

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

перший (бакалаврський)
(рівень вищої освіти)

Автоматизація ідентифікації деформацій елементів функціональної
електроніки на мікроструктурованих оптичних волокнах
(тема)

Виконав:
студент 4 курсу, групи ТРІТЗР-20-1

Назаренко С.В.

Спеціальності 172 Телекомунікації та
радіотехніка

Тип програми Освітньо-професійна

Освітня програма Інтелектуальні технології
засобів радіоелектроніки

Керівник доц. Сичова О.В.

Допускається до захисту
Зав. кафедри КІТАР

(підпис)

Невлюдов І. Ш.
(прізвище, ініціали)

2024р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ _____
Кафедра _____ КІТАР _____
Рівень вищої освіти _____ перший (бакалаврський) _____
Спеціальність _____ 172 Телекомунікації та радіотехніка _____
Тип програми _____ Освітньо-професійна _____
Освітня програма _____ Інтелектуальні технології засобів радіоелектроніки _____
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри КІТАР _____
(підпис)
« _____ » _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Назаренко Сергію Валерійовичу _____
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Автоматизація ідентифікації деформацій елементів функціональної електроніки на мікроструктурованих оптичних волокнах

Затверджена наказом по університету від 20.05.2024 №479Ст

2. Термін подання студентом роботи до екзаменаційної комісії 27.06.2024 р.

3. Вихідні дані до роботи Мова програмування – Java

БД – PostgreSQL

Операційна система – Windows

Фреймворки – OpenCV, ImajeJ, JavaFX, TensorFlow, Deeplearning4j,
PostgreSQL JDBC Driver,

Інструмент для управління залежностями – Maven

4. Перелік питань, що потрібно опрацювати в роботі

4.1 Вступ

4.2 Аналіз предметної області дослідження

4.3 Аналіз літератури та вимог технічного завдання

4.4 Розробка програмного модуля

4.5 Розробка структури бази даних

4.6 Висновки

4.7 Додатки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – --- с. формату А4

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Вступ	25.04 – 28.04.24	Виконано
2	Аналіз предметної області дослідження	01.05 – 05.05.24	Виконано
3	Аналіз літератури та вимог технічного завдання	8.05 – 12.05.24	Виконано
4	Розробка програмного модуля	15.05 – 19.05.24	Виконано
5	Розробка структури бази даних	24.05 – 14.06.24	Виконано
6	Висновки	15.06 – 16.06.24	Виконано
7	Додатки	17.06 – 18.06.24	Виконано
8	Подача на перевірку	21.06 – 25.06.24	Виконано
9	Захист дипломної роботи	27.06.2024	Виконано

Дата видачі завдання 21.03.2024 р.

Студент _____ Назаренко С.В. (підпис)

Керівник роботи _____ доц. Сичова О.В.
(підпис) (посада, прізвище, ініціали)

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«13» червня 2024 р.

Назаренко С.В.

РЕФЕРАТ

Пояснювальна записка: 52 с., 5 табл., 28 рис., 2 дод., 17 джерел.

МІКРОСТРУКТУРОВАНЕ ОПТИЧНЕ ВОЛОКНО, ДЕФОРМАЦІЯ, ІДЕНТИФІКАЦІЯ, АВТОМАТИЗОВАНА СИСТЕМА

Метою роботи є підвищення якості мікроструктурованого оптичного волокна за рахунок розробки і використання програмного засобу автоматизації процесу ідентифікації деформацій.

Об'єктом розробки є автоматизація технологічного процесу контролю оптико-геометричних параметрів мікроструктурованих оптичних волокон;

Предметом розробки є програмний засіб для ідентифікації деформацій мікроструктурованих оптичних волокон.

В даній кваліфікаційній роботі було досліджено методи аналізу зображень та визначено оптимальні підходи для ідентифікації дефектів у мікроструктурованих оптичних волокнах. Здійснено аналіз вимог до системи автоматизації та проаналізовано конкурентоспроможні рішення в даній галузі. Розроблено структурну схему автоматизованої системи аналізу, що включає модулі обробки зображень та ідентифікації дефектів. Після аналізу було обрано програмні та апаратні засоби для створення прототипу системи. Визначено оптимальну схему підключення компонентів, що забезпечує надійність та ефективність роботи системи. На наступному етапі було розроблено та налаштовано інтерфейс користувача для зручної роботи із системою. Після успішної реалізації програмних засобів було проведено тестування системи в різних умовах експлуатації. Далі було розроблено цифровий прототип системи та здійснено його тестування з метою виявлення можливих недоліків.

ABSTRACT

Explanatory note: 52 p., 5 tables, 28 figs., 2 appendices, 17 sources.

MICROSTRUCTURED OPTICAL FIBRE, DEFORMATION, IDENTIFICATION, AUTOMATED SYSTEM.

The aim of the work is to develop software for automating the identification of deformation of microstructured optical fibres;

The object of study is the automation of the technological process of controlling the optical and geometric parameters of microstructured optical fibres;

The subject of development is a software tool for identifying deformations of microstructured optical fibres.

In this qualification work, image analysis methods were investigated and optimal approaches for identifying defects in microstructured optical fibers were determined. The requirements for the automation system were analyzed and competitive solutions in this field were analyzed. The structural diagram of the automated analysis system, which includes image processing and defect identification modules, has been developed. After the analysis, software and hardware tools were selected to create a system prototype. The optimal connection scheme of the components has been determined, which ensures the reliability and efficiency of the system. At the next stage, the user interface was developed and configured for convenient work with the system. After the successful implementation of the software, the system was tested in various operating conditions. Next, a digital prototype of the system was developed and tested to identify possible flaws.

ЗМІСТ

Перелік умовних скорочень.....	9
Вступ.....	10
1 Аналіз мікроструктурованих оптичних волокон.....	12
1.1 Загальні характеристики оптичних волокон.....	12
1.2 Типи деформацій.....	17
1.3 Аналіз впливу деформацій на дисперсію та затування.....	18
1.4 Аналіз методів обробки зображення.....	19
2 Аналіз структури програми.....	24
2.1 Вибір мови програмування.....	24
2.2 Структура бекенду.....	26
2.3 Структура фронтенду.....	30
3 Алгоритм роботи програми.....	32
3.1 Блок-схема.....	32
3.2 Архітектура MVC.....	33
3.3 Структура роботи штучного інтелекту.....	36
3.4 Структура бази даних.....	37
4 Експериментальні випробування розробленого програмного забезпечення.....	43
4.1 Тестування функціоналу за допомогою програмного Забезпечення.....	43
5 Охорона праці.....	47
5.1 Цілі безпеки.....	47
5.2 Організація робочого простору.....	47
5.3 Аналіз безпеки.....	48
5.4 Аудит документації та безпеки.....	48
6 Висновки.....	49
Перелік джерел посилання.....	50

Додаток А Лістинг коду – вхідна точка коду для запуску сервера.....	52
Додаток Б Лістинг коду – підключення до БД.....	53
Додаток В Демонстраційний матеріал.....	54

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – база даних;

МОВ – мікроструктуроване оптичне волокно;

НМ – нейромережа;

ПЗ – програмне забезпечення.

ВСТУП

Завдяки розвитку технологій автоматизація стала головним трендом у всіх професіях, де нераціонально використовується праця людини. Розробка системи автоматизації ідентифікації деформацій функціональних елементів електроніки на мікроструктурованих оптичних волокнах має велике значення в сучасних умовах. Основною метою цього проекту є забезпечення ефективного контролю та виявлення дефектів оптичних волокон для досягнення високої якості продукції. Це, у свою чергу, призводить до зниження собівартості виробництва та підвищення надійності електронних компонентів. Завдяки використанню даної системи вдається досягти високої точності та стабільності в процесі виявлення дефектів. Крім того, система забезпечить зручне та швидке управління, що сприятиме підвищенню ефективності виробничих процесів.

Ця робота стане незамінним інструментом для підвищення функціональності систем контролю якості в електронній промисловості. Це дозволить забезпечити оптимальний рівень контролю, швидке виявлення дефектів і зниження виробничих витрат, що сприятиме економії ресурсів і скороченню витрат на ремонт і заміну несправних компонентів.

Враховуючи важливість цієї системи, можна зрозуміти, що її впровадження сприятиме оптимізації процесів управління якістю продукції в автоматизованих виробничих системах. Це стане основою для підвищення продуктивності та довговічності електронних компонентів, економічної ефективності, а також забезпечить високу якість кінцевої продукції.

Таким чином, розробка системи автоматизації ідентифікації деформацій функціональних елементів електроніки на мікроструктурованих оптичних волокнах є актуальною та необхідною. Ця система принесе багато переваг, зокрема, підвищення ефективності виробничих процесів, зниження витрат і підвищення якості продукції. Завдяки його використанню можна забезпечити

точне і стабільне виявлення дефектів, а також зручне управління системою, що позитивно позначиться на виробничих процесах.

До основних економічних факторів автоматизації відносять підвищення продуктивності, підвищення якості та швидкості реагування, що знижує витрати та кількість помилок.

Метою роботи є підвищення якості оптичного сигналу мікроструктурованого оптичного волокна за рахунок розробки і використання програмного засобу автоматизації процесу ідентифікації деформацій.

Об'єктом розробки є автоматизація технологічного процесу контролю оптико-геометричних параметрів мікроструктурованих оптичних волокон;

Предметом розробки є програмний засіб для ідентифікації деформацій мікроструктурованих оптичних волокон.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати типи оптичних волокон;
- провести аналіз впливу деформацій структури мікроструктурованих оптичних волокон на їх характеристики;
- розробити структуру та алгоритм роботи програмного засобу;
- розробити структуру бази даних;
- розробити графічний інтерфейс програми;
- провести експериментальні дослідження роботи програмного засобу для ідентифікації деформацій мікроструктурованих оптичних волокон;
- оформити пояснювальну записку згідно з вимогами ДСТУ 3008:2015 [1], методичних вказівок з підготовки кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти відповідної спеціальності 172 Телекомунікації та радіотехніка освітньої програми «Інтелектуальні технології засобів радіоелектроніки» [2] та положення про академічну доброчесність [3].

1 АНАЛІЗ СТРУКТУРИ МІКРОСТРУКТУРОВАНИХ ОПТИЧНИХ ВОЛОКОН

1.1 Загальні характеристики оптичних волокон

Оптичні волокна відіграють ключову роль у сучасних телекомунікаційних системах, медичних приладах, сенсорних системах та інших галузях високих технологій. Вони забезпечують високу пропускну здатність і надійність передачі даних на великі відстані з мінімальними втратами. Для досягнення вище згаданих характеристик оптичні волокна мають різні конструктивні особливості та застосування [6].

Більшість оптичних волокон проектується в залежності від потреб сфер де вони застосовуються.

Одномодові волокна (Single-Mode Fiber), використовуються для передачі світлового сигналу на великі відстані. Мають малий діаметр сердцевини, що дозволяє підтримувати лише одну моду поширення світла, що зменшує дисперсійні втрати, завдяки чому широко застосовуються у: телекомунікації та високошвидкісній мережі передачі даних. Прикладом такого волокна служить ІТУ-Т (рис. 1.1).

Серцевина є центральною частиною волокна. Виготовляється з чистого кварцового скла та має високий показник заломлення, що дозволяє світловому сигналу поширюватися всередині. Після сердцевини, в структурі, іде оболонка, що оточує сердцевину і виготовлена з кварцового скла з трохи нижчим показником заломлення, завдяки чому забезпечує повне внутрішнє відбиття світлового сигналу, утримуючи його всередині сердцевини. Останнім елементом волокна є покриття, що служить зовнішнім захисним шаром, котрий виготовлений з полімерного матеріалу. Захищає волокно від механічних пошкоджень, вологи та інших зовнішніх впливів.

Ця структура забезпечує низькі втрати сигналу та високу пропускну

здатність, що робить одномодове волокно ІТУ-Т G.652 ідеальним для телекомунікаційних систем, де необхідно передавати дані на великі відстані з мінімальними втратами.

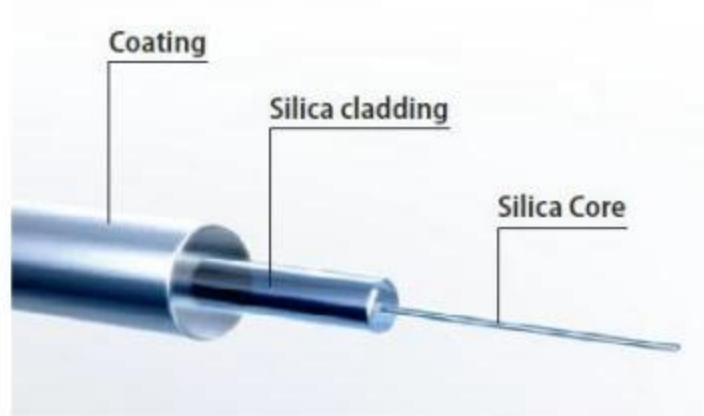


Рисунок 1.1 – Одномодове волокно ІТУ-Т

Багатомодові волокна (Multi-Mode Fiber, MMF), мають більший діаметр серцевини, що дозволяє підтримувати кілька мод поширення світла. Це спрощує з'єднання волокон і знижує вартість системи, але збільшує дисперсійні втрати на великих відстанях. Через це широко застосовується у локальних мережах, коротких з'єднаннях в центрах обробки даних, системах освітлення.

Гарним прикладом для цього є OM1, OM2, OM3 та OM4 волокна, які використовуються в локальних і корпоративних мережах (рис. 2.2).

OM1 має діаметр серцевини 62,5 мкм, та діаметр оболонки 125 мкм. Частіше за все його використовують на коротких відстанях в локальних мережах. Підтримують швидкість передачі даних до 1 Гбіт/с на відстані до 300 метрів.

OM2 має діаметр серцевини 50 мкм, та діаметр оболонки 125 мкм. Використовується на коротких відстанях, але можуть забезпечувати вищу швидкість передачі даних порівняно з OM1. Підтримують до 10 Гбіт/с на відстані до 82 метрів.

OM3 має діаметр серцевини 50 мкм, та діаметр оболонки 125 мкм. Цей тип волокон оптимізований для лазерних систем і підтримує високошвидкісні застосування, такі як 10 Гбіт/с на відстані до 300 метрів і 40-100 Гбіт/с на відстані до 100 метрів.

OM4 має діаметр серцевини 50 мкм та діаметр оболонки 125 мкм. Завдяки чому цей тип ОВ має покращену продуктивність порівняно з OM3, підтримуючи 10 Гбіт/с на відстані до 550 метрів і 40-100 Гбіт/с на відстані до 150 метрів.

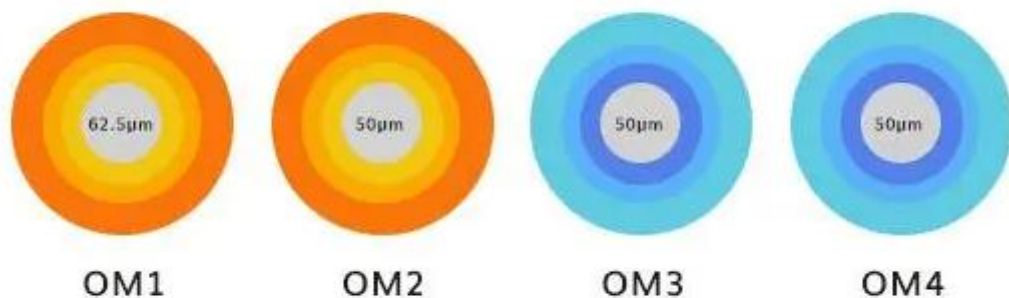


Рисунок 1.2 – OM1, OM2, OM3 та OM4 волокна

Фотонно-кристалічні волокна мають складну структуру з періодичною зміною показника заломлення в поперечному перерізі, що дозволяє керувати поширенням світла у волокні. Це забезпечує високу ступінь гнучкості в налаштуванні дисперсійних характеристик і допомагає підтримувати передачу світла з низькими втратами навіть у складних умовах. Через це широко застосовується у високочутливих сенсорах, нелінійній оптиці, спектроскопії [6].

На рисунку нижче зображенні поперечні перерізи фотонно-кристалічного волокна з масивом повітряних отворів, основні параметри структури яких включають (рис. 1.3);

- Λ : відстань між центрами сусідніх повітряних отворів;
- d : діаметр кожного повітряного отвору.

Ці параметри критично впливають на дисперсійні властивості волокна.

Графік ліворуч в центрі показує залежність ефективного показника заломлення від довжини хвилі для різних конфігурацій (Λ , d) (рис. 1.3). Він демонструє, як зміна параметрів структури волокна впливає на його дисперсійні характеристики.

Графік праворуч в центрі показує залежність між довжиною хвилі накачки та фазовим узгодженням для різних структур (рис. 1.3). Він ілюструє, як різні конфігурації волокна можуть бути оптимізовані для досягнення ефективного фазового узгодження під час чотирихвильового змішування.

Також, на рисунку можна побачити 1,2621 груп фазового узгодження, що знаходяться у правій частині рисунку, позначені параметрами структури PCF (рис. 2.3). Цей набір даних використовується для навчання нейронної мережі, щоб вона могла прогнозувати оптимальні параметри для досягнення бажаних характеристик FWM.

Схема праворуч показує процес перетворення довжини хвилі (λ_{pump} , $\lambda_{\text{converted}}$) за допомогою чотири хвильового змішування (рис. 1.3). Перетворювач довжини хвилі є ключовим елементом в оптичних системах, де необхідно змінювати довжину хвилі світлового сигналу для різних застосувань.

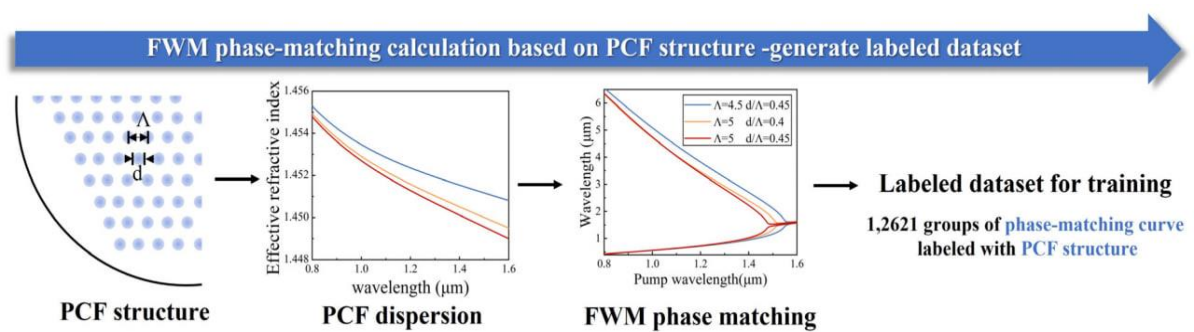


Рисунок 1.3 – Структура фотонно-кристалічного волокна

Мікроструктуровані оптичні волокна характеризуються наявністю складних структур у поперечному перерізі, які дозволяють досягти специфічних оптичних властивостей, таких як низькі втрати, висока дисперсія або можливість керування поляризацією (рис. 1.4). Через що, основними застосуваннями для них є передові сенсорні системи та телекомунікаційні мережі [6].

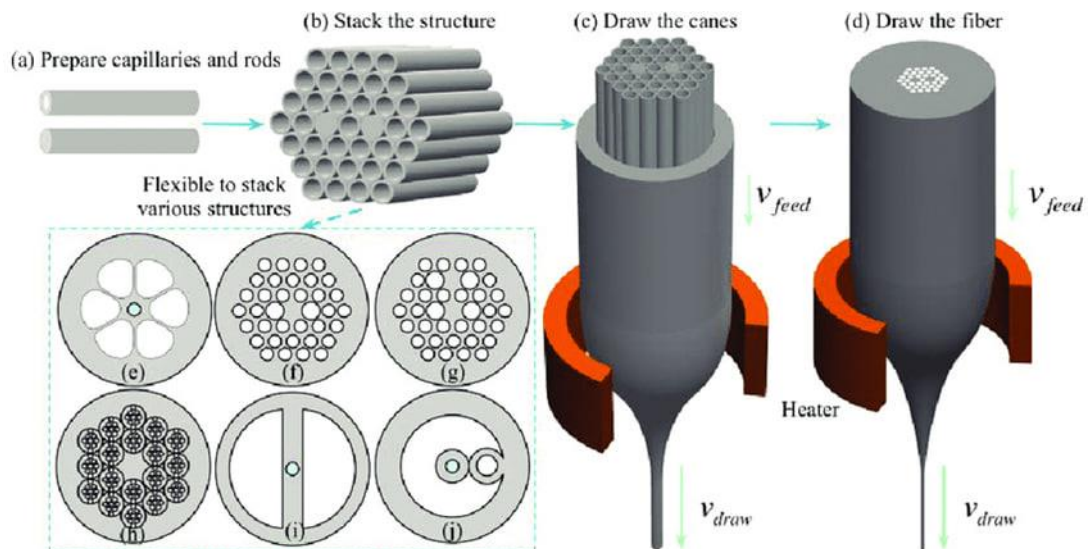
Серцевина є центральною частиною волокна, де проходить основний світловий сигнал (рис. 1.4). Виготовлена вона з високоякісного кварцового скла з вищим показником заломлення порівняно з оболонкою. Оболонка оточує серцевину і виготовлена з кварцового скла з нижчим показником заломлення. Ця різниця в показниках заломлення між серцевиною і оболонкою забезпечує явище повного внутрішнього відбиття, яке утримує світло всередині серцевини.

Структура ОВ включає в себе масив повітряних отворів, рівномірно розташованих навколо серцевини. Вони можуть бути різної форми і розміру, що дозволяє створювати специфічні оптичні властивості волокон. Залежно від конфігурації, мікроструктури можуть забезпечувати різну дисперсію, нелінійні властивості та поляризаційну залежність.

Існують наступні види структур:

- (а) кільцеві структури, які використовуються для налаштування дисперсійних властивостей;
- (б) стовпчасті структури що містять ряди вертикальних стовпів, які забезпечують високий ступінь контролю над світловими хвилями;
- (в-г) інші конфігурації, у яких можливі різні комбінації отворів та твердих стрижнів для досягнення специфічних оптичних ефектів.

Зовнішнє покриття волокна, яке захищає його від механічних пошкоджень і впливу навколишнього середовища, що також сприяє збереженню гнучкості волокна.



а) кільцеві структури; б) стовпчасті структури; в-г) інші конфігурації;

Рисунок 1.4 – Види структур оптичних волокон

1.2 Типи деформацій мікроструктурованих оптичних волокон

Існує декілька основних типів деформацій, кожна з яких має свої особливості та підтипи. Розглянемо їх усі для того, щоб краще зрозуміти об'єкт аналізу:

- стиснення та розтягнення виникає, коли волокно механічно навантажується, змінюючи його довжину і викликаючи зміни показника заломлення матеріалу. Наприклад, коли під час прокладання або експлуатації оптичного кабелю виникають напруження;

- вигин впливає на напрямок поширення світла, що може призвести до збільшення втрат сигналу. Це особливо важливо при прокладанні волокон в обмеженому просторі;

- теплове розширення і стиснення виникає, коли підвищення або зниження температури спричиняє зміни довжини волокна, що впливає на його дисперсійні властивості та загасання;

Деформація при температурних циклах виникає тоді, коли постійні зміни температури під час циклів нагрівання та охолодження призводять до появи мікротріщин та інших пошкоджень у волокні.

– зміна геометрії мікроструктур, це будь яка зміна в мікроструктурі волокна, коли деформації повітряних каналів або фотонних кристалів, можуть суттєво вплинути на передачу та ефективність волокна;

– дефекти матеріалу, це коли з'являються сторонні частинки або порожнечі, під час виробництва або експлуатації, впливаючи на оптичні властивості волокна.

Механічні деформації, такі як стиснення і розтягнення, можуть викликати зміну показника заломлення, що змінює швидкість поширення світла і, відповідно, дисперсію. Теплові деформації змінюють довжину волокна, впливаючи на часові характеристики сигналу і викликаючи додаткові втрати. Мікроструктурні деформації можуть змінювати ефективний показник заломлення, що впливає на загальні оптичні властивості волокна [1].

1.3 Аналіз впливу деформацій на дисперсію та згасання

Хроматична дисперсія виникає через різну швидкість поширення світлових хвиль різних довжин, що може бути викликано термічними та мікроструктурними деформаціями. Наприклад, теплове розширення змінює довжину волокна та його показник заломлення, що впливає на дисперсію.

Поляризаційна модова дисперсія (PMD) викликається механічними деформаціями, такими як вигин або розтяг, що створює анізотропію у волокні. Це призводить до розщеплення мод і, як наслідок, до різниці в часі проходження сигналу для різних поляризаційних мод, що значно погіршує якість сигналу.

Зміна температури може спричинити термічне розширення або стиснення волокна, що змінює його фізичні властивості, у тому числі показник заломлення та геометрію мікроструктури. Ці зміни можуть впливати на дисперсію, збільшуючи або зменшуючи значення залежно від температурного градієнта.

Вигін, скручування та розтягування волокна призводять до зміни внутрішньої структури та геометрії волокна. Це викликає додаткові втрати на згині та зміни в дисперсійних характеристиках, оскільки різні моди світла починають взаємодіяти з новими межами мікроструктур, створеними механічними впливами.

Термоциклічність та механічні навантаження можуть викликати мікротріщини та внутрішні дефекти, які збільшують втрату сигналу. Наприклад, постійні коливання температури можуть викликати мікротріщини, що призводить до розсіювання світла та збільшення загасання.

Мікроструктурні дефекти можуть виникати через виробничі похибки або зовнішні механічні впливи, такі як тиск або стиск. Ці дефекти змінюють ідеальну структуру волокна, створюючи додаткові шляхи розсіювання світла і збільшуючи загальні втрати сигналу.

При згинанні волокна змінюється кут відбиття світла всередині структури волокна. Це призводить до втрат на вигині, коли частина світла виходить за межі ядра волокна, що збільшує згасання сигналу. З іншого боку, змінюється розподіл мод, що впливає на дисперсійні характеристики.

Такий тип деформації як розтяг, призводить до зміни відстані між повітряними проміжками в мікроструктурі, що змінює його геометрію і, як наслідок, впливає на дисперсію. Розтягування може призводити до анізотропії, що збільшує поляризаційну модову дисперсію (PMD).

Підвищення або зниження температури змінює фізичні параметри матеріалу волокна, що призводить до зміни показника заломлення. Це впливає на хроматичну дисперсію і може призводити до змін у ширині сигналу, що передається.

1.4 Аналіз методів обробки зображення

У процесі автоматизації ідентифікації деформацій функціональних елементів електроніки на мікроструктурованих оптичних волокнах важливу

роль відіграє вибір методів обробки зображень. Існує кілька основних підходів до обробки зображень, кожен із яких має свої переваги та обмеження.

Традиційні методи обробки зображень включають фільтрацію, сегментацію, аналіз контурів і морфологічні операції.

Фільтрування, наприклад, дозволяє видалити шум із зображень, що покращує якість даних для подальшого аналізу (рис 1.5).



Рисунок 1.5 – Фільтрування зображення

Це можна виконати за допомогою різних фільтрів, таких як фільтри Гаусса, медіанний або середньозважений фільтр. З іншого боку, сегментація дозволяє виділити певні ділянки зображення, які містять потрібні об'єкти, такі як дефекти оптичних волокон.

Контурний аналіз корисний для ідентифікації меж об'єктів і оцінки їх форми та розміру (рис. 1.6).

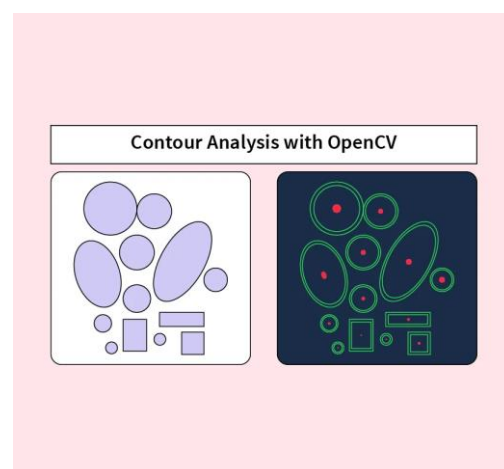


Рисунок 1.6 – Контурний аналіз зображення за допомогою OpenCV

Морфологічні операції, такі як ерозія та розширення, дозволяють змінювати структуру об'єктів на зображенні для покращення їх візуалізації та аналізу (рис 1.7).

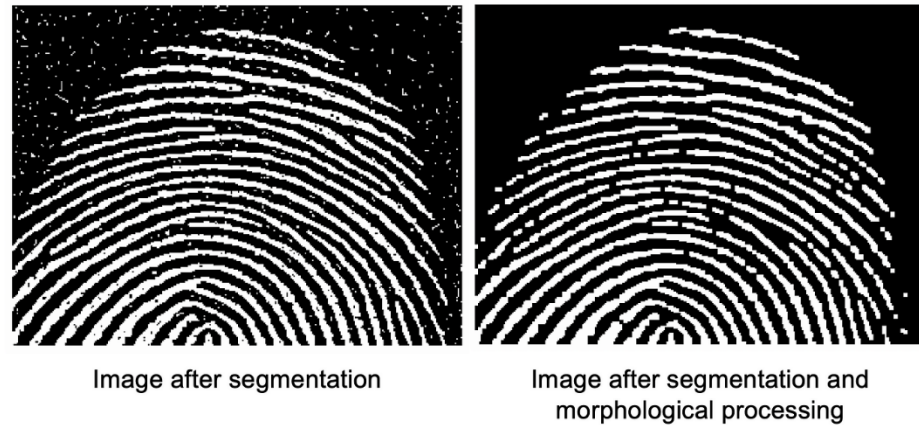


Рисунок 1.7 – Результат роботи морфологічної операції

Сучасні методи обробки зображень базуються на використанні машинного навчання та штучного інтелекту. Зокрема, згорткові нейронні мережі (CNN) стали популярним інструментом для автоматичної обробки та аналізу зображень. Вони здатні навчатися на великих обсягах даних і виявляти складні шаблони, що робить їх особливо ефективними для вирішення проблем, пов'язаних з ідентифікацією дефектів.

У даній роботі використовуються методи обробки зображень на основі OpenCV та ImageJ. OpenCV (Open Source Computer Vision Library) надає широкий спектр інструментів для обробки зображень, включаючи фільтрацію, сегментацію та аналіз контурів. ImageJ, у свою чергу, є інструментом для аналізу біомедичних зображень і має широкий спектр плагінів для спеціалізованих завдань.

Для цієї роботи були обрані наступні методи обробки зображень: гауссовий і медіанний фільтри для зменшення шуму, методи сегментації для

виділення дефектних областей і методи контурного аналізу для оцінки розмірів і форм дефектів (рис 1.8), (рис 1.9), (рис 1.10).

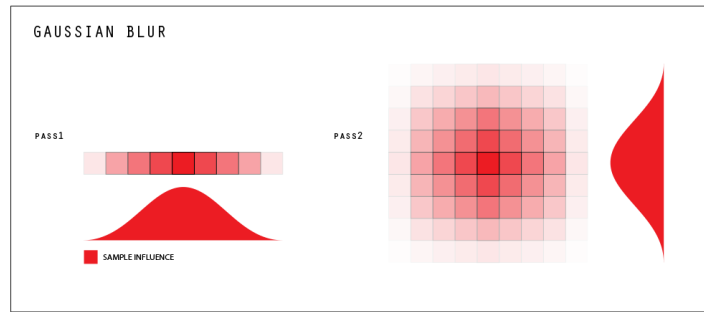


Рисунок 1.8 – Приклад роботи Гаусового фільтру

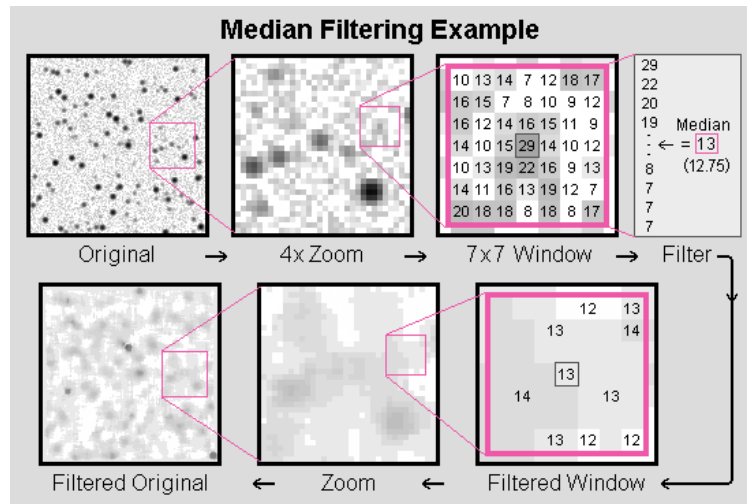


Рисунок 1.9 – Приклад роботи медіанного фільтру

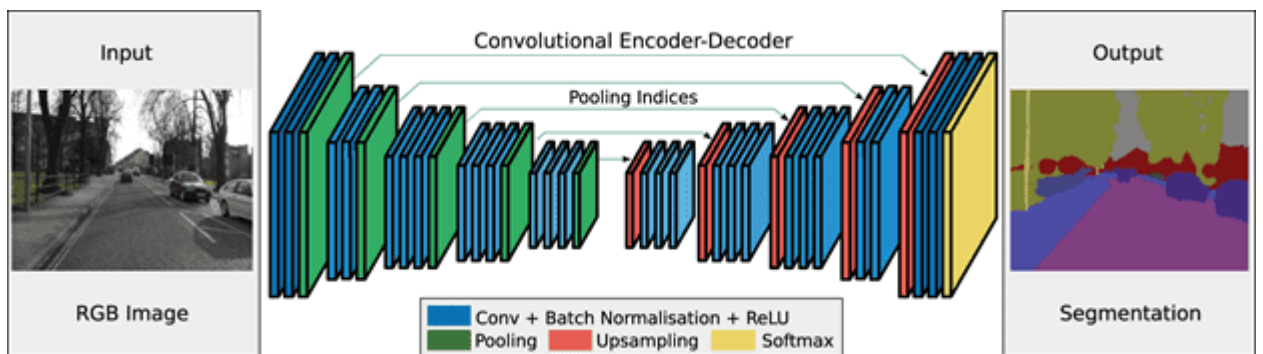


Рисунок 1.10 – Приклад роботи сегментації зображення

Використання цих методів дозволяє забезпечити високу точність виявлення дефектів і стабільність результатів.

Обґрунтування вибору методів полягає в їх ефективності та доступності для реалізації в рамках використовуваного програмного забезпечення. Гауссовий і медіанний фільтри забезпечують ефективне видалення шумів без істотних спотворень корисної інформації. Методи сегментації та аналізу контурів дозволяють точно визначити межі та характеристики дефектів, що важливо для задач, поставлених у роботі.

Таким чином, вибір методів обробки зображень базується на їх здатності забезпечувати високу якість аналізу та відповідати конкретним вимогам проекту.

2 АНАЛІЗ СТРУКТУРИ ПРОГРАМИ

2.1 Вибір мови програмування

За для аналізу зображень мікроструктурованих оптичних волокон важливо розглянути різні мови програмування, щоб визначити найбільш доречною для виконання даної роботи.

Кожна мова має свої особливості, які можуть вплинути на ефективність розробки та продуктивність програми.

Python:

- відома своєю легкістю у вивченні та читанні, а також великою кількістю бібліотек для обробки зображень та машинного навчання;
- широко використовується в наукових обчисленнях, веб-розробці та штучному інтелекті;
- продуктивність може бути недостатньою для завдань, які вимагають обробки великих обсягів даних у реальному часі. Це може обмежити її використання для аналізу зображень оптичних волокон, де продуктивність є критично важливою.

C++:

- пропонує високу продуктивність та можливості керування пам'яттю, що робить її популярною для розробки вбудованих систем, ігор та операційних систем;
- дозволяє отримати найкращу продуктивність, але складність синтаксису та довший час розробки можуть стати перешкодою для швидкого створення прототипів та кастомізації проекту.

C#:

- широко використовується на платформі Microsoft для створення десктопних додатків, забезпечує хороший баланс між простотою та потужністю;

- обмежена крос-платформенна сумісність може бути недоліком, оскільки більшість C# використовується в Windows, хоча .NET Core покращує ситуацію з крос-платформенністю.

Go:

- фокусується на швидкості та простоті синтаксису. Часто цю мову використовують для створення веб-серверів та мікросервісів, вона забезпечує ефективне управління паралелізмом;

- молодша екосистема та обмежений набір бібліотек для обробки зображень і машинного навчання порівняно з Java і Python можуть бути обмежувальними факторами.

На основі розглянутих мов був зроблен висновок що Java є найкращим вибором для цієї роботи, виходячи з її цілей. Вона пропонує високу портативність завдяки віртуальній машині Java (JVM), яка дозволяє запускати додатки на різних платформах без зміни коду. Має широкий спектр бібліотек для обробки зображень і машинного навчання, таких як OpenCV і DeepLearning4j. Крім того, добре підходить для інтеграції з такими базами даних, як PostgreSQL, що важливо для зберігання та аналізу результатів обробки.

Java також пропонує баланс між продуктивністю, масштабованістю і простотою розробки, що робить її найкращим вибором для реалізації складних алгоритмів аналізу зображень і машинного навчання. Використання Java дозволяє досягти високої якості результатів з мінімальними витратами на адаптацію та супровід коду, забезпечуючи стабільність та надійність програмного забезпечення.

Отже, на основі аналізу мов програмування можна зробити висновок, що Java є найкращим вибором для цього проекту. Вона надає всі необхідні інструменти та можливості для ефективної обробки зображень, інтеграції з

базами даних та реалізації складних алгоритмів машинного навчання, що робить її ідеальним рішенням для поставлених завдань.

2.2 Структура бекенду

Для забезпечення ефективної роботи проекту був створений бекенд, який обробляє зображення, зберігає результати аналізу і взаємодіє з базою даних. Структура бекенду включає кілька ключових компонентів для забезпечення високої продуктивності та надійності системи.

Сервер є ядром бекенду і відповідає за прийом запитів користувачів, обробку даних і взаємодію з іншими компонентами системи. Вибір технології для серверної частини є критично важливим, оскільки вона повинна забезпечувати високу продуктивність і масштабованість. У цьому проекті використовується мова Java з її потужною екосистемою та можливістю інтеграції з різними бібліотеками та інструментами.

Основними компонентами серверної є декілька компонентів. Розглянемо їх усі.

Контролери, що отримують HTTP-запити від клієнтів і перенаправляють їх на відповідні сервіси для обробки (рис. 2.1);

```
1  @RestController
2  @RequestMapping("/api")
3  public class ImageController {
4
5      @Autowired
6      private ImageService imageService;
7
8      @PostMapping("/upload")
9      public ResponseEntity<String> uploadImage(@RequestParam("file") MultipartFile file) {
10         String result = imageService.processImage(file);
11         return new ResponseEntity<>(result, HttpStatus.OK);
12     }
13
14     @GetMapping("/results")
15     public ResponseEntity<List<AnalysisResult>> getResults() {
16         List<AnalysisResult> results = imageService.getResults();
17         return new ResponseEntity<>(results, HttpStatus.OK);
18     }
19 }
```

Рисунок 2.1 – Компонент серверу – Контролер

Сервіси, що містять бізнес-логіку та забезпечують основні функції додатку, такі як обробка зображень та аналіз даних (рис. 2.2);

```

11 @Service
12 public class ImageService {
13
14     @Autowired
15     private ImageProcessing imageProcessing;
16
17     @Autowired
18     private FiberClassification fiberClassification;
19
20     @Autowired
21     private ImageRepository imageRepository;
22
23     public String processImage(MultipartFile file) {
24         try {
25             File savedFile = new File("uploads/" + file.getOriginalFilename());
26             file.transferTo(savedFile);
27
28             Mat image = Imgcodecs.imread(savedFile.getAbsolutePath());
29
30             Mat processedImage = imageProcessing.applyMedianFilter(image);
31             processedImage = imageProcessing.equalizeHistogram(processedImage);
32             processedImage = imageProcessing.regionSplitting(processedImage);
33
34             int defectTypeId = fiberClassification.classify(processedImage);
35
36             ImageResult result = new ImageResult();
37             result.setFilePath(savedFile.getAbsolutePath());
38             result.setDefectTypeId(defectTypeId);
39             imageRepository.save(result);
40
41             return "Image processed and analyzed successfully. Defect Type ID: " + defectTypeId;

```

Рисунок 2.2 – Компонент серверу - Сервіс

Сховища, що відповідають за взаємодію з базою даних, виконання запитів на зберігання та отримання даних (рис. 2.3).

```

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface ImageRepository extends JpaRepository<ImageResult, Long> {
}

```

Рисунок 2.3 – Компонент серверу - Сховище

Для обробки зображень використовується бібліотека OpenCV. Основні завдання, що виконуються на цьому етапі, включають фільтрацію шуму, гістограмне вирівнювання та сегментацію зображення (рис 2.4).

```
20 public static Mat regionSplitting(Mat image) {
21     Mat gray = new Mat();
22     Imgproc.cvtColor(image, gray, Imgproc.COLOR_BGR2GRAY);
23
24     Mat binary = new Mat();
25     Imgproc.threshold(gray, binary, 0, 255, Imgproc.THRESH_BINARY_INV + Imgproc.THRESH_OTSU);
26
27     Mat labels = new Mat();
28     int nLabels = Imgproc.connectedComponents(binary, labels);
29
30     Mat colorLabels = Mat.zeros(image.size(), image.type());
31     Scalar[] colors = new Scalar[nLabels];
32     for (int i = 0; i < nLabels; i++) {
33         colors[i] = new Scalar(Math.random() * 255, Math.random() * 255, Math.random() * 255);
34     }
35
36     for (int row = 0; row < labels.rows(); row++) {
37         for (int col = 0; col < labels.cols(); col++) {
38             int label = (int) labels.get(row, col)[0];
39             if (label == 0) continue;
40             colorLabels.put(row, col, colors[label].val);
41         }
42     }
43
44     return colorLabels;
45 }
```

Рисунок 2.4 – Приклад методу обробки зображення, “Розділення на регіони”

Модуль машинного навчання відповідає за класифікацію дефектів на зображеннях. Використовується бібліотека Deeplearning4j, яка надає можливість створювати і використовувати нейронні мережі для аналізу даних. Модель нейронної мережі навчається на даних про дефекти, а потім використовується для аналізу нових зображень (рис. 2.5).

```

40 public String getDefectDescription(int defectTypeId) {
41     String sql = "SELECT description FROM defect_types WHERE id = ?";
42     try (Connection conn = Database.connect();
43         PreparedStatement pstmt = conn.prepareStatement(sql)) {
44         pstmt.setInt(1, defectTypeId);
45         ResultSet rs = pstmt.executeQuery();
46         if (rs.next()) {
47             return rs.getString("description");
48         }
49     } catch (SQLException e) {
50         e.printStackTrace();
51     }
52     return "Description not found.";
53 }
54
55 public String getDefectImpact(int defectTypeId) {
56     String sql = "SELECT influence_dispersion, influence_attenuation FROM defects WHERE defect_type_id = ?";
57     try (Connection conn = Database.connect();
58         PreparedStatement pstmt = conn.prepareStatement(sql)) {
59         pstmt.setInt(1, defectTypeId);
60         ResultSet rs = pstmt.executeQuery();
61         if (rs.next()) {
62             return "Influence on Dispersion: " + rs.getString("influence_dispersion") +
63                 ", Influence on Attenuation: " + rs.getString("influence_attenuation");
64         }
65     } catch (SQLException e) {
66         e.printStackTrace();
67     }
68     return "Impact details not found.";
69 }

```

Рисунок 2.5 – Приклад машинного навчання

База даних PostgreSQL забезпечує надійне зберігання результатів аналізу та метаданих про зображення і дефекти (рис. 2.6).

Структура бази даних включає наступні таблиці:

- fibre_samples, зберігає інформацію про зразки волокон, включаючи ідентифікатори та шляхи зображень;
- defect_types, містить опис різних типів дефектів, їх характеристики та вплив на оптичні властивості волокна;
- analysis_results, зберігає результати аналізу, включаючи ідентифікатори та опис дефектів.

Бекенд взаємодіє з клієнтською частиною через REST API, забезпечуючи обмін даними між користувачем і сервером. Контролери обробляють запити на завантаження зображень, виконують аналіз і отримують результати, забезпечуючи швидкий доступ до функціоналу.

```

7 public class Database {
8
9     private static final String URL :string = "";
10    private static final String USER :string = "";
11    private static final String PASSWORD :string = "";
12
13    public static Connection connect() throws SQLException {
14        return DriverManager.getConnection(URL, USER, PASSWORD);
15    }
16
17    public static void saveFiberSample(String fiberType, String imagePath) {
18        String sql = "INSERT INTO fiber_samples(fiber_type_id, image_path) VALUES ((SELECT id FROM fiber_types WHERE name = ?), ?)";
19        try (Connection conn = connect();
20            PreparedStatement pstmt = conn.prepareStatement(sql)) {
21            pstmt.setString(1, fiberType);
22            pstmt.setString(2, imagePath);
23            pstmt.executeUpdate();
24        } catch (SQLException e) {
25            System.out.println(e.getMessage());
26        }
27    }
}

```

Рисунок 2.6 – Приклад коду взаємодії з базою даних

2.3 Структура фронтенду

Інтерфейс є важливою частиною програмного забезпечення або веб-додатку, оскільки він відповідає за представлення користувацького інтерфейсу та забезпечення взаємодії з користувачем. Фронтенд визначає, як інформація та функціональність відображається на екрані комп'ютера.

Основні завдання фронтенду включають:

- структурування та стилізація елементів на веб-сторінці або в додатку, включаючи текст, зображення, кнопки, форми та інші компоненти;
- надання інструментів для взаємодії користувача з додатком, такі як кнопки, поля введення, меню та інші елементи, які дозволяють користувачеві взаємодіяти з програмою;
- обробка таких подій, як натискання кнопок, наведення миші, введення даних та інші дії користувача;
- надсилання запиту на сервер для отримання даних, які потім відображаються на сторінці.

Для реалізації фронтенду в цьому проекті був обраний React як основний фреймворк. React – це бібліотека користувацького інтерфейсу JavaScript з відкритим кодом, розроблена компанією Facebook. Вона дозволяє створювати інтерактивні інтерфейси та спрощує розробку складних додатків завдяки компонентному підходу.

Технології які були використані у цьому проекті:

- react, дозволяє створювати динамічні та інтерактивні веб-додатки за допомогою компонентів, що полегшує розробку та підтримку коду;
- axios, бібліотека для виконання HTTP-запитів, що забезпечує зручний спосіб взаємодії з бекендом та отримання даних з сервера;
- bootstrap, CSS-фреймворк, який забезпечує адаптивний і зручний інтерфейс, що підтримує сучасні стандарти веб-дизайну.

Основні компоненти проекту, які забезпечують взаємодію з користувачем;

- завантаження зображень, яке містить форму для вибору файлу та кнопку для завантаження зображення на сервер. Компонент використовує Axios для надсилання зображення на сервер за допомогою HTTP-запиту;
- результати аналізу, що отримують дані з сервера і відображають список проаналізованих зображень з описом знайдених дефектів. Компонент використовує Axios для отримання даних з бекенду через API;

Для здійснення HTTP-запитів до бекенду був обраний ApiService бо він містить методи для завантаження файлів та отримання результатів аналізу.

3 АЛГОРИТМ РОБОТИ ПРОГРАМИ

3.1 Блок-схема

База даних складається з 3 таблиць – зразки волокон, типи дефектів та результати аналізу. Система дозволяє завантажувати зображення для аналізу, зберігати результати та переглядати інформацію про дефекти. Для взаємодії з базою даних використовується REST API, що забезпечує зручну та ефективну роботу з даними.

Після завантаження зображення користувач бачить сторінку результатів аналізу, на якій відображаються виявлені дефекти та детальна інформація про них.

Алгоритм програми включає наступні кроки:

- користувач завантажує зображення мікроструктурованого оптичного волокна через інтерфейс;
- зображення передається на сервер для обробки;
- сервер виконує обробку зображення за допомогою бібліотеки OpenCV, включаючи фільтрацію шуму, вирівнювання гістограми та сегментацію;
- машинне навчання, включаючи модель на основі нейронної мережі, аналізує зображення і виявляє дефекти;
- результати аналізу зберігаються в базі даних, включаючи інформацію про тип дефекту та його характеристики;
- користувач може переглянути результати аналізу через інтерфейс, який відображає всі знайдені дефекти;

Блок-схема – це графічне представлення алгоритму або процесу, яке використовує блоки для представлення кроків або дій і стрілки для відображення взаємозв'язків між ними. Блок-схеми використовуються для візуального представлення порядку виконання завдань або послідовності

подій у програмуванні, інженерії, бізнес-процесах та інших сферах. Алгоритм показаний на рисунку 3.1.

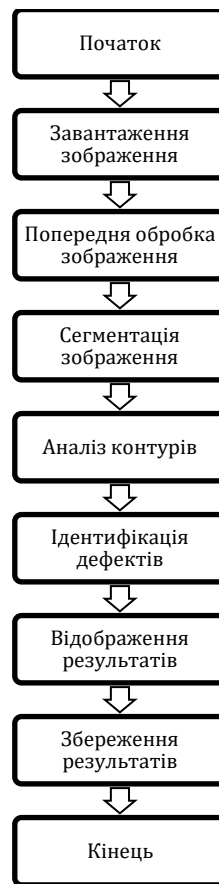


Рисунок 3.1 – Блок-схема

3.2 Архітектура MVC

Архітектура MVC (Model-View-Controller) - це фундаментальний патерн проектування, який використовується для організації програмного коду шляхом поділу його на три основні компоненти: модель, представлення та контролер. Цей патерн забезпечує чіткий розподіл обов'язків і допомагає краще керувати складними програмними проектами (рис. 3.2).

Модель представляє базу даних і основну логіку програми, яка відповідає за роботу з даними, обробку запитів, виконання обчислень і зберігання результатів. У цьому проекті модель містить компоненти для роботи

з базою даних, а саме для зберігання результатів аналізу зображень та інформації про знайдені дефекти.

Представлення відповідає за інтерфейс користувача, тобто всі візуальні елементи, з якими він взаємодіє. У цій роботі представлення включає в себе веб-інтерфейс, за допомогою якого користувач може завантажувати зображення для аналізу, переглядати результати обробки та взаємодіяти з додатком.

Контролер у цій роботі є прошароком між моделлю і представленням, який реагує на дії користувача, обробляє вхідні дані і викликає відповідні методи моделі для оновлення даних і відображення результатів. Контролер обробляє запити від веб-інтерфейсу, передає їх моделі для обробки і повертає результати для відображення на екрані.



Рисунок 3.2 – Архітектура MVC

У патерні MVC контролер і представлення залежать від моделі, а модель не залежить від них. Контролер містить логіку отримання даних з моделі та передачі їх до представлення. Контролер обробляє запити, згенеровані моделлю View, і передає їх моделі, де зберігаються всі дані. Модель змінюється відповідно до запиту. Представлення "дізнається" про необхідні зміни, оскільки воно підписане на події моделі і відображає нові дані.

У проекті чітко визначені компоненти контролера та моделі для взаємодії з базою даних.

Контролер відповідає за обробку запитів користувача, взаємодію з моделлю для отримання або модифікації даних і повернення результатів у подання. Він визначає, які дані отримувати з моделі і як їх відображати.

Модель відповідає за зберігання та обробку даних. Вона містить всю бізнес-логіку і взаємодіє з базою даних. У роботі модель обробляє дані зображення, аналізує його на наявність дефектів і зберігає результати в базі даних.

Представлення відповідає за представлення даних користувачеві та взаємодію з ним. Він отримує дані від контролера і генерує користувацький інтерфейс, який забезпечує зручний спосіб взаємодії з додатком.

Під час роботи програми користувач взаємодіє з інтерфейсом, реалізованим у представленні. Представлення надсилає запити до контролера, який обробляє їх, взаємодіє з моделлю для отримання або модифікації даних і повертає результати назад до представлення. Представлення оновлюється відповідно до отриманих даних і відображається користувачеві.

На рисунках 3.3 та 3.4 показано схематичне представлення контролера та моделі в проекті.

Контролер
- Обробка запитів
- Взаємодія з моделлю
- Повернення даних

Рисунок 3.3 – Контролер

Модель
- Збереження даних
- Обробка даних
- Взаємодія з базою даних

Рисунок 3.4 – Модель

3.3 Структура роботи штучного інтелекту

У даній роботі використовується згортова нейронна мережа (CNN) для автоматизації ідентифікації дефектів на зображеннях мікроструктурованих оптичних волокон. Головною метою є точна класифікація та виявлення дефектів. Модель навчається на попередньо розміченому наборі даних, який включає зображення з різними типами дефектів та без них. Попередня обробка зображень, така як зміна розміру, нормалізація та збільшення даних, покращує загальну здатність моделі.

Архітектура ResNet (Residual Network) дозволяє ефективно навчати глибокі моделі, усуваючи проблему зникнення градієнтів. Процес навчання включає мінімізацію функції втрат, яка вимірює різницю між прогнозованими та фактичними мітками дефектів. Після навчання модель аналізує нові зображення, використовуючи згорткові та повністю зв'язані шари для виділення функцій та класифікації. На виході модель видає ймовірність приналежності зображення до класів: “без дефектів” або одного з типів дефектів, а також локалізує області дефектів для подальшого аналізу.

Для покращення прогнозів модель інтегрована з методами аналізу зображень, такими як розмиття за Гауссом, медіанний фільтр, метод розділення регіонів та контурний аналіз. Ці методи допомагають згладжувати зображення, зменшувати шум, виділяти краї та локалізувати дефекти, підвищуючи точність класифікації.

Використання фреймворків TensorFlow і Keras для побудови та навчання моделі, а також OpenCV для обробки зображень та оцінки продуктивності, забезпечує ефективне вирішення задачі автоматичного виявлення та класифікації дефектів на зображеннях мікроструктурованих оптичних волокон. Архітектура ResNet забезпечує високу точність, стійкість до перенавчання та здатність до узагальнення нових даних..

3.3 Структура бази даних

База даних проекту складається з п'яти основних таблиць: Користувачі, Зображення, Дефекти, Результати та Зображення без дефектів. Кожна таблиця має свою специфічну функцію та структуру для забезпечення зберігання та обробки даних.

Таблиця користувачів зберігає інформацію про користувачів, включаючи унікальні ідентифікатори, імена, адреси електронної пошти та хешовані паролі.

Таблиця зображень зберігає інформацію про завантажені зображення, включаючи унікальні ідентифікатори, імена файлів та шляхи до них.

У таблиці дефектів зберігається інформація про можливі типи проблем, включаючи їхні унікальні ідентифікатори та описи.

Таблиця результатів містить інформацію про результати аналізу зображень, включаючи ідентифікатори зображень, типи дефектів і додаткову інформацію про вплив на дисперсію і загасання сигналу.

Таблиця зображень без дефектів зберігає інформацію про завантажені зображення, включаючи унікальні ідентифікатори та шляхи до них.

Така структура бази даних забезпечує ефективну організацію та зберігання даних, що дозволяє системі швидко виконувати операції аналізу зображень.

У таблиці 3.5, users, будуть наведені дані структури та зв'язок з іншими таблицями, де:

- id, це унікальний ідентифікатор, який генерується PostgreSQL і необхідний для впізнавання користувача та зв'язку таблиці users з іншими таблицями;
- name поле відповідає за ім'я користувача;
- password, пароль користувача, зберігається у хешованому форматі; для хешування використовується бібліотека bcrypt;
- email - електронна пошта користувача; це поле має валідацію за допомогою регулярних виразів у JavaScript;
- token потрібен для аутентифікації користувача, який генерується під час логіна і зберігається для перевірки при подальших запитах.

Таблиця 3.5 – Склад таблиці users

Id	Name	Email	Password	Token
1	Jhon	Jhon_Example@gmail.com	passForJhon1	token1
2	Jane	jane_Example@gmail.com	Pass_For_Jane@	token2
3	Mike	mIke_E_xample@gmail.com	PassFor!M!ike	token3

Таблиця users з'єднана з таблицею images через поле user_id. Це забезпечує зв'язок між користувачами та завантаженими ними зображеннями.

У таблиці 3.6, images, будуть наведені дані структури та зв'язок з іншими таблицями, де:

- image_id - унікальний ідентифікатор зображення, згенерований PostgreSQL за допомогою вбудованої функції UUID, це поле використовується для підключення до таблиць дефектів і результатів;
- file_name - ім'я файлу зображення;

- file_path - шлях до файлу зображення;
- user_id - зовнішній ключ, який посилається на власника цього зображення у таблиці users.

Таблиця images з'єднана з таблицями defects, no_defect_results та results через поле image_id. Це забезпечує зв'язок між зображеннями та результатами їх аналізу.

Таблиця 3.6 – Склад таблиці images

Image_id	user_id	file_name	file_path
1	1	image1.png	/path/to/image1.png
2	2	image2.png	/path/to/image2.png
3	3	image3.png	/path/to/image3.png

У таблиці 3.7, defects, будуть наведені дані структури та зв'язок з іншими таблицями, де:

- defect_id – унікальний ідентифікатор дефекту, який генерується PostgreSQL і використовується для з'єднання з таблицею results;
- defect_description - опис дефекту.

Таблиця defects зв'язується з таблицею results через поле defect_id.

Таблиця 3.7 – Склад таблиці defects

defect_id	defect_description
1	Microcrack
2	Impurity
3	Structural deformation

У таблиці 3.8, result, будуть наведені дані структури та зв'язок з іншими таблицями, де:

- result_id - унікальний ідентифікатор результату, який генерується PostgreSQL;
- image_id - це зовнішній ключ, який посилається на поле image_id у таблиці images;
- defect_id - це зовнішній ключ, який посилається на поле defect_id у таблиці defects;
- impact_on_dispersion - вплив дефекту на дисперсію;
- impact_on_attenuation - вплив дефекту на затухання.

Таблиця 3.8 – Склад таблиці result

result_id	image_id	defect_id	impact_on_dispersion	impact_on_attenuation
1	1	1	Low	Medium
2	2	2	High	Low
3	3	3	Medium	High

У таблиці 3.9, no_defect_result, будуть наведені дані структури та зв'язок з іншими таблицями, де:

- no_defect_result_id, це унікальний ідентифікатор результату, який генерується PostgreSQL;
- image_id, це зовнішній ключ, який посилається на поле image_id у таблиці images;
- analysis_summary, це короткий підсумок аналізу зображення без дефектів.

Таблиця 3.9 – Склад таблиці no_defect_result

no_defect_result_id	image_id	analisis_summery
1	4	No defect found
2	5	Image is clearly of any defect

Під час логіну, здійснюється перевірка токенів для забезпечення безпеки та успішної аутентифікації користувача. Процес включає декілька кроків. Спочатку користувач вводить свої облікові дані (електронну пошту та пароль). Після цього сервер отримує ці дані та шукає відповідний запис у таблиці users. Якщо користувача знайдено, сервер порівнює введений пароль з хешованим паролем у базі даних. Після цього у разі якщо паролі співпадають, сервер генерує новий токен для користувача та оновлює відповідний запис у таблиці users. У кінці токен передається користувачу та використовується для подальших запитів до сервера для аутентифікації.

Нижче приведено приклад коду запиту на логін:

```
const loginUser = async (email, password) => {
  const response = await fetch('/api/login', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({ email, password }),
  });

  const data = await response.json();
  if (data.token) {
    localStorage.setItem('token', data.token);
    return true;
  }
}
```

```
} else {  
    return false;  
}  
};
```

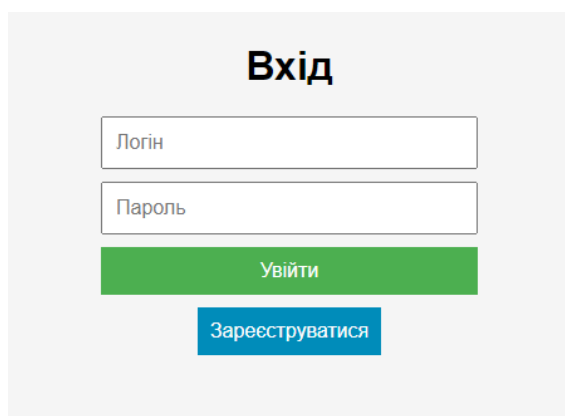
Токен – це рядок, що складається з трьох частин, розділених крапкою. Кожна з цих частин містить інформацію про користувача. Перша частина містить ідентифікатор користувача (id). Друга частина містить секретний ключ. Третя частина містить час життя токена. У цьому проекті токен є вічним. Для хешування використовується бібліотека `bcrypt`.

Хешування – це процес перетворення даних змінної довжини в рядок фіксованої довжини. Результат такого перетворення називається "хеш". Хеш-функції широко використовуються в криптографії, базах даних, структурах даних і в різних алгоритмах для забезпечення швидкого пошуку даних.

4 ЕКСПЕРЕМЕНТАЛЬНІ ВИПРОБУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ

4.1 Тестування функціоналу за допомогою програмного забезпечення

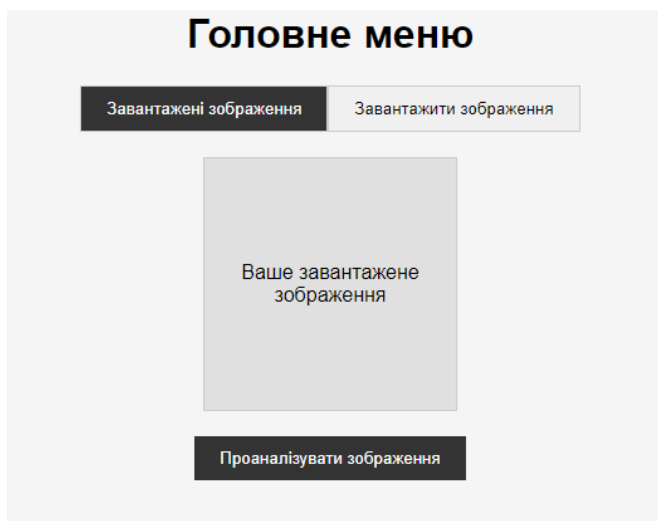
Після запуску програми користувач потрапляє у вікно реєстрації як на рисунку нижче (рис. 4.1).



The screenshot shows a login interface with the title "Вхід". It contains two text input fields labeled "Логін" and "Пароль". Below these fields are two buttons: a green button labeled "Увійти" and a blue button labeled "Зареєструватися".

Рисунок 4.1 – Вікно логіну

Після успішного входу користувач потрапляє у головне меню, де можна як завантажити зображення для аналізу, так і обрати завантаженне (рис. 4.2).



The screenshot shows a main menu interface with the title "Головне меню". It features two buttons at the top: "Завантажені зображення" and "Завантажити зображення". Below these is a large gray box containing the text "Ваше завантажене зображення". At the bottom is a button labeled "Проаналізувати зображення".

Рисунок 4.2 – Головне меню

Завантажимо зображення. Щоб це зробити, треба натиснути на кнопку завантажити зображення та обрати файл (рис. 4.3).

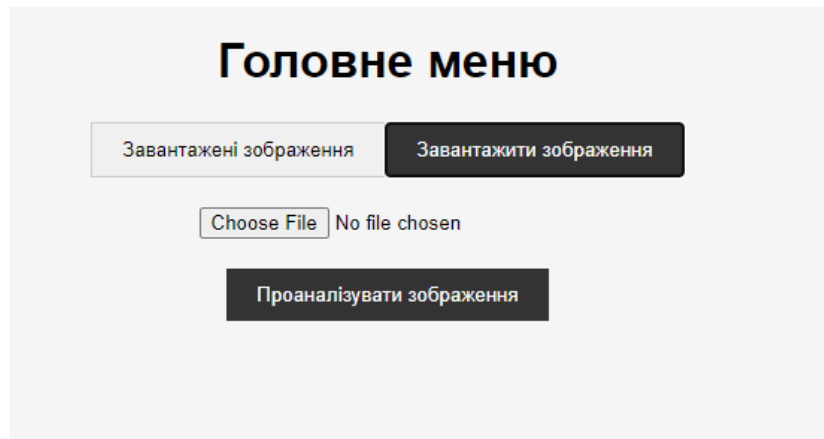


Рисунок 4.3 – Завантажити зображення

Завантажене зображення відображається у інтерфейсі, за бажанням його можна видалити та обрати інше (рис. 4.4).

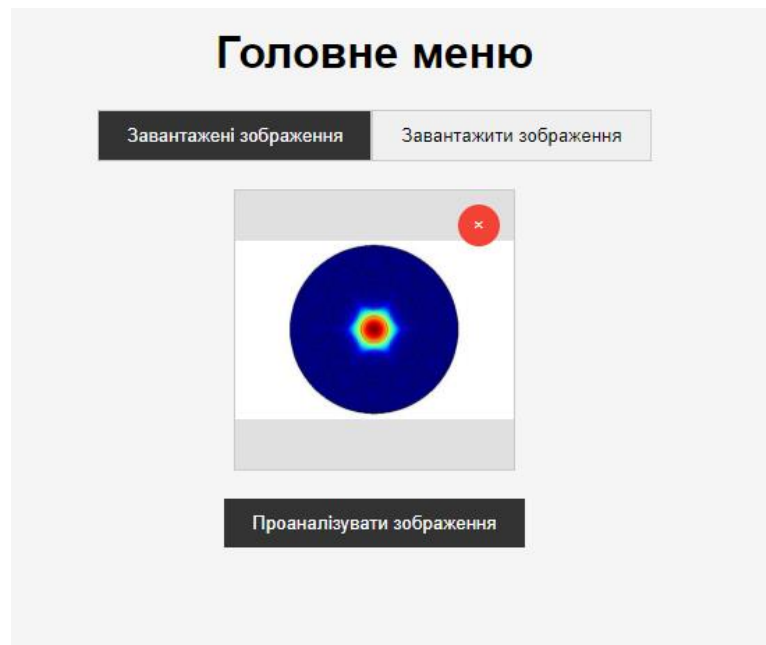


Рисунок 4.4 – Список зображень

Користувач може натиснути на кнопку “Проаналізувати зображення” для аналізу, після чого система виконує обробку зображення та відображає результати аналізу (рис. 4.5).

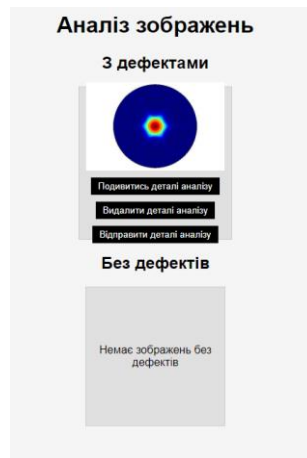


Рисунок 4.5 – Аналіз зображень

Користувач може переглянути результати аналізу(рис. 4.6).

Деталі аналізу

Номер ідентифікатора зображення: 3
Відсоток відмінностей від ідеалу: 13%
Відсоток погрішностей при аналізі: 2%
Тип дефекту: зміна структури волокна
Вплив на якість сигналу: дисперсія сигналу

Закрити

Рисунок 4.6 – Результати аналізу

Користувач може поділитися результатами аналізу або видалити їх. Наприклад, можна поділитися результатами через електронну пошту, як на рисунку нижче (рис. 4.7).

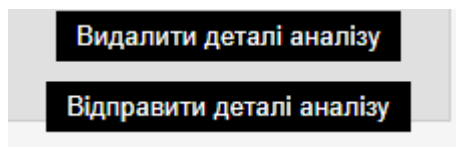


Рисунок 4.7 – Можливість поділитись зображенням або видалити

У головному меню є пункт "Графіки", де можна переглянути графічне представлення аналізу зображень, включаючи розподіл по типам дефектів, проценту кожного дефекту та їх наявності, в залежності від загальної кількості. Також можна переглянути всю історію аналізів та отримати більш розгорнуту інформацію по статистиці (рис. 4.8).



Рисунок 4.8 – Графіки

5 ОХОРОНА ПРАЦІ

У сучасному світі, коли технології та інформаційна безпека стають все більш важливими, важливо не забувати про важливість безпеки в процесі розробки та розвитку програмного забезпечення безпеки. Обеспечение ефективних підходів к защите процесса создания и эксплуатации добавок для автоматизации анализа дефектов микроструктурированных световодов и необходимого складского успеха их функционирования. Охорона здоров'я і безпеки жителів.

5.1 Цілі безпеки

Аналіз ризиків та оцінка результатів безпеки є критичним етапом забезпечення безпеки та стабільності програмного забезпечення. Цей процес включає в себе виявлення потенційних загроз безпеки даних, можливих випусків систем безпеки програмного забезпечення та ризиків, пов'язаних із здоров'ям клієнтів. На основі отриманих результатів формулюються та реалізуються різні підходи до забезпечення безпеки, такі як розробка механізмів контролю доступу, безпечне шифрування даних, регулярні аудити безпеки та розвиток підтримки клієнтів для безпечної доставки додаткових даних. Це дозволяє знизити ризик безпеки нових інцидентів і забезпечує надійність і конфіденційність даних.

5.2 Організація робочого простору

Організація робочого простору включає в себе розвиток принципів ергономіки при проектуванні інтерфейсів, щоб мінімізувати ризик травм або стресу для працівників. Також важливо забезпечити достатню вентиляцію та освітлення в робочій зоні для підтримки здоров'я працівників. Ефективна організація робочого процесу включає в себе розробку чотирьох інструкцій

для правильної роботи програмного забезпечення та розвиток навиків ручного доступу до необхідної інформації та інструментам, що сприяє підвищенню продуктивності. продуктивність і забезпечення комфортної роботи ума.

5.3 Аналіз безпеки

Моніторинг та аналіз безпеки - важливий крок у забезпеченні надійності та безпеки програмного забезпечення. Цей процес включає в себе постійне вдосконалення процедур безпеки та аналіз інцидентів для оперативного виявлення та пом'ягшення потенційних розливів. Регулярні аудити безпеки та оновлення підходів до забезпечення безпеки до нових загроз допомагають забезпечити стійкість безпеки програмного забезпечення до потенційних атак і зберегти безпеку даних. Це підвищує рівень кращих даних і забезпечує безпеку тих, хто не має доступу до своїх даних.

5.4 Аудит документації та безпеки

Процес документування та аудиту безпеки включає оцінку сили потенційних загроз і воздействий. Вірогідність конкретної загрози визначає ймовірність конкретної загрози, при цьому можна оцінити масштаб шкоди, яка може нанести цю загрозу. Це дозволяє виявити найбільш критичні ризики та визначити пріоритети міри безпеки для їх усунення або скорочення потоку. Учет цих факторів допомагає забезпечити адекватний захист добавок та їх компонентів від можливих загроз.

Ефективна реалізація найважливіших підходів для забезпечення стабільної та безпечної роботи програмного забезпечення для автоматизації аналізу дефектів мікроструктурованих оптичних волокон, зміни ризику виникнення проблем з безпекою даних та забезпечення здоров'я та безпеки персоналу.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи розробив систему автоматизації ідентифікації деформацій функціональних електронних елементів на мікроструктурованих оптичних волокнах. Були виконані наступні завдання:

- аналіз теоретичної частини мікроструктурованих оптичних волокон та методів їх дослідження;
- розглянути та проаналізувати різні методи виявлення дефектів з використанням машинного навчання;
- описати важливість та методи точного вимірювання деформації в оптичних волокнах;
- продемонструвати важливість автоматизації процесів виявлення дефектів і деформації;
- розроблено програмний код для автоматизації процесу виявлення деформацій;
- проведено експериментальні випробування розробленої моделі в імітаційному середовищі;

Основне завдання модуля автоматизації виконано - користувач має всі важливі інструменти, які дозволяють швидко та ефективно виявляти деформації та дефекти в мікроструктурованих оптичних волокнах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с
2. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти спеціальності 172 Телекомунікації та радіотехніка, освітньої програми «Інтелектуальні технології засобів радіоелектроніки» / Упоряд. Н. П. Демська, В. В. Євсєєв, О. М. Замірець, В. В. Невлюдова, Ю. М. Олександров. Харків : ХНУРЕ, 2022. 38 с.
3. Положення про академічну доброчесність [Електронний ресурс]: Наказ ХНУРЕ від 02 лютого 2021 р. No 50. – Режим доступу: https://nure.ua/wpcontent/uploads/Main_Docs_NURE/polozhennja-pro-akademichnudobrochesnist.pdf.
4. Xu, M. G., et al. "Discrimination between strain and temperature effects using dual-wavelength fibre grating sensors." *Electronics Letters* 30.13 (1994): 1085-1087.
5. Lee, B., et al. "Interferometric fiber optic sensors." *Sensors* 12.3 (2012): 2467-2486.
6. Rao, Y. J. "In-fibre Bragg grating sensors." *Measurement science and technology* 8.4 (1997): 355.
7. Kersey, A. D., et al. "Fiber grating sensors." *Journal of lightwave technology* 15.8 (1997): 1442-1463.
8. Othonos, A., and K. Kalli. "Fiber Bragg gratings: fundamentals and applications in telecommunications and sensing." Artech House, 1999.
9. Jiang, Y., et al. "Recent Advances in Microstructured Optical Fiber-Based Sensors for Biochemical Applications." *Sensors* 2017, 17(2), 183.

10. Xu, M., et al. "Optofluidics in Microstructured Optical Fibers." *Micromachines* 2018, 9(4), 145.
11. Li, Y., et al. "Progress on the Microcavity Lasers Based on Microstructured Optical Fiber." *Electronics* 2023, 12(8), 1761.
12. Smith, R., et al. "Advances in Microstructured Optical Fibers for Environmental Sensing." *Journal of Lightwave Technology* 2019, 37(6), 1452-1463.
13. Zhao, X., et al. "Recent Developments in Microstructured Optical Fiber Sensors for Industrial Applications." *Applied Sciences* 2020, 10(12), 4147.
14. Chen, L., et al. "Microstructured Optical Fibers for Medical Diagnostics." *Biomedical Optics Express* 2021, 12(3), 789-801.
15. Wang, T., et al. "Microstructured Optical Fiber Sensors for Structural Health Monitoring." *Optics Express* 2018, 26(4), 4789-4800.
16. Liu, H., et al. "Development and Applications of Microstructured Optical Fiber-Based Biosensors." *IEEE Sensors Journal* 2016, 16(10), 3847-3853.
17. Стищенко Т.Є., Пронюк Г.В., Сердюк Н.М., Хондак І.І. «Безпека життєдіяльності»: навч. посібник / Т.Є Стищенко, Г.В. Пронюк, Н.М. Сердюк, І.І. Хондак. – Харків: ХНУРЕ, 2018. – 336 с.3. Кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки (КІТАР) [Електроний ресурс]. – Режим доступу: <https://nure.ua/department/kafedra-komp-yuterno-integrovanih-tehnologiy-avtomatizatsiyi-ta-mehatroniki-kitam> – 19.05.20