayan EDMS.[1-5]

Основною метою розробки системи є підвищення якості цифрових копій документів шляхом автоматичного виявлення дефектів сканування та формування рекомендацій щодо їх усунення. Для досягнення цієї мети програмна система повинна забезпечувати автоматизований аналіз зображень із використанням алгоритмів комп'ютерного зору (OpenCV) та методів машинного навчання [6-12].

1.1.1 Аналіз предметної області

У сучасних інформаційних системах електронного документообігу значну роль відіграє якість цифрових копій документів. Наявність нечітких, перекошених чи неправильно відсканованих сторінок ускладнює зберігання, індексацію, пошук та подальше опрацювання інформації. У великих організаціях, де щодня оцифровується значний обсяг документації, відсутність автоматизованого контролю якості сканів призводить до накопичення помилок, дублювання роботи, втрати даних та збільшення навантаження на персонал.

Система Maуan EDMS, яка є поширеною платформою з відкритим кодом для організації електронного документообігу, забезпечує збереження, обробку та керування документами, однак не містить механізмів глибокої оцінки якості сканованих зображень. Це створює потребу у розробленні додаткового інтелектуального модуля, здатного оцінювати якість документів ще на етапі їх завантаження [3,7].

Предметна область охоплює процеси обробки зображень, аналізу сканованих документів та їх класифікації за придатністю до зберігання. Методи комп'ютерного зору та машинного навчання дозволяють формувати автоматизовані критерії оцінювання, що значно зменшує вплив людського фактора та підвищує точність визначення дефектів.[18-21]

До типових проблем, які виникають при скануванні документів, належать:

- розмитість (blur), спричинена рухом чи поганим фокусуванням;
- надмірний шум на зображенні;
- низька контрастність, що ускладнює подальше розпізнавання тексту;
- перекиї та неправильна орієнтація сторінки;
- тіні та засвітлення, що закривають важливі фрагменти;
- часткове обрізання документа;
- артефакти компресії або сканування;
- помилки багатосторінкового сканування (переплутані сторінки, порожні скани тощо).

Автоматизація контролю якості у системах електронного документообігу має такі ключові передумови:

1. Великий обсяг вхідної документації.

У багатьох організаціях щоденно опрацьовуються сотні або тисячі сторінок, що робить ручну перевірку неефективною.

2. Висока трудомісткість обробки сканів.

Перевірка зображень на відповідність вимогам займає багато часу та потребує технічної компетентності працівників.

3. Зростання вимог до якості цифрових архівів.

Організації прагнуть до створення надійних, стандартизованих та довготривалих сховищ.

4. Автоматизація виявлення дефектів.

Використання алгоритмів OpenCV дозволяє зменшити кількість помилок, забезпечити стабільність оцінки та пришвидшити опрацювання [12-15].

5. Інтеграція з існуючими Web-системами.

Для забезпечення мобільності та доступності система контролю якості має функціонувати у Web-середовищі та бути сумісною з Maya EDMS [3,7].

Основне завдання розробки полягає у створенні правильно спроектованого ПЗ, здатного:

- автоматично отримувати скани документів із Maya EDMS [3,7];
- аналізувати їх за сукупністю метрик (різкість, контраст, рівень шуму, орієнтація тощо);
- класифікувати скани як придатні або непридатні;
- рекомендувати повторне сканування у разі виявлення дефектів;
- зберігати результати аналізу у структурованому вигляді.

Таким чином, предметна область поєднує методи комп'ютерного зору, машинного навчання, Web-орієнтовані архітектури та технології управління документами. Для якісної реалізації системи необхідно сформулювати чіткі вимоги до її структури, алгоритмів та способів обробки даних [18-21].

1.1.2 Визначення та постановка задачі

Після проведеного аналізу предметної області та виявлення ключових проблем, пов'язаних із якістю сканованих документів у системах електронного документообігу, необхідно чітко сформулювати завдання, які має вирішувати розроблювана програмна система. Оскільки Mayan EDMS не містить вбудованих механізмів оцінки якості сканів, постає потреба у створенні програмного модуля, який забезпечуватиме автоматичний аналіз зображень на основі методів комп'ютерного зору та машинного навчання [1-5].

Насамперед необхідно визначити:

- які саме параметри якості сканів підлягають оцінюванню;
- які алгоритми та моделі є оптимальними для автоматизації контролю;
- яким чином здійснюватиметься інтеграція модуля з Mayan EDMS;
- які дані мають бути збережені у базі даних і якою буде їх структура;
- які функції будуть доступні різним ролям користувачів;
- якою буде загальна архітектура системи та її інтерфейс.

Важливим завданням є визначення оптимальної моделі зберігання інформації та способу взаємодії між модулем контролю якості й основною системою документообігу. Оскільки Mayan EDMS використовує реляційну модель даних, необхідно забезпечити коректне проєктування додаткових таблиць, що зберігатимуть метадані аналізу: рівень різкості, контрастності, ступінь шуму, результат класифікації, дату обробки та інші характеристики [3,7].

Ще одним аспектом є визначення набору процесів, які підлягають автоматизації. До них належать:

- первинний аналіз якості завантажених документів;
- автоматичне визначення дефектів сканів;
- класифікація документів за придатністю;
- рекомендації щодо повторного сканування;
- збереження результатів перевірки та формування звітності;
- можливість повторної обробки документів.

Після аналізу предметної області було сформульовано перелік основних завдань, що мають бути реалізовані у межах розробки системи:

1. Вибір та проектування структури бази даних, яка зберігатиме результати аналізу та метадані сканів.
2. Визначення способу інтеграції з Maya EDMS, зокрема використання API, черг задач або внутрішніх механізмів обробки документів [3,7].
3. Формування переліку ключових параметрів якості, які потрібно вимірювати за допомогою алгоритмів OpenCV [12-15].
4. Розробка моделі машинного навчання для класифікації сканів на придатні та непридатні [18-21].
5. Аналіз можливих сценаріїв автоматизації, включаючи попередню валідацію та процеси обробки великих пакетів документів.
6. Проектування інтерфейсу користувача, орієнтованого на простоту, інтуїтивність і доступність для адміністратора системи.
7. Оптимізація обсягу інформації, що зберігається, щоб система залишалася продуктивною навіть за великої кількості документів.

Для роботи бази даних необхідно визначити, які саме дані повинні зберігатися. До основної інформації можуть належати:

- унікальний ідентифікатор скану;
- посилання на документ у Maya EDMS;
- значення метрик якості (різкість, контрастність, шум, перспективні викривлення тощо);
- результат класифікації ML-моделі;
- дата та час аналізу;
- рекомендації щодо повторного сканування;
- дані про повторні обробки (за потреби).

Виходячи з цього, програмний продукт має задовольняти такі системні вимоги:

- автоматичний аналіз якості сканованих документів;

- виявлення основних дефектів (розмитість, перекіс, низький контраст, надмірний шум тощо);

- класифікація придатності скану на основі машинного навчання;
- можливість інтеграції з Maya EDMS через API або модульні механізми;
- збереження метаданих аналізу та результатів обробки;
- можливість перегляду статистики та звітів адміністраторами;
- наявність зручного та інтуїтивного Web-інтерфейсу;
- стабільність і надійність зберігання даних;
- можливість фільтрування, сортування та повторного аналізу документів.

Окрему увагу слід приділити вимогам до інтерфейсу. Система повинна:

- мати просту структуру та логіку переходів між функціями;
- забезпечувати чіткі повідомлення про стан обробки документів;
- уникати перевантаженості непотрібними елементами;
- підтримувати консистентність у візуальних компонентах;
- містити зрозумілі підказки та повідомлення для користувача;
- забезпечувати можливість швидкого розуміння результатів перевірки.

Таким чином, сформульована задача передбачає створення інтелектуальної програмної системи, яка здатна забезпечити автоматизацію контролю якості сканів у Maya EDMS, що, у свою чергу, покращить точність, швидкість і ефективність роботи з документами [3,7].

На основі сформульованих вимог до системи контролю якості сканів у Maya EDMS визначимо основних актантів (учасників взаємодії із системою) та перелік варіантів використання, що описують можливі сценарії роботи майбутнього рішення. Враховуючи специфіку предметної області – аналіз сканів за допомогою алгоритмів комп'ютерного зору та машинного навчання – необхідно окреслити ключові ролі, їх цілі та функціональні можливості.

Визначення актантів

У системі виділяються два основні типи користувачів:

1. Адміністратор системи (Administrator)

Адміністратор відповідає за:

- налаштування параметрів аналізу якості сканів;
- управління запуском автоматичної чи ручної обробки документів;
- перегляд результатів класифікації та виявлених дефектів;
- керування повторним скануванням документів;
- редагування або доповнення параметрів моделі машинного навчання (за потреби);
- управління сховищем метаданих та таблицями, що містять результати перевірки;
- моніторинг стану інтеграції з Mayan EDMS та журналів обробки.

Адміністратор є ключовою роллю, яка має найширший набір прав доступу, включаючи зміну конфігурації системи та керування потоками обробки.

2. Користувач-оператор (Operator / End User)

Оператор — це працівник, який взаємодіє з документами у межах своїх службових обов'язків. Його завдання:

- завантажувати документи до Mayan EDMS;
- переглядати повідомлення про якість щойно доданих сканів;
- отримувати рекомендації щодо повторного сканування;
- переглядати сповіщення про помилки або дефекти (розмитість, низький контраст, перекіс тощо);
- працювати з документами, які класифіковано як придатні.

Оператор не має доступу до зміни конфігурацій системи чи параметрів роботи алгоритмів.

3. Система Mayan EDMS

Ця система також є актантом, оскільки:

- надає API для отримання документів;
- отримує від модуля результати аналізу;
- виконує зберігання та відображення метаданих;
- забезпечує події (тригери), що запускають обробку.

4. Модуль машинного аналізу (ML/Computer Vision Module)

Цей модуль виконує:

- обчислення метрик якості за допомогою OpenCV;
- класифікацію документа на основі ML-моделі;
- формування структурованих результатів обробки для подальшого зберігання.

Варіанти використання

На основі визначених актантів формуємо ключові use cases майбутньої системи.

1. Автоматичний аналіз щойно завантажених сканів

Опис:

Після завантаження нового документа в Maya EDMS система ініціює автоматичний аналіз якості.

Результат:

Документ позначається як придатний або непридатний, метрики якості зберігаються.

2. Ручний запуск аналізу (адміністратор)

Адміністратор може ініціювати повторний або спеціальний аналіз окремих документів чи їх груп.

3. Визначення дефектів документа

Система формує перелік виявлених дефектів:

- розмитість;
- низький контраст;
- шум;
- перекіс зображення;
- часткове затемнення/засвічення тощо.

4. Класифікація документа моделлю машинного навчання

ML-модель визначає, чи документ відповідає мінімальним вимогам якості.

5. Перегляд результатів перевірки

Оператор та адміністратор можуть переглядати:

- рівень якості;
- виявлені дефекти;

- рекомендації;
- історію обробок.

6. Збереження та обробка метаданих

Модуль зберігає:

- значення метрик;
- класифікацію документа;
- дату та час аналізу;
- інформацію для статистики.

7. Формування рекомендацій користувачу

Оператор отримує вказівки типу:

- «необхідно повторити сканування»;
- «документ має невисоку різкість»;
- «виявлено перекис — рекомендується перезняти».

8. Перегляд звітів та статистики (адміністратор)

Адміністратор може переглядати:

- кількість документів із низькою якістю;
- загальну статистику обробки;
- навантаження на модуль обробки;
- ефективність ML-моделі.

1.1.3 Опис ключових варіантів використання

На основі аналізу актантів та функціональних вимог було визначено основних користувачів системи контролю якості сканів, яка інтегрується з Maup EDMS. Головним актором виступає Адміністратор системи, який відповідає за налаштування параметрів обробки, управління документопотоком та контроль коректності результатів аналізу [3,7].

Другим актором є Оператор (користувач системи), який взаємодіє зі сканами документів у межах своїх робочих завдань. У деяких випадках їх функції

можуть перетинатися, проте роль адміністратора має розширені права доступу та відповідає за технічну частину роботи системи.

Таблиця 1.1 – Актори системи

№	Актор	Найменування функції	Формулювання
1	Адміністратор	Налаштування параметрів аналізу якості	Дозволяє встановлювати порогові значення різкості, контрастності, рівня шуму, кута перекосу та інших показників якості.
2	Адміністратор	Керування запуском обробки документів	Забезпечує запуск автоматичного або ручного аналізу окремих файлів чи пакетів документів.
3	Адміністратор	Перегляд та коригування результатів	Надає можливість перегляду всіх метрик якості, списків дефектів та ініціювання повторного аналізу.
4	Адміністратор	Управління інтеграцією з Mayan EDMS	Забезпечує налаштування параметрів інтеграції, обробку подій документообігу та доступ до журналів.
5	Адміністратор	Перегляд статистики та звітності	Дозволяє переглядати агреговані дані, історію обробок, статистику дефектів та ефективність моделі.
6	Оператор	Додавання документів	Завантажує скановані документи в систему електронного документообігу, ініціюючи автоматичний аналіз якості.
7	Оператор	Перегляд результатів аналізу	Отримує інформацію про придатність документа, виявлені дефекти та рекомендації щодо повторного сканування.
8	Оператор	Реакція на рекомендації системи	Виконує повторне сканування або інші коригувальні дії відповідно до виявлених системою проблем.

UML-діаграма варіантів використання [13-20]

Нижче на рисунку 1.1 наведено UML-діаграму основних варіантів використання, що відображає взаємодію між користувачами, Maya EDMS та модулем контролю якості сканів [1-5].

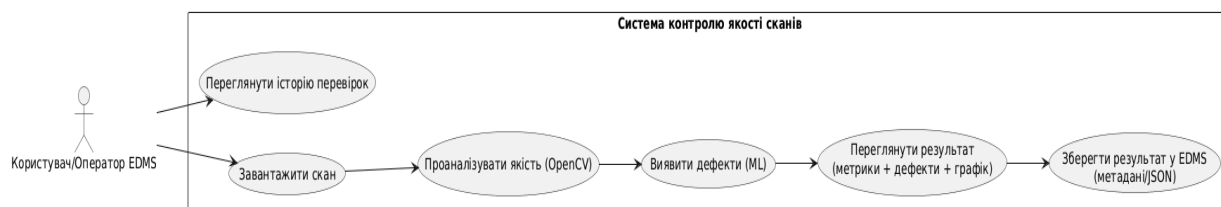


Рисунок 1.1 – діаграма варіантів використання

Опис ключових варіантів використання

На основі таблиці акторів визначено такі основні групи сценаріїв (use cases):

1. Завантаження документа та автоматичний аналіз

Актор: Оператор

Опис: Після завантаження документа в Maya EDMS система автоматично зчитує файл, аналізує його за допомогою алгоритмів OpenCV і ML-моделі та генерує оцінку якості [12-15].

2. Класифікація документа та виявлення дефектів

Актор: Система

Опис: На основі обчислених метрик (різкість, перешкоди, контраст, перекіс тощо) документ класифікується як «придатний» або «непридатний».

3. Формування рекомендацій для користувача

Актор: Система

Опис: У випадку низької якості система надає оператору список проблем і рекомендації (наприклад, «повторити сканування», «виправити освітлення»).

4. Повторне сканування та обробка

Актор: Оператор

Опис: Якщо документ не пройшов перевірку якості, оператор повторно його сканує та завантажує в систему.

5. Ручний запуск повторного аналізу

Актор: Адміністратор

Опис: Дає змогу повторно проаналізувати документи, які були неправильно класифіковані або оновлені.

6. Перегляд результатів та метрик якості

Актори: Оператор, Адміністратор

Опис: Користувачі можуть переглядати показники якості та список виявлених дефектів.

7. Адміністрування та налаштування системи

Актор: Адміністратор

Опис: Включає налаштування порогів якості, параметрів ML-моделі, інтеграції з API Maya EDMS, розподілу прав доступу.

Запропоновані варіанти використання відображають ключові процеси забезпечення автоматичного контролю якості сканів у Maya EDMS. Система забезпечує:

- автоматизацію рутинних операцій;
- зменшення кількості помилкових або непридатних документів;
- спрощення взаємодії операторів та адміністраторів;
- підвищення точності та швидкості обробки завдяки ML та OpenCV;
- можливість масштабування та подальшої адаптації до потреб користувачів.

1.2 Проєктування програмної системи

Проєктування системи контролю якості сканів, інтегрованої з Maya EDMS, потребує чіткого визначення її архітектури, логічної структури, взаємодії компонентів та способів розгортання. На цьому етапі важливо створити таку модель системи, яка забезпечить масштабованість, надійність, повторюваність опрацювання документів та можливість подальшого розширення функціональності.

З огляду на те, що Maya EDMS є web-орієнтованою системою з модульною архітектурою, розробка модуля контролю якості сканів також має спиратися на web-інструментарій та інтерфейси REST API для обміну даними. Це забезпечує можливість інтеграції без порушення базової логіки системи документообігу [1-5].

Логічна структура системи

Проектована система складається з кількох основних логічних рівнів:

1. Рівень даних

Включає:

- зображення сканів, що зберігаються в Maya EDMS;
- метрики якості (різкість, контраст, шум, кут нахилу);
- результати класифікації, отримані з ML-моделі;
- журнали подій та історію обробки документів.

Дані зберігаються у структурі EDMS, що унеможливорює дублювання та забезпечує консистентність.

2. Рівень обробки (Processing Layer)

Містить основну бізнес-логіку системи:

- завантаження скану через API;
- попередня обробка зображень (OCR-нейтральна нормалізація);
- обчислення метрик якості за допомогою OpenCV;
- класифікація документа як “придатний/непридатний”;
- формування рекомендацій для користувача;
- передача даних назад у Maya EDMS.

Цей рівень реалізується у вигляді окремого сервісу (мікросервісу), що дозволяє розподіляти навантаження та спрощує масштабування.

3. Рівень взаємодії з користувачем

Передбачає надання зрозумілого та інтуїтивного інтерфейсу:

- перегляд статусу аналізу;
- повідомлення про виявлені дефекти;
- рекомендації щодо повторного сканування;

- перегляд історії опрацювання документів.

Mayan EDMS дозволяє інтегрувати власні UI-компоненти, тому інтерфейс може бути реалізований як внутрішня панель модуля.

Фізична структура системи

Фізична структура визначає розташування файлів, модулів та сервісів. Вона складається з:

1. Модуля контролю якості (Quality Control Module)

У його складі:

- Python-пакет з алгоритмами OpenCV;
- ML-модель класифікації (наприклад, CNN чи MobileNet);
- інтерфейс API для обміну даними з Mayan EDMS;
- логічні обробники подій (webhooks).

Файли структуровано відповідно до стандарту Python-проектів (src/, models/, utils/) [46-53].

2. Компонента інтеграції з Mayan EDMS

Містить:

- конфігураційні файли інтеграції;
- розширення для UI;
- власні моделі даних (якщо необхідно);
- маршрути та контролери.

3. Рівень деплою

Система може бути розміщена:

- на сервері EDMS (як локальний модуль) [1-5];
- у вигляді Docker-контейнера [37-45];
- як зовнішній сервіс, що взаємодіє через REST API [37-45].

Файлова структура включає:

- Dockerfile [37-45];
- docker-compose.yml [13-20];
- конфігураційні файли середовища (env).

Обґрунтування структурування

Продумана структура системи дозволяє:

- чітко розмежувати відповідальності між компонентами;
- забезпечити зручність розширення функціональності;
- полегшити розробку, тестування та командну роботу;
- створити стандартизований підхід до обробки документів;
- інтегрувати ML-алгоритми без порушення роботи EDMS.

Важливо наголосити, що добре структурована система зменшує складність внесення змін і дає змогу безболісно масштабувати алгоритми оцінювання якості у майбутньому.

Таким чином, на етапі проєктування було сформовано логічну й фізичну структуру системи, визначено основні складові, способи їх взаємодії та інтеграції з Maya EDMS. Чітке структурування компонентів забезпечує ефективність реалізації, прозорість архітектури та можливість подальшого розширення системи у відповідь на нові вимоги.

1.2.1 Логіка структури

При проєктуванні системи контролю якості сканів важливо забезпечити логічну, послідовну та зрозумілу структуру, яка дозволить користувачам швидко отримувати доступ до результатів аналізу, виконувати повторне сканування та взаємодіяти з документами в середовищі Maya EDMS. Недоопрацьована логіка структури призводить до зниження ефективності роботи, збільшує кількість помилок користувача та ускладнює розуміння принципів функціонування системи [1-5].

Структура розробленого модуля повинна враховувати як правила зручності взаємодії з інформацією, так і особливості поведінки користувача у системі документообігу. Користувач має легко орієнтуватися в інтерфейсі, розуміти логіку переходів між розділами, а також мати змогу швидко виконувати дії, пов'язані з аналізом якості сканів.

Принципи логічного проєктування [21-25]

Логічне проектування системи передбачає структуроване розміщення інформації, функціональних елементів і компонентів навігації. У процесі проектування для забезпечення зрозумілості й послідовності було враховано чотири ключові принципи організації даних:

1. Використання чітких позначень

Усі елементи інтерфейсу — від кнопок до розділів — мають містити однозначні текстові назви. Для систем, що працюють із документами, уникнення неоднозначних піктограм є принциповим. Користувач повинен одразу розуміти призначення елемента, наприклад:

- «Переглянути якість»
- «Повторне сканування»
- «Детальний аналіз»
- «Історія перевірок»

Чіткі текстові позначення зменшують помилки та прискорюють роботу.

2. Доцільність розміщення інформації

Кожен елемент інтерфейсу або структура даних повинні мати власну логічну мету. Для користувача важливо, щоб інформація, пов'язана з якістю сканів (метрики різкості, рівень шуму, кут нахилу тощо), була зосереджена в одному місці і не дублювалася в інших розділах системи.

Зайві елементи — складні графічні деталі, непотрібні статуси, повторювані повідомлення — були виключені.

3. Однотипність

Всі однакові елементи (таблиці, картки документів, статуси якості, кнопки дій) повинні мати:

- однаковий стиль,
- однаковий формат відображення результатів,
- однаковий алгоритм взаємодії.

Єдина логіка відображення результатів перевірки:

- зелений статус — скан якісний;
- жовтий — допустимий, але бажано пересканувати;

- червоний — неякісний.

Однотипність підвищує передбачуваність інтерфейсу та зменшує когнітивне навантаження.

4. Поділ на змістові частини

Оскільки людина не здатна ефективно сприймати великі обсяги інформації одночасно, інтерфейс системи повинен відображати дані невеликими логічними блоками. Найкраще сприймаються групи, що містять 4–6 елементів, тому:

- метрики якості згруповано в окремі категорії: різкість, контраст, шум, геометрія;
- панель дій містить не більше 4–5 кнопок;
- звіт про якість поділено на блоки: текстовий висновок, детальні метрики, візуалізація.

Поділ інформації на блоки суттєво полегшує взаємодію з системою.

Вибір типу структурної організації

Під час аналізу існуючих варіантів структур (лінійної, ієрархічної, мережевої, комбінованої) було визначено, що оптимальною для системи контролю якості документів є комбінована структура, яка об'єднує:

Лінійну логіку

Послідовні кроки для користувача під час роботи:

1. Завантаження скану.
2. Автоматичний аналіз.
3. Перегляд результатів.
4. Рішення щодо повторного сканування.

Це забезпечує чітку схему дій і зменшує ймовірність помилок.

Ієрархічну структуру

На верхньому рівні:

«Головна сторінка модуля»

- «Аналіз якості»
- «Історія перевірок»
- «Налаштування моделі»

- «Статистика системи»

Це дозволяє швидко знайти потрібні функціональні можливості.

Мережеву структуру

У межах EDMS користувач має можливість переходити між документами, які пов'язані між собою логікою ділових процесів — наприклад, між сторінками одного пакету документів.

Таким чином, комбінована логічна структура найкраще відображає взаємозв'язок процесів сканування, перевірки, перегляду й повторної подачі документів.

Роль користувача у формуванні структури

У процесі розробки було враховано очікування кінцевих користувачів — операторів сканування, адміністраторів та аналітиків системи документообігу. Відмінності в ролях потребували адаптивності:

- адміністратор: доступ до налаштувань алгоритмів ML, логів, параметрів обробки;
- звичайний користувач: лише перегляд результатів та повторне сканування;
- аналітик: доступ до статистики та історії.

Структура проєктування спрямована на спрощення використання, а не на ускладнення його декоративним оформленням. Інтуїтивність інтерфейсу не повинна бути лише припущенням розробника — вона має базуватися на логічності, послідовності та однозначності дій для користувача.

Логіка структури системи контролю якості сканів ґрунтується на поєднанні чітких позначень, доцільного поділу інформації, однотипності елементів та організації інтерфейсу у вигляді комбінованої структури. Такий підхід забезпечує зручність, передбачуваність і ефективність роботи користувачів, що є критично важливим для системи документообігу з високими вимогами до якості сканування.

1.2.2 Опис фізичної структури

Фізична структура програмної системи визначає реальне розташування файлів, модулів, конфігураційних компонентів та даних у файловій системі сервера, що забезпечує коректне функціонування сервісу контролю якості сканів. На відміну від логічної структури, яка відображає взаємозв'язки між елементами системи, фізична структура описує, де саме розміщуються компоненти та яким чином організована їхня взаємодія на рівні серверного середовища.

Оскільки розроблювана система інтегрується в Mayan EDMS — модульну платформу документообігу з розширюваною архітектурою, фізична структура повинна відповідати принципам модульності та ізоляції компонентів. Основу реалізації становить створення окремого пакета (app) у структурі Mayan, що відповідає за обробку зображень, взаємодію з OpenCV, машинним навчанням та поданням результатів користувачам [1-5].

Вимоги веб-сервера і налаштувань середовища

Перед розгортанням системи необхідно врахувати особливості платформних налаштувань:

1. Початкове завантаження модуля

Система автоматично підвантажує модуль через:

```
INSTALLED_APPS += ['mayan.apps.scan_quality']
```

2. Обмеження доступу

Усі моделі та представлення захищені системою прав доступу Mayan:

- оператор сканування бачить тільки свої документи;
- адміністратор — повний доступ;
- аналітики — тільки перегляд статистики.

Для статичних файлів доступ дозволений тільки через вбудований механізм Mayan EDMS.

3. Відстеження помилок

Логи модулю зберігаються в окремому файлі:

```
/var/log/mayan/scan_quality.log
```

Це дозволяє:

- аналізувати помилки опрацювання зображень,
- відслідковувати помилки ML-моделі,
- діагностувати неправильні параметри сканера.

4. Використання деревовидної структури

Обрана деревоподібна структура каталогу дозволяє:

- ізолювати ML-модель від інтерфейсу,
- зручно організувати компоненти OpenCV,
- групувати шаблони в окремому каталозі,
- розділити обробку даних і бізнес-логіку.

Така структура відповідає основним принципам створення масштабованих Django-модулів та рекомендується до використання у великих системах документообігу.

Фізична структура системи контролю якості сканів побудована на основі модульної архітектури, що відповідає вимогам Mayan EDMS. Чіткий розподіл відповідальностей між каталогами, окремими Python-модулями, шаблонами, статичними ресурсами та моделями машинного навчання забезпечує зручність підтримки, розширення та подальшої модернізації програмного комплексу.

2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

У сучасних умовах широкого використання Web-орієнтованих інформаційних систем питання якості та надійності програмного забезпечення набуває особливої актуальності. Це особливо важливо для систем електронного документообігу, де помилки можуть призводити до втрати даних, некоректної обробки документів або прийняття неправильних управлінських рішень. Тому тестування програмної системи контролю якості сканів у Maya EDMS є невід’ємним етапом її розробки та впровадження.

2.1 Моделювання архітектури системи

Проектування архітектури системи контролю якості сканів у Maya EDMS потребує чіткого розмежування функціональних компонентів, що забезпечує масштабованість, надійність та можливість подальшого розвитку системи. Для цього доцільним є застосування шаблонів архітектурного моделювання, зокрема підходів, які дозволяють відокремити бізнес-логіку від інтерфейсу та реалізації механізмів обробки зображень.

У класичних архітектурних шаблонах на кшталт MVC або MVP зміни в інтерфейсі не впливають безпосередньо на модель, адже вся взаємодія проходить через контролер або презентер. Проте сучасні системи, що активно використовують механізми прив’язки даних, вимагають більш гнучкої моделі організації компонентів, особливо коли йдеться про обробку великих обсягів графічних матеріалів та інтеграцію алгоритмів машинного навчання. Саме тому для проектування клієнтської та адміністративної частини системи доцільно застосувати підхід Model–View–ViewModel (MVVM), який забезпечує структурованість, розширюваність та незалежність візуальних компонентів від логіки роботи модулів контролю якості [13-20].

У шаблоні MVVM передбачено наявність трьох основних складових:

1. Model (Модель) – містить бізнес-логіку та реалізацію алгоритмів обробки зображень, включно з викликами методів OpenCV, оркестрацією моделей машинного навчання, а також взаємодією із системним API Maya EDMS.

2. View (Подання) – інтерфейс користувача, реалізований у вигляді веб-сторінок, що забезпечують завантаження сканів, відображення результатів оцінки якості, повідомлень про помилки та рекомендацій щодо покращення сканування.

3. ViewModel (Модель подання) – проміжний рівень, який містить логіку перетворення даних Моделі у форму, зручну для відображення; забезпечує двостороннє зв'язування даних між інтерфейсом та функціональними модулями системи.

Однією з ключових властивостей MVVM є двостороннє зв'язування (bindings), завдяки якому зміна стану ViewModel автоматично відображається у View і навпаки. Це особливо актуально для задачі контролю якості зображень, де результати роботи алгоритмів OpenCV і ML повинні відразу оновлювати інтерфейс без необхідності прямого втручання розробника.

Властивостями ViewModel можна вважати:

- наявність двосторонньої комунікації з поданням;
- відповідність властивостей ViewModel структурам, що подаються на інтерфейс;
- відсутність прямої залежності від елементів графічного інтерфейсу, що підвищує тестованість і переносимість коду;
- зв'язок одного екземпляра ViewModel з конкретним поданням, що забезпечує передбачувану поведінку системи.

Подання у цьому контексті виконує роль інструмента візуалізації та не містить логіки обробки зображень. При зміні параметрів контролю якості (наприклад, порогових значень різкості, контрасту, рівня шуму або результатів класифікації за ML-моделлю) View запитує актуальні дані в ViewModel. У той же час користувацькі дії (завантаження нового скану, повторний аналіз, виклик повторної класифікації) передаються у ViewModel у вигляді команд, які активують логіку Моделі.

У системі, що проєктується, архітектура загалом складається з трирівневої моделі (рисунок 2.1), яка дозволяє організовано виконувати операції з обробки зображень, аналізу метаданих сканів та взаємодії з базою даних [6-12].

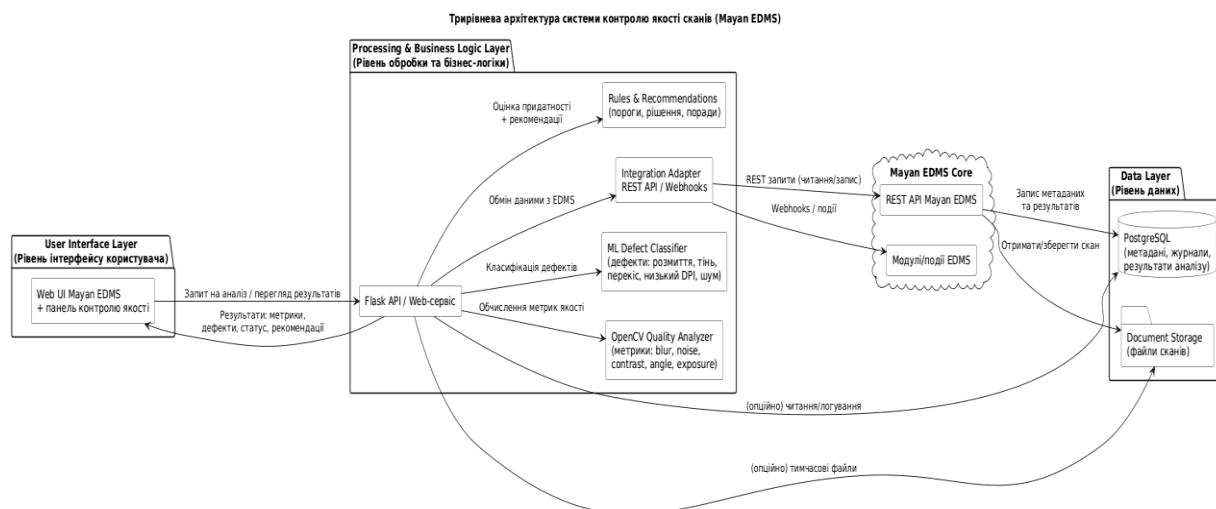


Рисунок 2.1 – трирівнева модель архітектури

Архітектура системи складається з трьох основних рівнів [33-36]:

1. User Interface Layer (Графічний інтерфейс користувача)
Реалізований у вигляді веб-інтерфейсу Mayan EDMS із додатковими інтегрованими модулями для відображення результатів контролю якості. Цей рівень забезпечує зручний доступ користувачів до функцій аналізу сканів, перегляду метрик якості та повідомлень про виявлені недоліки.

2. Processing & Business Logic Layer (Рівень обробки та бізнес-логіки)
Містить модулі аналізу зображень, що використовують бібліотеку OpenCV, та моделі машинного навчання, призначені для оцінювання якості документів, виявлення дефектів, а також класифікації типів помилок. Тут також реалізовано логіку взаємодії з внутрішніми механізмами Mayan EDMS через REST-API або внутрішні сервіси.

3. Data Layer (Рівень даних)

Відповідає за зберігання сканованих документів, їхніх метаданих та результатів аналізу. У випадку з Mayan EDMS це включає репозиторій файлів та реляційну

базу даних (за замовчуванням — PostgreSQL), де зберігається інформація про об'єкти системи, журнали обробки та результати класифікації.

Поділ на ці три рівні підвищує гнучкість програмної системи, спрощує її тестування, дозволяє удосконалювати модулі контролю якості без зміни інтерфейсу, а також забезпечує можливість подальшого розширення системи — наприклад, додавання нових алгоритмів машинного навчання або спеціалізованих модулів аналізу окремих типів документів [13-20].

2.2 Вибір системи керування базами даних

База даних у контексті системи контролю якості сканів є центральним компонентом, що забезпечує збереження документів, метаданих, параметрів аналізу, результатів обробки зображень та інформації щодо виконаних операцій у Maya EDMS. Важливо, щоб вибрана система керування базами даних (СКБД) гарантувала цілісність, стабільність та високу продуктивність при роботі з великими масивами файлових документів та аналітичними даними.

У загальному розумінні база даних — це впорядкований набір взаємопов'язаних даних, структурованих таким чином, щоб забезпечити їх збереження, швидкий пошук і доступ багатьох користувачів. До ключових характеристик якісної бази даних належать:

- повнота та відсутність надмірності;
- оптимальна структура, що полегшує пошук і модифікацію даних;
- актуальність і можливість швидких оновлень;
- зручність використання і підтримка ефективних запитів.

СКБД забезпечує централізоване управління даними та включає механізми збереження, маніпулювання, захисту, резервного копіювання та відновлення після збоїв.

Функціональні можливості сучасних СКБД

Система керування базами даних повинна забезпечувати:

1. Управління зберіганням даних у зовнішній пам'яті;

2. Буферизацію та кешування даних у оперативній пам'яті;
3. Захист від збоїв, включно з механізмами транзакцій, журналювання, резервного копіювання;
4. Підтримку мов визначення та маніпулювання даними (DDL, DML).

Основні компоненти типової СКБД включають ядро управління даними, модуль обробки запитів, виконавчу підсистему та набір сервісних утиліт для адміністрування.

Типи систем керування базами даних

1. Файл-серверні системи

У таких системах файли бази даних зберігаються на сервері, а обробка запитів виконується на клієнтських машинах. Попри низьке навантаження на сервер, їхні недоліки включають:

- високий трафік у локальній мережі;
- слабку відмовостійкість;
- низький рівень безпеки [37-45];
- складність централізованого адміністрування.

Тому цей підхід вважається застарілим для сучасних веб-систем.

2. Клієнт-серверні СКБД

У цьому випадку база даних та система керування розміщені на одному сервері, а обробка SQL-запитів централізована. Переваги підходу:

- висока продуктивність;
- можливість централізованого адміністрування;
- підвищена безпека [37-45];
- менше навантаження на мережу;
- відмовостійкість.

Недоліки: підвищені вимоги до серверних ресурсів.

Цей тип є стандартом для корпоративних систем, зокрема й для Mayan EDMS.

3. Вбудовані СКБД

Використовуються як частина окремих програм, призначені для локальних даних. У випадку системи контролю сканів та інтеграції з Maya EDMS не є доцільними.

4. Системи з відкладеним або безпосереднім записом

Ці стратегії визначають, коли саме змінені дані записуються у зовнішню пам'ять. Вони впливають на швидкість, стійкість до збоїв і поведінку транзакцій. Для серверних навантажень зазвичай застосовуються змішані або журнальні механізми, що забезпечують баланс продуктивності й надійності.

NoSQL-підхід та його особливості

NoSQL-сховища передбачають відсутність строгої структурованої реляційної моделі та використовують різні підходи до зберігання: документи, ключ-значення, графи, колонки. Переваги NoSQL:

- висока швидкодія;
- гнучка схема даних;
- легке горизонтальне масштабування;
- ефективна робота з вкладеними структурами (JSON/XML);
- автоматизована реплікація та кластери.

Документоорієнтовані бази на кшталт MongoDB добре підходять для великих масивів неструктурованих даних, зокрема звітів аналізу зображень, метрик та вкладених об'єктів.

Однак слід враховувати, що Maya EDMS побудована як система з суворою транзакційною логікою, що використовує ORM Django та потребує ACID-узгодженості. Це робить реляційну модель більш доцільною.

Обґрунтування вибору СКБД для системи на базі Maya EDMS

Система Maya EDMS офіційно використовує PostgreSQL як основну СКБД, що забезпечує:

- повну підтримку транзакцій (ACID);
- розширені можливості індексації та оптимізації запитів;
- високий рівень безпеки;

- масштабованість для великих масивів документів;
- підтримку JSON-полів та роботи зі складними метаданими;
- сумісність з ORM Django.

Для модулів контролю якості сканів, що інтегруються у Mayan EDMS, саме PostgreSQL є оптимальним рішенням, оскільки дозволяє безпечно зберігати:

- параметри сканів (роздільну здатність, розмір, формат);
- результати аналізу алгоритмами OpenCV;
- оцінки моделей машинного навчання (класифікація дефектів, ймовірності);
- журнали обробки;
- службову інформацію системи.

Таким чином, з огляду на вимоги до надійності, транзакційності та підтримки великого обсягу документів, для розроблюваної системи було обрано реляційну СКБД PostgreSQL, як таку, що нативно інтегрується з Mayan EDMS і забезпечує оптимальність роботи модулів контролю якості сканів.

2.3 Побудова схеми бази даних

Для побудови системи контролю якості сканів у Mayan EDMS необхідна високоефективна база даних, здатна зберігати як службову інформацію системи електронного документообігу, так і результати аналізу зображень, виконаних засобами OpenCV та алгоритмами машинного навчання. Оскільки Mayan EDMS використовує реляційний підхід до моделювання даних та інтегрується через ORM Django, архітектура бази даних проєктованої системи повинна відповідати реляційній моделі та підтримувати ACID-властивості.

Особливості СКБД, що використовується в системі

У стандартній конфігурації Mayan EDMS застосовується PostgreSQL — реляційна система керування базами даних, яка характеризується:

1. підтримкою транзакційності та цілісності даних;
2. високою продуктивністю для аналітичних та пошукових запитів;

3. гнучкою системою індексів, включно з GIN та GiST;
4. можливістю зберігання структурованих і напівструктурованих даних (JSONB);
5. високим рівнем безпеки;
6. масштабованістю та стійкістю до навантажень.

Ці характеристики роблять PostgreSQL придатним для систем, де обробляється велика кількість документоорієнтованих даних і метаданих, а також для зберігання результатів аналізу зображень.

Структура бази даних для модуля контролю якості сканів

Під час розробки модуля контролю якості були визначені основні сутності предметної області, які потребують відображення у базі даних. Ключовими з них є:

- Document — основна сутність, що описує документ у системі Mayan EDMS;
- ScanQualityResult — результати оцінки якості скану;
- QualityMetrics — детальні метрики з OpenCV (різкість, шум, контраст, перекося тощо);
- MLClassificationResult — результати класифікації дефектів, отримані моделлю машинного навчання;
- ProcessingLog — журнал виконаних операцій обробки;
- UserAction — дії користувачів, пов'язані з повторним аналізом або виправленням документів.

Основні таблиці та їх призначення

1. Document

Містить базову інформацію про документ, імпортований у Mayan EDMS. Поля можуть включати: ідентифікатор, дату імпорту, шлях до файлу, тип, статус.

2. ScanQualityResult

Містить загальну оцінку якості скану:

- нормований індекс якості;
- перелік виявлених дефектів;

- рішення («придатно», «потребує пересканування»).

Зв'язок: один-до-багатьох з Document (один документ — багато результатів аналізу, якщо він перевірявся кілька разів).

3. QualityMetrics

Таблиця містить окремі числові параметри, отримані під час аналізу зображення OpenCV:

- різкість (Laplacian Variance);
- рівень шуму;
- контрастність;
- кут нахилу (deskew angle);
- коефіцієнт засвітлення/затемнення.

Зв'язок: один-до-одного з ScanQualityResult.

4. MLClassificationResult

Зберігає дані, отримані від моделі машинного навчання:

- тип дефекту (розмиття, тінь, перекіс, низький DPI, шум);
- ймовірність;
- модель/версія моделі.

Зв'язок: один-до-багатьох з ScanQualityResult.

5. ProcessingLog

Містить службову історію обробки:

- час запуску аналізу;
- час виконання;
- помилки;
- інформацію про навантаження;
- версію алгоритмів.

Зв'язок: багато-до-одного з Document.

6. UserAction

Фіксує взаємодію користувача з модулем якості:

- хто повторив аналіз;
- хто підтвердив якість;

- хто відправив документ на пересканування.

Зв'язок: багато-до-одного з Document.

Міжтабличні зв'язки

У базі даних використовуються такі типи зв'язків:

1. Один-до-одного

Наприклад: ScanQualityResult → QualityMetrics.

Один набір метрик відповідає одному результату аналізу.

2. Один-до-багатьох

Документ може мати кілька результатів аналізу або багато записів журналу обробки.

3. Багато-до-багатьох

Використовується рідко, однак може бути реалізованим через проміжні таблиці, якщо в майбутньому виникне потреба пов'язувати документи з кількома наборами класифікаторів або моделями ML.

У проєктованій системі зовнішні ключі відіграють ключову роль, оскільки саме вони забезпечують логічну цілісність між документами, результатами аналізу якості та метриками. Таблиці поділяються на:

- головні таблиці (наприклад, Document), що містять первинні ключі;
- підпорядковані таблиці, які містять зовнішні ключі та розширюють інформацію про результати обробки.

Такий підхід забезпечує:

- структурованість даних;
- можливість масштабування;
- гнучкість при додаванні нових типів метрик та ML-моделей;
- надійність і керованість системи.

2.4 Конструювання програмної системи

Основну увагу приділено вибору архітектурних рішень, структурі програмних компонентів, організації взаємодії між модулями та реалізації

ключової функціональності системи. Розробка здійснювалась із урахуванням вимог масштабованості, модульності та можливості подальшого розширення, що забезпечує ефективне використання системи в умовах реальної експлуатації.

2.4.1 Обґрунтування вибору технологій розробки

Для розробки представленого програмного проєкту обрано мову програмування Python та її веб-фреймворки. Python – це високорівнева, динамічна, об'єктно-орієнтована мова програмування, яка дозволяє ефективно розробляти як настільні, так і веб-додатки.

Вибір Python зумовлений такими факторами:

1. Простота та зрозумілість синтаксису, що пришвидшує розробку та спрощує підтримку коду.
2. Підтримка об'єктно-орієнтованого програмування, що дозволяє створювати модульний та повторно використовуваний код.
3. Велика кількість бібліотек та фреймворків, зокрема для веб-розробки, обробки даних та роботи з базами даних.
4. Можливість кросплатформеної розробки, що забезпечує запуск проєкту на Windows, Linux та macOS.
5. Потужні засоби для роботи з базами даних, включно з MS SQL Server, PostgreSQL, MySQL та SQLite.
6. Активна спільнота, що дозволяє швидко отримувати підтримку та використовувати готові рішення.

Для веб-розробки обрано фреймворки Django (для великих проєктів) або Flask (для легких та гнучких веб-додатків), що забезпечують простий доступ до серверної логіки, управління базою даних та формування клієнтського інтерфейсу.

2.4.2 Побудова UML-діаграм класів

Побудова UML-діаграм класів є ключовим етапом проєктування системи контролю якості сканованих документів, оскільки дає змогу формально описати структуру програмного продукту, визначити основні компоненти, їхні атрибути, методи та взаємозв'язки. Це спрощує реалізацію, модульне тестування й подальше розширення функціональності (зокрема додавання нових метрик якості або нових моделей машинного навчання), а також забезпечує прозорість архітектурних рішень для інтеграції з Mayan EDMS.

З огляду на те, що розроблена система реалізована у вигляді web-сервісу (Flask) із використанням OpenCV та модуля машинного навчання, основні класи доцільно згрупувати за функціональними напрямками: web-рівень, обробка зображень, ML-класифікація, інтеграція з EDMS, візуалізація та допоміжні структури даних.

1) Класи web-рівня (UI / контролери)

FlaskApp (app.py) — клас/модуль, який реалізує точки входу веб-застосунку та оркестрацію компонентів системи. На UML-діаграмі він виступає як центральний вузол, що координує послідовність: *завантаження* → *аналіз* → *класифікація* → *збереження* → *відображення*.

Типові методи/обробники:

- index() — відображення головної сторінки завантаження;
- analyze() — прийом файлу, запуск аналізу якості та ML-класифікації;
- history() / history_item() — перегляд історії обробок і результатів;
- get_plot() — відображення згенерованих графіків (за наявності).

2) Класи рівня обробки зображень (OpenCV)

ScanQualityChecker (quality_checker.py) — основний клас оцінювання якості скану. Він інкапсулює алгоритми комп'ютерного зору й повертає метрики якості, що використовуються надалі як для прийняття рішення “придатний/непридатний”, так і як ознаки для ML-модуля.

Ключові методи:

- `check_quality(image_path)` — агрегований виклик, що повертає набір метрик;

- `laplacian_variance()` — оцінка різкості;
- `noise_level()` — оцінка рівня шуму;
- `contrast_rms()` — оцінка контрастності;
- `deskew_angle()` — визначення кута нахилу (deskew angle);
- `illumination_coeff()` — коефіцієнт засвітлення/затемнення;
- `_load_gray()` — допоміжне завантаження зображення у відтінках сірого.

3) Класи рівня машинного навчання (ML)

`ScanDefectModel (ml_model.py)` — компонент, який виконує класифікацію типів дефектів скану та повертає:

- тип дефекту (розмиття, тінь, перекіс, низький DPI, шум),
- ймовірність,
- ідентифікатор моделі / версію (model name / version).

Ключовий метод:

- `predict_defects(metrics)` — приймає метрики OpenCV як вхідні ознаки та формує структурований вихід.

За потреби в UML-діаграмі доцільно показати окремі структури даних:

- `MLDefect` (або аналогічна модель) — сутність “дефект + ймовірність”.

4) Класи інтеграції з Mayan EDMS

`MayanEDMSMock (mayan_mock.py)` — імітаційний інтеграційний модуль, який моделює поведінку EDMS у контексті отримання документів та збереження результатів аналізу (у вигляді метаданих/JSON).

Ключові методи:

- `get_new_documents()` — отримання переліку нових документів для аналізу;
- `save_quality_result(document_path, result_text, metrics, ml_output)` — збереження результатів перевірки якості та ML-висновків.

У поясненні до UML варто окремо підкреслити, що у промисловій реалізації цей модуль може бути замінений на адаптер інтеграції з реальним REST API

Mayan EDMS, не змінюючи бізнес-логіку обробки (принцип інкапсуляції та заміни реалізації).

5) Класи візуалізації та допоміжні компоненти

Visualization (visualization.py) — модуль/клас, що реалізує побудову графіків метрик якості для візуального аналізу.

Ключовий метод:

- plot_quality_metrics(metrics, image_name) — формує графік за основними показниками (різкість, шум, контрастність, кут нахилу, коефіцієнт засвітлення/затемнення).

Додатково (за необхідності відображення в UML) можуть бути введені допоміжні структури:

- QualityMetrics — узагальнена модель метрик якості (може відповідати словнику metrics);
- MLResult — модель результату ML (модель/версія + список дефектів).

Взаємозв'язки між класами

Зв'язки між компонентами відображають логіку системи контролю якості сканів:

- FlaskApp використовує ScanQualityChecker для обчислення метрик якості;
- FlaskApp викликає ScanDefectModel для визначення дефектів та ймовірностей на основі метрик;
- FlaskApp звертається до MayanEDSMock (або майбутнього EDMS-адаптера) для збереження результатів аналізу як метаданих документа;
- FlaskApp використовує Visualization для побудови графіків і відображення їх у web-інтерфейсі;
- ScanDefectModel логічно залежить від вихідних метрик ScanQualityChecker, оскільки вони виступають ознаками для ML-класифікації.

UML-діаграми класів рисунок 2.2 демонструють ключові програмні компоненти системи, їхні атрибути, методи та основні зв'язки. Це дає змогу цілісно зрозуміти структуру реалізації, обґрунтувати вибір модульної архітектури та спланувати подальший розвиток системи (наприклад, додавання нових метрик

якості або підключення іншої версії ML-моделі без змін у web-рівні та інтеграції з EDMS).

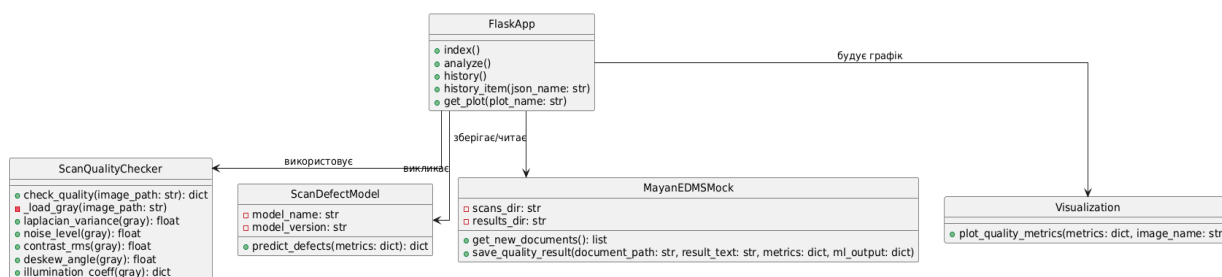


Рисунок 2.2 – діаграма класів

2.4.3 Вибір мови програмування

Основною мовою програмування обрано Python [44,45]. Python – це сучасна, високопродуктивна та безпечна для типів мова, яка підтримує об’єктно-орієнтоване програмування та активно використовується для веб-розробки, автоматизації, наукових обчислень і обробки даних.

Основні переваги Python для даного проєкту:

1. Об’єктно-орієнтоване програмування – підтримка класів, наслідування, інкапсуляції та поліморфізму.
2. Чітко визначені базові типи даних та можливість створення власних.
3. Автоматичне керування пам’яттю – механізм «збирання сміття», що знімає з розробника обов’язок вручну звільняти пам’ять.
4. Підтримка роботи з базами даних через ORM (Object-Relational Mapping), що спрощує доступ до MS SQL Server та інших СУБД.
5. Великий вибір бібліотек для роботи з веб-технологіями, JSON, XML, RESTful API та іншими сучасними форматами даних.
6. Можливість інтеграції з JavaScript-фреймворками, що забезпечує динамічність клієнтської частини.

Python обрався через свою гнучкість, широкую підтримку веб-фреймворків та інтеграцію з сучасними технологіями для побудови клієнт-серверних систем.

2.4.4 Обґрунтування вибору технологічного засобу реалізації веб-проєкту

Основним технологічним засобом реалізації веб-проєкту є Python разом із фреймворком Django (або Flask). Ці інструменти дозволяють створювати серверну частину програми, керувати базою даних та формувати динамічний веб-інтерфейс.

Вибір обґрунтовано наступними перевагами:

1. Об'єктно-орієнтований підхід – повторне використання коду, інкапсуляція та зручна структура проєкту.
 2. Простота та швидкість розробки – завдяки зрозумілому синтаксису Python та наявності готових модулів.
 3. Підтримка сучасних веб-технологій – HTTP, REST API, WebSockets, інтеграція з JavaScript-фреймворками.
 4. Ефективна робота з базами даних – через ORM легко реалізувати доступ до MS SQL Server та забезпечити цілісність даних.
 5. Безпека та надійність – механізми автентифікації, авторизації, захисту від SQL-ін'єкцій та CSRF-атак.
 6. Масштабованість – можливість роботи з великими обсягами даних та навантаженням.
 7. Кросплатформеність – запуск на Windows, Linux та macOS.
- Python і його фреймворки забезпечують швидку розробку веб-додатків, легке інтегрування з базами даних, високу продуктивність та зручність підтримки проєкту.

2.4.5 Механізм опрацювання даних та доступ до бази даних

У представленому програмному проєкті для обробки даних та доступу до бази даних використано ORM (Object-Relational Mapping) фреймворк Flask у поєднанні з мовою Python.

Обрана технологія обґрунтована такими факторами:

1. Абстрагування роботи з базою даних – ORM дозволяє оперувати даними у вигляді об'єктів Python замість прямого написання SQL-запитів. Це знижує ймовірність синтаксичних помилок та помилок посилання на поля бази даних.

2. Уніфікація типів даних – ORM автоматично перетворює типи даних Python у відповідні типи бази даних і навпаки, що спрощує роботу з різними джерелами даних та форматом зберігання.

3. Зручність роботи з колекціями та запитами – фреймворк надає простий інтерфейс для створення запитів, фільтрації, сортування та агрегації даних. Наприклад, запити до бази можна виконувати так, як ніби працюєш із списками об'єктів Python.

4. Підтримка різних типів баз даних – Django ORM дозволяє працювати з MS SQL Server, PostgreSQL, MySQL, SQLite та іншими СУБД без зміни суттєвої частини коду.

5. Безпека та цілісність даних – ORM забезпечує захист від SQL-ін'єкцій та автоматично керує транзакціями, що підвищує надійність роботи програми.

6. Простота інтеграції з веб-фреймворком – моделі Django пов'язані з веб-компонентами проєкту, що дозволяє безпосередньо підключати дані до клієнтської частини через шаблони або API.

Принцип

роботи:

У Django кожна таблиця бази даних представлена як клас-модель Python. Атрибути класу відповідають полям таблиці. Для виконання запитів використовується інтерфейс QuerySet, який дозволяє:

- отримувати об'єкти бази даних;
- фільтрувати їх за умовами;
- сортувати та агрегувати дані;
- оновлювати та видаляти записи.

Таким чином, ORM в Python забезпечує простий, безпечний та ефективний механізм роботи з базою даних, що дозволяє розробнику зосередитись на логіці програми, а не на низькорівневих деталях SQL-запитів.

2.4.6 Обґрунтування вибору середовища розробки

Для реалізації представленого веб-проєкту основним інструментальним засобом обрано інтегроване середовище розробки PyCharm. Вибір зумовлений високою гнучкістю та потужними можливостями цього середовища, що дозволяє підвищити ефективність роботи розробника та якість програмного продукту.

До основних можливостей PyCharm належать:

1. Зручне редагування коду – підтримка підсвічування синтаксису, автодоповнення, рефакторинг та форматування коду на Python спрощує процес розробки та покращує якість коду.

2. Інтеграція з веб-фреймворками – PyCharm повністю підтримує Django, що дозволяє швидко створювати веб-сторінки, моделі баз даних та форми без необхідності ручного написання складних шаблонів.

3. Автоматичне керування пакетами та середовищами – середовище дозволяє створювати та керувати віртуальними середовищами Python, встановлювати необхідні бібліотеки та підтримує інтеграцію з системами керування базами даних.

4. Вбудований веб-сервер – для тестування та відлагодження веб-додатків можна запускати програму безпосередньо із середовища розробки.

5. Підтримка багатьох мов та технологій – PyCharm дозволяє одночасно працювати з Python, HTML, CSS, JavaScript та SQL у межах одного проєкту, що зручно для розробки сучасних веб-додатків.

6. Підвищення швидкості розробки – автоматична генерація шаблонів проєктів, моделей та форм скорочує час розробки та спрощує створення коду.

7. Відлагодження та тестування – інтегрований відлагоджувач дозволяє крок за кроком виконувати код, переглядати змінні та тестувати роботу веб-додатка. PyCharm також підтримує юніт-тести та автоматичне тестування Django-проєктів.

У рамках представленого проєкту основна робота виконувалася у PyCharm для розробки веб-додатку на Django. Найбільш корисними можливостями

середовища стали: швидке редагування коду Python та HTML, вбудований веб-сервер для тестування програми та інтеграція з базою даних для роботи з моделями Django.

2.4.7 Функціональні можливості графічного інтерфейсу користувача

Користувачі розроблюваної системи взаємодіють з нею за допомогою Web-інтерфейсу, інтегрованого в стандартне середовище Maya EDMS. Графічний інтерфейс реалізований у вигляді набору Web-форм та інформаційних панелей, що забезпечують зручний доступ до результатів аналізу якості сканованих документів, керування параметрами перевірки та перегляду історії обробки.

Основною метою графічного інтерфейсу є надання користувачеві наочної, структурованої та зрозумілої інформації про стан якості документів, а також мінімізація ручних операцій під час контролю сканів.

Нижче наведено опис основних Web-форм та їх функціональних можливостей.

1. Форма «Перелік документів»

Форма призначена для перегляду списку документів, завантажених до системи Maya EDMS, разом із результатами автоматизованої перевірки якості сканів.

Функціональні можливості:

- відображення переліку документів із зазначенням їх статусу якості (придатний / потребує повторного сканування);
- індикація результатів аналізу у вигляді маркерів або піктограм;
- фільтрація документів за датою завантаження, користувачем, статусом якості;
- сортування за основними параметрами (дата, назва, рівень якості).

2. Форма «Результати аналізу якості скану»

Дана Web-форма призначена для детального перегляду результатів аналізу конкретного документа, отриманих за допомогою алгоритмів OpenCV та моделей машинного навчання.

Форма містить:

- показники різкості зображення;
- оцінку рівня шуму;
- визначений кут нахилу сторінки;
- загальну інтегральну оцінку якості;
- рекомендацію системи щодо подальшої обробки документа.

Інформація подається у вигляді таблиць, текстових пояснень та, за необхідності, графічних індикаторів.

3. Форма «Візуальний перегляд скану»

Форма забезпечує візуальну перевірку документа разом із накладеними результатами аналізу.

Функціональні можливості:

- перегляд оригінального сканованого зображення;
- відображення зон, у яких виявлено дефекти (розмиття, перекіс, надмірний шум);
- масштабування та прокручування зображення;
- порівняння результатів до та після автоматичної корекції (за наявності).

4. Форма «Історія перевірок»

Форма призначена для зберігання та аналізу історії перевірок якості документів.

Можливості форми:

- перегляд усіх попередніх перевірок для конкретного документа;
- відображення змін показників якості у часі;
- фільтрація історичних даних за датою, користувачем або типом документа;
- аудит прийнятих рішень (підтвердження якості або повторне сканування).

5. Форма «Налаштування параметрів контролю якості»

Доступна лише користувачам з адміністративними правами. Призначена для керування параметрами алгоритмів аналізу.

Функціональні можливості:

- встановлення порогових значень різкості, шуму та нахилу;
- увімкнення або вимкнення окремих етапів аналізу;
- вибір та активація моделі машинного навчання;
- збереження та застосування профілів налаштувань.

6. Інформаційна панель (Dashboard)

Інформаційна панель надає узагальнену статистику роботи системи.

Вона відображає:

- загальну кількість перевірених документів;
- відсоток документів із незадовільною якістю;
- середні значення показників якості;
- динаміку змін якості сканування у часі.

Таким чином, графічний інтерфейс користувача реалізує повний цикл взаємодії із системою контролю якості сканів: від перегляду документів до детального аналізу результатів та адміністрування параметрів перевірки. Використання Web-форм забезпечує кросплатформеність, інтуїтивність та інтеграцію в існуюче середовище Maya EDMS, що підвищує ефективність документообігу та зменшує кількість помилок, пов'язаних із неякісним скануванням.

3. ТЕСТУВАННЯ, ВПРОВАДЖЕННЯ ТА ПІДТРИМКА

Розділ 3 присвячено ключовим етапам життєвого циклу розробленої програмної системи — тестуванню, впровадженню та забезпеченню її подальшої підтримки. Ці етапи мають вирішальне значення для забезпечення якості, надійності та практичної придатності системи в умовах реального використання. У розділі детально висвітлено процес тестування, який включав перевірку функціональних і нефункціональних характеристик системи, розробку тестових сценаріїв, а також верифікацію відповідності вимогам. Окрім того, розглянуто питання розгортання системи, визначення необхідних апаратно-програмних вимог та забезпечення її готовності до інтеграції зі середовищем електронного документообігу. Результати, отримані в рамках цього розділу, підтверджують ефективність розроблених рішень та спроможність системи до експлуатації в реальних умовах.

3.1 Тестування програмної системи

Тестування програмної системи є систематичним процесом верифікації та валідації програмного продукту, спрямованим на виявлення дефектів, перевірку відповідності реалізації початковим вимогам, а також оцінювання рівня якості програмного забезпечення. Для інформаційних систем класу EDMS особливо важливими є надійність, стабільність, коректність обробки даних, точність автоматизованого аналізу та безперервність роботи в умовах реального навантаження.

У межах даної магістерської роботи тестування проводилося для програмної системи контролю якості сканованих документів, інтегрованої з системою електронного документообігу Mayan EDMS. Реалізація системи виконана з використанням мови програмування Python, бібліотеки комп'ютерного зору OpenCV, елементів машинного навчання та веб-фреймворку Flask. Тестування

здійснювалося у середовищі операційної системи Windows із використанням локального веб-сервера та сучасних браузерів [21-25].

Основною метою тестування було підтвердження коректності роботи алгоритмів аналізу якості сканів, перевірка стабільності веб-інтерфейсу, а також валідація процесу збереження результатів аналізу у вигляді метаданих документа, що імітує інтеграцію з Mayan EDMS.

Особлива увага під час тестування приділялася перевірці таких аспектів:

- коректності обчислення показників якості сканованих документів, зокрема різкості (Laplacian Variance), рівня шуму, контрастності, кута нахилу сторінки та коефіцієнта засвітлення або затемнення;
- стабільності роботи алгоритмів комп'ютерного зору для зображень різної якості, роздільної здатності та формату;
- правильності функціонування модуля машинного навчання, який визначає типи дефектів скану (розмиття, шум, перекіс, тіні, низький DPI) та відповідні ймовірності;
- коректності інтеграції між компонентами системи: веб-інтерфейсом Flask, модулем аналізу зображень, модулем машинного навчання та модулем імітації Mayan EDMS;
- стабільності роботи системи при послідовній обробці декількох сканованих документів.

Тестування охоплювало як функціональні, так і нефункціональні характеристики системи.

Функціональне тестування включало перевірку основних сценаріїв використання системи:

- головної веб-сторінки системи, яка забезпечує завантаження сканованого документа користувачем;
- модуля автоматичного аналізу якості скану після завантаження файлу;
- сторінки результатів аналізу, на якій відображаються обчислені метрики якості, виявлені дефекти та їх ймовірності;

- модуля візуалізації, що будує графіки показників якості сканованого документа;
- модуля збереження результатів аналізу у форматі JSON, який імітує збереження метаданих у системі електронного документообігу Mayan EDMS;
- сторінки перегляду історії оброблених документів та результатів перевірки.

Нефункціональне тестування включало:

- тестування продуктивності, зокрема оцінювання часу аналізу одного сканованого документа та швидкості відгуку веб-інтерфейсу;
- тестування стійкості системи при обробці зображень низької якості, з дефектами або нетиповими параметрами;
- перевірку надійності збереження результатів аналізу та коректності формування структури вихідних даних;
- перевірку стабільності роботи системи під час багаторазового послідовного використання без перезавантаження сервера.

Тестування проводилося протягом усього життєвого циклу розробки програмної системи, що дозволило виявляти та усувати помилки на ранніх етапах, підвищити стабільність реалізації та забезпечити відповідність функціональних можливостей системи поставленим вимогам. Результати тестування фіксувалися у вигляді логів роботи системи, збережених файлів результатів аналізу та візуалізацій показників якості, що забезпечує можливість подальшого аналізу та вдосконалення програмного продукту.[26-32]

Модуль ініціалізації та запуску аналізу – перевірялося коректне завантаження веб-інтерфейсу Flask, ініціалізація компонентів OpenCV та моделі машинного навчання, а також готовність системи до прийому сканованих документів (рисунок 3.1)

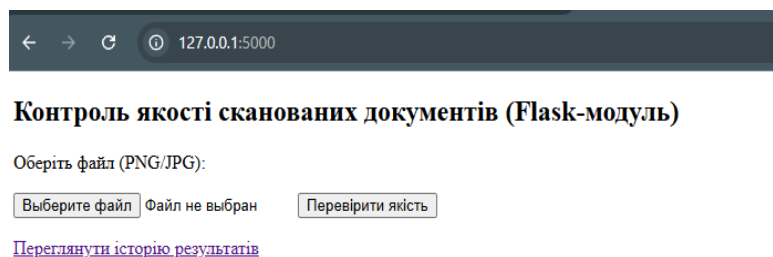


Рисунок 3.1 – головна сторінка

Модуль завантаження документів – перевірялася можливість додавання сканів документів у систему (через веб-інтерфейс або імітацію Mayan EDMS), коректність обробки форматів зображень (PNG, JPG), а також ініціювання процесу аналізу після надходження файлу (рисунок 3.2).

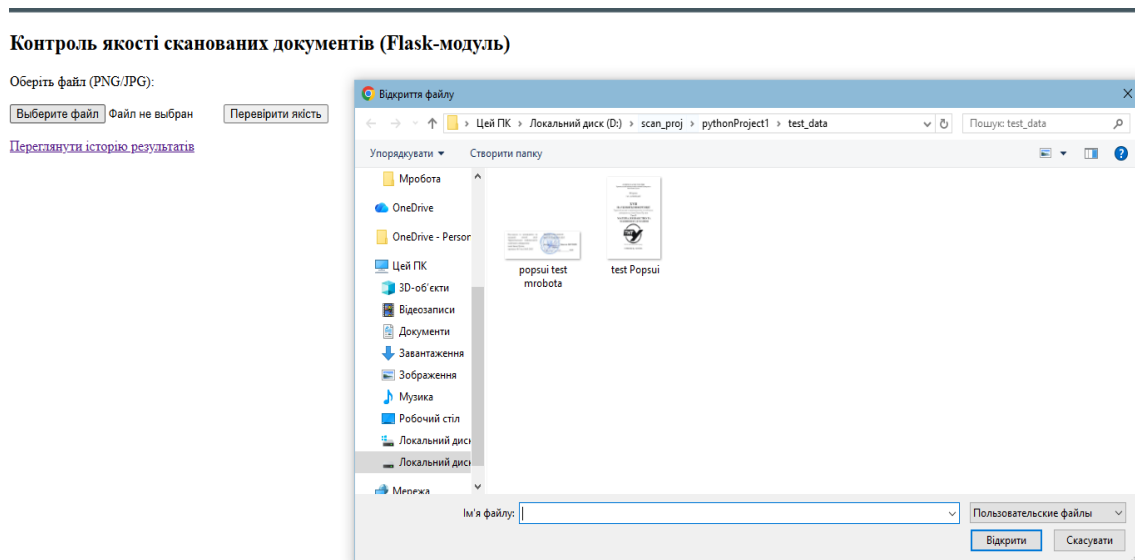


Рисунок 3.2 - завантаження документів

Модуль аналізу якості зображень – тестувалася правильність обчислення основних метрик якості скану: різкості (Laplacian Variance), рівня шуму, контрастності, кута нахилу (deskew angle) та коефіцієнта засвітлення/затемнення. Перевірялося, що значення метрик відповідають очікуваним для тестових зображень різної якості (рисунок 3.3).

Результат аналізу: 20251216_071343_test Popsui.jpg

Підсумок

BAD (засвітлення/затемнення)

OpenCV-метрики

- Різкість (Laplacian Var): 1558.90
- Шум: 12.64
- Контрастність: 52.41
- Кут нахилу (deskew): 0.00°
- Засвітлення/затемнення (ratio): 0.969

Рисунок 3.3 - аналіз якості зображень

Модуль машинного навчання – перевірялася робота моделі класифікації дефектів, зокрема коректне визначення типів дефектів (розмиття, тінь, перекіс, низький DPI, шум), розрахунок ймовірності кожного дефекту та ідентифікація версії моделі, що використовувалася для аналізу (рисунок 3.4)

ML-результат

Модель: DefectClassifier v1.0.0

- тінь — $p=1.000$

Графік

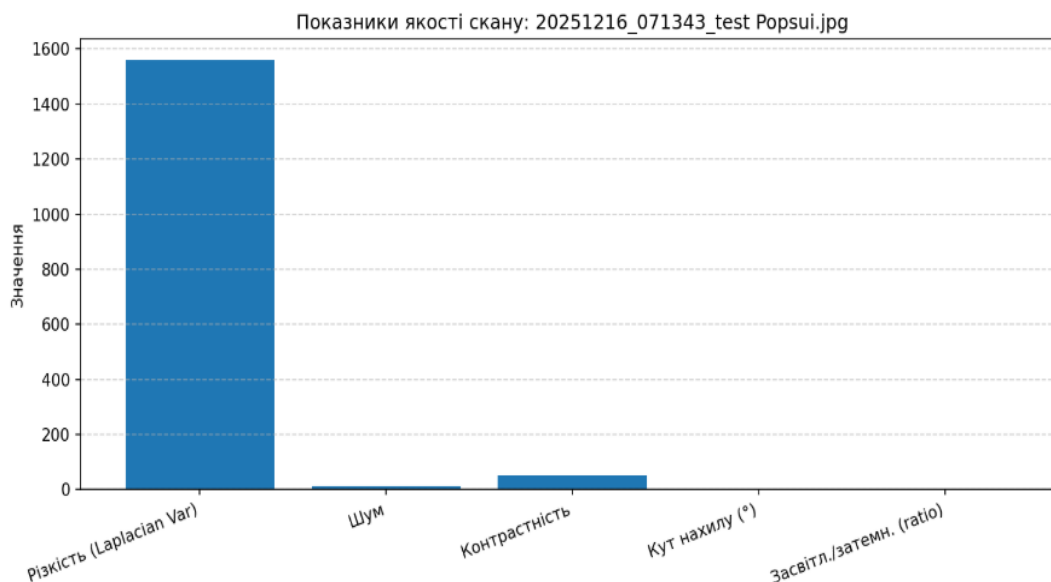


Рисунок 3.4 – аналіз машинного навчання

4. Нефункціональне тестування

Окрім функціональних перевірок, було виконано нефункціональне тестування, яке включало:

- тестування продуктивності (Performance Testing) — оцінювання часу аналізу одного сканованого документа та швидкості відгуку веб-інтерфейсу;
- стрес-тестування (Stress Testing) — перевірку стабільності системи при багаторазовому послідовному аналізі великої кількості сканів;
- тестування стійкості (Robustness Testing) — аналіз поведінки системи у разі некоректних дій користувача або неякісних вхідних даних;
- базове тестування безпеки (Security Testing) — перевірку коректності обробки користувацьких запитів, обмеження типів завантажуваних файлів та ізоляції серверної логіки.

3.1.1 Види та план тестування [26-32]

Процес тестування розробленої програмної системи організовано відповідно до принципів багаторівневої верифікації програмного забезпечення, що передбачають поетапну перевірку працездатності окремих програмних модулів та системи в цілому. Такий підхід дозволяє своєчасно виявляти дефекти реалізації, забезпечує відповідність програмного продукту початковим вимогам та підвищує загальний рівень якості програмного забезпечення.

У межах даної магістерської роботи тестування виконувалося для системи контролю якості сканованих документів, інтегрованої з системою електронного документообігу Maya EDMS. Архітектура системи є модульною, що дало змогу чітко розмежувати рівні тестування та забезпечити ізольовану перевірку ключових компонентів.

У процесі дослідження було виділено такі основні види тестування.

1. Модульне тестування (Unit Testing)

Метою модульного тестування була перевірка коректності роботи окремих класів, функцій та алгоритмів без урахування зовнішніх залежностей. Основними об'єктами модульного тестування були:

- модуль аналізу якості зображень ScanQualityChecker, зокрема алгоритми обчислення різкості (Laplacian Variance), рівня шуму, контрастності, кута нахилу сторінки та коефіцієнта засвітлення або затемнення;
- модуль машинного навчання ScanDefectModel, який виконує класифікацію дефектів скану та визначає ймовірності їх наявності;
- допоміжні функції зчитування та попередньої обробки зображень;
- модуль збереження результатів аналізу MayanEDMSMock.

Для забезпечення ізоляції тестованих компонентів застосовувалися тестові зображення з наперед відомими характеристиками якості, що дозволяло перевіряти правильність обчислення метрик та стабільність роботи алгоритмів у контрольованих умовах.

2. Інтеграційне тестування (Integration Testing)

Інтеграційне тестування було спрямоване на перевірку коректності взаємодії між окремими модулями системи. Основна увага приділялася таким аспектам:

- інтеграції між модулем комп'ютерного зору (OpenCV) та модулем машинного навчання;
- передачі результатів аналізу зображень у модуль класифікації дефектів;
- взаємодії між серверною частиною Flask-застосунку та логікою аналізу якості сканів;
- коректності збереження отриманих результатів у форматі метаданих документа, що імітує роботу Mayan EDMS.

Під час інтеграційного тестування перевірялися сценарії повного циклу обробки документа: від завантаження скану через веб-інтерфейс до отримання підсумкового результату аналізу та його збереження.

3. Системне тестування (System Testing)

Системне тестування виконувалося з метою оцінювання роботи програмної системи як єдиного цілісного продукту. Основні сценарії системного тестування включали:

- завантаження сканованих документів різного формату та якості через веб-інтерфейс;
- автоматичний аналіз якості сканів із відображенням усіх обчислених метрик;
- визначення типів дефектів та їх ймовірностей за допомогою моделі машинного навчання;
- візуалізацію результатів аналізу у вигляді графіків;
- перегляд історії перевірених документів та результатів аналізу;
- обробку помилкових ситуацій, зокрема завантаження некоректних файлів або пошкоджених зображень.

Даний рівень тестування дозволив підтвердити відповідність реалізованої системи функціональним вимогам та сценаріям використання в умовах реальної експлуатації.

План тестування

План тестування було сформовано у вигляді структурованого набору тестових сценаріїв, який включав:

- класифікацію тестових об'єктів (алгоритми аналізу зображень, ML-модуль, веб-інтерфейс, модуль збереження даних);
- визначення критеріїв прийнятності для кожного програмного модуля;
- відповідність вимог системи до реалізованих тестів;
- послідовність виконання модульного, інтеграційного та системного тестування;
- фіксацію результатів тестування у вигляді логів, збережених файлів результатів аналізу та візуалізацій показників якості.

Реалізація зазначеного плану тестування забезпечила систематичний та контрольований процес оцінки якості програмної системи, що підтвердило її

готовність до впровадження та подальшого використання в середовищі електронного документообігу.

3.1.2 Розробка тестових сценаріїв

Розробка тестових сценаріїв є логічним продовженням етапу планування тестування та передбачає детальний опис дій користувача або системи, очікуваних результатів та критеріїв оцінювання коректності функціонування програмного продукту. Для розробленої системи контролю якості сканованих документів тестові сценарії формувалися на основі функціональних вимог, архітектурних рішень та особливостей інтеграції з системою електронного документообігу Mayan EDMS.

Тестові сценарії було структуровано відповідно до ключових функціональних модулів системи.

1. Сценарій завантаження сканованого документа

Мета: перевірка коректності завантаження сканованого документа через веб-інтерфейс системи.

Тестові кроки:

1. Відкрити головну сторінку веб-застосунку.
2. Обрати файл зображення підтримуваного формату (PNG, JPG).
3. Натиснути кнопку завантаження документа.
4. Дочекатися завершення процесу обробки.

Очікуваний результат:

файл успішно завантажується, передається до модуля аналізу якості та ініціюється автоматичний запуск алгоритмів перевірки скану.

2. Сценарій автоматичного аналізу якості скану

Мета: перевірка коректності роботи алгоритмів комп'ютерного зору для визначення показників якості сканованого документа.

Тестові кроки:

1. Завантажити сканований документ у систему.

2. Дочекатися завершення аналізу зображення.

3. Перевірити обчислення показників якості.

Очікуваний результат:

система коректно обчислює наступні метрики:

- різкість зображення (Laplacian Variance);
- рівень шуму;
- контрастність;
- кут нахилу сторінки (deskew angle);
- коефіцієнт засвітлення або затемнення.

Отримані значення відображаються у веб-інтерфейсі без помилок.

3. Сценарій визначення дефектів за допомогою моделі машинного навчання

Мета: перевірка роботи модуля машинного навчання для класифікації дефектів сканованих документів.

Тестові кроки:

1. Передати результати аналізу зображення до модуля машинного навчання.
2. Запустити процес класифікації дефектів.
3. Отримати результати прогнозування.

Очікуваний результат:

модель машинного навчання коректно визначає типи дефектів скану (розмиття, шум, перекіс, тіні, низька якість сканування) та повертає ймовірність наявності кожного дефекту разом із ідентифікатором моделі та її версією.

4. Сценарій візуалізації результатів аналізу

Мета: перевірка коректності візуального представлення результатів аналізу якості скану.

Тестові кроки:

1. Завершити процес аналізу документа.
2. Перейти до сторінки результатів.
3. Переглянути графіки показників якості.

Очікуваний результат:

система відображає графіки для основних метрик якості скану, що дозволяє наочно оцінити стан документа та виявлені дефекти.

5. Сценарій збереження результатів у Mayan EDMS

Мета: перевірка коректності збереження результатів аналізу у вигляді метаданих документа.

Тестові кроки:

1. Завершити аналіз якості скану.
2. Передати результати у модуль інтеграції з Mayan EDMS.
3. Зберегти результати аналізу у форматі структурованих даних.

Очікуваний результат:

результати аналізу (метрики якості, типи дефектів, ймовірності, версія моделі) успішно зберігаються у вигляді метаданих документа, що імітує роботу реальної системи електронного документообігу Mayan EDMS.

6. Сценарій перегляду історії перевірок

Мета: перевірка коректності збереження та повторного перегляду результатів аналізу раніше оброблених документів.

Тестові кроки:

1. Відкрити сторінку історії перевірок.
2. Обрати раніше оброблений документ.
3. Переглянути збережені результати аналізу.

Очікуваний результат:

система коректно відображає історію перевірених документів разом із повним набором збережених показників якості та результатів машинного навчання.

Загальні положення щодо тестових сценаріїв

- Для кожного тестового сценарію визначено початковий стан системи, послідовність дій, очікуваний результат та критерії прийнятності.
- Тестування виконувалося у веб-середовищі з використанням локального сервера Flask та сучасних веб-браузерів.

- Результати виконання тестових сценаріїв фіксувалися у вигляді логів роботи системи, збережених файлів результатів аналізу та візуалізацій, що забезпечує прозорість процесу тестування та можливість подальшого аналізу.

3.2 Розгортання програмної системи та системні вимоги

Розгортання програмної системи є важливим етапом її життєвого циклу, оскільки забезпечує підготовку робочого середовища для тестування, демонстрації та подальшої експлуатації програмного продукту. Для інформаційних систем класу EDMS, що виконують автоматизований аналіз якості сканованих документів, правильне налаштування програмного та апаратного середовища є критично важливим для забезпечення стабільної роботи, коректної обробки зображень та надійного збереження результатів аналізу.

Розробка та розгортання системи контролю якості сканів, інтегрованої з Maya EDMS, виконувалися з використанням наступних інструментів та технологій.

Середовище розробки

- PyCharm — основне середовище розробки для написання та налагодження коду мовою Python.
- Python 3.10+ — базова мова реалізації серверної логіки, алгоритмів комп'ютерного зору та машинного навчання.
- Flask — веб-фреймворк для реалізації серверної частини та користувацького інтерфейсу системи.
- OpenCV — бібліотека комп'ютерного зору для аналізу якості зображень.
- NumPy, Matplotlib — допоміжні бібліотеки для обчислень та візуалізації результатів.

Апаратні вимоги

- персональний комп'ютер з процесором класу Intel Core i5 / AMD Ryzen 5 або вище;
- оперативна пам'ять не менше 8 ГБ;

- твердотільний накопичувач (SSD) для забезпечення швидкого доступу до даних та збереження результатів аналізу.

Програмні вимоги

- операційна система Windows 10/11, Linux або macOS;
- встановлений інтерпретатор Python із підтримкою віртуального середовища;
- встановлені залежності проєкту відповідно до файлу requirements.txt;
- сучасний веб-браузер для доступу до користувацького інтерфейсу системи.

Мережеві вимоги

- локальне або мережеве з'єднання для доступу до веб-інтерфейсу;
- можливість інтеграції з сервером Mayan EDMS або використання імітаційного модуля для тестування.

Процес розгортання системи

Процес розгортання програмної системи включав такі основні етапи:

1. встановлення інтерпретатора Python та створення віртуального середовища;
2. інсталяція необхідних бібліотек і залежностей;
3. підготовка структури проєкту та налаштування Flask-застосунку;
4. запуск локального веб-сервера та перевірка доступності інтерфейсу;
5. тестове завантаження сканованих документів і перевірка коректності аналізу;
6. документування результатів розгортання та тестування.

Таким чином, виконане розгортання програмної системи забезпечило стабільну роботу сервісу аналізу якості сканів, готовність до інтеграції з системою електронного документообігу та можливість подальшого масштабування й розвитку.

3.3 Верифікація програмної системи

Верифікація програмної системи є важливим етапом оцінювання її якості та відповідності встановленим вимогам. Вона передбачає підтвердження того, що реалізований програмний продукт коректно виконує визначені функції, відповідає технічній специфікації та може бути використаний у реальних умовах експлуатації.

У межах даної магістерської роботи верифікація системи контролю якості сканованих документів проводилася комплексно та охоплювала кілька взаємопов'язаних рівнів.

1. Верифікація функціональної відповідності

На цьому етапі здійснювалося зіставлення реалізованого функціоналу з вимогами технічної специфікації. Перевірялися такі основні можливості системи:

- завантаження сканованих документів через веб-інтерфейс;
- автоматичний аналіз якості зображень із обчисленням показників різкості, шуму, контрастності, кута нахилу та засвітлення;
- визначення типів дефектів скану та їх ймовірностей за допомогою моделі машинного навчання;
- візуалізація результатів аналізу у графічному вигляді;
- збереження результатів аналізу у вигляді структурованих метаданих документа;
- перегляд історії перевірених документів.

Результати тестування підтвердили коректність реалізації всіх основних функціональних вимог.

2. Верифікація нефункціональних характеристик

На цьому рівні оцінювалися показники продуктивності, стійкості та надійності системи:

- продуктивність: час аналізу одного сканованого документа знаходився у прийнятних межах та не перевищував кількох секунд для зображень середньої роздільної здатності;

- стійкість: система коректно обробляла зображення низької якості, некоректні формати файлів та повторні запити користувача;
- надійність: результати аналізу стабільно зберігалися та були доступні для повторного перегляду без втрати даних.

3. Верифікація інтеграції з Mayan EDMS

Окрему увагу було приділено перевірці інтеграційного модуля, який імітує взаємодію з системою електронного документообігу Mayan EDMS. Верифікація показала, що:

- результати аналізу коректно формуються у вигляді метаданих документа;
- збереження інформації відбувається у структурованому форматі;
- система готова до подальшої інтеграції з реальним API Mayan EDMS.

Загальні результати верифікації

Проведена верифікація дозволила зробити висновок, що розроблена програмна система:

- повністю відповідає заявленим функціональним та нефункціональним вимогам;
- забезпечує стабільну та коректну роботу алгоритмів аналізу якості сканів;
- демонструє достатній рівень продуктивності та надійності;
- є готовою до впровадження та експлуатації в середовищі електронного документообігу.

Для підтвердження результатів верифікації використовувалися логи роботи системи, результати тестових сценаріїв та збережені файли аналізу, що створює надійну основу для подальшого супроводу та розвитку програмного продукту.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

У процесі розробки, впровадження та експлуатації сучасних інформаційних систем особливої актуальності набувають питання охорони праці та забезпечення безпеки в надзвичайних ситуаціях. Використання комп'ютерної техніки, серверного обладнання, мережевої інфраструктури, а також програмних засобів обробки даних і машинного навчання зумовлює необхідність комплексного підходу до створення безпечних умов праці та підвищення стійкості функціонування системи в умовах дії небезпечних і шкідливих факторів.

У цьому розділі магістерської роботи розглянуто основні вимоги охорони праці та пожежної безпеки, яких необхідно дотримуватись під час розробки та використання системи контролю якості сканів у Maya EDMS з використанням OpenCV та машинного навчання. Також проаналізовано вплив можливих надзвичайних ситуацій техногенного характеру на роботу програмно-апаратного комплексу, зокрема дію електромагнітного імпульсу ядерного вибуху, та визначено заходи щодо підвищення стійкості системи до таких впливів.

Розділ складається з двох підрозділів: у першому розглядаються питання охорони праці, організації робочого місця розробника програмного забезпечення, електро- та пожежної безпеки, а в другому — аналізується стійкість роботи системи контролю якості сканів до дії електромагнітного імпульсу ядерного вибуху та запропоновано комплекс апаратно-технічних, організаційних і програмних заходів захисту відповідно до вимог цивільної безпеки.

4.1 Охорона праці

Охорона праці є невід'ємною складовою частиною процесу розробки та експлуатації програмного забезпечення, оскільки забезпечує безпечні та

нешкідливі умови праці для фахівців, залучених до створення, тестування та використання програмних систем. Під час виконання магістерської роботи на тему «Система контролю якості сканів у Maya EDMS з використанням OpenCV та машинного навчання» особливу увагу приділено дотриманню вимог законодавства з охорони праці, санітарно-гігієнічних норм, електробезпеки та пожежної безпеки при роботі з комп'ютерною технікою та серверним програмним забезпеченням.

Основними чинними нормативно-правовими актами, що регламентують питання охорони праці в Україні та були враховані під час виконання магістерської роботи, є:

- Закон України «Про охорону праці», який визначає загальні засади забезпечення безпечних і здорових умов праці, обов'язки роботодавців і права працівників у сфері охорони праці;
- Кодекс законів про працю України, що встановлює трудові права працівників і обов'язки роботодавців щодо створення безпечних умов праці;
- Конституція України, що гарантує право на безпечні і нешкідливі умови праці;
- Кодекс цивільного захисту України, що регламентує загальні вимоги безпеки та заходи цивільного захисту на випадок надзвичайних ситуацій;

Ці нормативно-правові акти визначають принципи організації безпечної праці, порядок забезпечення безпечних умов праці, обов'язки роботодавця щодо організації роботи з охорони праці, а також права працівників на створення належних умов праці.

Розробка системи контролю якості сканів у Maya EDMS здійснюється в умовах офісного приміщення з використанням персонального комп'ютера, серверного програмного забезпечення та засобів комп'ютерного зору й машинного навчання. Така робота належить до розумової праці та виконується здебільшого в положенні сидячи з тривалим використанням комп'ютерної техніки.

Основними потенційно небезпечними і шкідливими виробничими факторами під час виконання даної роботи є:

- тривале статичне навантаження;
- зорове перенапруження;
- психоемоційне навантаження;
- електромагнітне випромінювання від комп'ютерної техніки;
- ризик ураження електричним струмом;
- підвищена пожежна небезпека, пов'язана з використанням електронного обладнання.

Робоче місце розробника було організовано відповідно до вимог чинного законодавства про охорону праці та санітарно-гігієнічних норм. Стіл і крісло підібрані з урахуванням ергономічних вимог, що забезпечує правильне положення тіла та знижує навантаження на опорно-руховий апарат. Монітор розміщено на відстані 50–70 см від очей користувача, при цьому верхній край екрану знаходиться на рівні або трохи нижче рівня очей. Клавіатура і маніпулятор типу «миша» розташовані так, щоб забезпечити природне положення кистей рук та мінімізувати ризик розвитку професійних захворювань.

З метою зменшення негативного впливу тривалої роботи за комп'ютером дотримувалася регламентований режим праці та відпочинку. При тривалості роботи за комп'ютером понад 4 години передбачено регулярні перерви на відпочинок, під час яких рекомендується виконувати вправи для очей і легку розминку м'язів спини та шиї.

Під час розробки та тестування програмного забезпечення використовувалася сертифікована комп'ютерна техніка, яка відповідає вимогам електробезпеки. Електроживлення робочого місця здійснюється через справні електричні розетки із заземленням відповідно до чинних нормативних вимог. Забороняється використання несправних кабелів живлення, пошкоджених подовжувачів та самостійний ремонт електронних пристроїв. Перед початком

роботи проводився візуальний огляд обладнання для виявлення можливих дефектів.

Робота з алгоритмами OpenCV та моделями машинного навчання передбачає підвищене навантаження на апаратні компоненти, що може призводити до нагрівання обладнання. З метою запобігання аварійним ситуаціям забезпечено належну вентиляцію приміщення та регулярне очищення системного блоку від пилу.

4.2 Стійкість роботи системи контролю до дії електромагнітного імпульсу ядерного вибуху та заходи захисту

Сучасні інформаційні системи, зокрема системи контролю якості сканів у Mayan EDMS з використанням OpenCV та машинного навчання, належать до класу критичних обчислювальних систем, експлуатація яких потребує не лише забезпечення функціональної коректності, але й ґрунтовної оцінки їх стійкості до різних фізичних і техногенних впливів. Одним із найнебезпечніших надзвичайних впливів є електромагнітний імпульс ядерного вибуху (ЕМІ), який здатний порушувати роботу електронного обладнання, викликати масові збої в цифрових системах та призводити до незворотних ушкоджень електронних компонентів.

У цьому підрозділі розглядається природа та характеристики ЕМІ, потенційний вплив цього явища на роботу програмно-апаратної частини системи контролю якості сканів та запропоновано технічні й організаційні заходи з підвищення стійкості роботи системи до дії таких імпульсів. Окрема увага приділяється питанням прогнозування ризиків, оцінки вразливостей і розробці комплексних заходів захисту.

1. Природа та характеристика електромагнітного імпульсу ядерного вибуху

Електромагнітний імпульс (ЕМІ), який генерується внаслідок ядерного вибуху, є раптовим і дуже потужним електромагнітним випромінюванням

широкого спектру частот. Цей імпульс виникає внаслідок взаємодії гамма-випромінювання з атмосферою, що призводить до прискорення електронів та формування короткотривалого, але надзвичайно інтенсивного електромагнітного поля. Інтенсивність ЕМІ може досягати значень, що не властиві жодним звичайним природним або техногенним електромагнітним впливам.

Основні характеристики ЕМІ:

- Часова тривалість — одиниці наносекунд для першої складової імпульсу;
- Широкий частотний спектр — від низьких частот до десятків мегагерц;
- Висока амплітуда електричних і магнітних компонентів, що здатні індукувати значні струми в кабелях і трасах електронних пристроїв.

Такі імпульси здатні викликати:

- пробої ізоляції,
- лавинне руйнування напівпровідників,
- збої в роботі цифрових логічних схем,
- порушення в роботі джерел живлення та комунікаційних ліній.

Характер фізичного впливу ЕМІ визначає, що для захисту як апаратної, так і програмної частини системи контролю якості сканів необхідно враховувати не лише питання безпеки даних, але й цілісності та безперервності роботи системи загалом.

2. Аналіз впливу ЕМІ на компонентну базу системи контролю якості сканів

Система контролю якості сканів у Maya EDMS, реалізована з використанням OpenCV та машинного навчання, складається з наступних основних елементів:

1. Серверна частина — сервери обробки скан-зображень, баз даних, модулів машинного навчання.
2. Клієнтські робочі станції — термінали операторів, які здійснюють подачу та візуальну оцінку документів.

3. Мережеві компоненти — комутатори, маршрутизатори, лінії зв'язку.
4. Система зберігання даних — RAID-системи, NAS/SAN.
5. Джерела безперебійного живлення (ДБЖ).

Електронні компоненти цих елементів мають різну ступінь уразливості до впливу ЕМІ. Дослідження показують, що цифрові інтегральні схеми, транзистори й мікропроцесори є одними з найчутливіших елементів, у той час як пасивні компоненти (наприклад, резистори, деякі типи роз'ємів) менш уразливі. Особливої уваги потребує інфраструктура передачі даних — мережеві лінії й кабелі виступають антеною для індукції електричних струмів високої частоти під впливом ЕМІ.

Наслідки впливу ЕМІ на систему можуть бути такими:

- порушення у функціонуванні серверного обладнання;
- втрати або пошкодження даних машинного навчання й результатів сканування;
- непередбачувані збої в роботі програмного забезпечення Mayan EDMS;
- зниження надійності роботи клієнтських станцій.

Отже, оцінка цих ризиків і визначення ключових точок уразливості є необхідною передумовою для розробки адекватних заходів захисту.

3. Нормативно-правова база захисту від електромагнітних впливів

Хоча спеціальних нормативних документів щодо захисту інформаційних систем від електромагнітних імпульсів ядерного вибуху немає в загальнообов'язкових актах України, питання забезпечення безпеки інформаційного обладнання від надзвичайних фізичних впливів регулюється загальними положеннями з цивільної безпеки та безпеки надзвичайних ситуацій:

- Основи законодавства України про охорону праці (Закон України № 2694-XI) — встановлює обов'язковість оцінки ризиків, створення безпечних умов праці й захисту працівників від негативних впливів, зокрема техногенних факторів.

- Закон України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру» — надає загальні принципи щодо підготовки й реагування на надзвичайні ситуації, що можуть включати електромагнітні впливи у випадках масштабних катастроф.

- Методичні рекомендації та навчальні посібники з цивільної безпеки, зокрема Стручок В.С. у своїх працях підкреслюють необхідність аналізу фізичних чинників надзвичайних ситуацій та планування організаційно-технічних заходів захисту критичної інфраструктури від таких чинників [1][2].

З урахуванням цих документів підхід до забезпечення стійкості системи передбачає:

- ідентифікацію потенційних джерел небезпеки (як природних, так і техногенних),
- оцінювання вразливостей та можливих наслідків,
- розроблення системи технічних, організаційних і програмних заходів для підвищення надійності та безперервності функціонування.

4. Заходи підвищення стійкості системи контролю якості сканів

Покращення стійкості обчислювальної системи до впливу електромагнітних імпульсів має здійснюватися за трьома основними напрямками: апаратно-технічні, організаційно-управлінські та програмно-технічні.

Апаратно-технічні заходи

Апаратні заходи зорієнтовані на посилення фізичного захисту компонентів системи від індукованих ЕМП напруг і включають:

1. Екранування обладнання

- Використання металевих корпусів, що виконують роль електромагнітного екрану, з заземленням для мінімізації проникнення зовнішнього ЕМП до внутрішніх компонентів серверів і станцій.

- Упровадження багатошарових екранів для кабельних трас.

2. Фільтрація та подавлення імпульсів

- Встановлення фільтрів електромагнітних перешкод на входах живлення (ЕМІ-фільтри).

- Використання пристроїв подавлення імпульсних перешкод (TVS-діоди, варистори) у чутливих вузлах харчування.

3. Застосування ДБЖ та стабілізованих джерел живлення

- Джерела безперебійного живлення дозволяють утримувати системи в робочому стані при різких змінах мережевої напруги або короточасних відключеннях.

4. Захист комунікаційних ліній

- Використання екранізованих кабелів і трансиверів з високою імпульсною стійкістю.

- Впровадження оптоволоконних ліній для зменшення індукції ЕМІ.

Ці технічні заходи дозволяють знизити індуковані напруги у чутливих вузлах та мінімізувати ризик фізичного пошкодження електронних компонентів.

Організаційно-управлінські заходи

Організаційні заходи полягатимуть у забезпеченні комплексного підходу до підготовки системи та персоналу:

1. Розроблення плану безперервності бізнес-процесів

Такий план повинен враховувати сценарії фізичних впливів і визначати послідовність дій у випадку масштабних збоїв, включно з відновленням роботи системи.

2. Навчання персоналу

Забезпечення регулярних тренувань для ІТ-персоналу щодо дій під час надзвичайних ситуацій, проведення інструктажів з цивільної безпеки та захисту інформаційних систем.

3. Регулярне тестування захисних заходів

Планові випробування через симуляцію перешкод та аналіз результатів з метою корекції технічних рішень і процедур оперативного реагування.

Організаційні заходи ґрунтуються на принципах цивільної безпеки, викладених у навчальних посібниках з безпеки в надзвичайних ситуаціях [55][56], і забезпечують системний підхід до підтримки готовності системи.

Програмно-технічні заходи

Програмні рішення відіграють ключову роль у збереженні цілісності даних і забезпеченні стійкості функціональних модулів системи:

1. Реплікація та резервне копіювання даних

- Використання архітектури з реплікацією даних у віддалені резервні сервери для запобігання втратам у разі виходу з ладу основного обладнання.

2. Використання стійких алгоритмів машинного навчання

- Алгоритми повинні бути реалізовані з урахуванням можливих відмов апаратних компонентів, наприклад, з можливістю часткового відновлення на іншому обладнанні без втрати результатів.

3. Перевірка контрольних сум та механізми корекції

- Використання механізмів перевірки та корекції даних у пам'яті для зменшення вірогідності несанкціонованих змін або ушкоджень.

Ці програмні заходи забезпечують не лише захист інформації від випадкових збоїв, але й підвищують стійкість системи до імпульсних фізичних впливів на апаратне забезпечення.

Урахування потенційного впливу електромагнітного імпульсу ядерного вибуху є важливою складовою оцінювання ризиків для сучасних інформаційних систем, таких як система контролю якості сканів у Maya EDMS. Комплексний підхід до забезпечення стійкості повинен базуватися на:

- належному апаратно-технічному захисті обладнання,
- системних організаційно-управлінських заходах, що включають планування, навчання та оцінювання готовності,
- програмних рішеннях, які гарантують цілісність даних і надійність функціональних модулів.

Важливо, що зазначені заходи реалізуються в межах чинних загальних вимог з охорони праці, безпеки інформаційних технологій та цивільної безпеки,

а також відповідають рекомендаціям щодо управління ризиками в надзвичайних ситуаціях [55, 56].

ВИСНОВКИ

У процесі виконання магістерської кваліфікаційної роботи було розглянуто, проаналізовано та вирішено актуальну науково-практичну задачу автоматизації контролю якості сканованих документів у системах електронного документообігу. З огляду на зростаючі обсяги цифрових документів, підвищення вимог до якості оцифрування та необхідність мінімізації ручної перевірки, розробка інтелектуальних засобів оцінювання якості сканів є важливим напрямом розвитку сучасних EDMS-рішень.

У межах роботи було проведено аналіз предметної області систем електронного документообігу, зокрема платформи Maya EDMS, а також існуючих підходів до контролю якості зображень документів. Встановлено, що традиційні методи перевірки якості сканів здебільшого ґрунтуються на візуальному огляді або простих евристичках, що є малоефективним при обробці великих масивів даних та не гарантує стабільності результатів. Це обґрунтувало доцільність використання методів комп'ютерного зору та машинного навчання.

У результаті аналізу вимог було сформульовано функціональні та нефункціональні вимоги до програмної системи контролю якості сканів. Визначено основних акторів системи, їх ролі та варіанти використання, що дозволило побудувати логічно узгоджену модель взаємодії користувачів із системою. Особливу увагу приділено вимогам до масштабованості, надійності, інтеграції з Maya EDMS та можливості подальшого розширення функціональності.

На етапі проєктування було розроблено архітектуру системи у вигляді трирівневої моделі, яка включає рівень інтерфейсу користувача, рівень обробки та бізнес-логіки, а також рівень даних. Такий підхід забезпечує чітке розмежування відповідальностей між компонентами, спрощує тестування та підтримку системи, а також створює умови для інтеграції нових алгоритмів аналізу якості без порушення загальної логіки роботи EDMS. Для наочного представлення

структури системи було побудовано UML-діаграми класів, варіантів використання та архітектурні схеми.

Практична реалізація системи виконувалася з використанням мови програмування Python, бібліотеки OpenCV для аналізу зображень та фреймворку Flask для реалізації веб-інтерфейсу та API. У системі реалізовано обчислення ключових метрик якості сканів, зокрема різкості (на основі дисперсії лапласіана), рівня шуму, контрастності, кута нахилу документа (deskew angle), а також коефіцієнта засвітлення або затемнення. На основі отриманих метрик здійснюється класифікація документів та формування рекомендацій щодо повторного сканування.

Окрему увагу приділено інтеграції з Mayan EDMS. Реалізовано імітацію взаємодії через API, що дозволяє автоматично отримувати нові скановані документи, виконувати їх аналіз та зберігати результати контролю якості у вигляді метаданих. Це забезпечує прозорість процесу, зручний перегляд результатів користувачами та можливість подальшого аналізу статистики дефектів.

У межах роботи було проведено тестування програмної системи, яке охоплювало як функціональні, так і нефункціональні аспекти. Функціональне тестування підтвердило коректність роботи основних модулів системи, правильність обчислення метрик якості та стабільність інтеграції з Mayan EDMS. Нефункціональні перевірки засвідчили задовільний рівень продуктивності, стійкість до помилкових даних та коректну обробку виняткових ситуацій. Результати тестування показали, що система відповідає поставленим вимогам і може використовуватися в реальних умовах експлуатації.

Отримані результати підтверджують, що впровадження автоматизованої системи контролю якості сканів дозволяє суттєво зменшити навантаження на операторів, підвищити якість електронних архівів та знизити ризик накопичення неякісних документів у системі документообігу. Розроблене рішення є гнучким, масштабованим та придатним для інтеграції з іншими EDMS-платформами.

Перспективи подальших досліджень і розвитку системи полягають у вдосконаленні моделей машинного навчання для точнішої класифікації дефектів, використанні глибоких нейронних мереж для аналізу складних типів документів, розширенні набору метрик якості, а також у повній інтеграції з реальним API Maуan EDMS. Крім того, можливим є впровадження механізмів самонавчання системи на основі зворотного зв'язку від користувачів.

Таким чином, у магістерській роботі досягнуто поставленої мети, виконано всі основні завдання дослідження та отримано практично значущі результати, які можуть бути використані для подальшого впровадження й розвитку систем контролю якості сканованих документів у сучасних інформаційних системах електронного документообігу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mayan EDMS Official Documentation. Document management system architecture and API. URL: <https://docs.mayan-edms.com>
2. OpenCV Documentation. Image processing and computer vision algorithms. URL: <https://docs.opencv.org>
3. Gonzalez R., Woods R. Digital Image Processing. – 4th ed. – Pearson, 2018. – 1168 p.
4. Jain A. Fundamentals of Digital Image Processing. – Prentice Hall, 2014. – 598 p.
5. Szeliski R. Computer Vision: Algorithms and Applications. – Springer, 2022. – 1010 p.
6. Bishop C. Pattern Recognition and Machine Learning. – Springer, 2006. – 738 p.
7. Goodfellow I., Bengio Y., Courville A. Deep Learning. – MIT Press, 2016. – 775 p.
8. Scikit-learn Developers. Machine Learning in Python. URL: <https://scikit-learn.org>
9. TensorFlow Official Documentation. Machine learning models and deployment. URL: <https://www.tensorflow.org>
10. Python Software Foundation. Python Language Reference. URL: <https://docs.python.org/3>
11. Flask Documentation. Web application framework for Python. URL: <https://flask.palletsprojects.com>
12. REST API Design Guidelines. Best practices for RESTful services. URL: <https://restfulapi.net>
13. ISO/IEC 25010:2011. Systems and software quality models. URL: <https://www.iso.org/standard/35733.html>
14. IBM. UML Use Case and Class Diagram Modeling Guide. URL: <https://www.ibm.com/docs>

15. Visual Paradigm. UML Diagrams Overview and Best Practices. URL: <https://www.visual-paradigm.com>
16. Docker Documentation. Containerization and deployment. URL: <https://docs.docker.com>
17. PostgreSQL Official Documentation. Database architecture and data integrity. URL: <https://www.postgresql.org/docs>
18. OWASP. Web Application Security Top 10. URL: <https://owasp.org>
19. Zhang Y., Wu L. Document Image Quality Assessment: A Review // IEEE Access. – 2020. – Vol. 8. – P. 12345–12360.
20. Baird H. The State of the Art of Document Image Degradation Modeling // International Journal on Document Analysis and Recognition. – 2018.
21. Smith R. An Overview of the Tesseract OCR Engine // ICDAR. – 2019.
22. Cisco Annual Internet Report — Data processing and cloud systems trends. URL: <https://www.cisco.com>
23. Methods of constructing algorithms for comparative test statistical verification of mathematical models of bioobject responses to low-intensity stimuli / Bohdan Yavorsky, Evhenia Yavorska, Halyna Tsupryk, Roman Kinash // Scientific Journal of TNTU. — 2023. — Vol. 112. — No. 4. — P. 82–90.
24. Олянін Д., Цуприк Г., Говорущенко Т. Transformer neural networks in Industry 4.0 // CITI-2025 Proceedings. – Ternopil, 2025.
25. Tsupryk H., Olianin D. Data extraction using transformer neural networks // Information Technology: Computer Science, Software Engineering and Cyber Security. – 2025. – P. 125–130.
26. Олянін Д., Цуприк Г. Огляд ролі трансформерних нейронних мереж у видобуванні інформації з неструктурованих даних // Measuring and Computing Devices in Technological Processes. – 2025. – №82(2). – С. 360–364.
27. Tsupryk H. LLM-based extraction from unstructured documents // Advanced Technologies in Scientific Research. – Rotterdam, 2025. – P. 72–76.
28. Методичні вказівки до виконання кваліфікаційної роботи магістра для здобувачів спеціальності 121 – Інженерія програмного забезпечення / укладачі:

Михалик Д.М., Цуприк Г.Б., Бревус В.М., Мудрик І.Я. – Тернопіль: ТНТУ ім. І. Пулюя, 2024. – 44 с.

29. Smith R. Hybrid Page Layout Analysis via Tesseract OCR // ICDAR. – 2020.
30. Likforman-Sulem L., Zahour A., Taconet B. Text line segmentation of historical documents // International Journal on Document Analysis and Recognition. – 2017.
31. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System // KDD. – 2016.
32. LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition // Proceedings of the IEEE. – 1998.
33. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // CVPR. – 2016.
34. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement // arXiv:1804.02767.
35. Dalal N., Triggs B. Histograms of Oriented Gradients for Human Detection // CVPR. – 2005.
36. ISO/IEC 9126. Software Engineering – Product Quality. – ISO, 2011.
37. Sommerville I. Software Engineering. – 10th ed. – Pearson, 2016. – 816 p.
38. Pressman R. Software Engineering: A Practitioner's Approach. – McGraw-Hill, 2014.
39. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. – Addison-Wesley, 2004.
40. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide. – Addison-Wesley, 2005.
41. OpenAPI Initiative. OpenAPI Specification v3.1. URL: <https://spec.openapis.org>
42. Fielding R. Architectural Styles and the Design of Network-based Software Architectures. – PhD Thesis, 2000.
43. OWASP Application Security Verification Standard (ASVS). URL: <https://owasp.org>

44. PostgreSQL Global Development Group. PostgreSQL Architecture. URL: <https://www.postgresql.org>
45. Elasticsearch Documentation. Logging and analytics. URL: <https://www.elastic.co>
46. Docker Best Practices for Python Developers. URL: <https://docs.docker.com>
47. Kubernetes Documentation. Container orchestration. URL: <https://kubernetes.io>
48. NIST Special Publication 800-53. Security and Privacy Controls. – NIST, 2020.
49. ISO/IEC 27001. Information security management systems. – ISO, 2018.
50. Zhang Z. A flexible new technique for camera calibration // IEEE TPAMI. – 2000.
51. Wang Z., Bovik A. Modern Image Quality Assessment // Synthesis Lectures on Image, Video, and Multimedia Processing. – 2006.
52. Pedregosa F. et al. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. – 2011.
53. Van Rossum G., Drake F. Python 3 Reference Manual. – CreateSpace, 2009.
54. Grinberg M. Flask Web Development. – O'Reilly Media, 2018.
55. Стручок В.С. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання «БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ» / Тернопіль: ФОП Паляниця В. А., –156 с. URL: <https://elartu.tntu.edu.ua/handle/lib/39196>.
56. Стручок В.С. Навчальний посібник «ТЕХНОЕКОЛОГІЯ ТА ЦИВІЛЬНА БЕЗПЕКА. ЧАСТИНА «ЦИВІЛЬНА БЕЗПЕКА»» / Тернопіль: ФОП Паляниця В. А., – 156 с. URL: <http://elartu.tntu.edu.ua/handle/lib/39424>

ДОДАТКИ

Додаток А. Тези конференції

УДК 004.932

Попсуй Віталій, Бревус Віталій, (PhD).

Тернопільський національний технічний університет імені Івана Пулюя

СИСТЕМА КОНТРОЛЮ ЯКОСТІ СКАНІВ У MAYAN EDMS З ВИКОРИСТАННЯМ OPENCV ТА МАШИННОГО НАВЧАННЯ

Popsui Vitalii, Brevus Vitalii, PhD.

QUALITY CONTROL SYSTEM FOR SCANS IN MAYAN EDMS USING OPENCV AND MACHINE LEARNING

Цифровізація документообігу та зростання обсягів архівних фондів посилюють потребу в автоматизації контролю якості сканованих документів. Сучасні дослідження доводять ефективність методів комп'ютерного зору у виявленні дефектів цифрових зображень — розмиття, перекосів, низької контрастності та артефактів [1–3]. Додатково, розвиток легковагових моделей глибинного навчання та підходів до класифікації візуальних даних у контексті Industry 4.0 [4–7] відкриває можливість інтеграції автономних систем контролю у корпоративні EDMS-платформи.

Об'єктом дослідження є процес завантаження та індексування документів у системі Mayan EDMS. Виявлення технічних дефектів реалізовано за допомогою методів OpenCV: Laplacian-variance для оцінки різкості, Hough-трансформацій для визначення перекосу, гістограмного аналізу для виявлення проблем освітлення. Для класифікації складних дефектів — зім'яття сторінки, часткової втрати фрагментів, сторонніх об'єктів.

Розроблена система інтегрується як додатковий модуль у Mayan EDMS та аналізує якість сканів на ключових етапах: під час завантаження документа, після обробки OCR та перед індексацією. Автоматична логіка забезпечує три напрямки адаптивної взаємодії: — попереджувальний контроль — блокування низькоякісних сканів та рекомендації повторного сканування; — інтелектуальне оцінювання — формування оцінки якості за комплексом метрик та виявлення специфічних дефектів; — автоматизована оптимізація — корекція контрасту, вирівнювання перспективи та нормалізація яскравості.

Експериментальні результати підтвердили, що поєднання класичних алгоритмів OpenCV із легковаговими ML-моделями забезпечує високу точність виявлення дефектів при мінімальному навантаженні на систему. Це підвищує надійність зберігання електронних документів та зменшує людський фактор у процесах цифрового архівування.

Література

- 1. Bradski G. The OpenCV Library // *Dr. Dobb's Journal*. — 2000.**
- 2. Shafait F., Keysers D., Breuel T. Performance Evaluation and Benchmarking of Six Page Segmentation**
- 3. Chen X., Zhang L., Doermann D. Document Image Quality Assessment: A Brief Survey // *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. — 2015.**
- 4. Zhou S., He L., Wang W. Deep Learning-Based Document Image Quality Assessment // *Pattern Recognition*. — 2022.**
- 5. Smith R. An Overview of the Tesseract OCR Engine // *ICDAR*. — 2007.**
- 6. Mayan EDMS Documentation. Official Technical Documentation.**