Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп’ютеризованих технологій
(повна назва)

Кафедра Комп’ютерно-інтегрованих технологій, автоматизації та мехатроніки
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

(тема)
Автоматизована система виявлення браку деталей з використанням методу розпізнавання образів `

Виконав: студент 2 курсу, гр. КТРСм-19-1
Стеблянко Б.О.
(прізвище, ініціали)

Спеціальність 151 Автоматизація та комп’ютерно-інтегровані технології

освітньої програми Комп'ютеризовані та робототехнічні системи
(код і повна назва)

Тип програми освітньо-професійна
(повна назва освітньої програми)

Керівник доцент Бабак І.М.
(посада, прізвище, ініціали)

Допускається до захисту
зав. кафедри _____ Невлюдов І.Ш.
(підпис) (прізвище, ініціали)

2020

Харківський національний університет радіоелектроніки

Факультет	Автоматики і комп'ютеризованих технологій
Кафедра	Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
Рівень вищої освіти	другий (магістерський)
Спеціальність	151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми	освітньо-професійна
Освітня програма	Комп'ютеризовані та робототехнічні системи

(код і повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____»_____ 2020 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Стеблянку Богдану Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Автоматизована система виявлення браку деталей з використанням методу розпізнавання образів
затверджена наказом по університету від 02.11. 2020 р. № 1509 Ст.
2. Термін подання студентом роботи до екзаменаційної комісії 09.12.2020 р.
3. Вихідні дані до роботи Система розпізнавання деталей типу обертанням 71 классу. Системи розпізнавання образів на основі OpenCV. Сканування деталей на виявлення браку в умовах виробництва.
4. Перелік питань, що потрібно опрацювати в роботі
4.1 Вступ.
4.2 Аналіз програмних бібліотек розпізнавання образів
4.3 Аналіз методів обробки візуальних образів
4.4 Знаходження контурів і операції з ними
4.5 Удосконалення алгоритму знаходження браку
4.6 Тестування розробленої системи розпізнавання браку
4.7 Перевірка чутливості системи
4.8 Охорона праці під час виконання проєкту
4.9 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 11 с. формату А4

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання	04.11.2020 р.	Виконано
2	Огляд та аналіз інструментів для	06.11.2020 р.	Виконано
	створення програмного забеспечення		
3	Розробка программного забеспечення	11.11.2020 р.	Виконано
4	Проведення єксперемнатльного	28.11.2020 р.	
	аналізу створеної системи		Виконано
5	Оформлення пояснювальної	08.12.2020 р.	Виконано
	записки		
6	Подання роботи до екзаменаційної	16.12.2020 р.	Виконано
	комісії		

Дата видачі завдання 05.10.2020 р.

Студент _____ Стеблянко Б.О.

(підпис) (прізвище, ініціали)

Керівник роботи доцент Бабак І.М.

(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить: 68 с., 25 рис., 2 дод., 15 джерел.

СИТЕМА ВИЯВЛЕННЯ БРАКУ, OPENCV, РОЗПІЗНАВАННЯ ОБРАЗІВ, КАМЕРИ, АВТОМАТИЗАЦІЯ ВИРОБНИЦТВА

Об'єкт дослідження – автоматизація процесу контролю якості на виробництві

Предмет дослідження – метод розпізнавання образів.

Мета атестаційної магістерської роботи – скорочення часу на виявлення браку деталей завдяки використанню автоматизованої системи розпізнавання образів.

Методи дослідження та використане обладнання: для побудови системи були використані дані наукових досліджень в сфері та методи системного аналізу з використанням методів копьютерного зору SHIFT FAST; тестування систем проводилося на персональній IBM-сумісній ЕОМ (ОЗУ – 8192 *Мб*, тактова частота процесору – 3,4 *ГГц*, Обсяг жорсткого диску – 1 *Тб*) з підключеною веб камерою Logitech Webcam C210.

Результати: науковий – проведений аналіз інструментів та методів для опізнавання образів та створенна системи перевірки деталей на брак.

Сфера застосування – використання на автоматизованому виробництві, промислового призначення. З метою прискорення перевірки продуктів на брак.

ABSTRACT

The explanatory note contains: 68 pages, 25 figures, 2 appendices, 15 sources.

MARRIAGE DETECTION SYSTEM, OPENCV, PATTERN RECOGNITION, CAMERAS, PRODUCTION AUTOMATION

The object of study - automation of quality control in production

The subject of research is the method of pattern recognition.

The purpose of the master's thesis is to reduce the time to detect the lack of details with an automated pattern recognition system.

Research methods and equipment used: data from scientific research in the field and methods of system analysis were used to build the system;with the usage of SHIFT and FAST methods of computer vison, systems were tested on a personal IBM-compatible computer (RAM - 8192 MB, processor clock speed - 3.4 GHz, hard disk capacity - 1 TB) with a connected webcam Logitech Webcam C210.

Results: scientific - analysis of tools and methods for pattern recognition and created a system for checking parts for defects.

Scope - use in automated production, industrial use. In order to speed up the inspection of products for lack.

ЗМІСТ

Перелік скорочень…… 6
Вступ…… 8
1 Аналіз сучасних засобів розпізнавання образів…… 9
1.1 Аналіз програмних бібліотек розпізнавання образів…… 9
1.2 Аналіз методів обробки візуальних образів…… 15
1.3 Знаходження контурів і операції з ними…… 16
1.4 Використання розпізнавання образів на виробництві…… 18
1.5 Постановка задач дослідження…… 25
2 Розробка архітектури та програмного забеспечення системи…… 26
2.1 Удосконалення алгоритму знаходження браку …… 26
2.2 Вибір формату збереження контурів…… 32
2.3 Принцип роботи структурна схема программи…… 38
2.4 Висновки по дургому розділу магістерської атестаційної роботи…… 41
3 Експерементальна перевірка системи…… 42
3.1 Тестування розробленої системи розпізнавання браку…… 42
3.2 Перевірка чутливості системи…… 45
3.3 Охорона праці під час виконання проєкту…… 49
3.4 Висновки по третьому розділу магістерської атестаційної роботи…… 52
Висновки…… 53
Перелік джерел посилань…… 54
Додаток А Демонстраційний матеріал у вигляді презентації…… 56
Додаток Б Відомість атестаційної роботи…… 68

ПЕРЕЛІК СКОРОЧЕНЬ

БЗ – база знань

НПАОП – нормативно-правові акти з охорони праці

ПК – персональний комп’ютер

PNG – Portable Network Graphics

JPEG – "Joint Photographic Experts Group

HP – High Precision

BMP – BitMaP

DoG – Difference of Gaussians

ВСТУП

Виявлення браку на виробництві завжди було важливою проблемою у всіх галузях промисловості. Автоматизація виробництва дає змогу виконувати цей процес автоматично, а розвиток сучасних методів розпізнавання образів може дати можливість впровадити методи, що забезпечуватимуть досить високий рівень надійності розпізнавання. Для впровадження методів розпізнавання браку на виробництві має актуальність дослідження методів та бібліотек, щоб обґрунтовано використовувати їх на виробництві.

Моделі розпізнавання образів неоднозначно розглядають потенційні рішення без потрібної кількості початкових параметрів. Проте моделювання візуального світу у всій своїй різноманітності набагато складніше, ніж, скажімо, моделювання голосового апарату , що генерує певні звуки. Моделі, які ми використовуємо при комп'ютерному зорі, зазвичай розробляються в фізиці (оптика, сенсорний дизайн тощо) та комп'ютерній графіці. Обидва ці напрями мають задачу моделювати як об'єкти рухаються та контактують, як приклад світло, що відбивається від їх поверхонь, розсіюється атмосферою, переломлюється через лінзи камери (або людські очі), і нарешті проектується на площину зображення. У комп'ютерному розумінні ми намагаємося описати світ, який ми бачимо в одному або декількох зображеннях і описати його властивості, такі як форма, освітленість та колір. Алгоритми зору настільки схильні до помилок, що часто невелика зміна у пікселях зображення призводить до кардинально інакших результатів.

Таким чином комп’ютерне бачення – це набір методів та алгоритмів для взаємодії з візуальним навколишнім середовищем, ціллю яких є розпізнавання та аналіз графічних об’єктів.

Об'єкт дослідження – автоматизація контролю якості на виробництві.

Предмет дослідження – метод розпізнавання образів.

Мета атестаційної роботи – скорочення часу на виявлення браку деталей завдяки використанню автоматизованої системи розпізнавання образів.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

– проаналізувати принципи роботи методів розпізнавання образів;

– розробити алгоритм системи для виявлення браку;

– обґрунтувати вибір програмних і апаратних засобів;

– розробити програму для розпізнавання образів деталей на виробництві та провести дослідження;

– оформити атестаційну роботу магістра згідно ДСТУ 3008:2015 [1] навчального посібника з дипломного проектування [2], методичних вказівок до випускної кваліфікаційної роботи рівня «Магістр» [3] та положення про протидію академічному плагіату [4].

Матеріали магістерської атестаційної роботи опубліковано в [5].

1 АНАЛІЗ СУЧАСНИХ ЗАСОБІВ РОЗПІЗНАВАННЯ ОБРАЗІВ

1.1 Аналіз програмних бібліотек розпізнавання образів

OpenCV (Open Source Computer Vision Library) являє собою бібліотеку програмного забезпечення для комп'ютерного зору з відкритим вихідним кодом і комп'ютерного навчання. OpenCV була створена для забезпечення загальної інфраструктури додатків для комп'ютерного зору і прискорення використання сприйняття машини в комерційних продуктах. Будучи ліцензованим BSD продуктом, OpenCV спрощує бізнес для використання і модифікації коду. Метою розробки даної бібліотеки є підвищення ефективності обчислень в додатках реального часу. Мова С, на якому була написана бібліотека, є оптимізованим. Бібліотека OpenCV здатна використовувати багатоядерні процесори. У разі, якщо знадобиться автоматична оптимізація на різних апаратних платформах Intel, можливо додаткове придбання бібліотеки IPP, з англійської Integrate Performance Primitives. До складу даної бібліотеки входять процедури з низкорівневою оптимізацією, які можуть застосовуватися для різноманітних алгоритмічних областей. В спектр можливостей OpenCV входить автоматичне застосування IPP під час виконання будь-якої програми. На рис. 1.1 приведена діаграма, що порівнює швидкодію бібліотек OpenCV, OpenCV + IPP, VXL і LTI .

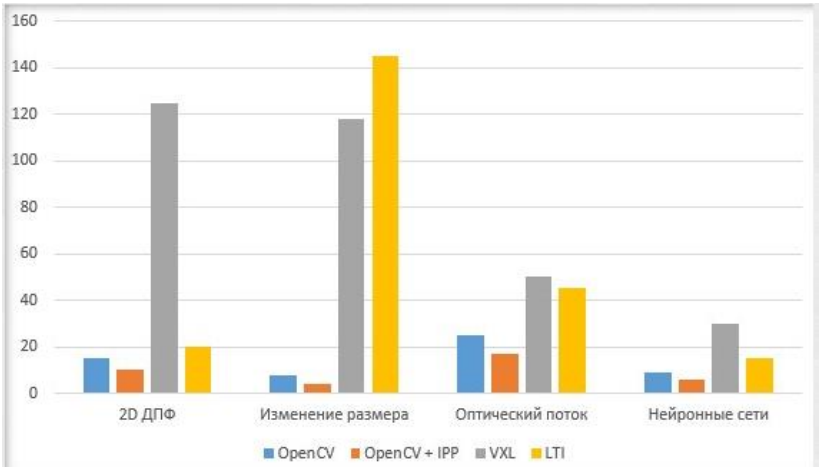


Рисунок 1.1 – Порівняння бібліотек комп'ютерного розпізнавання образів

У бібліотеці понад 2500 оптимізованих алгоритмів, які включають в себе повний набір класичних і сучасних алгоритмів комп'ютерного зору і машинного навчання. Ці алгоритми можуть використовуватися для виявлення і розпізнавання осіб, ідентифікації об'єктів, класифікації дій людини в відео, відстеження руху камери, відстеження рухомих об'єктів, вилучення 3D-моделей об'єктів, створення 3D-хмар точок з стереокамер, зшивання зображень разом для отримання зображення високої роздільної здатності для всій сцени, знайти схожі зображення з бази даних зображень, видаляти червоні очі з зображень, зроблених за допомогою спалаху, стежити за рухами очей, розпізнавати декорації і встановлювати маркери, щоб накладати їх на доповнену реальність і т. д. OpenCV має більше 47 тисяч користувачів спільноти і передбачувана кількість завантажень, що перевищують 14 мільйонів. Бібліотека широко використовується в компаніях, дослідницьких групах і урядових органах .Поряд з відомими компаніями, такими як Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, які використовують бібліотеку, є багато стартапів, таких як Applied Minds, VideoSurf і Zeitera, які широко використовують OpenCV. Розгорнуті застосування OpenCV охоплюють діапазон від зшивання зображень вуличного огляду разом, виявляючи вторгнення в відео спостереження в Ізраїлі, контролюючи обладнання шахт в Китаї, допомагаючи роботам пересуватися і збирати об'єкти в Willow Garage, виявляти нещасні випадки в басейнах в Європі, запускати інтерактивне мистецтво в Іспанії і Нью-Йорку, перевіряючи злітно-посадочні смуги на наявність сміття в Туреччині, перевіряючи етикетки на продуктах на заводах по всьому світу, щоб швидко виявити особа в Японії.

Бібліотека має інтерфейси для C ++, Python, Java і MATLAB і підтримує Windows, Linux, Android і Mac OS. OpenCV орієнтується в основному на додатки в режимі реального часу і використовує переваги команд MMX і SSE, коли вони доступні. В даний час активно розвиваються повнофункціональні інтерфейси CUDA і OpenCL. Існує більше 500 алгоритмів і близько 10-кратного числа функцій, які становлять або підтримують ці алгоритми.

OpenCV написаний на мові C ++ і має шаблонний інтерфейс, який працює без проблем з контейнерами STL. На рис. 1.2 представлена загальна структура проекту OpenCV.

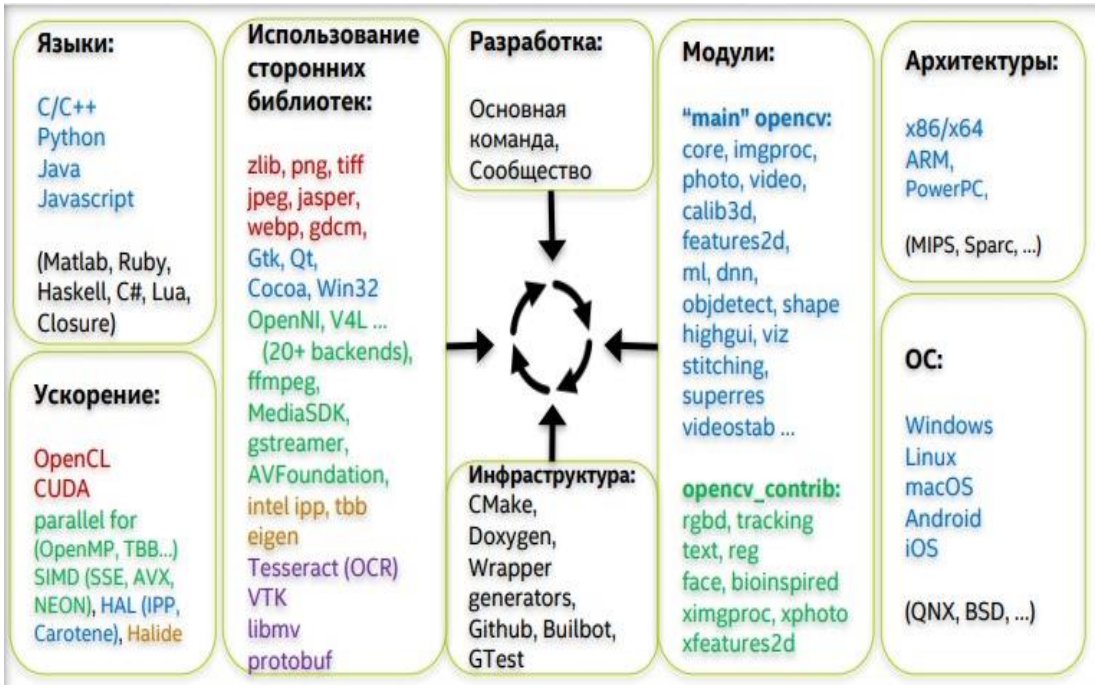


Рисунок 1.2 – Карта проекту OpenCV

AForge.NET є бібліотекою з відкритим вихідним кодом, створеної на мові C#, яка призначена для розробників і дослідників в області комп'ютерного зору. Крім того, в бібліотеці є функціонал для розробників в області штучного інтелекту. Спектр можливостей бібліотеки досить широкий: обробка зображень, нейронні мережі, генетичні алгоритми, нечітка логіка, машинне навчання, робототехніка і багато іншого.

Бібліотека включає кілька основних компонентів. AForge.Imaging - бібліотека підпрограм для обробки зображень і фільтрів. AForge.Vision - бібліотека комп'ютерного зору. AForge.Video - набір бібліотек для роботи з відео. AForge.Neuro - бібліотека для виконання різноманітних дій і операцій з нейронними мережами. AForge.Genetic - бібліотека підпрограм для використання генетичних алгоритмів для вирішення різних завдань. AForge.Fuzzy - бібліотека для роботи з нечіткою логікою. AForge.Robotics –

бібліотека, яка забезпечує підтримку деяких методів, що застосовуються в сфері робототехніки. AForge.MachineLearning – бібліотека для роботи з елементами машинного навчання AForge.NET постійно поліпшується і прогресує. З цієї бібліотеці існує велика кількість прикладів, які демонструють її роботу, і html-документація, яка підтримується в актуальному стані і допомагає розробникам у використанні даного фреймворка. Існує, так само як і у бібліотеки OpenCV, активна спільнота, в якому можна зачерпнути необхідну інформацію, ставлячи питання розробникам, або поділитися власними напрацюваннями. Але, на жаль, кількість учасників цієї спільноти поступається своєю кількістю аналогічним по OpenCV. Ще одним обмеженням на шляху розробника є той факт, що вся документація по бібліотеці написана тільки на англійській мові. Зважаючи на це можливі труднощі у вивченні і освоєнні даного фреймворка. На рис. 1.3 приведена загальна структура бібліотеки AForge.NET.

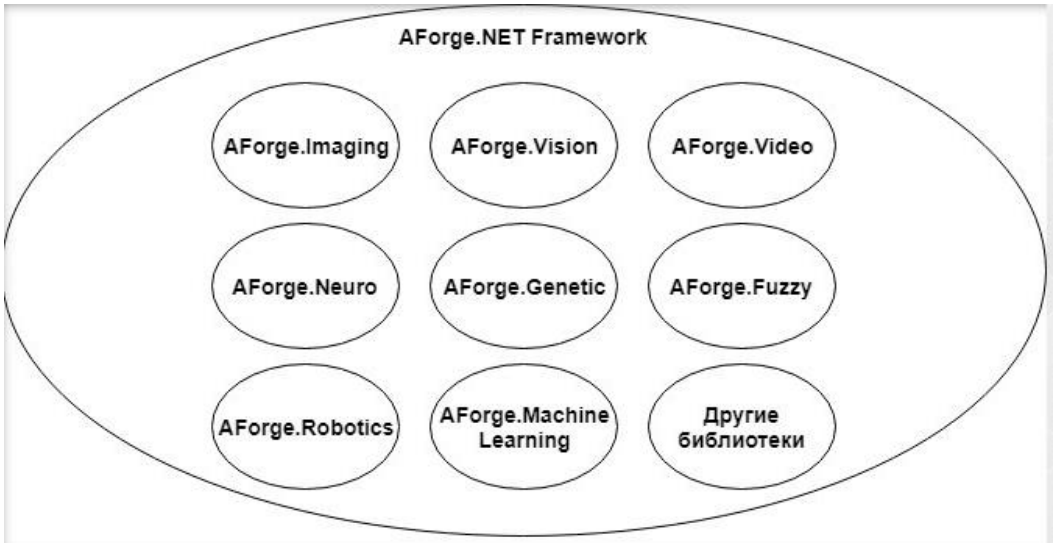


Рисунок 1.3 – Структура фреймворка AForge.NET

VXL це набір бібліотек, написаних на мові C ++, які призначені для наукових досліджень і реалізації технологій комп'ютерного зору. VXL була написана в ANSI / ISO C ++ і призначена для портативних платформ. Бібліотека складається з декількох основних складових: VNL (числа) чисельні алгоритми і контейнери, наприклад, матриці, вектори, оптимізатори

і т.д., VIL (зображення) завантаження, збереження і редагування зображень у багатьох найбільш поширених форматах (також існує можливість роботи з дуже великими зображеннями), VGL (геометрія) – геометрія точок, кривих та інших елементарних об'єктів в одно-, дво- і тривимірному просторах, VSL (вхідний і вихідний потоки), VBL (основні шаблони), VUL (утиліти) – різний функціонал для незалежних платформ. Крім основних бібліотек, що входять до складу VXL, існують і додаткові. Вони відповідають за такі поняття, як чисельні алгоритми, обробка зображень, системи координат, геометрія камери, стерео, маніпуляції з відео потоком, відновлення структури при русі камери, графічний дизайн, функції відстеження, топологія, класифікатори, 3d візуалізація і багато іншого. Особливість бібліотеки полягає в тому, що кожен її компонент може використовуватися окремо, не посилаючись на інші компоненти. Таким чином, в додатку можна використовувати тільки те, що дійсно необхідно. VXL використовується по всьому світу. Бібліотека застосовується в сфері навчання і промисловості, деякі провідні світові експерти в сфері комп'ютерного зору користуються даною бібліотекою. Існує документація по VXL з описом кожного класу і функцій. Однак у даній бібліотеки існує такий же недолік, як і у AForge.NET. Розробнику, що не знає англійську мову буде вельми проблематично освоїти і використовувати даний набір бібліотек. На рис. 1.4 наведено ієрархічну будову ядра VXL .

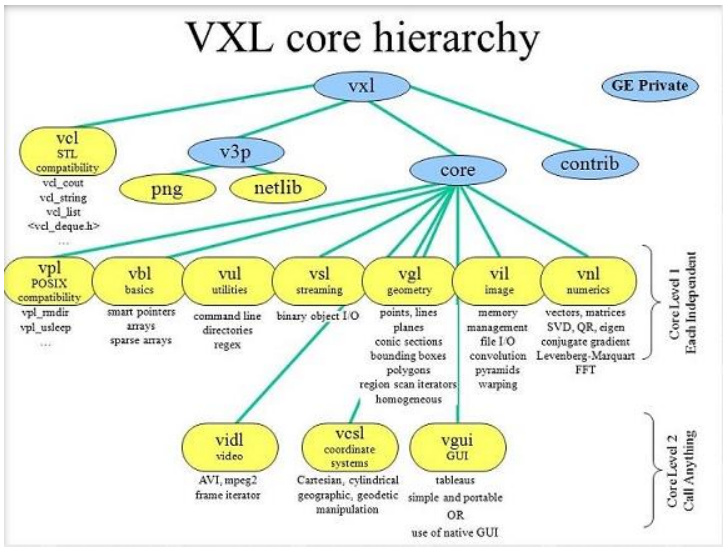


Рисунок 1.4 – Ієрархічна будова ядра VXL

LTI або LTI-lib– об'єктно-орієнтована бібліотека алгоритмів і структур даних. Вона часто застосовується при обробці зображень і в сфері комп'ютерного зору. LTI–lib була розроблена як частина науково-дослідних проектів в області комп'ютерного зору з технологіями робототехніки, розпізнавання об'єктів, голосу і жестів. Основною метою розробки даної бібліотеки є створення об'єктно-орієнтованої бібліотеки на мові C++, що багато в чому спрощувало б використання коду і його обслуговування, але при цьому були б забезпечені швидкі алгоритми, які можна було б використовувати в реальних додатках.

Бібліотека була розроблена з застосуванням GCC (набір компіляторів, застосовуваний для різноманітних мов програмування) під Linux і Visual C++ під Windows NT. Багато класів інкапсулюють Windows / Linux функціональність для того, щоб спростити вирішення системних або апаратних завдань (наприклад, класи для многопоточності і синхронізації, виміру часу і доступ до послідовного порту).

Бібліотека забезпечена документацією , яка підтримується розробниками в актуальному стані і може бути знайдена на головній сторінці проекту. Вся інформація в ній надається англійською мовою, як і в більшій частині інших бібліотек. Бібліотека LTI, так само як і бібліотека OpenCV, є вільно поширюваним програмним забезпеченням, згідно GNU Lesser General Public License. Існує величезна кількість проектів, при розробці яких застосовувалися технології комп'ютерного зору, для яких використовувалася саме бібліотека LTI.

Після детального дослідження бібліотек розпізнавання образів мною була обрана бібліотека OpenCV, на мою думку вона найкраще підходить для створення системи автоматизації контролю якості на виробництві. OpenCV являється най швидшою бібліотекою серед представлених через це швидкість виробництва буде максимальною без збільшення ризику пропуску браку, також можна відмітити гнучкість даної бібліотеки, програма написана на мові C++ з її використанням буде працювати на Windows, Linux, Android і Mac OS, не потребуючи модифікування програмного коду.

1.2 Аналіз методів обробки візуальних образів

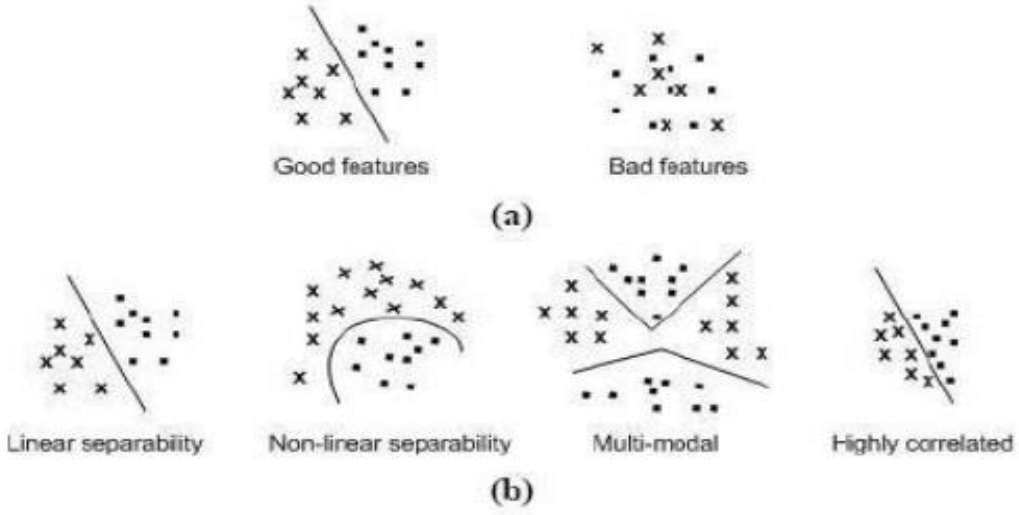
Моделі, які ми використовуємо при комп'ютерному зорі, зазвичай розробляються в фізиці (оптика, сенсорний дизайн тощо) та комп'ютерній графіці. Обидва ці напрями мають задачу моделювати як об'єкти рухаються та контактують, як приклад світло, що відбивається від їх поверхонь, розсіюється атмосферою, переломлюється через лінзи камери (або людські очі), і нарешті проеціюється на квартиру (або вигнутий) площину зображення. У комп'ютерному баченні ми намагаємося описати світ, який ми бачимо в одному або декількох зображеннях і описати його властивості, такі як форма, освітленість та колір. Алгоритми зору настільки схильні до помилок, що часто невелика зміна у пікселях зображення призводить до кардинально інакших результатів.

Таким чином комп’ютерне бачення – це набір методів та алгоритмів для взаємодії з візуальним навколишнім середовищем, ціллю яких є розпізнавання та аналіз графічних об’єктів.

Ознаки (features) можна визначити як будь-який відмітнний аспект, якість або характеристику, які можуть бути символічними (наприклад, колір) чи чисельними (наприклад, висота). Комбінація функцій d представляється як вектор d-розмірного стовпця, який називається вектором властивостей.

D-мірний простір, визначений функцією вектора, називається простором ознак (feature space). Об'єкти представлені у вигляді точок у просторі ознак. Це подання називається графіком розсіювання (scatter plot) .

Шаблон (pattern) визначається як комбінація ознак, характерних для особи. У класифікації шаблон - це пара змінних {x, w}, де x - це сукупність спостережень або ознак (вектор ознак), а w – концепція, що лежить в основі спостереження(label). Якість вектора властивостей пов'язана з його здатністю відрізняти приклади з різних класів (рис. 1.5). Приклади з того ж класу повинні мати подібні значення ознак, а приклади з різних класів мають різні значення ознак



a - відмінність між хорошими (good) та поганими (bad) ознаками; b - властивості ознак за способом розділення: лінійні, нелінійні, мультимодальні, висококорельовані

Рисунок 1.5 – Ознаки (характеристики)

1.3 Знаходження контурів і операції з ними

Контурний аналіз – це один з важливих і дуже корисних методів опису, зберігання, розпізнавання, порівняння та пошуку графічних образів / об'єктів. Контур – це зовнішні обриси (обвід) предмета / об'єкта.

При проведенні контурного аналізу:

–покладається, що контур містить достатню інформацію про форму об'єкта;

–внутрішні точки об'єкта до уваги не беруться.

Вищенаведені положення, зрозуміло, накладають суттєві обмеження на область застосування контурного аналізу, які, в основному, пов'язані з проблемами виділення контуру на зображеннях:

–через однакової яскравості з фоном об'єкт може не мати чіткої межі, або може бути зашумлений перешкодами, що призводить до неможливості виділення контуру; перекриття об'єктів або їх угруповання призводить до того, що контур виділяється неправильно і не відповідає кордоні об'єкта.

Однак, перехід до розгляду тільки контурів об'єктів дозволяє піти від простору зображення – до простору контурів, що істотно знижує складність алгоритмів і обчислень. Таким чином, контурний аналіз має досить слабку стійкість до перешкод, і будь-який перетин або лише часткова видимість об'єкта призводить або до неможливості детектування, або до помилкових спрацьовувань, але простота і швидкодія контурного аналізу, дозволяють цілком успішно застосовувати даний підхід (при чітко вираженому об'єкті на контрастному тлі і відсутності перешкод). Отже, ми визначилися, що контур – це якась межа об'єкта, яка відділяє його від фону (інших об'єктів). детектор кордонів Кенні, а потім можна привести будь-які інші методи отримання довічного зображення:граничне перетворення,виділення об'єкта за кольором. У всіх цих випадках ми отримуємо бінарне зображення, яке явно задає нам межі об'єкта. Ось ця сукупність пікселів, що становлять кордон об'єкта і є контур об'єкта. Щоб оперувати отриманими контуром, його необхідно якось уявити (закодувати). Наприклад, вказувати вершини відрізків, що становлять контур. Інший відомий спосіб кодування контуру – це ланцюгової код Фрімена.

Ланцюгові коди застосовуються для подання кордону у вигляді послідовності відрізків прямих ліній певної довжини і напряму. В основі цього уявлення лежить 4 – або 8 – зв'язкова решітка. Довжина кожного відрізка визначається здатністю решітки, а напрямки задаються обраним кодом. (Для подання всіх напрямків в 4 – зв'язковий решітці досить 2-х біт, а для 8 зв'язковий решітки ланцюгового коду потрібно 3 біта (рис. 1.6).

Після знаходження контуру виробу програма зрівнює його за прикладами правильних та неправильних виробів, та подає сигнал на розподілення, бракований виріб відправлятися на переробку а правильний пропускається далі по конвеєру для використання в більш складній системі або на фасування та продаж.

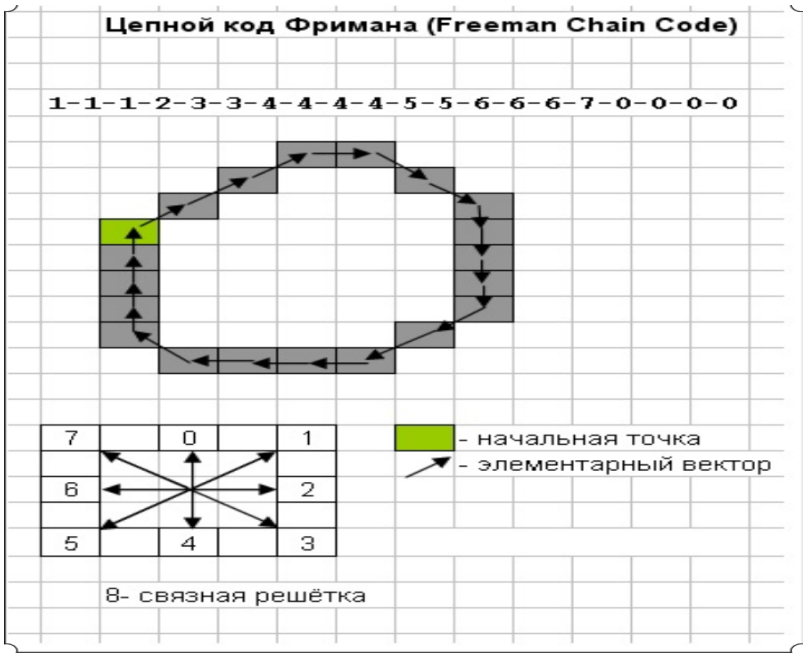


Рисунок 1.6 – Ланцюговий код Фрімена (Фрідмана)

Якщо за результатом перевірки продукт не підходить по контуру ні до правильного або неправильного зображення цієї деталі зберігається програма передає сигнал, після якого деталь відправляться до спеціальної лінії, яка буде перевірна спеціалістом для ідентифікації її як правильної або бракованої. Потім спеціаліст перевіряє деталь та додає її до прикладу правильних або бракованих деталей, внаслідок чого якість та точність перевірки стає кращою з кожною перевіркою.

1.4 Використання розпізнавання образів на виробництві

На заводі SEAT в Марторелле, Барселона, Іспанія, що нараховує більш ніж 12000 співробітників і має річний обсяг виробництва, у 2012 році переважав 375 000, компанією Axis Communications була реалізована ідея візуального контролю.

У 2011 році завод SEAT став майданчиком для виробництва Audi Q3, першого автомобіля класу "люкс", зібраного в Іспанії. Використання штрих-кодів на автомобільних заводах для відстеження деталей в процесі виробництва є стандартною процедурою. На заводі SEAT хотіли вжити додаткових заходів,

щоб уникнути затримок і полегшити робочий процес. В результаті, понад 100 відеокамер спостереження виробництва компанії Axis було встановлено для здійснення візуальної перевірки на додаток до існуючої системи сканування штрих-кодів. Зображення штрих-кодів, які визначаються як, можливо, містять помилки, відправляються на комп'ютер аварійного введення даних - так система забезпечує візуальний контроль, який служить в якості резервного для сканування штрих-кодів. Маючи зображення штрих-кодів, в разі помилки їх сканування, транспортні засоби не повинні переміщатися операторами, щоб бути повторно відсканованими. Крім того, камери були також використані для моніторингу транспортування кузовів автомобілів по всьому об'єкту.

З невпечатляющімі числами зростання, виробники в даний час більш ніж коли-небудь шукають способи не тільки підвищити ефективність, але і скоротити непотрібні витрати.

Автомобільна промисловість має багато жорстких вимог, що стосуються виробничих процесів. Безпека повинна бути на першому місці, коли справа доходить до виготовлення деталей і збірки. Тому вкрай важливо, щоб навіть найменші дефекти були виявлені, перш ніж стали частиною готового продукту. Ретельна перевірка виробничої лінії, проте, може бути дорогою і трудомісткою, тому ефективність є ключовим фактором. Час - гроші, і з такою економічною ситуацією ніхто не може дозволити собі витрачати його даремно.

Використання камер відеоспостереження не тільки для цілей безпеки, а й для підвищення ефективності - не є новою концепцією. Проте, коли виробники краще дізналися про можливі вигоди і їх здатності бути настільки ж корисними, коли мова йде про підвищення ефективності виробництва, все більше і більше компаній зацікавилися цим рішенням.

Переваги відеокамер спостереження традиційно, промислові камери використовувалися у виробництві для забезпечення якості і управління технологічними процесами. Точно так же, камери відеоспостереження, як правило, використовувалися безпосередньо для відеоспостереження - спостереження за людьми, для гарантії того, що в приміщеннях з обмеженим доступом немає персоналу, для щоденної записи того, що відбувається і т.д.

"Відеоспостереження забезпечує контроль за дотриманням правил безпеки, за тим, що інструменти та обладнання обробляються правильно, а також за тим, що виробничий персонал завжди дотримується певний порядок процесів," - каже Андреа Coppi, директор з розвитку бізнесу в компанії Axis Communications. Проте, стає все більш поширеним явищем серед виробників поєднання використання промислових камер з мережевими камерами відеоспостереження на виробничих потужностях - класичні промислові камери використовуються для задач контролю, а відеокамери спостереження - для моніторингу процесу. Саме використання камер відеоспостереження в цих рамках привернуло увагу виробників автомобілів. Венді Берк, директор по маркетингу компанії IQinVision, зазначила, що через те, що "камерам не потрібно робити перерву, вони не мають поганих днів, не відволікаються і не засинають, вони завжди включені, завжди в роботі", камери відеоспостереження, в зокрема, HD камери високої роздільної здатності, є надзвичайно корисними для виробників, допомагаючи відстежити дорогі виробничі дефекти і інші проблеми, перш ніж ці проблеми повторяться. Аналогічно тому, як відеоспостереження використовується в автомобільній промисловості для підвищення ефективності виробництва, мегапіксельні відеокамери спостереження IQinVision знайшли застосування в сталеливарній промисловості. Несправності в виробництві є дорогими помилками для сталеварів. Таким чином, Ellwood Group, американський виробник сталі, встановив 250 мегапіксельних відеокамер IQinVision на всіх своїх 10 виробничих підприємствах, щоб здійснювати не тільки відеоспостереження, але і виявляти виникаючі в процесі виробництва проблеми. "Якщо буде виявлений дефект стали," - говорить Берк, - "персонал контролю якості здатний швидко переглянути всі відеоматеріали цього виробничого процесу і знайти помилку". Потім, коли несправність буде знайдена, вона буде виправлена, що дозволить відновити виробництво, заощадивши час і гроші.

Оскільки в більшості виробничих потужностей використовується стандартна інфраструктура IP, відеокамери спостереження, які побудовані на

відкритій платформі, можуть бути легко інтегровані в існуючу систему промислового об'єкта. Це одне з найбільших переваг мережевих відеокамер спостереження, вважає Coppi. "Мережеві відеокамери спостереження можуть бути легко інтегровані в промислові виробничі середовища, такі як виробничі лінії для автоматичного виконання візуального огляду, щоб контролювати ефективність виробничої лінії, а також для віддаленої допомоги в обслуговуванні".

Крім того, "IP-відеокамери спостереження можуть отримувати зображення з високою роздільною здатністю з високою частотою кадрів, що дозволяють точно аналізувати швидку передачу об'єктів з промислової виробничої лінії". Так як мережеві відеокамери спостереження можуть підтримувати інтелектуальну обробку відео і сучасну відеоаналітіка, вони також можуть бути використані для виконання візуального контролю конкретного процесу на виробничій лінії, повідомляючи про відсутність етикетки, неправильно закритій кришці або про відсутність припою, за словами Coppi. Одним з найбільших переваг використання камер спостереження у виробництві, як вказав Coppi, є те, що вони "можуть бути легко інтегровані в промислові виробничі середовища, такі як виробничі лінії для автоматичного виконання візуального огляду. Вони зможуть контролювати ефективність роботи виробничої лінії і здійснювати віддалену допомога ".

Після установки відеокамер спостереження встановлюється програмне забезпечення для управління відео (VMS), яке допомагає керувати, отримувати доступ і працювати з зображеннями і записом. Однак, його використання в автомобільній промисловості допомогло не тільки вирішити вищезазначені завдання, а й упорядкувати та удосконалити виробничий процес. Коли в компанії Nissan Motor Manufacturing UK шукали шляхи підвищення ефективності свого виробництва, вони хотіли знайти рішення, яке можна було б інтегрувати не тільки з їх поточними системами, а й встановити більш 300 мережевих відеокамер спостереження від різних виробників для моніторингу виробничих ліній. Підрозділ Nissan у Великобританії завжди використовувало

інструменти машинного зору, в тому числі вузькоспеціалізовані промислові камери для перевірки окремих процесів. Проте, ці камери не дають загального уявлення про процес, так як вони призначені для того, щоб зосередитися на одному конкретному русі або завданню. Для того, щоб отримати більш повну картину, в Nissan UK було прийнято рішення впровадити мережеві камери і програмне забезпечення VMS від Milestone. Відкритий характер VMS від Milestone дав Nissan можливість перегляду відзнятого матеріалу з камер від різних виробників і виявлення перебоїв в технологічному процесі, здійснюючи моніторинг в режимі реального часу і миттєву запис і відтворення. Це допомогло Nissan UK запобігти виробничі проблеми і усунути причини тимчасової зупинки на своїх виробничих лініях. Крім того, зручність, яке дозволяє отримати доступ до зображення з камери за допомогою ПК, а також можливість порівнювати два відео пліч-о-пліч, щоб вивчити, як різні співробітники виконують один і той же процес, полегшило управління об'єктом. Переваги VMS допомогли не тільки поліпшити якість і ефективність заводу Nissan у Великобританії, але і дали йому систему, яка може масштабироваться разом з об'єктом.

Ефективність важлива в будь-якій галузі, але в висококонкурентному виробничої галузі, такий, як автомобільна промисловість, для досягнення успіху ефективність має першорядне значення. У той час як принципи бережливого виробництва допомагають виробникам у всіх галузях промисловості скорочувати відходи і підвищувати ефективність, додаткові переваги традиційних продуктів безпеки, таких як відеокамери спостереження і VMS, крім забезпечення безпеки, довели, що індустрія систем безпеки не обмежується тільки затриманням зловмисників і контролем доступу . Додаткові переваги - підвищення ефективності виробництва і поліпшення управління робочим процесом, - доводять, що сфери застосування продуктів безпеки воістину безмежні.

З точки зору інтелектуальних інформаційних систем, образ - це сукупність даних про реальний або абстрактному об'єкті (процесі, явище), що дозволяє виділяти його з усім тим натовпом аналізованих даних і групувати з

іншими об'єктами відповіднодо вимог розв'язуваної задачі. Вимоги конкретного завдання дозволяють уникнути надмірностів описі образу, отже, спростити його аналіз при розпізнаванні. Це особливо важливо в тих випадках, коли при розпізнаванні

доводиться обробляти великі обсяги даних, що характерно длязадач тематичного дешифрування аерокосмічних знімків. Характеристики об'єкта, необхідні для вирішення конкретного задачі розпізнавання, називають ознаками.Часто в задачах розпізнавання доводиться виділяти кілька груп об'єктів певного типу. В цьому випадку необхідно використання ознак, що дозволяють відрізнити кожну групу від всіх інших. Такі групи в розпізнаванні прийнято називати класами. В принципі, завдання може складатися у виділенні тільки одного класу (Цільового), а все інше можна розглядати як клас «інше». Тому в будь-якому випадку розпізнавання можна розглядати як задачу класифікації безлічі образів.

З цієї точки зору будь-який алгоритм розпізнавання являє собою абстрактну систему R, що складається з трьох множин:R AS P = ,,,де A - перелік виділених класів, S - набір ознак, що дозволяють відносити образи до того чи іншого класу з безлічі A, P -правила прийняття рішення при віднесенні образа до класу. Безлічі A і S складають інформаційну компоненту системи розпізнавання і тісно взаємопов'язані. Залежно від того, як описані ознаки класів, можуть застосовуватися різні підходи до організації процесу розпізнавання. Основні такі підходи безпосередньо пов'язані з математичними методами опису множин. Принцип порівняння з еталоном. Застосовується, коли клас описується одним або декількома еталонними образами, тобто безліч образів одного класу задається перерахуванням його елементів. Принцип кластеризації. Відповідає предикатні способузавдання множин - системою обмежень за значеннями ознак.Кожному класу в цьому випадку зіставляються певні інтервали значень ознак.Принцип спільності властивостей. Відповідає способу завдання множин породжує процедурою, яка і визначає в підсумкувластивості образів - елементів даної множини.Правила прийняття рішень є методичною компонентою системи розпізнавання. Серед них також

виділяють три групи : 1) евристичні; 2) математичні; 3) лінгвістичні (синтаксичні) методи. Вибір правила прийняття

рішення пов'язаний, перш за все, з властивостями використовуваних ознак. Якщо ж для конкретного набору ознак існують альтернативні правила, то головним критерієм вибору є ймовірність помилкових рішень. Евристичні правила грунтуються на апріорних припущеннях щодо властивостей образів в кожному з класів, і задаються самим аналітиком даних. У цьому випадку ймовірність помилкових рішень можна визначити тільки шляхом виконання чисельних експериментів і оцінки підсумкової помилки розпізнавання. До евристичних правил можна також віднести логічні методи розпізнавання, так як прийняття рішення в них грунтуються на взаємозв'язках між логічними змінними, визначених самим аналітиком даних. Математичні правила ґрунтуються на математичних методах аналізу даних, що дозволяють апріорі оцінити і мінімізувати очікувану величину помилки розпізнавання. При їх використанні, однак, завжди необхідно переконатися, що образи в кожному з класів задовольняють вимогам, що пред'являються конкретним методом. Серед математичних методів виділяють детерміновані і статистичні, але переважна частина тих і інших методів відносяться до дискримінантний аналіз - одному з розділів багатовимірного статистичного аналізу даних. Синтаксичні правила, на відміну від попередніх, нерозривно пов'язані з принципом спільності властивостей, тобто в цьому випадку безліч образів одного класу задається породжує процедурою. Для того щоб визначити, належить образ класу чи ні, необхідно виконати його розбір за допомогою правил породжує процедури, подібно до того, як здійснюється синтаксичний розбір пропозиції. Звідси і назва цієї методології. У синтаксичному розпізнаванні ймовірність помилок повністю залежить від властивостей аналізованих даних і коректності опису класів. Виходячи з наведених особливостей методів аналізу образів, можна визначити, для яких типів даних вони найбільш придатні. Очевидно, що математичні методи застосовуються в тих випадках, коли образ описується набором параметрів, отриманих в результаті вимірювань або чисельного моделювання. Зокрема, логічні методи застосовуються в тих випадках, коли

ознаки способу описуються висловлюваннями, кожному з яких в певному класі можна привласнити значення

«істина» (1) або «брехня» (0). В деяких випадках вислів теж можна розглядати як параметр, який приймає два зазначених значення.Синтаксичні правила застосовуються в тих випадках, коли образявляє собою складну конфігурацію, задану непохідними (Атомарними) елементами, між якими існують певнізв'язку. Непохідні елементи можуть бути задані параметрами, логічними висловлюваннями і навіть конфігураціями. зв'язки різнихтипів можуть бути описані за допомогою символів-дескрипторів, в томучислі і при аналізі сцен. Однак в останньому випадку деякі типизв'язків можуть неявно враховуватися в процедурі аналізу сцени при вирішенні задачі розпізнавання.Евристичні правила можуть застосовуватися на будь-яких типах даних,проте не слід забувати, що в цьому випадку неможливо гарантуватиоптимізацію процедури прийняття рішення щодо очікуванихпомилок розпізнавання.Таким чином, щоб вибрати оптимальний правило прийняттярішення в задачі розпізнавання, необхідно спочатку скласти формалізовані описи образів і тих класів, які підлягають виділенню. Тобто процес розпізнавання розпадається на два основнихетапу: синтез образів і аналіз образів. При цьому етап синтезу образівчасто виявляється більш складним, ніж вибір методу їх аналізу.

1.5 Постановка задач дослідження

В результаті виконання першого розділу магістерської атестаційної роботи була доведенна актуальність данного дослідженя, проведено аналіз доступних інструментів для знаходження браку, були розглянуті бібліотеки компьютерного зору, OpenCV, AForge.NET, розглянуто та визначено найкращий метод аналізу деталей та методів обробки візуальних образів. Для розробки ПО потрібно обрати та дослідити алгоритми знаходження браку, обрати тип файлу для збереження контурів, обрати файлове розширення для збереження контурних зображень деталей.

2 РОЗРОБКА АРХІТЕКТУРИ ТА ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ СИСТЕМИ

2.1 Удосконалення алгоритму знаходження браку

Перший спосіб який , було розглянуто, знайти на зображенні регіон, який найбільш схожий на брак в термінах різниці кольорів пікселів. Це і робить template matching - метод, заснований на знаходженні місця на зображенні, найбільш схожому на шаблон. "Схожість" зображення задається певною метрикою. Тобто, шаблон "накладається" на зображення, і вважається розбіжність між зображенням і і шаблоном. Положення шаблону, при якому ця розбіжність буде мінімальним, і буде означати місце шуканого об'єкта.

Як метрики можна використовувати різні варіанти, наприклад - сума квадратів різниць між шаблоном і картинкою (sum of squared differences, SSD) формула (2.1).

$$SSDi,j = \sum_{a=0\ldots m,b=o\ldots n} (f_{i+a,j+b} - g_{a,b})^2 \quad (2.1)$$

Або використовувати крос-кореляцію (cross-correlation, CCORR) формула (2.2)

$$CCORR_{i,j} = \sum_{a=0\ldots m,b=0\ldots n} (f_{i+a,j+b} g_{a,b})^2 \quad (2.2)$$

Нехай f і g - зображення і шаблон розмірами (k, l) і (m, n) відповідно (канали кольору поки будемо ігнорувати); i, j - позиція на зображенні, до якої ми "доклали" шаблон.

Крос-кореляція насправді є сверткой двох зображень. Згортки можна реалізувати швидко, використовуючи швидке перетворення Фур'є. Згідно з теоремою про згортку, після перетворення Фур'є згортки перетворюється в просте поелементне множення формула (2.3)

$$CCORR_{i,j} = f * g = IFFT\big(FFT(f * g)\big) = IFFT(FFT(f)FFT(g)) \qquad (2.3)$$

де $\circledast$ - оператор згортки.

Таким чином ми можемо швидко порахувати крос-кореляцію. Це дає загальну складність O (kllog (kl) + mnlog (mn)), проти O (klmn) при реалізації на пряму. Квадрат різниці також можна реалізувати за допомогою згортки, так як після розкриття дужок він перетвориться в різницю між сумою квадратів значень пікселів зображення і крос-кореляції формула (2.4)

$$SSD_{i,j} = \sum_{a=0\ldots m, b=0\ldots n} \big(f_{i+a,j+b} - g_{a,b}\big)^2 =$$

$$= \sum_{a=0\ldots m,b=0\ldots n} f^2_{i+a,j+b} - 2f_{i+a,j+b}g_{a,b} + g^2_{a,b} = \qquad (2.4)$$

$$= \sum_{a=0\ldots m, b=0\ldots n} f^2_{i+a,j+b} + g^2_{a,b} - 2CCOR_{i.j}$$

В бібліотеці OpenCV, реалізований пошук шаблону за допомогою методу matchTemplate (до речі використовується саме реалізація через FFT, хоча в документації це ніде не згадується), який використовує різні метрики розбіжностей:

– CV_TM_SQDIFF - сума квадратів різниць значень пікселів

– CV_TM_SQDIFF_NORMED - сума квадрат різниць квітів, отнормірованих в діапазон 0..1.

– CV_TM_CCORR - сума поелементний творів шаблону і сегмента картинки

– CV_TM_CCORR_NORMED - сума поелементний творів, отнормірованих в діапазон -1..1.

– CV_TM_CCOEFF - крос-коррелація зображень без середнього

– CV_TM_CCOEFF_NORMED - крос-кореляція між зображеннями без середнього, отнормірovани в -1..1 (кореляція Пірсона)

Ці алгоритими було обрано для визначення общої форми деталі, для знаходження браку є вибір між алгоритми Harris corner detector, FAST та SHIFT

Одним з найбільш базових алгоритмів вважається Harris corner detector

Для картинки (тут і далі ми вважаємо, що оперуємо "інтенсивністю"зображенням, перекладеної в grayscale) він намагається знайти точки, в околицях яких перепади інтенсивності більше певного порогу. Алгоритм виглядає так:

Від інтенсивності I знаходяться похідні по осі X і Y (I_x та I_y відповідно). Їх можна знайти, наприклад, застосувавши фільтр Собеля.

Для пікселя вважаємо квадрат Ix, квадрат I_y і твори I_x і I_y. Деякі джерела позначають їх як I_{xx} I_{xy} та I_{xy}- що не додає зрозумілості, так як можна подумати, що це другі похідні інтенсивності (а це не так).

Для кожного пікселя вважаємо суми в якійсь околиці (більше 1 пікселя) w наступні характеристики формула (2.5):

$$A = \sum_{x,y} w(x,y) I_x I_x$$

$$B = \sum_{x,y} w(x,y)\, I_x I_y \qquad (2.5)$$

$$C = \sum_{x,y} w(x,y) I_y I_y$$

Для кожного пікселя порахувати значення * евристики R формула 2.6

$$R = Det(H) - k(T_r(H))^2 = (AB - C^2) - (A + B)^2 \qquad (2.6)$$

Значення k підбирається емпірично в діапазоні [0.04, 0.06] Якщо $ R $ у якогось пікселя більше певного порогу, то околиця $ w $ цього пікселя містить кут, і ми відзначаємо його як ключову точку.

Попередня формула може створювати кластери лежать поруч один з одним ключових точок, в такому випадку варто їх прибрати. Це можна зробити перевіривши для кожної точки є у неї значення R максимальним серед безпосередніх сусідів. Якщо немає - то ключова точка фільтрується. Ця процедура називається non-maximum suppression.

Ця матриця багатьма властивостями і формою схожа на матрицю коваріації. Наприклад, вони обидві позитивно полуопределённые матриці, але цим подібність не обмежується. Нагадаю, що у матриці коваріації є геометрична інтерпретація. Власні вектора матриці коваріації вказують на напрямки найбільшою дисперсії вихідних даних (на яких ковариация була порахована), а власні числа - на розкид уздовж осі (рисунок 2.1):

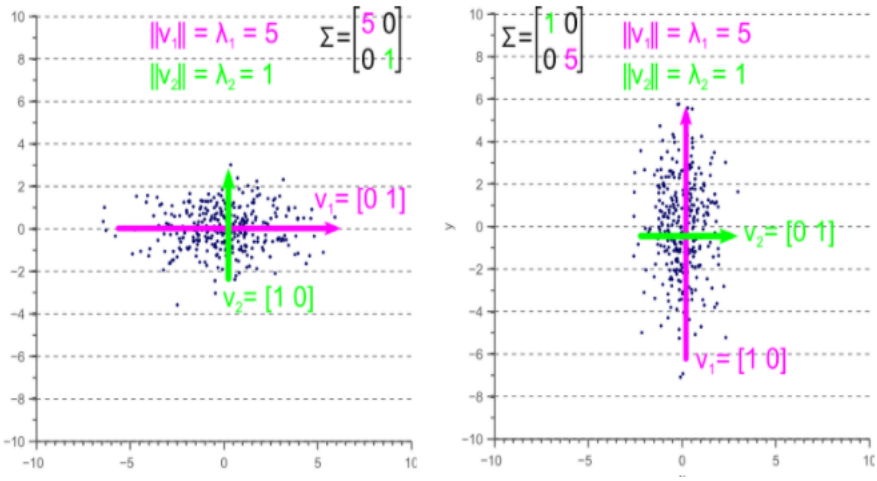


Рисунок 2.1 – Матриця Harris corner detector

Точно так само поводяться і власні числа структурного тензора: вони описують розкид градієнтів. На рівній поверхні власні числа структурного тензора будуть маленькими (бо розкид самих градієнтів буде маленьким). Власні числа структурного тензора, побудованого на шматочку картинки з гранню, будуть сильно відрізнятися: одне число буде великим (і відповідати власному вектору, спрямованому перпендикулярно грані), а друге - маленьким. На тензор кута обидва власних числа будуть великі. Виходячи з цього, ми

можемо побудувати евристику (λ_1, λ_2 - власні числа структурного тензора) формула (2.6)

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad (2.6)$$

FAST являється швидкаим методом знаходження точок Як підказує назва, FAST працює набагато швидьше ніж методу Харриса. Цей алгоритм намагається знайти точки, які лежать на краях і кутах об'єктах, тобто в місцях перепаду контрасту. Їх перебування відбувається наступним чином: FAST будує навколо пікселя-кандидата коло радіуса R, і перевіряє, чи є на ній безперервний відрізок з пікселів довжини t, який темніше (або світліше) пікселя-кандидата на K одиниць. Якщо ця умова виконується, то піксель вважається "ключовою точкою". При певних t ми можемо реалізувати цю евристику ефективно, додавши кілька попередніх перевірок, які будуть відсікати пікселі гарантовано не є кутами. Наприклад, при $ R = 3 $ і $ t = 12 $, досить перевірити, чи є серед 4 крайніх пікселів 3 послідовних, які строго темніше / світліше центру на K (на картинці - 1, 5, 9, 13). Ця умова дозволяє ефективно відсікти кандидатів, точно не є ключовими точками як показано на рисунку 2.2

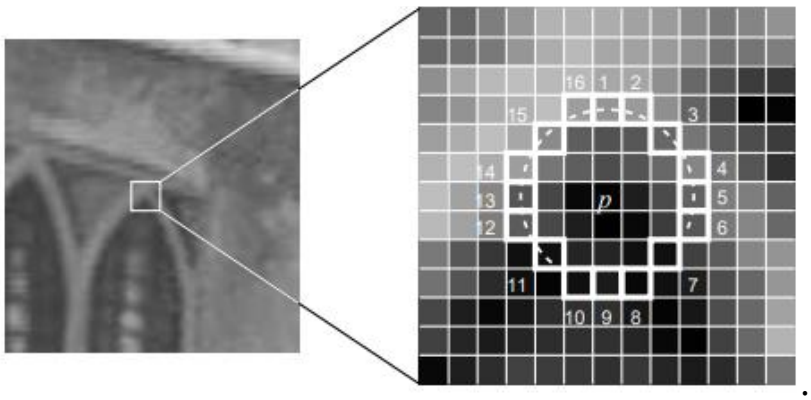


Рисунок 2.2 – Пікселі що перевіряються алгоритмом FAST

Обидва попередніх алгоритму не стійкі до змін розміру картинки. Вони не дозволяють знайти шаблон на зображенні, якщо масштаб об'єкта був змінений.

SIFT (Scale-invariant feature transform) пропонує вирішення цієї проблеми. Візьмемо зображення, з якого витягаємо ключові точки, і почнемо поступово зменшувати його розмір з якимось невеликим кроком, і для кожного варіанту

масштабу будемо знаходити ключові точки. Масштабування - важка процедура, але зменшення в 2/4/8 / ... раз можна провести ефективно, пропускаючи пікселі (в SIFT ці кратні масштаби називаються "октавами"). Проміжні масштаби можна апроксимувати, застосовуючи до картинки гауссовский блюр з різним розміром ядра. Як ми вже описали вище, це можна зробити обчислювально ефективно. Результат буде схожий на той, як якщо б ми спочатку зменшили картинку, а потім збільшили її до вихідного розміру - дрібні деталі губляться, зображення стає "замиленим" (рисунок 2.3).

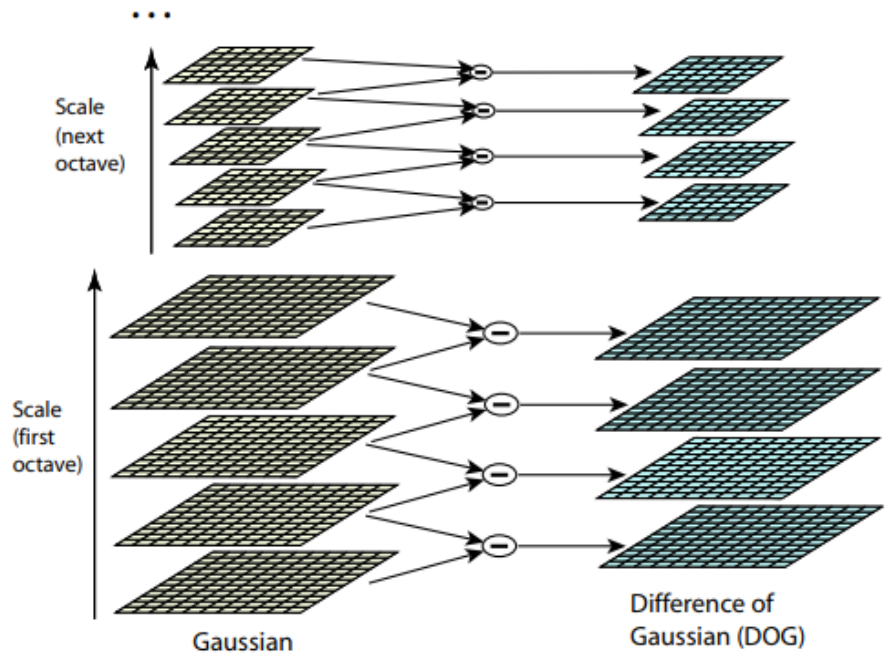


Рисунок 2.3 – Маштабування пікселей

Після цієї процедури порахуємо різницю між сусідніми масштабами. Великі (по модулю) значення в цій різниці вийдуть, якщо якась дрібна деталь

перестає бути видно на наступному рівні масштабу, або, навпаки, наступний рівень масштабу починає захоплює якусь деталь, яка на попередньому була видно. Цей прийом називається DoG, Difference of Gaussian. Можна вважати, що вел ике значення в цій різниці вже є сигналом того, що в цьому місці на зображенні є щось цікаве. Але нас цікавить той масштаб, для якого ця ключова точка буде найбільш виразною. Для цього будемо вважати ключовою точкою не

тільки точку, яка відрізняється від свого оточення, але і відрізняється найсильніше серед різних масштабів зображень. Іншими словами, вибирати ключову точку ми будемо не тільки в просторі X і Y, а в просторі (X, Y, Scale). У SIFT це робиться шляхом знаходження точок в DoG (Difference of Gaussians), які є локальними максимумами або мінімумами в 3x3x3 кубі простору (X, Y, Scale) навколо неї як показано на (рисунок 2.4).

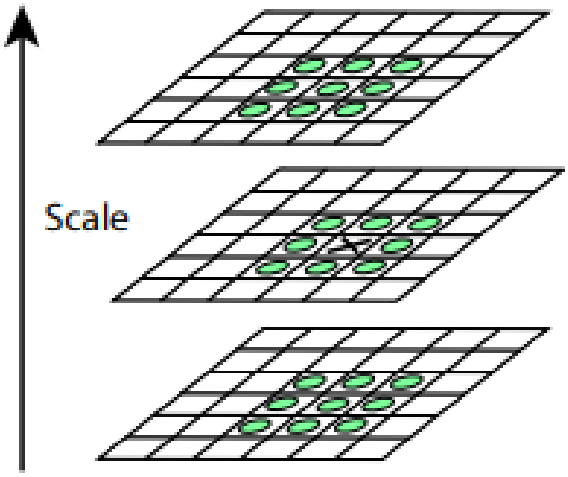


Рисунок 2.4 – Знаходження точок в DoG

2.2 Вибір формату збереження контурів

На виробництві найважливішими цілями завжди була максимізація швидкості без втрати якості, OpenCV підтримуе та може зберігати файли форматів:

– Растрові зображеня Windows - *.bmp, *.dib

– Файлы JPEG - *.jpeg, *.jpg, *.jpe

– Файлы JPEG 2000 - *.jp2

– Portable Network Graphics - *.png

– Формат портативного изображения - *.pbm, *.pgm, *.ppm

– Sun rasters - *.sr, *.ras

– Файлы TIFF - *.tiff, *.tif

Найпростіший формат запису растрових зображень. Розроблений фірмою Microsoft для збереження графіки в операційній системі Windows і сумісних з нею програмах. Для цього підтримка формату BMP була вбудована безпосередньо в ядро системи Windows. Також відомий під назвою DIB (Device Independent Bitmap - бітова матриця, яка не залежить від пристрою виводу). В операційній системі OS / 2 також є формат BMP. У Windows допускається робота з BMP-файлами стилю OS / 2, в яких використовуються різні формати інформаційного заголовка реєстрового масиву і таблиці кольорів.

Графіка зберігається у файлах з розширенням bmp або dib (зустрічається дуже рідко). Графічні дані можуть бути стиснуті з використанням найпростішого алгоритму RLE (Run Length Encoding - кодування із змінною довжиною рядка). У цьому випадку файл буде мати розширення .rle. Зазвичай стиснення не використовується, і при цьому розмір графічного файлу великий. В даний час - один з найпоширеніших графічних форматів. Підтримується практично всіма графічними програмами сумісними з Windows. Через свою простоту вимагає для виведення дуже мало системних ресурсів, тому основне його призначення - зберігання зображень, використовуваних як елементи призначеного для користувача інтерфейсу операційної системи.

Структура. Файл розбитий на чотири основні розділи: заголовок файлу растрової графіки, інформаційний заголовок реєстрового масиву, таблиця кольорів і власне дані реєстрового масиву. Тема файлу растрової графіки містить інформацію про фото, в тому числі адресу, з якого починається область даних растрового масиву. В інформаційному заголовку реєстрового масиву містяться відомості про зображення, яке зберігається у файлі, наприклад, його висоті і ширині в пікселях. У таблиці кольорів (як говорилося вище, таблиця

кольорів використовується, якщо кількість кольорів дорівнює або менше 256) представлені значення основних кольорів RGB (червоний, зелений, синій) для використовуваних в зображенні квітів. Програми, що зчитують і відображають BMP-файли, в разі використання відеоадаптерів, які не дозволяють відображати більше 256 квітів, для точної передачі кольору можуть програмно встановлювати такі значення RGB в колірних палітрах адаптерів. Формат власне даних растрового масиву в файлі BMP залежить від числа біт, що використовуються для кодування даних про колір кожного пікселя. При 256-кольоровому

зображенні кожен піксель в тій частині файлу, де містяться власне дані реєстрового масиву, описується одним байтом (8 біт). Це опис пікселя не представляє значний квітів RGB, а слугує дороговказом для входу в таблицю кольорів файлу. Значення пікселів зберігаються в порядку їх розташування зліва направо, починаючи (як правило) з нижнього рядка зображення. Якщо число байт в кожному рядку непарній, то до кожного рядка додається додатковий байт, щоб вирівняти дані реєстрового масиву по 16-біт кордонів.

Висновок на екран картинки, що зберігається в BMP-файлі, починається з читання заголовка файлу і інформаційного заголовка. Програма, таким чином, дізнається розміри зображення і кількість квітів. Потім програма читає колірну таблицю. Якщо комп'ютер виводить 256 квітів максимум, то програма заповнює колірну палітру значеннями з колірною таблиці. Таким чином, забезпечується правильна передача кольорів картинки. Якщо комп'ютер здатний відобразити тисячі або мільйони квітів одночасно, то колірну палітру заповнювати не потрібно. І, нарешті, зчитуються растрові дані. Як тільки рядок зі значеннями пікселів прочитана з файлу, вона передається в відеобуфер для отримання зображення на екрані. У таких графічних середовищах як Windows, програма пересилає значення кольорів не безпосередньо в відеобуфер, а в Windows, і вже Windows заносить їх в відеобуфер.

Файли BMP з глибиною 16 і 24 біт / піксель не мають таблиць кольорів; в цих файлах значення пікселів растрового масиву безпосередньо характеризують значення кольорів RGB.

Файл JPEG вдає із себе найпоширеніший графічний формат стисненого зображення на сьогоднішній день. Був створений корпорацією Joint Photographic Experts Group.

Файл PNG (з англ. Portable Network Graphic) відноситься до растрових зображень. Формат PNG містить певну палітру кольорів, що застосовуються в малюнку. Подібний графічний формат досить часто застосовують в мережі Всесвітньої павутини при наділення веб-сторінок різними зображеннями.

Завдяки використанню алгоритму стиснення Deflate, растрові зображення, мають розширення файлу PNG, доступні для стиснення без явних втрат якості.

Розробили даний формат файлу, щоб замінити формат GIF, адже останній тривалий час вимагав наявності платного програмного забезпечення. Серед власників веб-ресурсів, зображення PNG славляться відмінними характеристиками на тлі подібних форматів. PNG підтримує глибину кольору в межах до 48 біт. Основна відмінність GIF полягає в тому, що такий графічний файл обмежується лише 8 бітами (всього 256 кольорів). Слід знати, що на відміну від GIF, PNG не володіє підтримкою анімаційних ефектів.

Відкрити файл, що має розширення PNG, можна за допомогою фактично будь-якої програми перегляду. В операційній системі Windows, відкрити PNG можливо за допомогою простого подвійного клацання миші для перегляду зображень. Такий формат зображень запускається і в будь-яких веб-браузерах. Якщо користувачеві необхідно змінити збережене зображення в PNG версії, досить скористатися утилітами редагування зображень, такими як Adobe Photoshop або Microsoft Windows Photos, а також Corel PaintShop або ACD Systems.

Дане розширення файлу користується великою популярністю і несе в собі всю необхідну графічну інформацію для повнокольорових зображень хорошої якості.

JPEG використовує досить високу ступінь стиснення, при цьому може похвалитися підтримкою глибини кольору в 24 біта. Розглядається розширення застосовують в переважній більшості цифрових фотоапаратів і різних камер.

Широке поширення файлу .jpg або .jpeg спостерігається в середовищі зберігання цифрових фотографій в рамках мережі Всесвітньої павутини. Зображення в форматі JPG використовує значну кількість web-розробників, тому що є присутнім можливість зменшення розміру картинки без серйозної втрати якості і відтінків.

До недоліків відносять відсутність підтримки регулювання прозорості. Чим менше розмір файлу, тим вище значення ступеня його стиснення. Об'єднана група експертів, на стадії формування таких зображень, оголосила про те, що JPG

формат і JPEG є повністю рівнозначними, їх різниця полягає лише в одній букві. Варто відзначити, що jfif і .jpe так само відносяться до описуваного розширенню.

Відкрити файл .jpg можна за допомогою різноманітного програмного забезпечення в рамках різних операційних систем. Існує безліч програм для перегляду зображень, як штатного виробництва, так і випущеного сторонніми розробниками і іменитими виробниками. Виділяють такі утиліти як Adobe Photoshop, Paint, Corel PaintShop та інші. Тип файлу можна конвертувати в інші популярні формати.

Для вибору найкращого формату було варховано недоліки та переваги усіх підримуваемих ворматів, одним із найважливіши з яких була компрессія без значної втрати якості. Для цього було взяте контурне зображення деталі яке було збережене за допомогою OpenCV у трьох різних форматах

Зображення у форматі BMPбуло набільшим за розміром воно займало 737 КБ місця без візуальних відмінностей (рисунок 2.5).

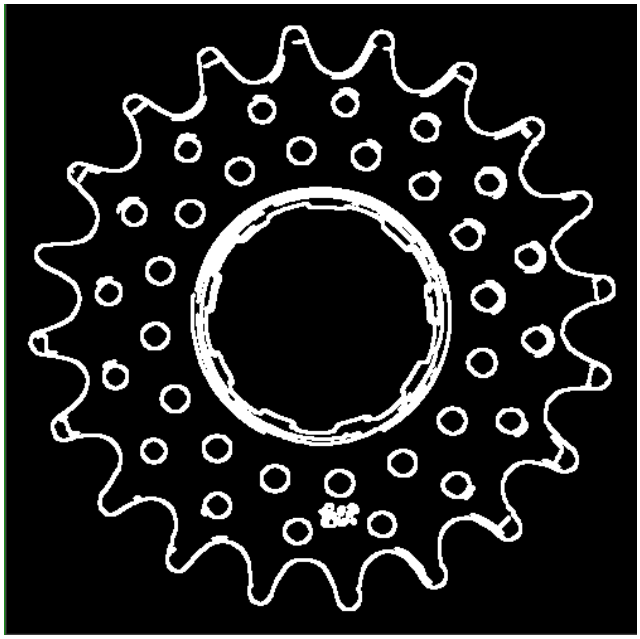


Рисунок 2.5 – Контур у форматі BMP

Зображення у форматі .png показало найкращу компресію серед усіх без особливої візуальної втрати якості, розмір зображення складав 6,89 кб, (рис. 2.6).

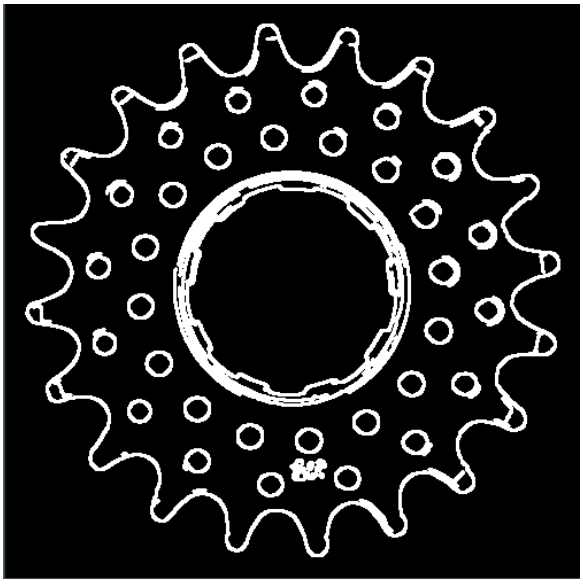


Рисунок 2.6 – Контур у форматі PNG

Зображення у форматі JPG показало меньшу компресію серед усіх та посередьню якість, розмір зображення складав 48,1 КБ (рисунок 2.7).

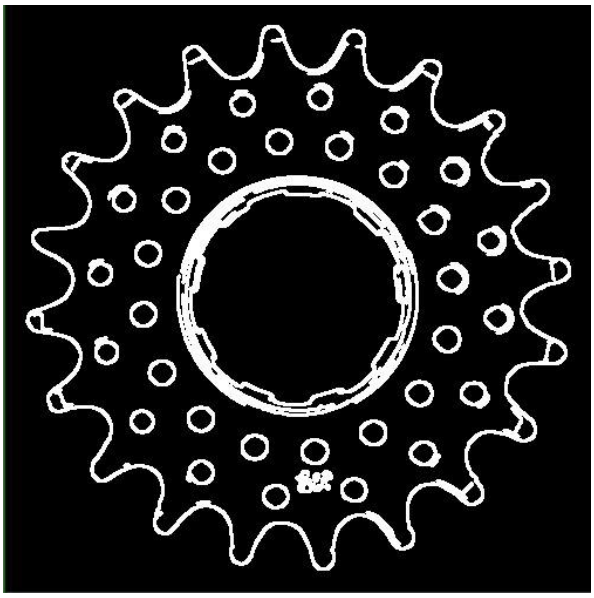


Рисунок 2.7 – Контур у форматі JPG

Усі зображення були збережені розміром 502x501 завдяки цьому ми маемо результати не привьязані до розрішення (рисунок 2.8).

Рисунок 2.8 – Файли збереженні у форматах BMP, JPG та PNG

Завдяки тому що контурні зображення зберігаються у чорному та білому кольорах найкращим файловим типом для збереження деталей являеться формат PNG.

2.3 Принцип роботи та створення структурна схема програми

Після запуску програма та підключення до камери програма просить зображення бракованих та правильних деталей або папки з готовими

контурами. Якщо користувач запускає програму перший раз він подає зображення бракованих та правильних деталей які конвертуються до контуру, та зберігаються до папок “Bad stuff”,яка містить зображення бракованих деталей, та “Good stuff”,яка містить зображення правильних деталей. Якщо папки зі зображеннями були створені за попередніх перевірок користувачем вводитися інтервал перевірки за яку деталь буде перевірена та відправлена до браку, перевірку або далі по конвеєру. Після виконання перевірок програма видає кількість правильних та бракованих деталей після чого показує деталі які ну підходять до бракованих або правильних, та надає можливість визначити їх до папок “ Good stuff” або “Bad stuff” для більш точних наступних перевірок. Якщо

программа робить 5 пустих фото вона зупиняе перевірку. Структурна схема роботи програми представленна на рисунку 2.9.

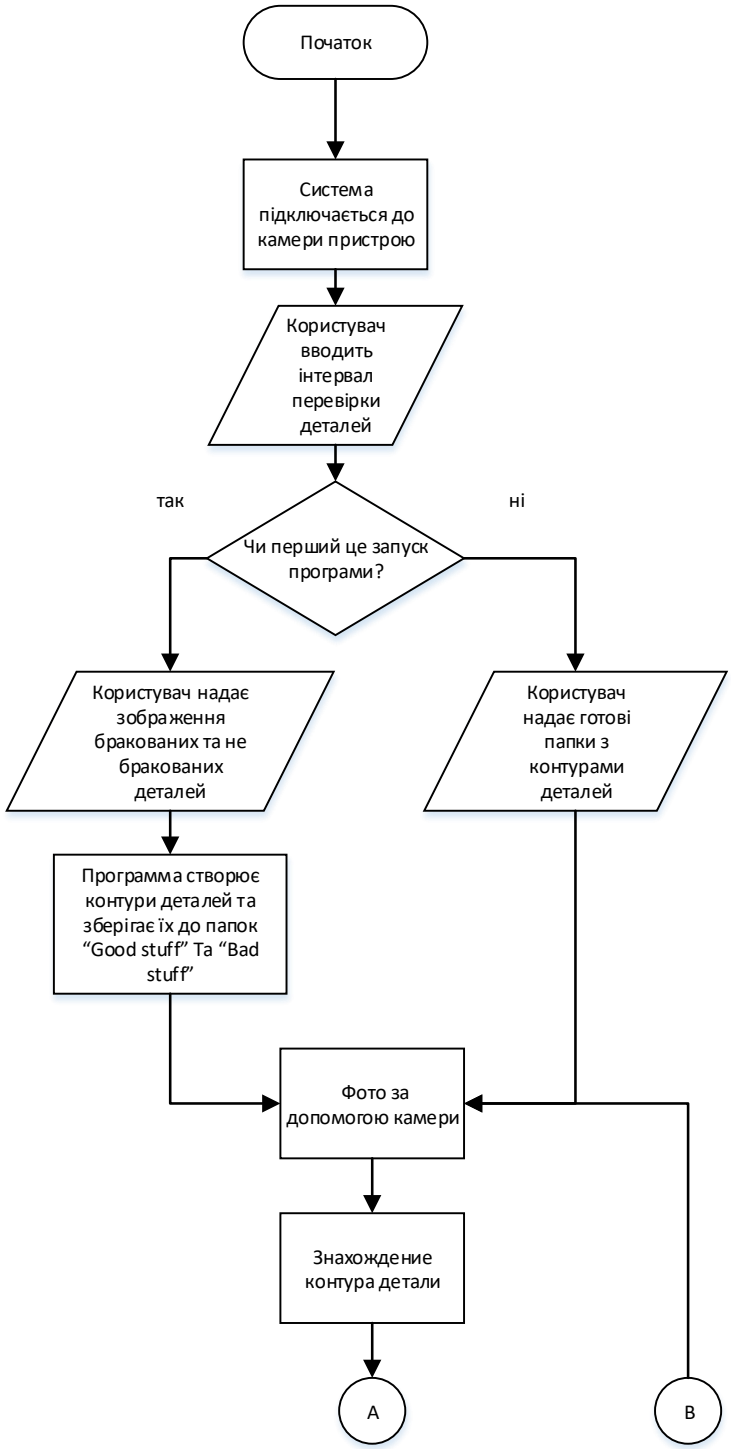


Рисунок 2.9 – Стурктурна схема роботи ПЗ

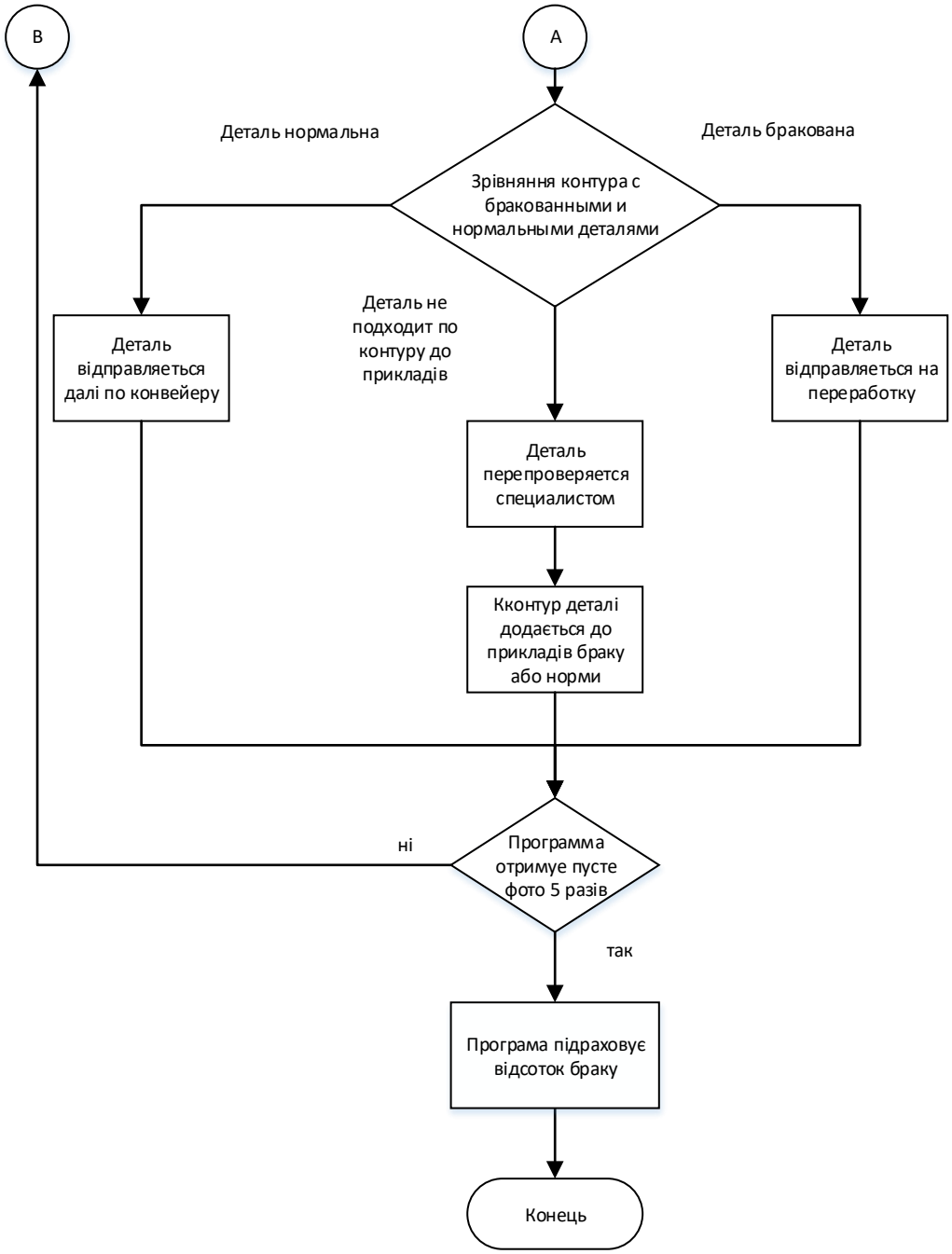


Рисунок 2.9, аркуш 2

2.4 Висновки по дургому розділу магістерської атестаційної роботи

В результаті виконання другого розділу магістерської атестаційної роботи був проведений аналіз методів розробки ПО для знаходження браку був удосконалений метод знаходження образів завдяки з'єднанню методів, template mathching, FAST та SHIFT. Було проведенно дослідження та обран оптимальний тип збереження файлу, розробленний алгоритм роботи метода, програми та обрані критерії для виконання поставленої задачі.

ЗЭКСПЕРЕМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

3.1 Тестування розробленої системи розпізнавання браку

Перший єксперемент проводився після загрузки контуру однієї правильної деталі та одіної бракованої перевірялись чотири деталі прваильна рисунок та три бракованих, одна з браком який программа розпізнає, друга така сама але повернута на 90 градусів та фото деталі з браком який программа не може розпізнати (рисунок 3.1).

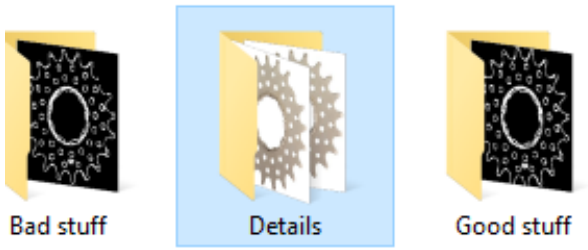


Рисунок 3.1 – Файли загружені в программу

Перевірка першої правильної деталі (рисунок 3.2).

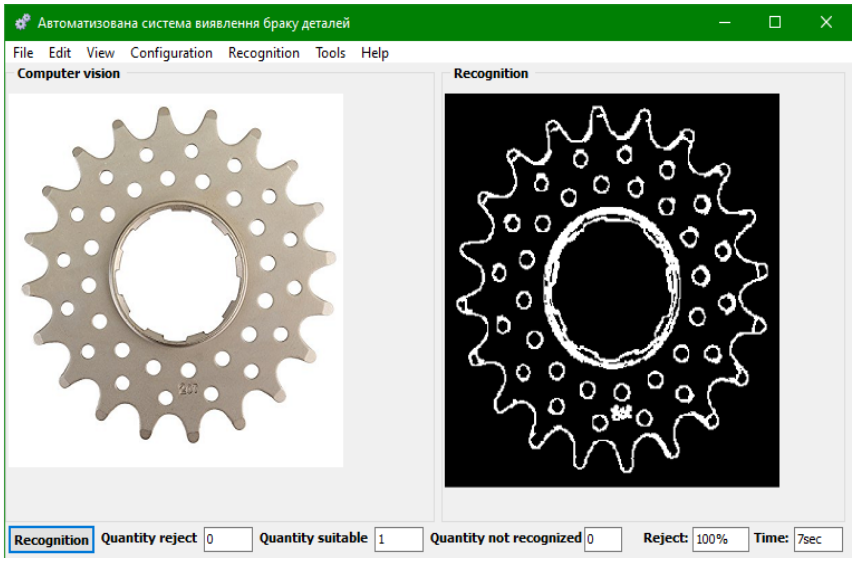


Рисунок 3.2 – Перевірка правильної деталі

При перевірці першої деталі программа знайшла деталь підходящу по контуру до прикладу правильної.Після перевірки першої деталі вона була додана до результату правильних, після цього пройшла перевірка бракованої деталі (рисунок 3.3).

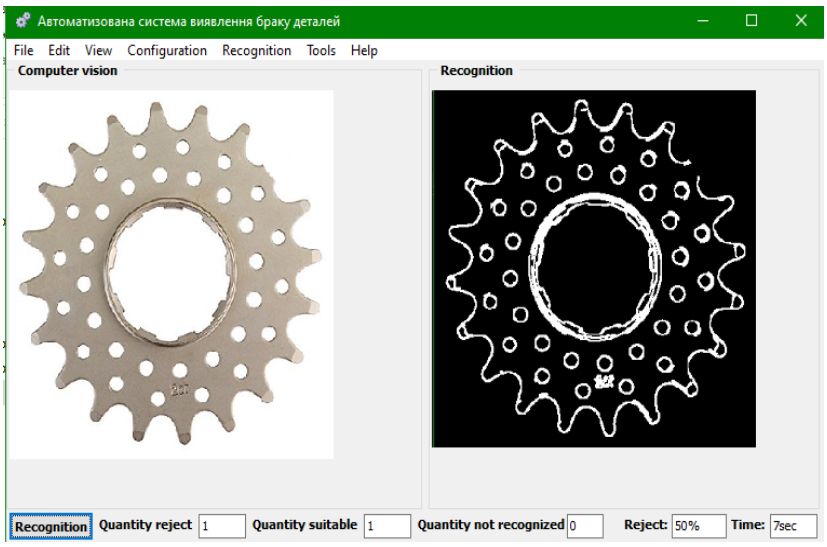


Рисунок 3.3 – Перевірка бракованї деталі

При перевіці першої неправильної деталі пргораммою було знайденно контур бракованої деталі.Після перевірки другої деталі вона була додана до результату правильних, після цього пройшла перевірка бракованої деталі під кутом девьяносто градусів на право (рис. 3.4).

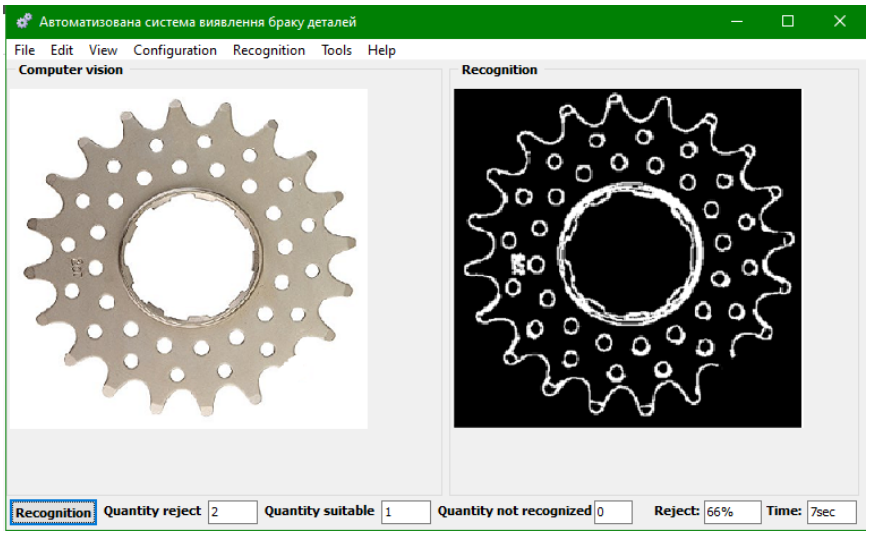


Рисунок 3.4 – Перевірка повернутої бракованї деталі

При перевіці першої модифікованої неправильної деталі пргораммою було знайденно контур бракованої деталі.Після перевірки третьої деталі вона була додана до результату бракованих, після цього пройшла перевірка іньшої бракованої деталі з браком який не був внесений при першому запуску (рис. 3.5).

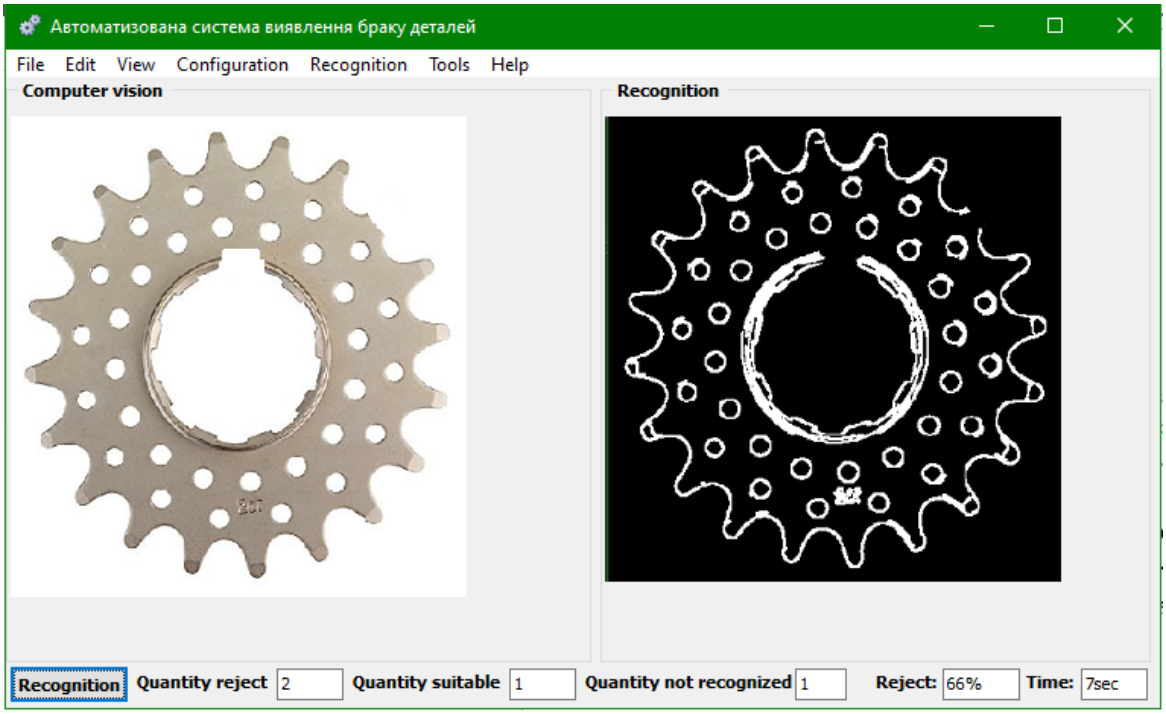


Рисунок 3.5 – Перевірка не знайомої бракованї деталі

При перевіці не знайомої неправильної деталі пргораммою не було знайденно контуру бракованої деталі але вона була достатньо відмінною від правильної для того щоб программа занесла її до списку не зрозумілих.Після перевірки третьої деталі вона була додана до не зрозумілих, після чого вона була збрежена користувачем до бракованих, при наступному запуску системи деталь була визначена як бракована (рисунок 3.6).

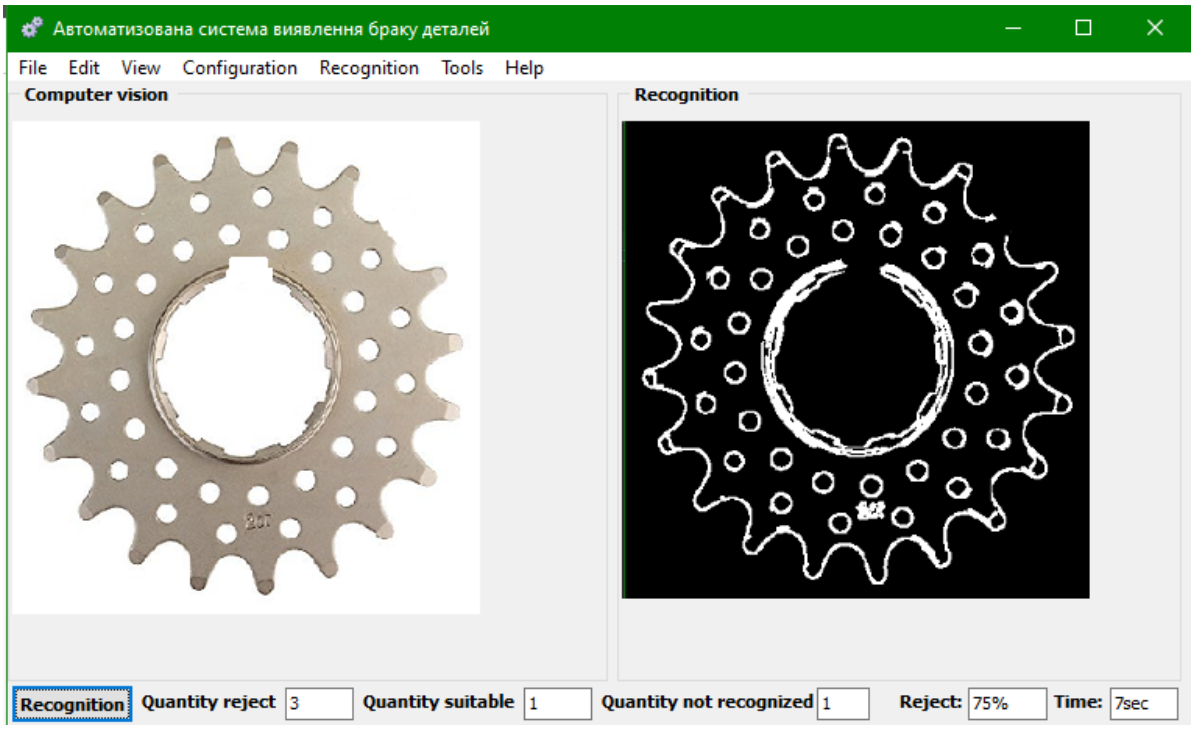


Рисунок 3.6 – Повторна перевірка бракованї деталі

3.2 Перевірка чутливості системи

Чутливість перетворення зображення деталі регулюється за допомогою конструкції CVcanny.

Детектор краю Canny був розроблений Джоном Ф. Кенні в 1986 році. Також відомий багатьом як оптимальний детектор, алгоритм Canny прагне задовольнити три основні критерії:

Низький рівень помилок: означає хороше виявлення лише існуючих країв.

Хороша локалізація: Відстань між виявленими крайовими пікселями та реальними пікселями крайових слід мінімізувати.

Мінімальна реакція: Тільки одна відповідь детектора на край.

Спершу фільтрується весь шум за допомогою гусового фільтру, приклад ядра Гауса розміром = 5, який може бути використаний, показаний нижче формула (3.1)

$$K = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (3.1)$$

Знайдіть градієнт інтенсивності зображення. Для цього ми дотримуємося процедури, аналогічної Собелю: Нанесіть пару масок згортки (у напрямках x та y) (3.2)

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$
$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ -1 & +2 & -1 \end{bmatrix} \quad (3.2)$$

Знайдіть силу та напрямок градієнта за допомогою формули (3.3):

$$G = \sqrt{G_x^2 + G_y^2}$$
$$0 = \arctan(\frac{G_y}{G_x}) \quad (3.3)$$

Напрямок округлено до одного з чотирьох можливих кутів (а саме 0, 45, 90 або 135)

Застосовується не максимальне придушення. Це видаляє пікселі, які не вважаються частиною ребра. Отже, залишаться лише тонкі лінії (краї кандидатів).

Гістерезис: Останній крок. Canny використовує два пороги (верхній і нижній):

Якщо градієнт пікселя перевищує верхній поріг, піксель приймається як ребро.

Якщо значення градієнта пікселів нижче нижнього порогу, воно відхиляється.

Якщо градієнт пікселя знаходиться між двома порогами, він буде прийнятий лише в тому випадку, якщо він підключений до пікселя, який знаходиться вище верхнього порогу.

Canny рекомендується співвідношення верхнє: нижнє між 2: 1 і 3: 1

Ця команда зменшує кількість шуму. При виборі занадто низького коефіцієнту зображення як частиною деталі будуть додані бліки світла та артефакти камери, приклад CVcanny (рисунок 3.7).

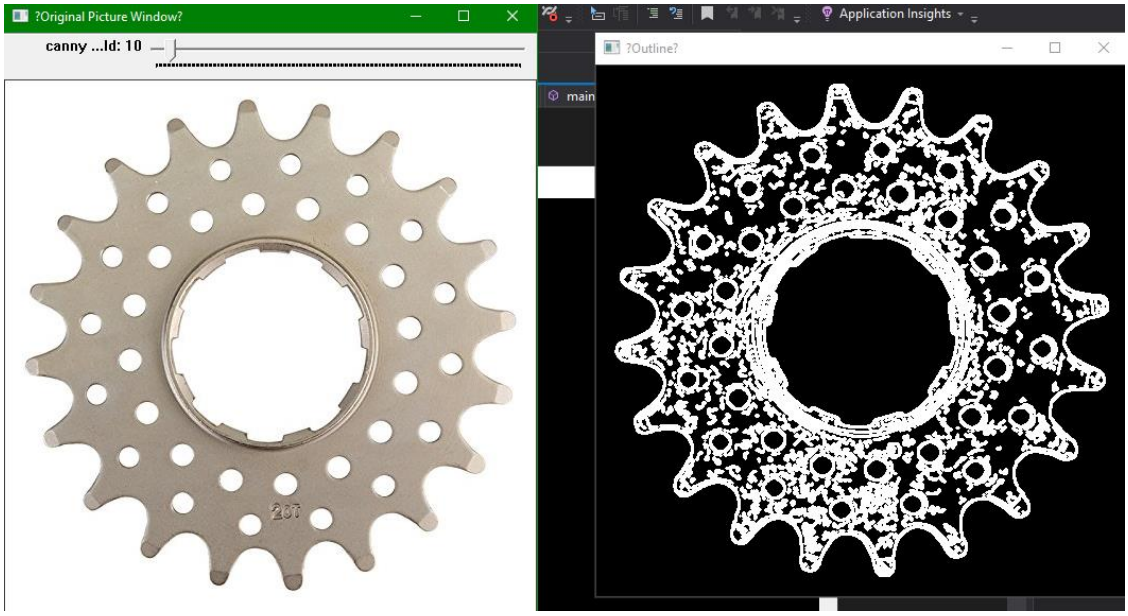


Рисунок 3.7 – Правильна деталь з низьким рівнем CVcany

При виборі занадто високого значення программа не буде обводити деякі частини деталей шо знижуе її ефктивність пошуку браку через те що OpenCV приймае частини деталей за шум (рисунок 3.8).

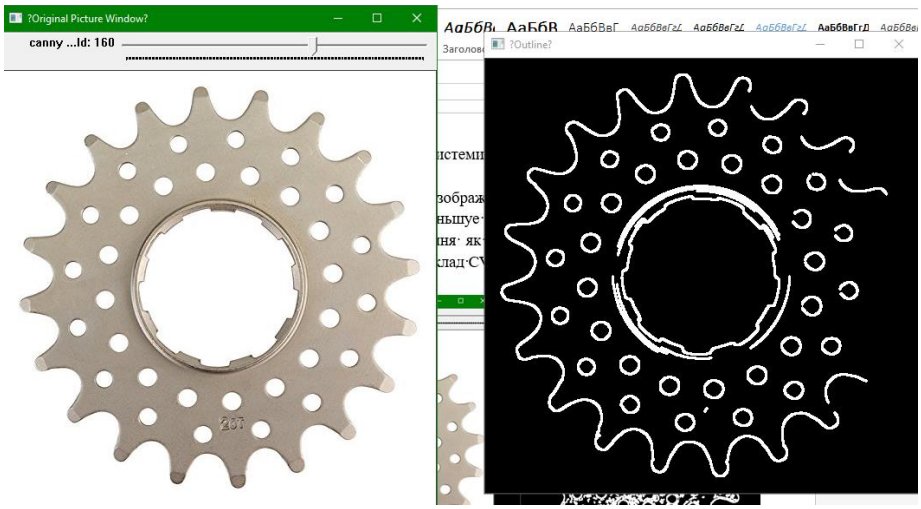


Рисунок 3.8 – Правильна деталь з високим рівнем CVcany

Для оптимальної роботи программи було обрано значення в 80 завдяки такій чутливості программа матиме змогу перевіряти деталь на брак з посередьним з оптимальним міксом шуму та деталізаці (рисунок 3.9).

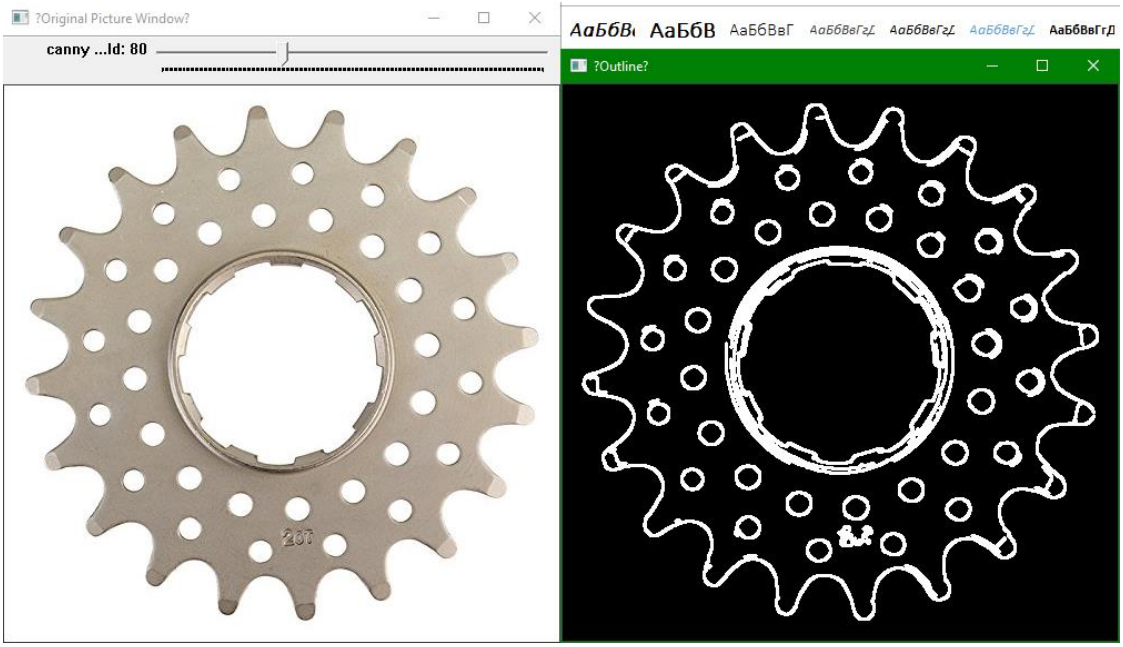


Рисунок 3.9 – Правильна деталь з оптимальним рівнем CVcany

3.4 Охорона праці під час виконання проєкту

Базовий показник для визначення складових витрат праці розраховується за формулою (3.4)

$$T=T_0 +T_д +T_a +T_п +T_н +T_{док}\ , \qquad (3.4)$$

де T – загальні витрати труда, люд. годин;

T_0 – витрати труда на опис завдання;

$T_д$ – витрати на розгляд предметної області;

T_a– витрати на розробку алгоритму рішення задачі;

$T_п$ – витрати на програмування;

$T_н$ – витрати на настроювання програми;

$T_{док}$ – витрати на підготовку документації.

Всі складові визначаються через умовну кількість операторів Q:

е q – кількість операторів;

c – коефіцієнт складності завдання (приймається від 1,25 до 2);

p – коефіцієнт корекції програми, який враховує новизну проекту (для абсолютно нової програми дорівнює 0,1).

Візьмемо кількість операторів 23. З них 19 – кількість студентів, 4 – викладачів і інженерів-лаборантів, які будуть використовувати даний програмний продукт в навчанні (лабораторній практиці). Коефіцієнт складності програми приймемо за 1,5, тому що програмне забезпечення середньої складності.

Програма нова, тому коефіцієнт корекції програми візьмемо 0,1.

Підставивши отримані дані в формулу (3.5), отримаємо:

$$Q = 23\ 1{,}5\ (1+ 0{,}1) = 38\ . \qquad (3.5)$$

Залежно від складності продукту T_0 візьмемо діапазон від 1 люд. год. до 5 люд. год. Програмний продукт середньої складності, тому T_0 візьмемо 2.люд. год.

$T_д$ визначається за формулою (3.6):

$$T_д = (Q \cdot B) / (S_д \cdot k) \qquad (3.6)$$

де B – коефіцієнт збільшення витрат роботи внаслідок недостатнього опису завдання (від 1,2 до 1,5);

$S_д$ – кількість операторів, яка виділяється на 1 люд. год;

k.– коефіцієнт кваліфікації працівника.

Таким чином, з огляду на стаж працівника до 2-х років, приймаємо коефіцієнт 0,8.(3.13)

T_a визначається за формулою (3.7):

$$T_a = \mathrm{Q} / (S_a \cdot k)$$
$$S_a = 1 - 2 \quad (3.7)$$

Де S_a– кількість операторів, зайнятих розробкою блок-схеми, на 1 люд. год.

$$T_a = 38/(2 \ \ 0{,}8) = 24 \ люд.\ год. \quad (3.8)$$

Витрати на програмування розраховуються за формулою (3.9):

$$T_п = Q/(S_п \quad k), \quad (3.9)$$

$$S_П = 1\text{-}2,$$

де $S_П$ – кількість операторів, зайнятих програмуванням, на 1 люд. год.

$$T_п = 38/(2 \ \ 0{,}8) = 24 \ люд.\ год. \quad (3.10)$$

Витрати на настройку програми розрахуємо наступним чином:

$$T_{н} = Q/(S_{н} \quad k) ,$$

(3.11)

$S_{н=1-2}$

де $S_н$ – кількість операторів, зайнятих налаштуванням, на 1 люд. Год.

Витрати на підготовку документації знаходять за формулою (3.12):

$$^{T}док = T_{др} + T_{док} , \qquad (3.12)$$

де $T_{др}$ – витрати праці на підготовку матеріалу рукописи знаходять за формулою (3.13):

$$T_{др} = Q/(S_{др} \quad k) , \qquad (3.13)$$

$S_{ДР=1-2}$

де $S_{др}$ – кількість операторів, зайнятих підготовкою матеріалів в рукописі, на люд. Год.

$$T_{др} = 38/(1\ 0.8) = 24 \text{ люд. Год} , \qquad (3.14)$$

$T_{док}$ – витрати праці на редагування, роздруківку і оформлення документів.

$T_{док}$ визначається за формулою (3.15):

$$T_{док} = 0{,}75 \cdot T_{др} = 0{,}75 \cdot 24 = 18 \text{ люд. Год.} \qquad (3.15)$$

$$T = 38+57 + 24+ 24+ 24+18 = 185 \text{ люд. год.} \qquad (3.16)$$

Отримане значення загальної трудомісткості необхідно відредагувати з урахуванням рівня мови програмування:

$$T_{кор} = T \quad k_{кор} = 185 \quad 0{,}9 = 167 \; люд.\; год \; . \tag{3.17}$$

де $k_{кор}$ – коефіцієнт, який враховує рівень мови програмування (від 0,8 до 1).

Час роботи персонального комп'ютера (фонд часу) під час створенняпрограмного продукту визначається за формулою (3.18):

$$Ф_ч = 1{,}15 \cdot (T_П + T_Д + T_Н) \cdot k_{кор} \tag{3.18}$$

3.5 Висновки по третьому розділу магістерської атестаційної роботи

В результаті виконання третього розділу магістерської атестаційної роботи розробленна программа була протестована та були обрані оптимальні настройки для максимально точного та швидкого результату.

ВИСНОВКИ

Виявлення браку на виробництві завжди було важливою проблемою у всіх галузях промисловості. Автоматизація виробництва дає змогу виконувати цей процес автоматично, а розвиток сучасних методів розпізнавання образів може дати можливість впровадити методи, що забезпечуватимуть досить високий рівень надійності розпізнавання. Для впровадження методів розпізнавання браку на виробництві має актуальність дослідження методів та бібліотек, щоб обґрунтовано використовувати їх на виробництві.

У магістерській роботі розглянуто овобливості актуальних методів розпізнавання образів та знаходження браку на виробництві.

Проаналізовано актуальні архитектури систем компьютерного зору з цілью зниження браку на виробництві.

Розробленна програмна реалізація систем опізнавання браку на овної бібліотеки OpenCV, що дозволяе редагувати програму для іншого типу деталей.

Результатом магістерьскої роботи є система за допомогою яка надає можливість реальному часі знаходить брак у простих вращающихся деталях типу 71,завдяки використанню метода FAST було отримано скорочення часу на виявлення браку деталей на виробництві.Программа розробленна на мові C++ надає змогу модивікації для інших типів деталей.В ході експерементальної перевірки були встановленна оптимальна величина заглушування шуму CVcany яка дорівнює 80. При використанні данної велечини важливі частини деталі не приймаються ПО за шум та гарантують точність перевірки на брак.

.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. - К.: ДП «УкрНДНЦ», 2016. - 31 с.

2. Невлюдов, І.Ш. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології» [Текст]:навч.посіб. / І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарєва, Г.В. Пономарьова. – Київ – 58, пр. Космонавта Комарова, 1, 2016. – 320 с.

3. Методичні вказівки «З розробки й оформлення магістерської атестаційної роботи для студентів другого (магістерського) рівня вищої освітигалузі знань 15 Автоматизація та приладобудування за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології освітні програми: «Автоматизоване управління технологічними процесами», «Комп'ютерноінтегровані технологічні процеси і виробництва», «Комп'ютеризовані таробототехнічні системи» [Текст] / Упоряд. І.Ш. Невлюдов, В.В. Косенко,В.В. Євсєєв. – Харків: ХНУРЕ, 2019. – 55 с.

4. Положення про протидію академічному плагіату в ХНУРЕ[Електронний ресурс].Режимдоступу:https://nure.ua/wpcontent/uploads/Main_Docs_NURE.html. – (дата звернення: 29.09.2020).

5. Невлюдов, І.Ш. Основи наукових досліджень [Текст]: навч. посібник / І.Ш. Невлюдов, Ю.М. Олександров, А.О. Андрусевич, О.О. Чала. – Кривій Ріг: Криворізький коледж НАУ, 2019. – 396 с.

6. Невлюдов, І.Ш. Техніко – економічне обґрунтування інженерних рішень в автоматизованому виробництві [Текст]: підручник / І.Ш. Невлюдов. – Кривий Ріг : Криворізький коледж НАУ, 2019. – 448 с.

7. Комарцова, Л.Г. Нейрокомпьютеры [Текст] / Л.Г. Комарцова,А.В. Максимов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 320 с.

8. Srivastava, N. Dropout: a simple way to prevent neural networks from overfitting. [Text] / N. Srivastava, G. E. Hinton // Journal of Machine Learning Research. – 2014. – Vol. 15. – № 1. – P. 1929–1958.

9. Методы робастного, нейро-нечеткого и адаптивного управления [Текст]
/ Под ред. Н.Д. Егупова. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 744 с.

10. Бодянський, Є. В. Нейро – фаззі моделі в системах штучного інтелекту
[Текст] / Є. В. Бодянський, Є. І. Кучеренко. – Харків: ХНУРЕ, 2006. – 177 с.

11. Руденко, О. Г. Основы теории искусственных нейронных сетей [Текст]
/ О.Г. Руденко, Е. В. Бодянский. – Харьков: Телетех, 2002. – 317 с.

12. Невлюдов, І.Ш. Розпізнавання голосових команд за допомогою багатошарового перцептрона [Текст] / І.Ш. Невлюдов, О.М. Цимбал, С.С. Мілютіна // Східно-європейський журнал передових технологій, Харків. – 2006. –№ 3/2 (16). – С. 13–16.

13. Цимбал, А.М. Использование искусственной нейронной сети вподсистеме ввода голосовой информации САПР ТП роботизированного производства [Текст] / И.Ш. Невлюдов, А.М. Цимбал, С.С. Милютина //Радиоэлектроника и информатика, Харьков. – 2007. – № 1. – С. 56-61.

14. Каллан, Р. Основные концепции нейронных сетей [Текст] // Р. Каллан. – New York: Williams, 2001. – С. 288.

15. Bergstra, J. Random search for hyper-parameter optimization [Text] /
J. Bergstra, Y. Bengio // Journal of Machine Learning Research. – 2012. – Vol. 13. – № 1. – P. 281–305.