

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)

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

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр

(назва освітнього ступеня)

на тему: Програмно- апаратне забезпечення самоадаптивного
керування вбудованими системами

Виконав: студент 6 курсу групи СІм-61

спеціальності _____

123 «Комп'ютерна інженерія»

(шифр і назва спеціальності (напряму підготовки))

Головатий І.Б.

(підпис)

(прізвище та ініціали)

Керівник

Баран І.О.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Луцик Н.С.

(підпис)

(прізвище та ініціали)

Завідувач

кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

Рецензент

Петрик М.Р.

(підпис)

(прізвище та ініціали)

Міністерство освіти і науки України

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра комп'ютерних систем та мереж

(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« »

20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня _____

магістр

(назва освітнього ступеня)

за спеціальністю _____

123 Комп'ютерна інженерія

студенту _____

Головату Ігорю Богдановичу

(прізвище, ім'я, по батькові)

1. Тема роботи _____

Програмно-апаратне забезпечення самоадаптивного

керування вбудованими системами

Керівник роботи _____

Баран Ігор Олегович., к.т.н. доцент

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

Затверджені наказом по університету від «_06_» грудня 2022 року №_4/7-986_

2. Термін подання студентом роботи 19.12.2022

3. Вихідні дані до роботи _____

наукові літературні джерела

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1 Аналітична частина. 2 Теоретична частина. 3. Практична частина.

4 Охорона праці та безпека в надзвичайних ситуаціях

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

1. Тема, мета, задачі, об'єкт, предмет дослідження. 2. Актуальність дослідження.

3 Програмне забезпечення вбудованих систем. 4. Еволюційні алгоритми та дерева рішень

5. Основні методи підвищення точності алгоритмів розпізнавання зображення особи.

6. Блок-схема генетичного алгоритму. Схема виявлення об'єкта.

7. Стенд аналізу орієнтації дисків. 8. Елементна база.

9. Результати випробувань. 10. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., доцент	07.12.22	12.12.22
Безпека в НС	Стручок В.С.		

7. Дата видачі завдання _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми кваліфікаційної роботи	06.12.22	Виконано
2	Аналіз літературних джерел	07.12-08.12.22	Виконано
3	Обґрунтування актуальності дослідження	08.12-09.12.22	Виконано
4	Аналіз предмету дослідження та предметної області	10.12-11.12.22	Виконано
5	Проведення дослідження методів та засобів аналітичного опрацювання даних	11.12-12.12.22	Виконано
6	Оформлення розділу «Аналітична частина»	12.12-13.12.22	Виконано
7	Оформлення розділу «Теоретична частина»	13.12-14.12.22	Виконано
8	Оформлення розділу «Практична частина»	14.12-15.12.22	Виконано
9	Оформлення розділу «Охорона праці та безпека в надзвичайних ситуаціях»	07.12-12.12.22	Виконано
10	Нормоконтроль	11.12-14.12.22	
11	Попередній захист роботи	15.12.22	Виконано
12	Захист кваліфікаційної роботи	20.12.22	

Студент _____
(підпис)

Головатий І.Б.

(прізвище та ініціали)

Керівник роботи _____
(підпис)

Баран І.О.

(прізвище та ініціали)

АНОТАЦІЯ

Програмно-апаратне забезпечення самоадаптивного керування вбудованими системами // Кваліфікаційна робота за освітнім рівнем «магістр» // Головатий Ігор Богданович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІм-61 // Тернопіль, 2022 // с. – 73, рис. – 31, табл. – 14 , аркушів А1 – 10 , бібліогр. – 35.

Ключові слова: ВБУДОВАНА СИСТЕМА, ОБРОБКА ЗОБРАЖЕНЬ, ГЕНЕТИЧНИЙ АЛГОРИТМ, АДАПТИВНЕ КЕРУВАННЯ, ФІТНЕС ФУНКЦІЯ

Кваліфікаційна робота присвячена дослідженню методів та засобів самоадаптивного управління вбудованими системами на основі еволюційних алгоритмів. Визначено сучасний шлях розвитку програмного забезпечення вбудованих систем – інтелектуальне управління із властивостями адаптивності. Проаналізовано алгоритми ідентифікації об'єктів бібліотеки OpenCV, визначено деструктивні параметри середовища, котрі чинять негативний вплив на якість цих алгоритмів. Вибрано ряд давачів, які потрібні для моніторингу зазначених факторів зовнішнього середовища. Пропонується подати обробку зображення у виді алгоритму, котрий складається з вибраних процедур фільтрацій та умов їх використання, котрими можуть бути покази підібраних давачів.

Запропоновано генетичний метод алгоритму роботи системи розпізнавання об'єктів. В його основі лежить представлення процедур бібліотеки обробки зображень у вигляді дерев рішень та використання до них різних генетичних алгоритмів.

Побудовано схему адаптивного управління роботою вбудованої системи, для котрої було підібрано відповідне обладнання, розроблено алгоритм функціонування системи та програмне забезпечення на його основі.

ANNOTATION

Hardware and software for self-adaptive control of embedded systems // Master thesis // Holovatiy Ihor // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Systems and Nets, group CIm - 61 // Ternopil, 2022 // p. – 73, fig. – 31 , table. – 14, Sheets A1 - 10 , Ref. - 35.

Keywords: EMBEDDED SYSTEM, IMAGE PROCESSING, GENETIC ALGORITHM, ADAPTIVE CONTROL, FITNESS FUNCTION

The thesis deals with the research of methods and means of self-adaptive control of embedded systems based on evolutionary algorithms. The modern way of software development of embedded systems is determined - intelligent management with adaptability properties. The object identification algorithms of the OpenCV library were analyzed, the destructive parameters of the environment, which have a negative impact on the quality of these algorithms, were determined. A number of sensors, which are needed for monitoring the specified factors of the external environment, have been selected. It is proposed to submit image processing in the form of an algorithm, which consists of selected filtering procedures and conditions of their use, which can be displays of selected sensors.

A genetic method of the object recognition system algorithm is proposed. It is based on the presentation of image processing library procedures in the form of decision trees and the use of various genetic algorithms for them.

A scheme of adaptive control of the work of the built-in system was built, for which appropriate equipment was selected, an algorithm of the system's functioning and software based on it were developed.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

АУ – адаптивне управління

АПК – апаратно-програмний комплекс

ВС – вбудована система

ГА – генетичний алгоритм

ГП – генетичне програмування

ЕП – еволюційне програмування

ПАЗ – програмно-алгоритмічне забезпечення

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

ПРК – промисловий контролер

ФФ – функція фітнесу

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ	6
ВСТУП.....	9
РОЗДІЛ 1. АНАЛІТИЧНА ЧАСТИНА	12
1.1. Аналіз предметної області дослідження.....	12
1.2. Програмне забезпечення вбудованих систем.....	13
1.3. Огляд алгоритмів обробки зображень та детектування об'єктів.....	16
1.3.1. Алгоритм співставлення шаблонів.....	17
1.3.2. Метод пошуку ключових точок.....	18
1.4. Вплив зовнішніх факторів на роботу алгоритмів розпізнавання	19
1.5. Попередня обробка зображень засобами бібліотеки OpenCV	21
1.5. Висновки до розділу	27
РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА	28
2.1. Еволюційні алгоритми та дерева рішень	28
2.2. Подання процедур обробки зображення за допомогою дерев рішень	29
2.3. Еволюція процедур обробки зображень	31
2.4. Функція фітнесу генетичного алгоритму	34
2.5. Генерація довільної процедури обробки зображення	36
2.6. Мутація.....	39
2.7. Схрещування.....	41
2.5. Висновки до розділу	41
РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА.	42
3.1. Практичне застосування розробки	42
3.2. . Апаратна частина стенду.....	44
3.2.1. Центральний обчислювач.....	45
3.2.2. Давач освітленості.....	47
3.2.3. Давач частинок	48

3.2.4. Давач відстані	48
3.2.5. IP –камера.....	49
3.3. Опис програмного забезпечення стенду.....	51
3.3.1. Класи дерев рішень	51
3.3.2. Операції над деревами рішень	53
3.3.3. Генерація системи виявлення об'єкта	54
3.4. Експериментальні дослідження.....	55
3.4.1. Методика проведення експерименту	56
3.4.2. Проведення експерименту та результати аналізу даних.....	58
3.6. Висновки до розділу	61
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	62
4.1. Охорона праці	62
4.2. Ергономічні вимоги до робочого місця користувача ПК	65
4.3. Висновки до розділу	68
ВИСНОВКИ.....	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	70
ДОДАТОК А. Тези конференції	
ДОДАТОК Б. Фрагмент лістингу програми генетичних операцій	

ВСТУП

Актуальність теми. Номенклатура та діапазон застосування ВС досить широкі: автоматизація виробництв, системи управління та контролю, автодорожні та транспортні системи, космічні системи, авіоніка та ракетобудування, військово-промислові комплекси та ін [1,2,3]. Традиційне представлення обчислювальної елементної бази таких систем виходить далеко за межі опису тільки конструкції та схемотехніки, торкаючись дедалі більше питань ускладнення алгоритмів роботи, підвищення рівня гнучкості та абстракції проектування ПЗ [2].

Технологія АУ також є об'ємною темою, найчастіше вона застосовується в системах обробки вбудованих зображень, пошуку та аналізу різних об'єктів на основі отримання візуальної інформації [4]. Дані системи є АПК і призначені для ідентифікації в режимі реального часу об'єктів контролю за допомогою розпізнавання образів на зображеннях, отриманих при допомозі відеокамер [5].

У тематиці АУ системами актуальним завданням є розробка нових способів управління ВС. Потрібний перехід від АУ апаратною частиною систем, котре, по суті, є підлаштуванням тієї чи іншої апаратури за обмеженими параметрами і для обмеженого набору функцій системи, до АУ системою в цілому. Отже, потрібна реалізація нового принципу програмування таких систем.

На підставі викладеного вище, можна зробити обґрунтоване припущення про те, що значущі результати можуть бути отримані при використанні ГП, наприклад, на основі ЕА. При цьому слід використовувати відомі рішення у сфері обробки зображень за допомогою вже реалізованих бібліотек алгоритмів [11]. Виникає потреба в інтегрованому підході та проведення досліджень для визначення технічних можливостей АУ системами обробки зображень, що вбудовуються, у поєднанні з інформацією про навколишні умови, показниками параметрів зовнішнього середовища, що критично впливають на основні алгоритми розпізнавання об'єктів [12].

Тобто, необхідно не тільки вдосконалювати бібліотеки або алгоритми

обробки зображень (що, безперечно, важливо), але в першу чергу розробляти нові принципи їх використання. Крім того, дуже важливо розробити алгоритми конструювання процедур розпізнавання із прив'язкою до конкретних умов експлуатації; потрібно особливе програмне рішення, що дозволяє забезпечити можливість аналізу отриманих даних, що дозволить здійснити розробку адекватних моделей розвитку ВС.

Мета дослідження: розробка методу ГП для формування системи виявлення об'єкта та АПК на основі модуля адаптивного інтелектуального агента.

В роботі поставлено та розв'язано **наступні задачі:**

- виконати огляд основних алгоритмів детектування об'єктів та алгоритмів обробки зображень, реалізованих у бібліотеці OpenCV, здатних підвищити ефективність роботи системи розпізнавання за негативного впливу навколишнього середовища на якість зображень;

- розробити метод автоматичного генерування ефективних процедур фільтрації зображень у вигляді дерев рішень з використанням адаптації до показників давачів навколишнього середовища та дерев рішень;

- розробити системи критеріїв та параметрів для оцінки якості роботи запропонованого способу;

- провести випробування одержаного підходу на діючому виробництві для збору, опрацювання та інтерпретації даних щодо якості роботи АПК.

Об'єкт дослідження: процес адаптивного керування ВС.

Предмет дослідження: технології для адаптивного керування ВС.

Методи дослідження: Методологічну базу дослідження містять фундаментальні положення комп'ютерної інженерії, а також наукові роботи українських та зарубіжних науковців і компаній у галузі управління ВС.

Наукова новизна отриманих результатів.

- запропоновано метод ГП, що базується на принципі пошуку дерева алгоритмів функціонування системи, що дозволяє здійснювати вибір процедури управління ВС у умовах навколишнього середовища;

– вдосконалено метод автоматичного генерування популяції шляхом застосування способу адаптивного вибору структури дерев рішень з процедур бібліотеки обробки зображень та показів давачів зовнішнього середовища, що дозволяє підвищити ефективність розв'язання задач розпізнавання об'єктів у складних умовах застосування.

– оптимізовано метод селекції та мутації для ГП із селективним способом формування та відбору наступних поколінь особин залежно від наведеної оцінки похибки якості роботи системи виявлення об'єктів.

– розроблено ПАЗ комплексу ВС.

Практичне значення одержаних результатів. Основним, практичним результатом дослідження є розроблене алгоритмічне, ПАЗ комплексу, котре може бути застосоване на реально діючому підприємстві при виробництві литих дисків.

Публікації. Результати дослідження апробовано на X науково-технічній конференції «Інформаційні моделі, системи та технології» у вигляді опублікованих тез [32]:

Головатий І.Б. Обробка зображень із використанням генетичного алгоритму. Інформаційні моделі, системи та технології: Праці X наук.-техн. конф. (Тернопіль, 07-08 грудня 2022 р.), Тернопіль, 2022. – С. 73.

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається з вступу, 4 розділів, висновків, списку використаної літератури та додатків. Обсяг роботи: пояснювальна записка – 72 арк. формату А4, графічна частина – 10 аркушів формату А1.

РОЗДІЛ 1

АНАЛІТИЧНА ЧАСТИНА

1.1. Аналіз предметної області дослідження

В даний час є кілька основних підходів до розробки ПЗ ВС. Один із них ґрунтується на використанні бібліотек процедур, що реалізують різні алгоритми обробки та виявлення об'єктів на зображенні [6]. Зазвичай такі бібліотеки містять кілька сотень процедур, які поділяються на групи: геометричні перетворення зображень, фільтрація, виявлення об'єктів за деякими ознаками тощо [5]. До бібліотек такого роду відносяться AForge.NET [7], VXL [8], LTI та бібліотека OpenCV [9], яку варто відзначити особливо, оскільки вона є бібліотекою з відкритим вихідним кодом та містить значну кількість вбудованих алгоритмів обробки зображень. Розробнику надається можливість самостійно вибрати процедури, що його цікавлять, і інтегрувати в ПЗ ВС.

В основі іншого підходу міститься ідея використання спеціалізованих АПК. Системи, побудовані за цим принципом, вже містять необхідний набір процедур обробки під якесь вузькоспеціалізоване завдання, але також дозволяють кінцевому користувачеві налаштовувати деякі наперед визначені параметри функціонування [10].

Дослідженням теорії управління зазначених систем присвячено роботи Ю. Баяковського, У. Претта, Т. Хуанга, Д. Форсайта, Р. Гонсалеса. Значний внесок у розробку методів та алгоритмів адаптивного управління зробили Д. Фогель, Б. Алпатов, С. Жовтов, В. Д. Курганов, В. К. Баклицький.

Тим не менш, відомі роботи не дозволяють однозначно стверджувати про створення ефективних, з сучасної точки зору, підходів до розробки ПЗ для АУ системами обробки зображень, що вбудовуються. Попередній аналіз дозволяє виділити ряд важливих положень, а саме:

- відома множина алгоритмів обробки зображень, проте для переважної

кількості цих алгоритмів не можна заздалегідь передбачити наскільки оптимальне їх застосування до певної конкретної задачі [11];

- майже для кожного завдання завжди можна підібрати ряд алгоритмів, для її вирішення, але вибір найбільш відповідного в більшості випадків ґрунтується на результатах тестування. При цьому розробник або користувач готової системи змушений створювати та тестувати різні комбінації існуючих чи вже реалізованих у системі алгоритмів;

- ПАЗ ставить певні вимоги до зображень, що містить образи об'єктів, що детектуються або аналізуються. Багато ВС застосовуються в складних нестационарних умовах експлуатації, наприклад, коли освітленість зони контролю є мінливою і непередбачуваною, а об'єкти контролю мають різні характеристики і можуть бути забруднені. При цьому, як правило, в системах використовуються засоби формування зображень (відеокамери, об'єктиви, засоби освітлення тощо) які не враховують особливості роботи конкретних алгоритмів розпізнавання [12]. Це призводить до того, що в складних умовах більшість засобів формування зображень являють собою низькоякісні зображення об'єктів, які малоприматні для роботи. Через це ефективність алгоритмів розпізнавання і, як наслідок, самої ВС управління значно знижується;

- адаптивні властивості систем, переважно, націлені на підвищення якості зображень розпізнаваних об'єктів (АУ параметрами засобів формування зображень), тоді як це властивість не поширюється самі алгоритми обробки і зачіпає вищого рівня поведінки ВС [11].

1.2. Програмне забезпечення вбудованих систем

Різноманітність завдань автоматизації виробничих процесів та способів їх вирішення породжує нові вимоги до систем керування. Тут окремим класом слід виділити ВС. З урахуванням існуючих технічних обмежень і фінансово-часових бюджетів, що виділяються, вибір варіанту реалізації може бути досить складним

[13, 14].

Як правило, ВС є програмовані в традиційному стилі звичайні комп'ютери, що відрізняються, наприклад, від ПК обмеженими обчислювальними ресурсами, використанням мов програмування нижчого рівня та обмеженим сервісом інструментальних засобів [2]. У переважній більшості випадків таке програмування виконується в традиційному стилі, не втручаючись в область апаратно-залежних частин і, тим більше, не припускаючи використання методів адаптації до навколишнього середовища. Такий підхід має низку переваг, з яких основним можна вважати просте і зрозуміле (спрощене) уявлення про місце, роль, способи створення ПЗ. Прийнятна якість проектування ПЗ у такій ідеології досягається тільки в рамках невеликих та нескладних проектів ВС, де прийнятне використання шаблонних рішень. У разі, коли від ПЗ та ВС загалом потрібна складна поведінка – такий підхід недоречний.

Стрімкий розвиток цифрової елементної бази, стилю, технології та парадигм програмування визначили необхідність виділення більшої частини сучасних ВС у спеціальну групу, для якої спочатку підкреслюється домінуюче значення програмних технологій на всіх рівнях організації системи. Такі ВС прийнято називати «Переважно програмними системами» (англ. «Software Intensive Systems») [15]. Щодо ВС загального призначення, в яких створення цільових додатків (тобто власне розв'язання прикладних завдань) досить чітко відокремлено від створення обчислювальної платформи (тобто найуніверсальнішої ВС) цей термін означає насамперед те, що акцент у вирішенні задачі зміщений у область програмної розробки. Крім ускладнення загальних принципів і алгоритмів роботи ПЗ для ВС, під цим варто розуміти застосування нових підходів, покликаних підвищити інтелектуальні властивості пристроїв, що розробляються. На перший план виходять вимоги до програмної частини комплексів у їх здатності самостійно приймати рішення на основі обробки інформації, що отримується, і адаптивності до навколишніх умов.

Часто такий підхід застосовують до ВС машинного зору, типовим завданням

яких, наприклад, є обробка та аналіз зображення для пошуку заданих об'єктів [16]. ПАЗ таких систем, в основному, складається зі спеціалізованих процедур та бібліотек обробки та аналізу зображень. Але на сучасному етапі розвитку додають:

- нейромережеві алгоритми, що передбачають навчання на прикладах штучної нейронної мережі [17];
- алгоритми машинного зору на основі моделей нечіткої логіки [18];
- методи візуального програмування процедур аналізу зображень [12].

Описані методи, що застосовуються в даний час, забезпечують основні вимоги до тенденцій розвитку подібного роду, але мають свої специфічні недоліки. Наприклад, нейромережні алгоритми вимагають складання великих навчальних вибірок, причому, при необхідності донавчання такого алгоритму до нових умов експлуатації та розширення навчальної вибірки, розробник буде змушений знову провести навчання нейронної мережі [17]. Примітно, що процедури обробки та аналізу виявлення у штучних нейронних мережах формуються всередині самої мережі, і цей процес у явному вигляді не доступний людині, що не надає можливості аналізу цього процесу [19].

У багатьох випадках корисним підходом є гібридизація, коли ПЗ ВС включає кілька інтелектуальних систем, у тому числі і різних типів. Цей підхід стає все більш поширеним із зростанням продуктивності обчислювальної техніки. Проте, розробка та налаштування навіть однієї такої системи – нетривіальне завдання. Для його вирішення необхідний ефективний інструментарій та підхід до його застосування, що включає здатність інтелектуального агента правильно застосовувати ці інструменти залежно від інформації про навколишні умови. Таким підходом можуть бути алгоритми ЕП, найбільш відомими з яких є ГА та алгоритм ГП [20]. Основоположні ідеї перших методів самоадаптації в ЕП було запропоновано Д. Л. Фогелем. Слід зазначити, що з самоадаптації в ЕП початку генерується нащадок, та був змінюються значення параметрів алгоритму, порядок виконання цих процесів істотно значення [21].

ГА є ефективною процедурою розв'язання складних задач оптимізації. Він

проводить пошук рішень у гіперплощині, що визначається бінарним рядком, яким кодується рішення. Завдання ГА - визначити, яка точка досліджуваної площини доставляють екстремум цільової функції. ГП відрізняється тим, що генотип індивіда представлений не лінійно (у вигляді бінарного рядка чи масиву чисел), а вигляді ієрархічної структури. Потенційно, розробник такої програмної системи не обмежує простір пошуку, а лише визначає елементи та зв'язки, за допомогою яких будуються рішення. Елементи, об'єднуючись в ієрархічні структури, породжують системи з новими властивостями, тому ГП є одним із найпривабливіших інструментів пошуку та адаптації у просторі складних структур [22].

Проте ефективне застосування ГА і ГП на даний момент потребує глибокого знання теорії еволюційного пошуку, що обмежує операції, якими оперують ці алгоритми. Як правило, генеровані програми складаються з найпростіших операцій, які утворюють разом алгоритм обчислень [23]. Використання більш складних операцій, таких як повноцінні бібліотечні функції і дозволило б суттєво розширити можливість застосування ГП у різних ВС. Одним з рішень цієї проблеми є розробка алгоритмів, що самоналаштовуються, які самостійно адаптуються під вирішуване завдання і зовнішні умови, вибираючи ефективні функції та їх параметри.

1.3. Огляд алгоритмів обробки зображень та детектування об'єктів

Оскільки в рамках цієї роботи розглядається завдання створення методу генерації оптимальної процедури обробки зображень за допомогою еволюційних методів, треба з'ясувати призначення головних груп та типів алгоритмів, що становлять цю множину. Потрібно проаналізувати алгоритми опрацювання зображень та детектування об'єктів бібліотеки OpenCV, взятої як елементну основу у розробці програмної частини моделі модуля адаптивного інтелектуального агента.

Практичну реалізацію у складі бібліотеки знаходять три основні групи

алгоритмів розпізнавання об'єктів:

- узгодженої фільтрації з перевіркою збігу, що використовують шаблони зображень об'єктів, які необхідно виявити [11];
- статистичного розпізнавання образів, які передбачають вибірку образів об'єктів з урахуванням їх характерних ознак [11];
- на основі моделей, за допомогою яких порівнюються характерні ознаки мети, що спостерігається з що зберігаються в пам'яті ВС моделями даного зображення;

Критерієм можливості застосування даних функцій бібліотеки в підході, що розробляється, є можливість оцінки точності їх роботи [12]. Це необхідно для цільової функції методу генерації оптимальної процедури обробки зображення, отже, у даній роботі логічно розглядати лише такі алгоритми, що задовольняють цьому критерію.

1.3.1. Алгоритм співставлення шаблонів

Для розпізнавання областей на вихідному зображенні за допомогою узгодженої фільтрації з перевіркою збігу за шаблоном у бібліотеці OpenCV реалізовано функцію `cv2.MatchTemplate()`. Функція накладає шаблон зображення на поточне зображення і згідно з вибраним алгоритмом виконує пошук кореляції між ними [26].

Нижче наведено приклад використання функції розпізнавання простого об'єкта квадратної форми (рис. 1.1)



Рис. 1.1. Результат використання функції

а) зображення шаблону об'єкта; б) аналізоване зображення; в) результат пошуку позначений квадратною рамкою на аналізованому зображенні

1.3.2. Метод пошуку ключових точок

Технологія відшукування ключових точок або тегів входить до складу методів, котрі спрямовані на знаходження абстракцій зображення та виокремлювання на ньому характерних особливостей. Ці особливості потім використовуються для зіставлення двох зображень для визначення у них спільних частин. Властиво не існує чіткого визначення, що є ключовою особливістю зображення. Нею можуть бути як ізольовані точки, так і криві чи деякі пов'язані області. Як приклади таких особливостей можуть бути межі об'єктів, та кути [24].

Метод дозволяє знаходити точки, будувати їх опис (дескриптори) та шукати схожі точки на іншому зображенні [25]. Після чого відфільтровує неправильно зіставлені точки, виходячи з гіпотези, що об'єкт жорсткий (недеформований) і тому всі точки на ньому повинні перетворюватися від одного кадру до іншого узгоджено. Що стосується плоского об'єкта це має бути перспективне перетворення (гомографія) [25].

Після того, як аналізовані зображення завантажені та передані функції виділення точок, проводиться пошук матриці найкращого перспективного перетворення, що здійснює цю відповідність. На рис. 1.2 відзначені "правильні" пари спеціальних точок, тобто пари, що підкоряються знайденому перспективному перетворенню.

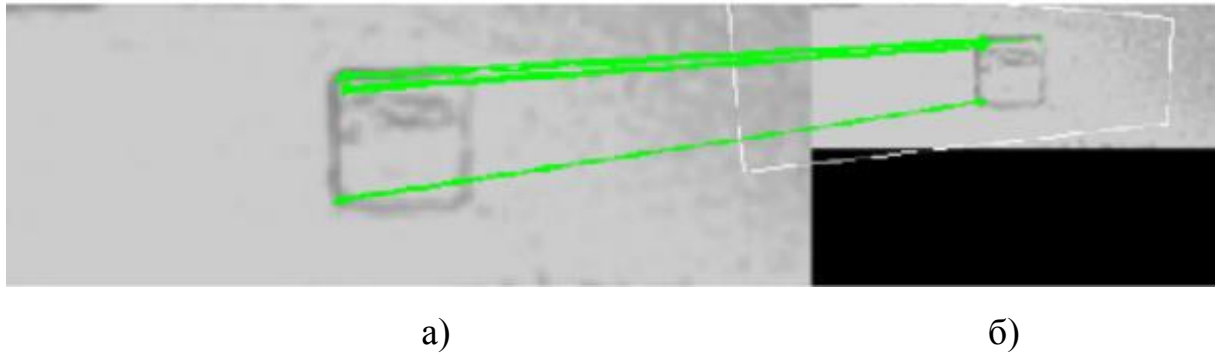


Рис. 1.2. Результат застосування функції detectAndCompare

а) виділені унікальні точки досліджуваного зображення, зіставлені з точками шаблону; б) зображення шаблону об'єкта

Після того, як матриця гомографії знайдена, саме розташування об'єкта у просторі може бути виділено функцією за допомогою середнього значення положення всіх унікальних точок [27].

1.4. Вплив зовнішніх факторів на роботу алгоритмів розпізнавання

Використання зазначених вище методів бібліотеки OpenCV призводить до появи специфічних помилок, які своїм виникненням завдячують впливу зовнішніх факторів чи проблем із апаратурою системи. Експериментально доведено, що зміна освітленості залежно від часу доби, стану атмосфери, відстані до аналізованого об'єкта - всі ці фактори створюють перешкоди, що призводять до хибних тривог та помилкових результатів розпізнавання з неприпустимим рівнем помилки [28].

Розглянемо докладніше, які є чинники, що впливають адекватність визначення і розпізнавання об'єктів на зображенні.

Зміна рівня освітленості, наприклад, спалах світла або відблиски від штучного освітлення виробничого цеху, призводять до появи тіні, високої контрастності та експозиції з фоновим засвіченням. Тінь спотворює форму об'єкта та створює ілюзію мінливості геометричних властивостей об'єкта. Висвітлення значною мірою впливає на можливість ідентифікувати та розпізнати заданий

об'єкт. Для отримання найкращого результату необхідно відстежувати протягом дня зміни інтенсивності та спрямованості сонячного світла, коригуючи положення камери та використовуючи додаткове освітлення, щоб уникнути ситуації з фоновим підсвічуванням. Від рівня освітлення залежить такий параметр камери як значення глибини різкості, воно збільшується при хорошому освітленні. Чим більший показник глибини різкості, тим більше область, в якій об'єкти знаходяться у фокусі (рис. 1.3).



Рис. 1.3. Подання глибини різкості

При низькому рівні освітленості може з'явитись шум на зображенні, що ускладнює ідентифікацію (рис. 1.4).



Рис. 1.4. Приклад шуму при низькому освітленні

- 2) Стан атмосфери (зависання часток, загазованість). Відбувається

розмивання форми об'єкта. Поверхні предметів, які потрапляють у кадр, також впливають на ідентифікацію об'єктів і можуть спричинити хибне спрацьовування. Мокрі поверхні, наприклад, посилюють відбите світло, матові поглинають деяку частину відбитого світла.

3) Спектральні зміни освітленості пов'язані з часом доби. Змінюється чутливість камери, змінюється чіткість об'єктів.

4) Ефекти, пов'язані з особливостями камери:

– внаслідок тривалої роботи відеообладнання у складі ВС (промислової) в систему відео об'єктива можуть потрапляти сторонні об'єкти, такі як частинки пилю, мастильно - охолоджувальні рідини і т.д.;

– розкручування деталей відеокамери через вібрацію промислового обладнання.

Перераховані ефекти будуть посилюватися з часом, що вимагатиме переналаштування алгоритмів під зміни як отримані зображення. Для можливості реалізації властивостей адаптивності методу, що розробляється, прийнято рішення аналізувати вищевказані дестабілізуючі фактори зовнішніх умов за допомогою низки давачів.

1.5. Попередня обробка зображень засобами бібліотеки OpenCV

Незважаючи на те, що методи детектування об'єктів бібліотеки OpenCV власними силами є потужними інструментами і для вирішення тривіальних завдань може вистачити однієї такої функції, це не є достатнім для застосування в агресивних умовах виробництва. Крім цих методів, новий підхід повинен боротися з негативними факторами, нейтралізуючи ефекти погіршення якості оброблюваних зображень за допомогою, наприклад, фільтрів. Фільтри здатні змінити одержувані кадри виключаючи перешкоди, причому, враховуючи особливість підібраних алгоритмів розпізнавання, робити це слід на всі зображення як шаблонні, так і тестовані.

Згладжуючі фільтри. Згладжування або розмиття зображення - це одна з найпростіших операцій обробки зображень, що часто використовуються. Як правило, розмиття застосовується, щоб зменшити шум або артефакти, які обумовлені вибором камери та зовнішнім освітленням. Також згладжування відіграє важливу роль при необхідності зменшити роздільну здатність зображення та отримати піраміду зображень різного масштабу (image pyramids) [27].

Розробники OpenCV реалізували кілька функцій розмиття зображення. Далі зупинимося на деяких із них, а саме розглянемо функції GaussianBlur та medianBlur [11]:

Функція GaussianBlur здійснює розмиття за допомогою обчислення матриці згортки зображення з дискретним ядром Гауса зі стандартними відхиленнями, рівними σ_x та σ_y по осях O_x та O_y відповідно. Зауважимо, що з виклику цієї функції накладається обмеження параметр kSize. Ширина і висота ядра повинні бути позитивними та непарними, або нульовими, якщо розмір ядра визначається стандартними відхиленнями. Формула ядра Гауса:

$$G_0(x, y) = Ae^{-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}} \quad (2.1)$$

Далі показаний приклад результату виконання функції GaussianBlur (рис. 1.5).



Рис. 1.5. Використання функції GaussianBlur

Функція `medianBlur` забезпечує розмиття за допомогою медіанного фільтра. Він, як правило, застосовується для зменшення коефіцієнту шуму чи т.зв. "згладжування" зображення. Вибирається певний шаблон, який накладається на всі пікселі зображення. Набір інтенсивностей пікселів, які накриті шаблоном, сортується і вибирається інтенсивність, що знаходиться в середині відсортованої множини. По суті, визначається медіана у відсортованому наборі даних. Розмір шаблону визначається параметром `kSize`. Фільтр застосовується до матриць різноманітного розміру, проте на противагу до матриці згортки, визначений розмір матриці має вплив лише на число пікселів, що розглядаються. Нижче показана робота медіанного фільтра для встановленого розміру ядра, що дорівнює трьом (рис. 1.6).



Рис. 1.6. Принцип зміни пікселів у роботі медіанного фільтра

Для поточного пікселя, саме ті пікселі, які «попадають» у матрицю, сортується, і вибирається середнє відсортованого набору. Саме це значення і буде вихідним для власне поточного пікселя. Далі показаний приклад використання функції `medianBlur` (рис. 1.7)



Рис. 1.7. Застосування функції medianBlur

Фільтри з використанням морфологічних перетворень. Дилатація (морфологічне розширення) - згортка зображення або виділення зображення з деяким ядром [1, 2]. Власне ядро може бути довільної форми та розміру. При цьому в ньому визначається єдина провідна позиція (anchor), котра співвідноситься із поточним пікселем для обчислення згортки. Для більшості випадків як ядро беруть квадрат чи коло з провідною позицією у центрі.

Властиво ядро можна визначати як шаблон або маску. Використання дилатації – прохід шаблоном (рис. 1.8) по всьому зображенню та застосуванню оператора відшукування локального максимуму до інтенсивності пікселів зображення, котрі потрапляють у шаблон.

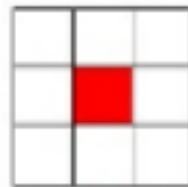


Рис. 1.8. Ядро морфологічних операцій

Така операція дає збільшення світлих фрагментів на зображенні (рис. 1.9). Тут сірим відмічені ті пікселі, котрі внаслідок використання дилатації стануть білими.

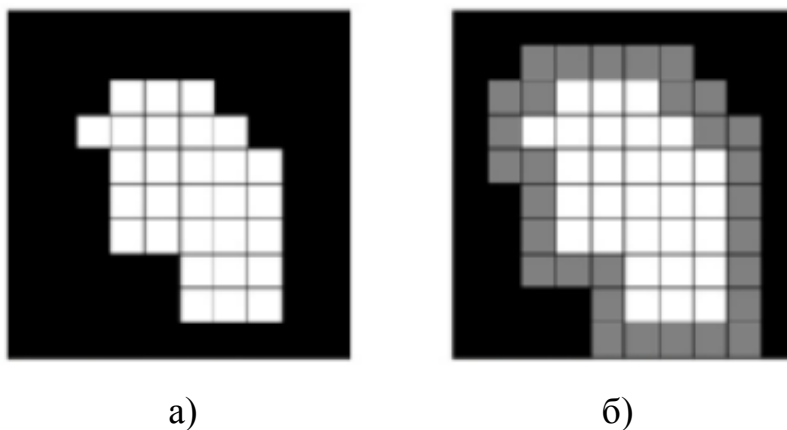


Рис. 1.9. Застосування морфологічних операцій
 а) оригінальне зображення; б) результат роботи дилатації

Ерозія (морфологічне звуження) – зворотна операція [1, 2]. Робота ерозії схожа до дилатації. Єдина відмінність – застосовується оператор відшукування локального мінімуму (рис. 1.10). Тут сірим показані пікселі, котрі будуть чорними в результаті роботи ерозії.

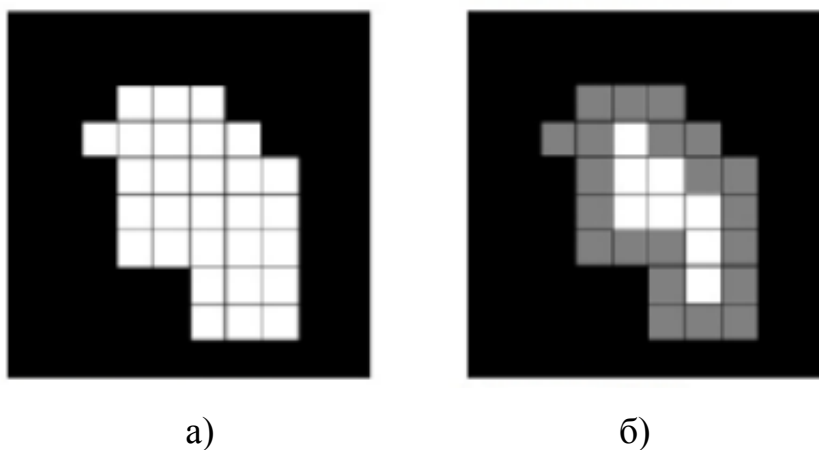


Рис. 1.10. Приклад застосування ерозії
 а) оригінальне зображення; б) результат роботи ерозії

Результат виконання програми показано на прикладі деякого зображення (рис. 1.11 та 1.12). Як видно з рисунків, застосування ерозії призвело до звуження контурів, дилатації – розширення.



Рис. 1.11. Результат застосування дилатації (ліворуч)



Рис. 1.12. Результат застосування ерозії (ліворуч)

Вирівнювання гістограм. Один з найпоширеніших дефектів фотографічних, сканерних та відео зображень це наявність слабого контрасту. Він багато в чому зумовлений малим діапазоном яскравостей, що відтворюються. Під контрастом, власне, мається на увазі різниця найбільшого та найменшого значень яскравості. Властиво контрастність зображення можна збільшити при зміні яскравості всіх елементів зображення та більшому діапазоні яскравостей.

Мета вирівнювання – усі рівні яскравості повинні мати однакову частоту, а гістограма відповідати рівномірному закону розподілу. У бібліотеці OpenCV реалізовано функцію `equalizeHist` [11], яка забезпечує покращення контрастності зображення (рис. 1.13) за допомогою вирівнювання гістограми [11, 16].



Рис. 1.13. Результат вирівнювання гістограми

1.6. Висновки до розділу

Було проведено аналіз предметної галузі застосування ВС, позначено актуальний вектор розвитку їх ПЗ щодо інтелектуального управління та властивостей адаптивності.

Наведено огляд алгоритмів детектування об'єктів бібліотеки OpenCV, виявлено деструктивні фактори зовнішнього середовища, що впливають на якість цих алгоритмів. Вибрано перелік давачів, необхідних для моніторингу зазначених факторів. Розглянуто процедури обробки зображень, які необхідно застосовувати спільно з алгоритмами виявлення з метою нівелювання впливів зовнішніх умов.

Для підвищення ефективності роботи алгоритмів розпізнавання, залежно від зовнішніх умов, можуть бути потрібні різні процедури фільтрації (згладжування, морфологічне перетворення тощо). В окремих випадках застосування алгоритмів розпізнавання, попередньої обробки зображень слід піддавати як зображення з камери, так і шаблон об'єкта, що розпізнається. Таким чином, обробку зображення можна подати у вигляді алгоритму, що складається з підібраних процедур фільтрацій та умов їх застосування, якими можуть бути покази давачів зовнішнього середовища. У разі необхідності використовувати такі засоби вибору процедур, які б повністю автоматизували процес генерації оптимальної процедури. З викладеного вище, можна робити цілком обґрунтований висновок, що ЕП, саме метод ГА, дозволять отримувати найбільший ефект.

РОЗДІЛ 2

ТЕОРЕТИЧНА ЧАСТИНА

2.1. Еволюційні алгоритми та дерева рішень

В рамках даної роботи пропонується новий метод кодування готових процедур обробки зображень для ГП, заснований на їхньому представленні за допомогою дерев рішень. Це дозволить встановити обмеження на побудову загальної процедури обробки на основі показань з давачів навколишнього середовища.

Грамотне обрання варіанту кодування програми в ГА є однією з головних проблем ГП [29]. Програма має бути представлена у кодї так, щоб у неї випадкові зміни була змога вносити (оператор мутації) та поєднувати два алгоритми в один (оператор схрещування). Саме це дозволить реалізувати принцип адаптивності передбачуваного методу. Результатом кодування буде дерево процедури.

Відповідно до запропонованого методу, спочатку, код програми процедури обробки зображення перетворюється на дерево розбору (рис. 2.1).

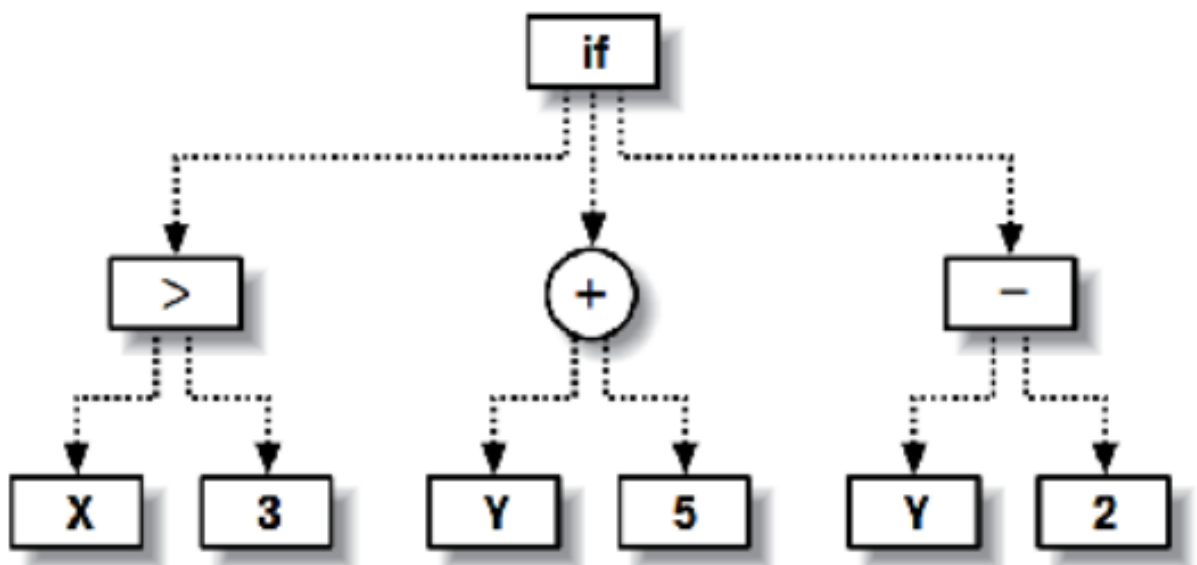


Рис. 2.1. Приклад програми, котра представлена у виді дерева

З рисунка видно, кожен вузол є або операцією над дочірніми вузлами, або «листовим», наприклад, параметром чи константою. Так, колом представлена операція підсумовування двох гілок, в даному випадку значень змінної Y і константи 5. Обчислена сума передається вище вузлу, який застосовує власну операцію до своїх нащадків.

Зверніть увагу, що один із вузлів відповідає операції `if`; це означає, що якщо значення лівої гілки дорівнює `true`, то він повертає свою центральну гілку, а якщо `false` – то праву гілку. Дерево може бути рекурсивним, якщо допускаються посилання на вузли, що розташовані вище. За допомогою таких дерев можна організовувати цикли та складніші керуючі структури.

2.2. Подання процедур обробки зображення за допомогою дерев рішень

У даному випадку кожна ланка дерева має один з трьох типів даних:

- логічний - фальш чи істина;
- числовий - дійсне число від 0 до 1;
- зображення.

Крім типів даних, кожна ланка дерева характеризується одним із трьох видів:

- функція – ланка з нащадками;
- змінна (показ одного з давачів) - лист дерева;
- константа (логічне значення, число або вихідне зображення) – лист дерева.

Використані функціональні ланки наведено у таблиці 2.1.

Функціональні ланки

Функція	Тип даних	Аргументи	Результат
Порівняння («>»)	Логічний	1) число 1 2) число 2	Істина, якщо число 1 більше, ніж число 2; інакше фальш
Умовний вираз (<if>)	Зображення	1) умова (логічний тип) 2) зображення 1 3) зображення 2	Якщо умова = істина, то зображення 1; інакше зображення 2
Розмиття Гаусса	Зображення	1) src - зображення 2) ksize – розмір ядра (числовий тип)	Зображення src, після застосування розмиття Гауса з ядром розміру [20ksize] x [20ksize]
Медіанний фільтр	Зображення	1) src - зображення 2) ksize – розмір ядра (числовий тип)	Зображення src, після застосування розмиття медіанним фільтр з ядром розміру [20ksize] x [20ksize]
Дилатація	Зображення	1) src - зображення 2) ksize – розмір ядра (числовий тип) 3) ktype – тип ядра (числовий тип)	Зображення src після застосування дилатації з круглим ядром розміру [20ksize] x [20ksize]. Ядро має квадратний тип, якщо ktype < 0,5 і круглий в іншому випадку
Ерозія	Зображення	src - зображення ksize - розмір ядра (числовий тип) ktype - тип ядра (числовий тип)	Зображення src після застосування ерозії з круглим ядром розміру [20ksize] x [20ksize]. Ядро має квадратний тип, якщо ktype < 0,5, і круглий інакше
Вирівнювання	Зображення	1) src - зображення	Зображення src після вирівнювання гистограми

Вподальшому в роботі будуть наведені приклади із використанням ланок.

2.3. Еволюція процедур обробки зображень

Принцип роботи ГА полягає у створенні набору випадкових рішень, що називається населенням. На кожному етапі оптимізації цільова функція обчислюється для всієї популяції, у результаті виходить ранжований перелік рішень. Після ранжування рішень формується нова популяція, котра носить назву наступного покоління. Спочатку до нової популяції включаються найкращі рішення з поточної. Цей процес зветься елітизм [30].

Крім них, до наступної популяції входять абсолютно нові рішення, що отримуються шляхом модифікації найкращих за допомогою генетичних операцій, таких як схрещування та мутація. Конкретні реалізації способів уявлення особин і генетичних операцій породжують різні різновиди ЕП. Потім цей процес повторюється - нова популяція ранжується та створюється чергове покоління.

Так триває задане число разів або до тих пір, поки протягом кількох поколінь не спостерігається жодних покращень [32]. Принцип методу показаний на рис. 2.2.

Для аналізованої задачі генерації системи виявлення об'єкта на зображенні, як генотип був обраний вектор $s = \{T_1, T_2, c\}$. Тут T_1 - дерево рішень, відповідне до обробки зображення з камери, T_2 - дерево рішень, відповідне до обробки шаблонного зображення, c - параметр алгоритму виявлення об'єкта.

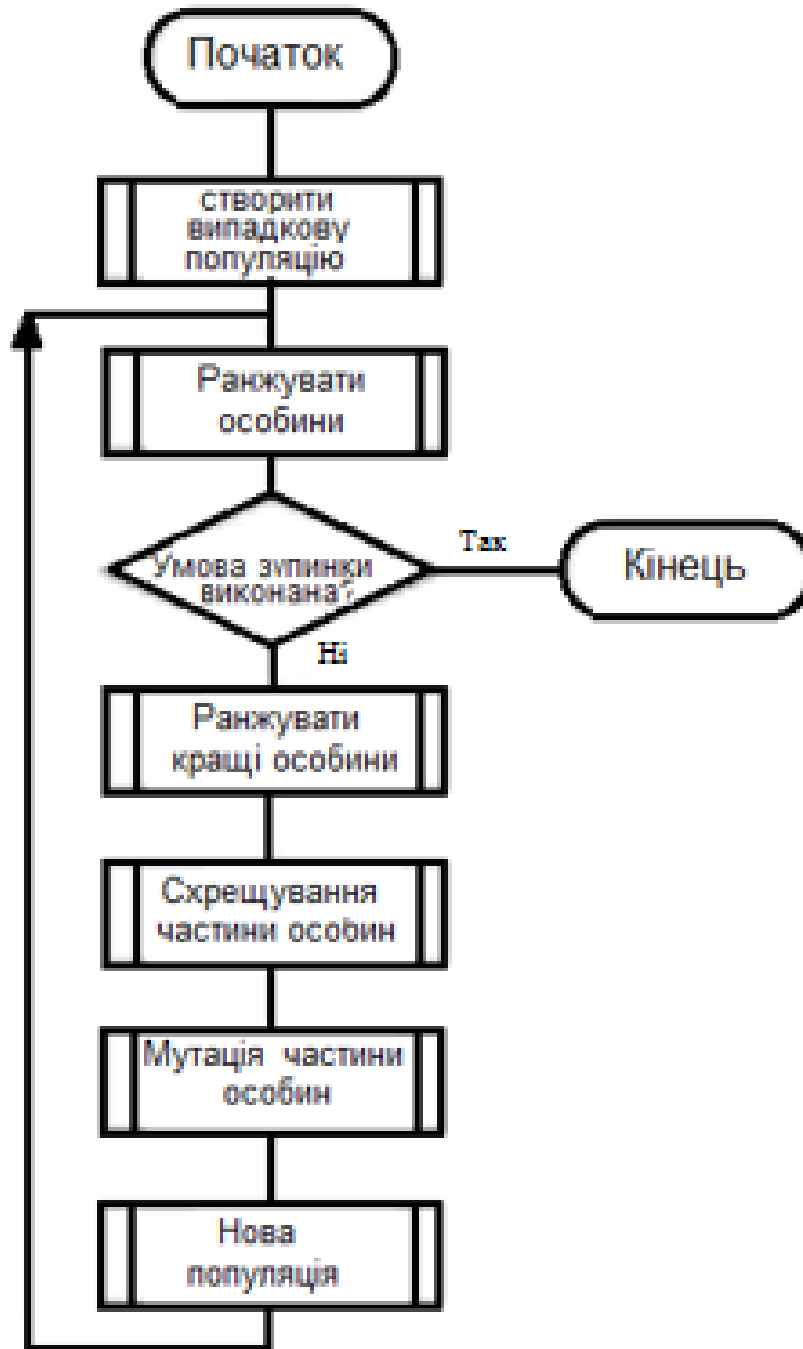


Рис. 2.2. Блок-схема ГА

Вважається, що в результаті виконання ГА буде отримана система, здатна максимально точно встановлювати положення об'єкта, що цікавить, на зображенні або виявляти факт його відсутності. Схема роботи цієї системи представлена на рис. 2.3.

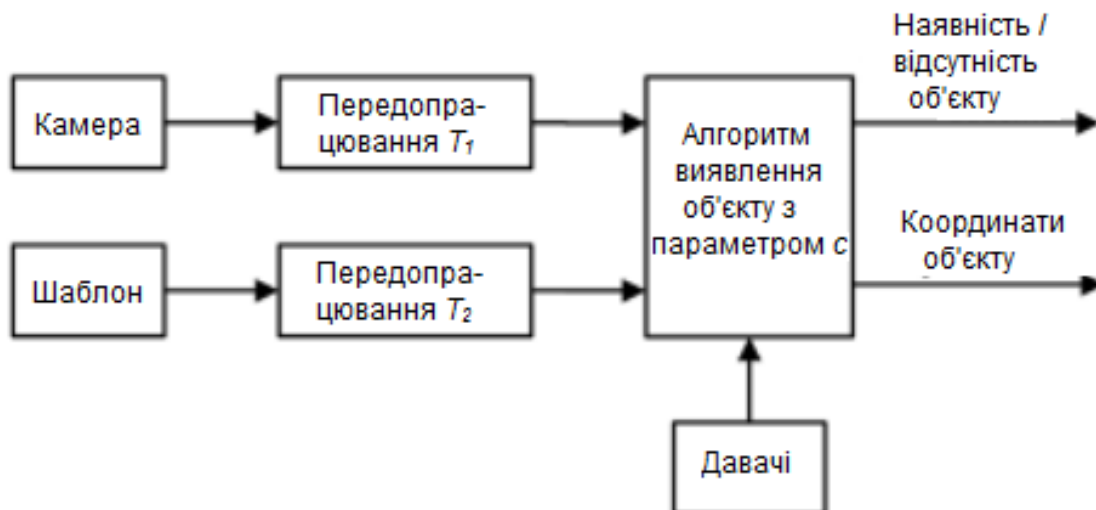


Рис. 2.3. Схема виявлення об'єкта

Для перевірки якості роботи системи використовується вибірка зображень, елементи якої складаються з:

- зображення, отриманого з камери;
- значень давачів зовнішнього середовища у момент отримання цього зображення;
- координат центральної точки об'єкта, що цікавить, на зображенні або вказівки факту його відсутності.

Система виявлення застосовується до кожного елемента вибірки. За результатами застосування системи до всієї вибірки складається таблиця спряженості (табл. 2.2).

Таблиця 2.2

Таблиця спряженості

	Об'єкт присутній	Об'єкт відсутній
Об'єкт виявлено	N_{in}	N_{xn}
Об'єкт не виявлено	N_{xn}	N_{in}

Тут N_{in} – число істинно позитивних випадків;

N_{xn} - число хибно позитивних випадків;

N_{in} - число істинно негативних випадків;

N_{xn} - число хибно негативних випадків.

Крім того, для істинно позитивних випадків також обчислюється похибка виявлення об'єкта ρ_i .

$$\rho_i = \begin{cases} \sqrt{(x_i^* - x_i)^2 + (y_i^* - y_i)^2}, & \text{якщо } i \text{ – істинно позитивний випадок;} \\ 0, & \text{у решті випадків;} \end{cases} \quad (2.1)$$

де i - номер елемента вибірки;

(x_i^*, y_i^*) - істинне положення центру об'єкта на i -му зображенні (у пікселях);

(x_i, y_i) - знайдене положення центру об'єкта на i -му зображенні (у пікселях).

Для ідеальної системи виявлення вірно таке:

– кількість помилок класифікації зображень (об'єкт, що цікавить, або відсутній) дорівнює нулю: $N_{xn} = N_{xn} = 0$;

– для кожного позитивного випадку (з номером i у вибірці) похибка виявлення дорівнює 0: $\rho_i = 0$.

Для отримання системи виявлення, найближчої до ідеальної, необхідно формалізувати критерії якості її роботи ФФ.

2.4. Функція фітнесу генетичного алгоритму

ФФ складається відповідно до таких вимог:

- чим нижче значення ФФ, тим якісніша система виявлення об'єкта;
- чим менше помилок класифікації зображень (хибно позитивних і хибно-негативних випадків), тим нижче значення ФФ;
- що менше похибка виявлення (відстань між істинним і знайденим

положеннями об'єкта, що цікавить), тим нижче значення ФФ;

– вважати результати виявлення однаково поганими при перевищенні похибкою ρ_i певного порога ρ_{max} ;

– хибнопозитивні та хибнонегативні випадки мають різні ваги.

Відповідно до вищевикладеного, було обрано таку ФФ:

$$Q(s) = w_{in} \sum_{i=1}^{N_{виб}} \rho_{i прив} + w_{хп} N_{хп} + w_{хн} N_{хн}, \quad (2.2)$$

де $\rho_{i прив}$ - наведене значення похибки визначення мітки в i -му зображенні навчальної вибірки $\rho_{i прив} \in [0, 1]$; обчислюється за формулою (2.3):

$$\rho_{i прив} = \left[\frac{\min(\rho_i, \rho_{max})}{\rho_{max}} \right]^{\gamma_{in}}; \quad (2.3)$$

залежність між ρ_i і $\rho_{i прив}$ показана на рис. 2.4;

$N_{виб}$ - розмір навчальної вибірки;

$\gamma_{in} \in (0, \infty)$ - показник, що регулює кривизну залежності ρ_i та $\rho_{i прив}$;

w_{in} - штрафний коефіцієнт похибки в позитивних випадках;

$w_{хп}$ - штрафний коефіцієнт хибно позитивних випадків;

$w_{хн}$ - штрафний коефіцієнт хибно негативних випадків.

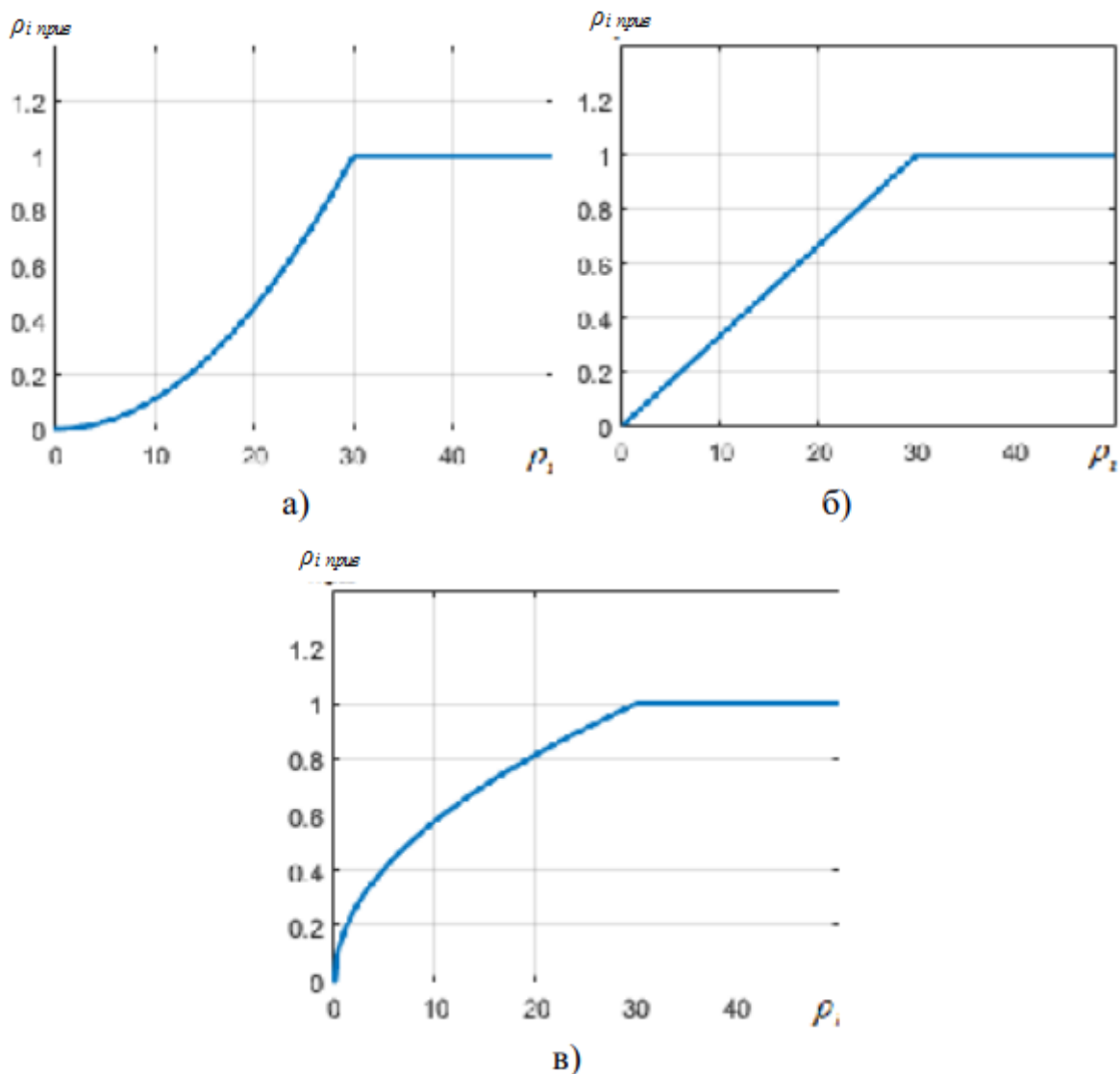


Рис. 2.4. Залежність між $\rho_i \text{ прив}$ та ρ_i
 а) $\gamma_{\text{прив}} > 1$; б) $\gamma_{\text{прив}} = 1$; в) $0 < \gamma_{\text{прив}} < 1$

2.5. Генерація довільної процедури обробки зображення

Для створення початкової популяції генетичного алгоритму, і, навіть, оператора мутації потрібен алгоритм генерації довільного дерева рішень. У ньому випадково вибираються ланки і з'єднуються так, щоб вийшло коректне дерево рішень, глибина якого не перевищує заданого максимального значення $N_{\text{макс. глїб}}$. Імовірності вибору кожної ланки при заданому типі даних наведено у таблицях 3.3, 3.4 та 3.5.

Таблиця 2.3

Ймовірності ланок логічного типу

Ланка	Ймовірність
Порівняння («>»)	0,5
Константа (випадковий вибір з 0 та 1)	0,5

Таблиця 2.4

Ймовірності ланок числового типу

Ланка	Ймовірність
Змінна (з випадковим номером)	0,5
Константа (випадкове число від 0 до 1)	0,5

Таблиця 2.5

Ймовірності ланок із типом даних «зображення»

Ланка	Ймовірність
Умовний вираз («if»)	0,2
Фільтр (випадковим чином вибирається один із наступних фільтрів: розмиття Гауса, медіанний фільтр, дилатація, ерозія, вирівнювання гистограми)	0,5
Константа (вихідне зображення)	0,3

Блок-схема алгоритму генерації довільної процедури попередньої обробки зображення наведено на рис. 2.5. Вона створює ланку зазначеного типу даних, і якщо ланка є функціональною, то рекурсивно створює гілки кожному аргументу функції. Як тип даних кореневої ланки потрібно вказувати зображення.

При генерації початкової популяції даний алгоритм (див. рис. 2.5) запускається двічі для кожної особи: перший раз – для створення дерева T_1 , другий

– для створення дерева T_2 . Крім цього, випадковим чином генерується параметр алгоритму виявлення об'єкта з діапазону $[0, 1]$.

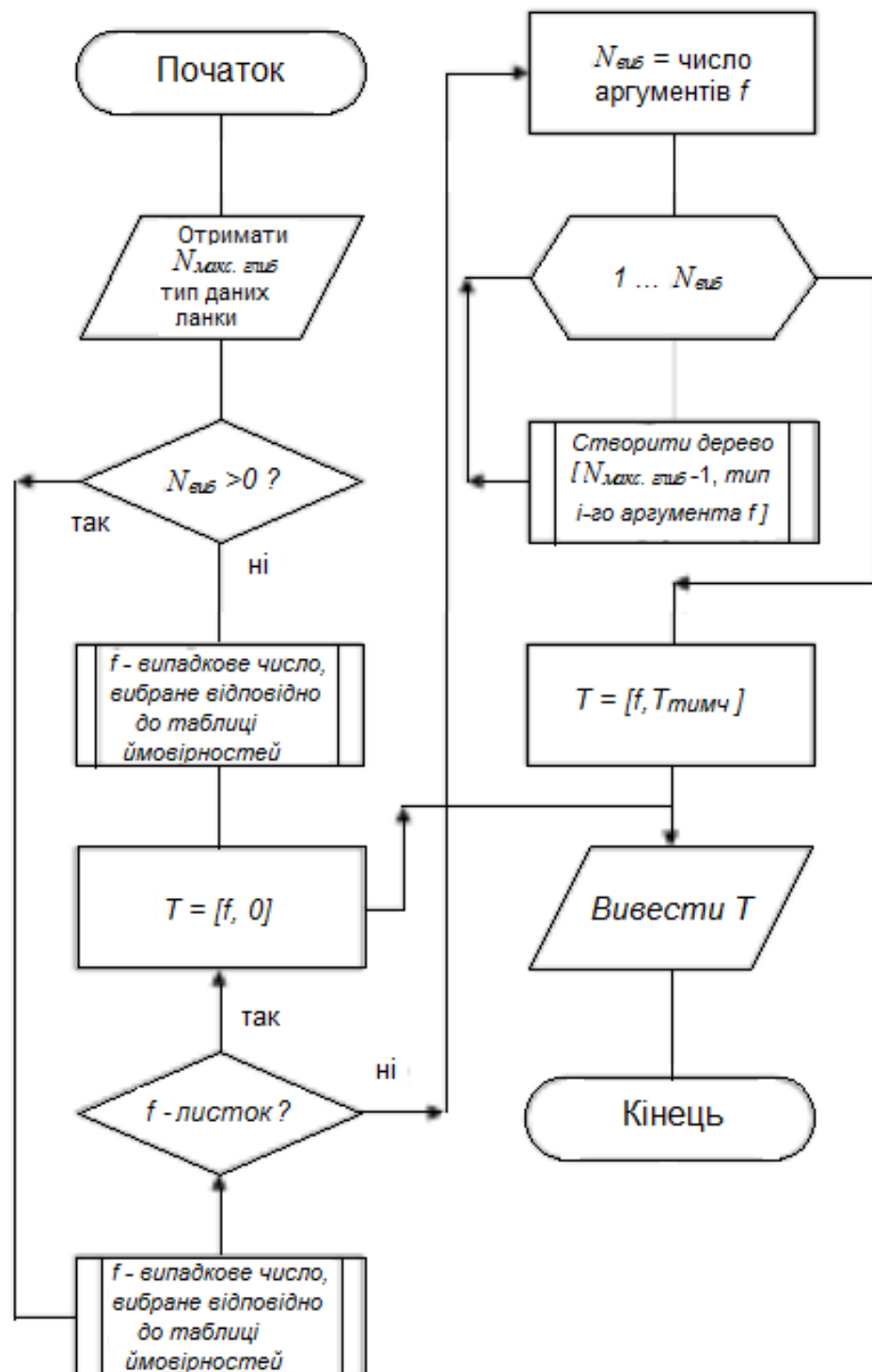


Рис. 2.5. Блок-схема алгоритму генерації довільної процедури передобробки зображення

2.6. Мутація

Алгоритм мутації використовує у своїй роботі алгоритм з рис. 2.5. У дереві рішень відбувається обхід ланок у глибину, і для кожної ланки є можливість мутувати, тобто замінитися довільно згенерованим деревом. В цьому випадку обхід ланок у цій гілці дерева припиняється. Також є обмеження, що глибина дерева, що мутує, не перевищує $N_{\text{макс. глб}}$.

Блок-схема мутації процедури передобробки зображення наведено рис. 2.6. Цей алгоритм виконується двічі для особи, що мутує: перший раз - для дерева T_1 , другий - для дерева T_2 .

Крім того, до параметра c додається випадкове число з центрованим нормальним розподілом. Якщо результат виявиться меншим за 0, то c замінюється значенням 0; якщо результат перевищить 1, то c замінюється значенням 1.

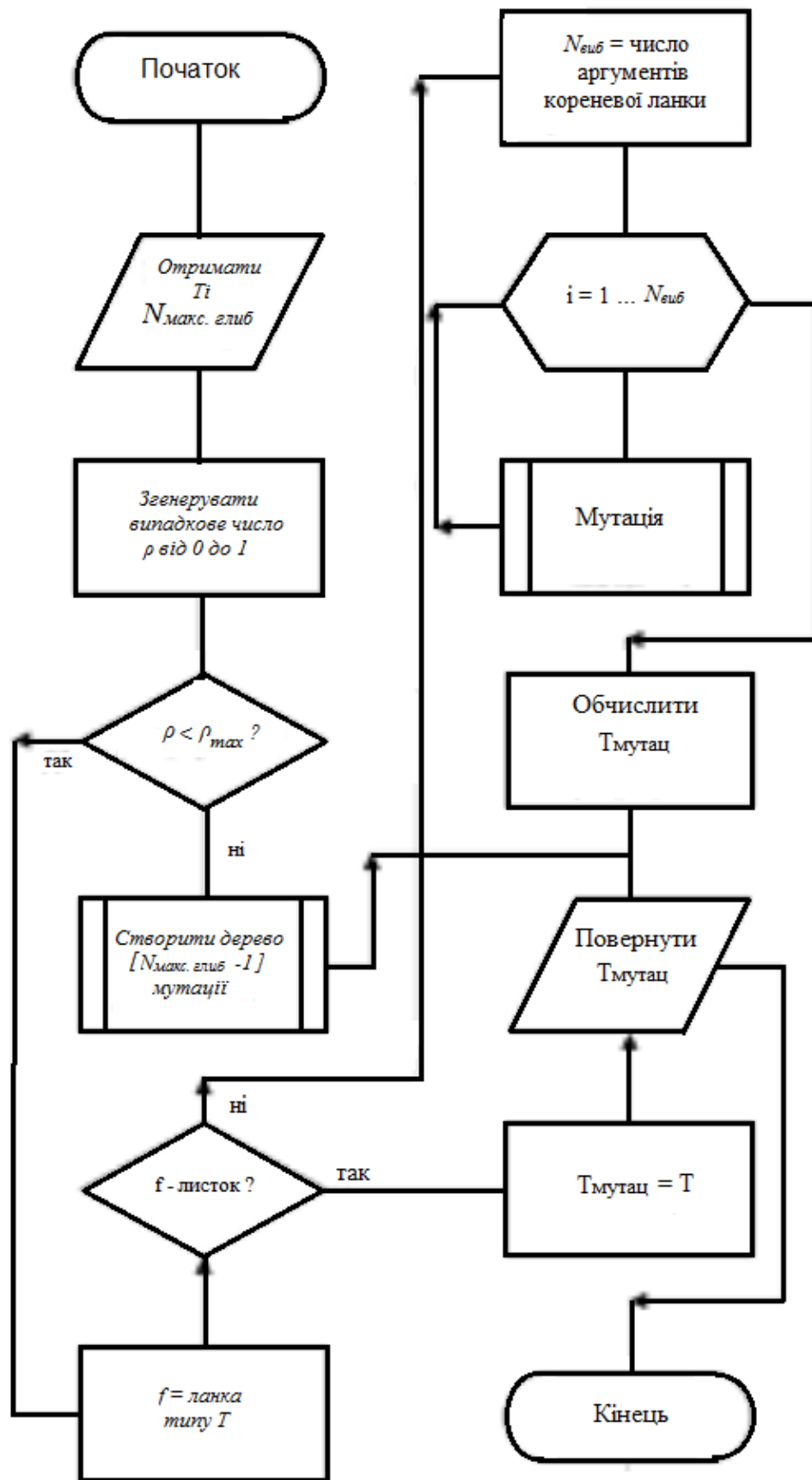


Рис. 2.6. Блок-схема алгоритму мутації дерева

3.7. Схрещування

Схрещування особин $s_A = \{T_{A1}, T_{A2}, c_A\}$ та $s_B = \{T_{B1}, T_{B2}, c_B\}$ відбувається наступним чином: з рівною ймовірністю витягуються один за одним відповідні елементи генотипів особин A або B та формується дочірня особина C . Приклади можливих генотипів особи C : $s_c = \{T_{B1}, T_{B2}, c_A\}$, $s_c = \{T_{A1}, T_{B2}, c_B\}$, $s_c = \{T_{A1}, T_{A2}, c_A\}$.

2.5. Висновки до розділу

Запропоновано метод ГП алгоритму роботи системи розпізнавання об'єктів. В основу методу покладено представлення процедур бібліотеки обробки зображень у вигляді дерев рішень та застосування до них ГА (схрещування, мутації, відбору).

Розроблено та реалізовано метод автоматичного генерування популяції, що відрізняється від відомих способом адаптивного вибору структури дерев рішень з процедур бібліотеки обробки зображень та показань давачів зовнішнього середовища, що дозволяє підвищити ефективність вирішення завдань розпізнавання об'єктів у складних умовах застосування.

Розроблено та реалізовано метод селекції для ГП, який відрізняється від відомих селективним способом формування та відбору наступних поколінь особин залежно від наведеної оцінки похибки якості роботи системи виявлення об'єктів.

Запропоновано спосіб мутації особин для ГА, заснований на розробленому способі автоматичного генерування популяції.

РОЗДІЛ 3

ПРАКТИЧНА ЧАСТИНА

3.1. Практичне застосування розробки

Отримані результати дисертаційних досліджень пропонується впровадити та випробувати на діючому виробництві. Наприклад, у цеху з виробництва литих дисків на автоматичній лінії механічної обробки виробів (рис. 3.1) можна розгорнути стенд комплексу розпізнавання спеціальних міток орієнтації автомобільного диска.



Рис. 3.1. Автоматична лінія виробництва

При обробці виробу на автоматичній лінії він послідовно переміщається від одного пристрою до іншого. Диск розміщується та закріплюється у визначено положенні для подальшої обробки за допомогою промислового маніпулятора, саме для управління цим процесом у лінію вбудована система технічного зору, що відповідає за правильну орієнтацію виробів у захопленні промислового робота. Для

контролю положення диска, на його боці (реборді) відлита спеціальна фізична мітка квадратної форми (рис. 3.2, б).

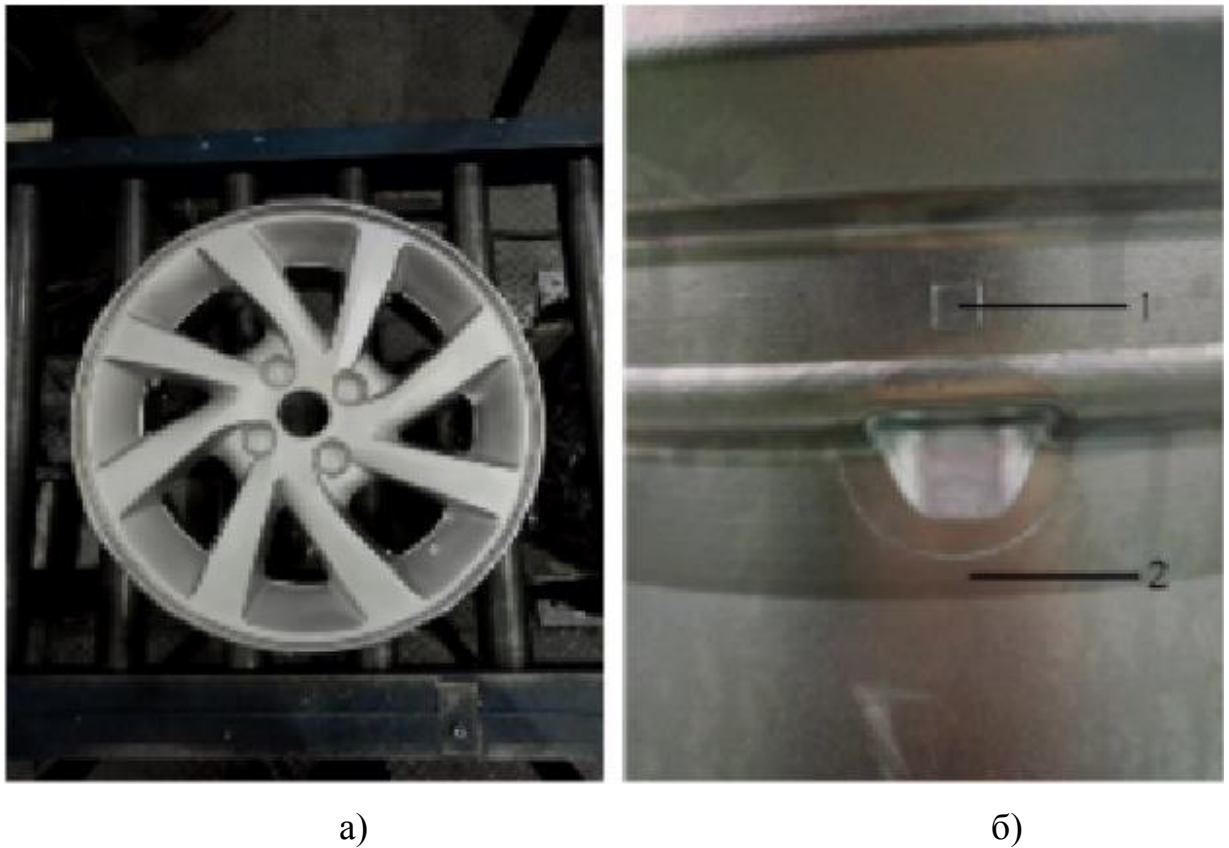


Рис. 3.2. Автомобільний диск

а) вид зверху; б) вид збоку: 1) мітка орієнтації; 2) реборд диска

Перед кожною технологічною операцією, виріб поміщається на спеціальний стенд (рис. 3.3), де встановлено об'єktiv ВС обробки зображень, вона повинна виявити фізичну мітку та повідомити координати мітки щодо кадру, отримані координати будуть передані ПРК як керуючий сигнал для наступних маніпуляцій.

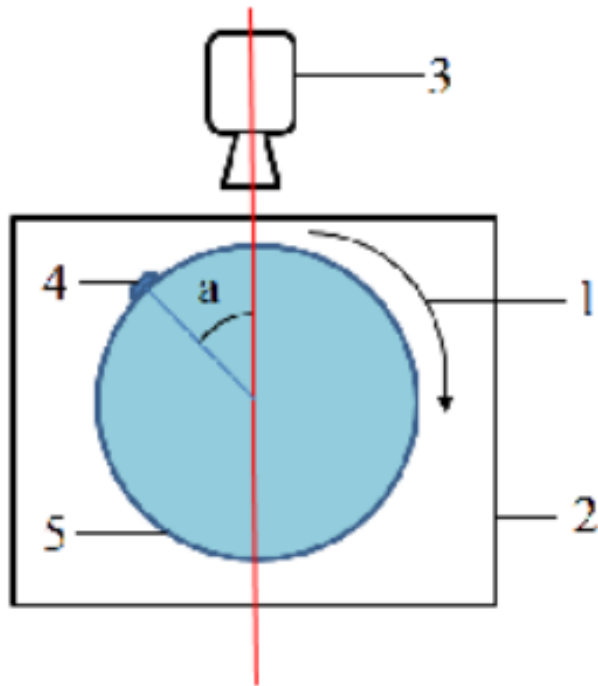


Рис. 3.3. Стенд аналізу орієнтації дисків

- 1) напрямок обертання диска; 2) поворотний стіл стенду; 3) об'єktiv IP-камери; 4) мітка орієнтації диска; 5) автомобільний диск

Необхідно врахувати, що окрім досить непростого типового завдання розпізнавання та виявлення об'єкта по візуальному зображенню, завдання ще більше ускладнює навколишнє середовище промислового виробництва. З'являється можливість перевірки адаптивної якості розробленого методу.

3.2. Апаратна частина стенду

Основний склад системи визначається відповідно до застосування розроблених алгоритмів і методів виявлення заданого об'єкта. Основний набір датчиків визначається відповідно до переліку вимірюваних показників, визначеному в розділі 1. Реєстрація кожного з показників здійснюється за допомогою відповідного датчика, а зображення об'єкта, що аналізується, надходить з IP-камери (рис. 3.4).

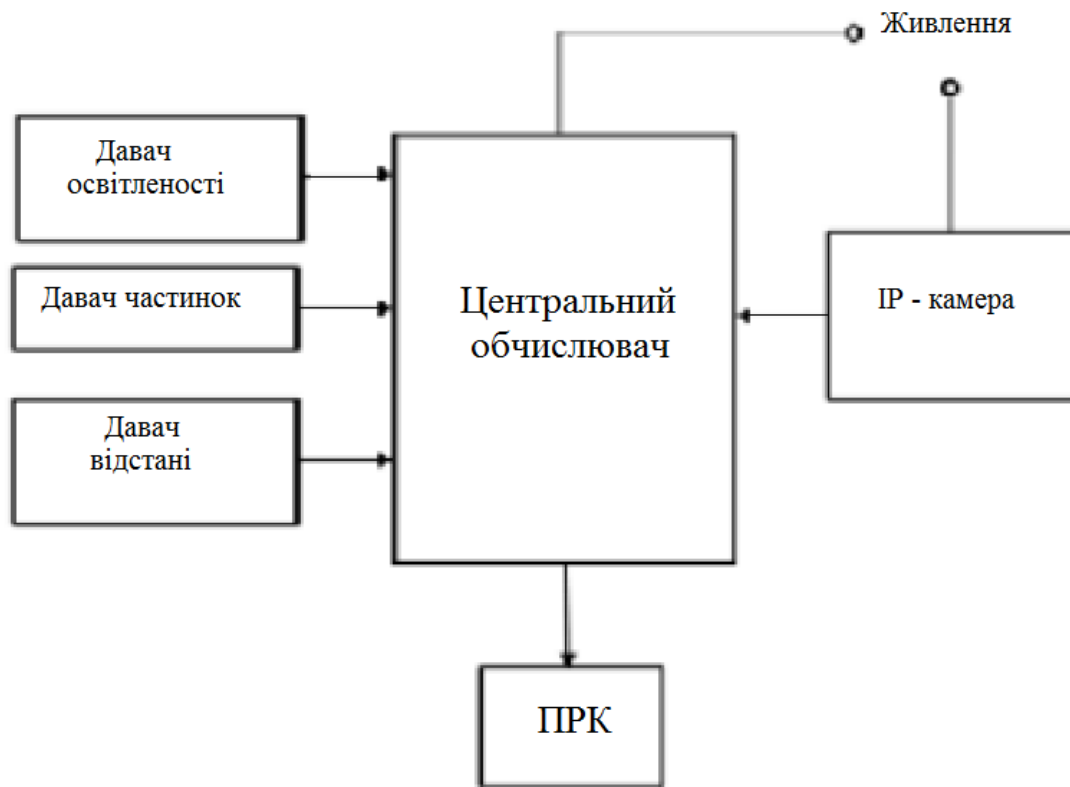


Рис. 3.4. Структурна схема стенду

Зазначимо, що давач відстані необхідний у складі апаратури для вимірювання відстані до об'єкта, оскільки останній знаходиться на боці автомобільного диска, через різні діаметри виробів, може змінюватися відстань від диска до об'єктиву камери.

3.2.1. Центральний обчислювач

Для зняття даних з давачів, отримання зображення з IP -камери, подальшої обробки зображення та відправки результату в ПЛР по TCP/IP протоколу використано промисловий одноплатний комп'ютер EPI-945GSE-A1-01R на базі мікропроцесора Intel ATOM N270 та операційної системи Debian Linux. Особливістю даного мікрокомп'ютера стала наявність контактів GPIO, що забезпечують можливість підключення давачів стенду на шині I2C. Зовнішній вигляд пристрою зображено рис. 3.5, а характеристики вказані у таблиці 3.1.

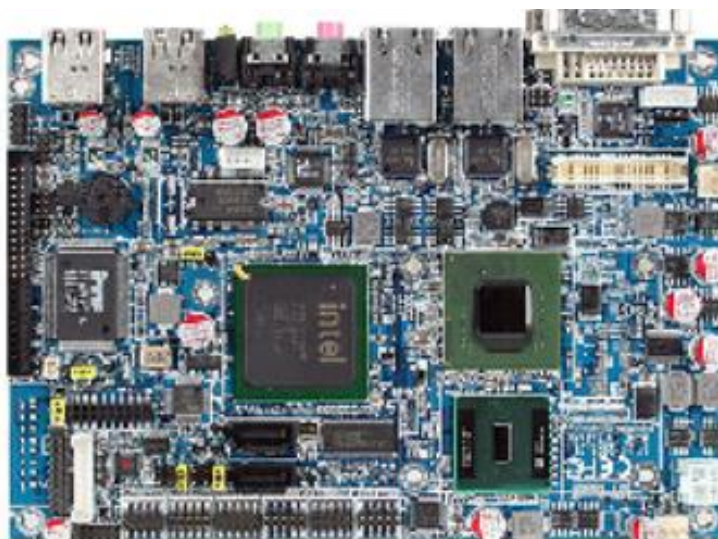


Рис. 3.5. Одноплатний комп'ютер EPI-945GSE-A1-01R

Таблиця 3.1

Основні технічні характеристики ЕОМ

CPU	Intel Atom N270 1.6GHz
BIOS	Award 4 Mbit Flash BIOS
GPU	Intel 945GSE
Інтерфейси	1 x EIDE, 2 x SATA, 1 x KB & Mouse (Optional) 5 x RS-232, 1 x RS-232/422/485 8-bit GPI, 8-bit GPO
Пам'ять (SDRAM, загальна)	One 200-pin SODIMM Socket Supports Up to 2GB DDR2 400/533 SDRAM
Максимальна робоча частота	1.6 ГГц
Порти USB 2.0	6
Відеовихід	DVI + LVDS
Мережа	Ethernet -порт RJ45 10/100 Мбіт/с
клас захисту	IP-52
Енергоспоживання	700 мА (3,5 Вт)
живлення	5 В через порт micro-USB або GPIO
Розміри	85,6x56x21 мм
Маса	54 г
Температурний режим	від -40° С до 85°

3.2.2. Давач освітленості

Як давач освітленості був обраний цифровий давач ВН1750. Перевага давача серед аналогів його категорії точності полягає в підтримці роботи з шиною І2С і переведенні показів в одиниці виміру Люкс.

Характеристики давача ВН1750 наведені в табл. 3.2.

Таблиця 3.2

Основні характеристики давача ВН1750

Характеристика	Показник
Напруга живлення	2,4 – 3,6 В
Струм споживання	120 мкА
Довжина хвилі	560 нм
Точність у режимі високої роздільної здатності	1 Лк
Точність у режимі низької роздільної здатності	4 Лк
Період вимірювання в режимі високої роздільної здатності	120 мс
Період вимірювання в режимі низької роздільної здатності	16 мс
Аналогово-цифровий перетворювач	16 біт

Зовнішній вигляд давача зображено рис. 3.6.

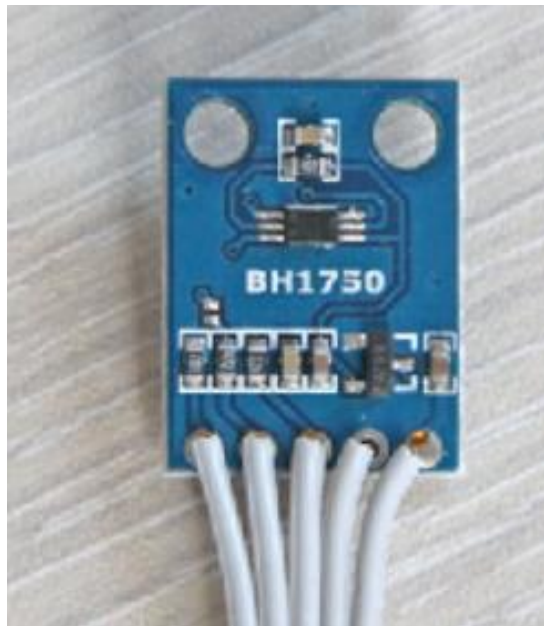


Рис. 3.6. Зовнішній вигляд давача освітленості BH1750 4.1.3

3.2.3. Давач частинок

Для відстеження стану атмосфери виробничих умов до складу стенду було включено дадаваччик пилу та частинок GP2Y1010AU0F від японського виробника електроніки Sharp [31]. Давач використовується в промисловому устаткуванні, роботах-пилососах, системах пожежної сигналізації та ін. Зовнішній вигляд давача зображений на рис. 3.6. Для передачі в ЕОМ використовується шина I2C. Технічні характеристики давача GP2Y1010AU0F:

- напруга живлення: $DC5 \pm 2$ В;
- струм споживання – 120 мкА;
- чутливість: 0.5 В/(0.1 мг/м³), напруга при чистому повітрі – 0.9 В;
- робоча температура: -10 ~ 65 °С;
- розмір: 46 мм x 30 мм x 17.6 мм



Рис. 3.7. Зовнішній вигляд давача пилу та частинок GP2Y1010AU0F 4.1.4

3.2.4. Давач відстані

Обрано цифровий давач AP3216. Модуль має дуже високі показники точності, температурну компенсацію, захист від перешкод та мерехтіння, може працювати за затемненим склом. Зовнішній вигляд модуля вимірювання відстані показано на рис. 3.8.

Характеристики давача AP3216:

- інтерфейс: I2C інтерфейс (до 400к ГЦ);
- вибір режиму: ALS, PS + IR;
- робоча напруга: 2,4 - 3.6VDC;
- робочий струм: 700µA макс.

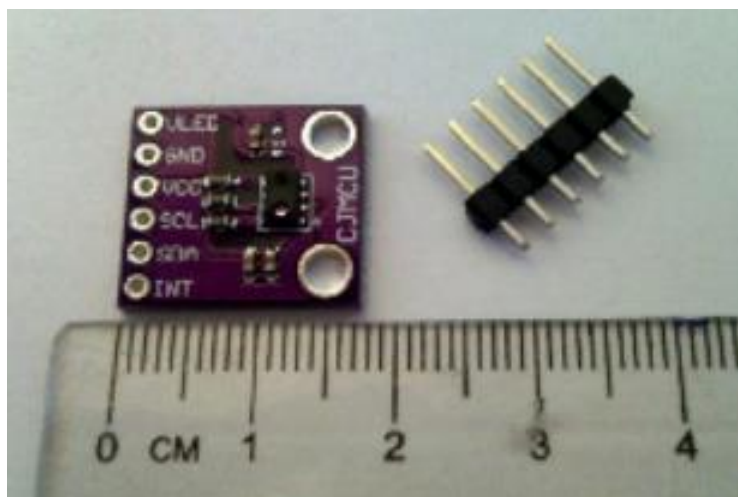


Рис. 3.8. Давач AP3216

3.2.5. IP -камера

Для отримання зображення та його подальшого аналізу до складу випробувального стенду входить IP -камера з об'єктивом фіксованого типу.

Протоколами транспортного рівня в таких пристроях можуть бути TCP, UDP та ін. Завдяки тому, що в IP- камерах немає необхідності передавати аналоговий сигнал у форматі PAL або NTSC, в них можуть застосовуватися великі роздільні здатності, в т.ч. мегапіксельні. Властиво типова роздільна здатність 640x480 пікселів. Проте є камери з мегапіксельними здатностями: 1280x1024, 1600x1200 та вищими.

Модель камери використовуваної у складі стенду є IP -камера ORIENT IP-36-720 . Зовнішній вигляд камери наведено на рис. 7.7.

Характеристики IP -камери:

- тип матриці: CCD;
- число пікселів матриці: 1 МП;
- фокусна відстань: 3.6 мм;
- максимальна роздільна здатність: 1280x720 пікселів;
- стандарт ступеня захисту: IP66.



Рис. 3.9. IP -камера ORIENT IP- 36-720

3.3 Опис програмного забезпечення стенду

Для вирішення поставленого завдання було використано такі програмні продукти.

Вона є високорівневою мовою, котра спрямована на покращення продуктивності програміста та читання коду. Варто зауважити, що синтаксис ядра Python є мінімальним. Властиво стандартна бібліотека містить велику множину спеціальних функцій, котрі досить просто використовувати у прикладних задачах.

Python має підтримку різних парадигм програмування, зокрема структурне, ООП функціональне та ін. Головними рисами архітектури є динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, технологія опрацювання винятків, використання багатопотокових обчислень, структури даних високого рівня. Дозволяється розбивати великий програмний код окремо на набір модулів, котрі потім можуть бути об'єднаними в пакети для подальшого їх використання. Ці властивості мови програмування полегшили завдання створення об'єктів алгоритмів ГП. Для розробки ПЗ використовувалася версія мови програмування Python 3.6.

Бібліотека OpenCV - бібліотека алгоритмів обробки та аналізу зображень, огляд алгоритмів бібліотеки, що використовуються, проведений у другому розділі даної роботи.

Бібліотека DEAP - бібліотека розподілених еволюційних алгоритмів у Python. Вона забезпечує основу роботи з ГА.

Розроблене ПЗ можна розбити на три частини: опис дерев рішень, операції над деревами рішень (створення та мутація) та генерація системи виявлення об'єкта. Фрагмент лістингу ПЗ представлений у додатку Б.

3.3.1. Класи дерев рішень

Для реалізації концепції дерева рішення розробили такі класи: FuncWrapper, FuncNode, SensorNode, ConstNode, ImageNode. Розглянемо кожен із них докладно.

FuncWrapper. Обгортковий клас для функцій. Кожен екземпляр цього класу має такі поля:

- function – функція, що лежить в основі класу;
- childCount - кількість аргументів, що функція приймає;
- childDataTypes - типи даних аргументів (0 - логічний, 1 - числовий, 2 – зображення);
- name – ім'я функції.

FuncNode. Функціональна ланка. Єдиний вид ланки, яка може мати нащадків. Кожен екземпляр цього класу має такі поля:

- fw – екземпляр обгорткового класу функцій;
- children - ланки-нащадки.

Крім того, цей клас, як і решта класів ланок, має метод evaluate. У випадку FuncNode, цей метод обчислює гілку дерева, рекурентно викликаючи методи , що визначають кожного нащадка. Цей метод приймає на вхід зображення, яке необхідно обробити, а також показання всіх давачів.

SensorNode. Ланка, що відповідає показанню будь-якого давача. Кожен

екземпляр цього класу має поле `index` – номер давача. При створенні екземпляра цього класу `index` ініціалізується вказаним значенням. Метод `evaluate` повертає показання давача з номером `index` (з масиву, переданого як другий аргумент методу).

`ConstNode`. Ланка, що відповідає логічній чи числовій константі. Кожен екземпляр цього класу має поле значення - значення константи. При створенні екземпляра цього класу `value` ініціалізується вказаним значенням. Метод `evaluate` повертає значення.

`ImageNode`. Ланка, що відповідає оброблюваному зображенню. Метод `evaluate` повертає зображення, що обробляється (перший аргумент методу).

3.3.2. Операції над деревами рішень

Для створення випадкового дерева використовується функція `createTree`. Як аргумент вона приймає максимальну глибину дерева і тип даних кореневої ланки (за замовчуванням - зображення). Крім того, для роботи цієї функції необхідно задати такі змінні:

- `createTree.nodeFactories` - список із трьох елементів, кожен із яких є списком фабрик ланок (об'єктів, що створюють ланки) відповідного типу даних (логічний, числовий, зображення);

- `createTree.nodeProbs` - список із трьох елементів, кожен з яких є списком ймовірностей вибору тієї чи іншої фабрики ланок;

- `createTree.leafFactories` і `createTree.leafProbs` - аналогічні змінні, але для випадку, коли максимальна глибина дерева досягнута (у цьому випадку не можна використовувати функціональні ланки).

Для створення ланок розроблено фабрики ланок. Усі вони мають метод `create()`, який створює відповідну ланку.

`FilterFactory`. Клас, що створює ланку `FuncNode`, що фільтрує зображення. Фільтр (а точніше екземпляр класу `FuncWrapper`, відповідний фільтру) випадково

вибирається зі списку `FilterFactory.filterWrapperList` з рівною ймовірністю. До складу цього списку входять:

- `gaussianBlurWrapper` - розмивання Гаусом;
- `medianBlurWrapper` – медіанний фільтр;
- `dilateWrapper` - дилатація;
- `erodeWrapper` - ерозія;
- `equalizeHistWrapper` - вирівнювання гістограми.

Метод `create` створює функціональну ланку без вказівки нащадків. Вважається, що нащадки будуть задані пізніше.

`FuncFactory`. Клас, що створює ланку `FuncNode` із зазначеною функцією. Під час створення цієї фабрики вказується екземпляр класу `FuncWrapper`. Метод `create` створює функціональну ланку без вказівки нащадків. Вважається, що нащадки будуть задані пізніше. У програмі цей клас використовується з наступними об'єктами класу `FuncWrapper`:

- `ifWrapper` - умовний вираз;
- `greaterThanWrapper` – функція порівняння «більше».

`SensorFactory`. Клас, що створює ланку `SensorNode`. Під час створення цієї фабрики вказується кількість давачів. Метод `create` створює ланку, яка відповідає давачу з випадковим номером.

`BoolConstFactory`. Клас, що створює ланку `ConstNode` із булевим значенням. Значення з рівною ймовірністю дорівнює 0 (фальш) або 1 (істина).

`NumConstFactory`. Клас, що створює ланку `ConstNode` із числовим значенням. Значення вибирається від 0 до 1 випадково.

`ImageFactory`. Клас, що створює ланку `ImageNode`.

Для мутації дерева використовується функція `mutatetree`. Як аргументи вона приймає дерево, мутацію якого необхідно зробити, можливість мутації ланки, максимальну глибину дерева і тип даних кореневої ланки (за замовчуванням - зображення).

3.3.3. Генерація системи виявлення об'єкта

Система виявлення об'єкта генерується відповідно до методики, описаної в розділі 2. Розглянемо необхідні для цього функції.

Distance - функція відстані між двома точками. Використовується для обчислення p_i у формулі (2.1).

distancePri - функція наведеної відстані між двома точками. Використовується для обчислення $p_{i\text{priv}}$ у формулі (2.3).

fitnessFcn ф ФФ. Обчислюється за такою формулою (2.2). На вхід приймає об'єкт, який повинен мати такі поля:

- cameraFilter - процедура Т 1 передобробки зображення з камери;
- templateFilter - процедура Т 2 передобробки шаблонного зображення;
- detectionParam - поріг c у алгоритмі виявлення об'єкта.

Крім того, для роботи цієї функції необхідно задати такі змінні:

- fitnessFcn.sample - навчальна вибірка; список елементів з наступними полями: cameraImage - зображення з камери; sensors - покази датчиків у момент взяття зображення; objectPresent - наявність або відсутність шуканого об'єкта; pos - положення шуканого об'єкта в пікселях (якщо відсутня, то одно None);

- fitnessFcn.templateImage - шаблонне зображення;
- fitnessFcn.detect - процедура виявлення об'єкта;
- fitnessFcn.distMax - параметр p_{max} у формулі (2.3);
- fitnessFcn.gamma - параметр γ_{in} у формулі (2.3);
- fitnessFcn.detectionErrorWeight - параметр w_{in} у формулі (2.3);
- fitnessFcn.falsePositivesWeight - параметр w_{xn} у формулі (2.3);
- fitnessFcn.falseNegativesWeight - параметр w_{xn} у формулі (2.3).

createIndividual. Функція, що створює випадкову особину.

Mate - функція, що виробляє нащадка двох батьків.

Mutate функція, що здійснює мутацію особини.

Main - функція для проведення експерименту. У ній необхідно вказати:

- popSize – розмір популяції;
- csProb - можливість схрещування;
- mutProb - ймовірність мутації;
- genCount – число поколінь.

3.4. Експериментальні дослідження

Для проведення випробувань розроблених алгоритмів потрібно скласти навчальну вибірку зображень об'єкта розпізнавання. Для створення навчальної вибірки використовуються засоби самого стенду. До навчальної вибірки увійшли знімки мітки орієнтації автомобільного диска та невелика область реборди навколо мітки, показання датчиків частинок, відстані та освітленості, записані у текстовий файл.

3.4.1. Методика проведення експерименту

Опис складеної вибірки:

- розмір: 200 зображень;
- кожен елемент вибірки складається з таких даних: зображення з камери, показання трьох датчиків у момент взяття зображення, наявність або відсутність мітки на зображенні, положення мітки в пікселях (якщо вона є);
 - наведені покази датчиків варіюються від 0 до 1:
 - для датчика відстані, що дорівнює 0, відповідає найменшому діаметру колеса (15 дюймів); значення 1 відповідає найбільшому діаметру (18 дюймів);
 - для датчика освітленості, що дорівнює 0, відповідає найнижчому освітленню у вибірці; значення 1 відповідає найвищому;
 - для датчика суспензії, що дорівнює 0, відповідає найчистішому повітрі у вибірці; значення 1 відповідає самому забрудненому.
- кількість зображень із міткою - 100, без мітки - 100; при (приблизно)

одних і тих же значення давача береться по два зображення одного і того ж колеса - з міткою і без мітки. Розподіл давачів у навчальній вибірці показано на рис. 3.10.

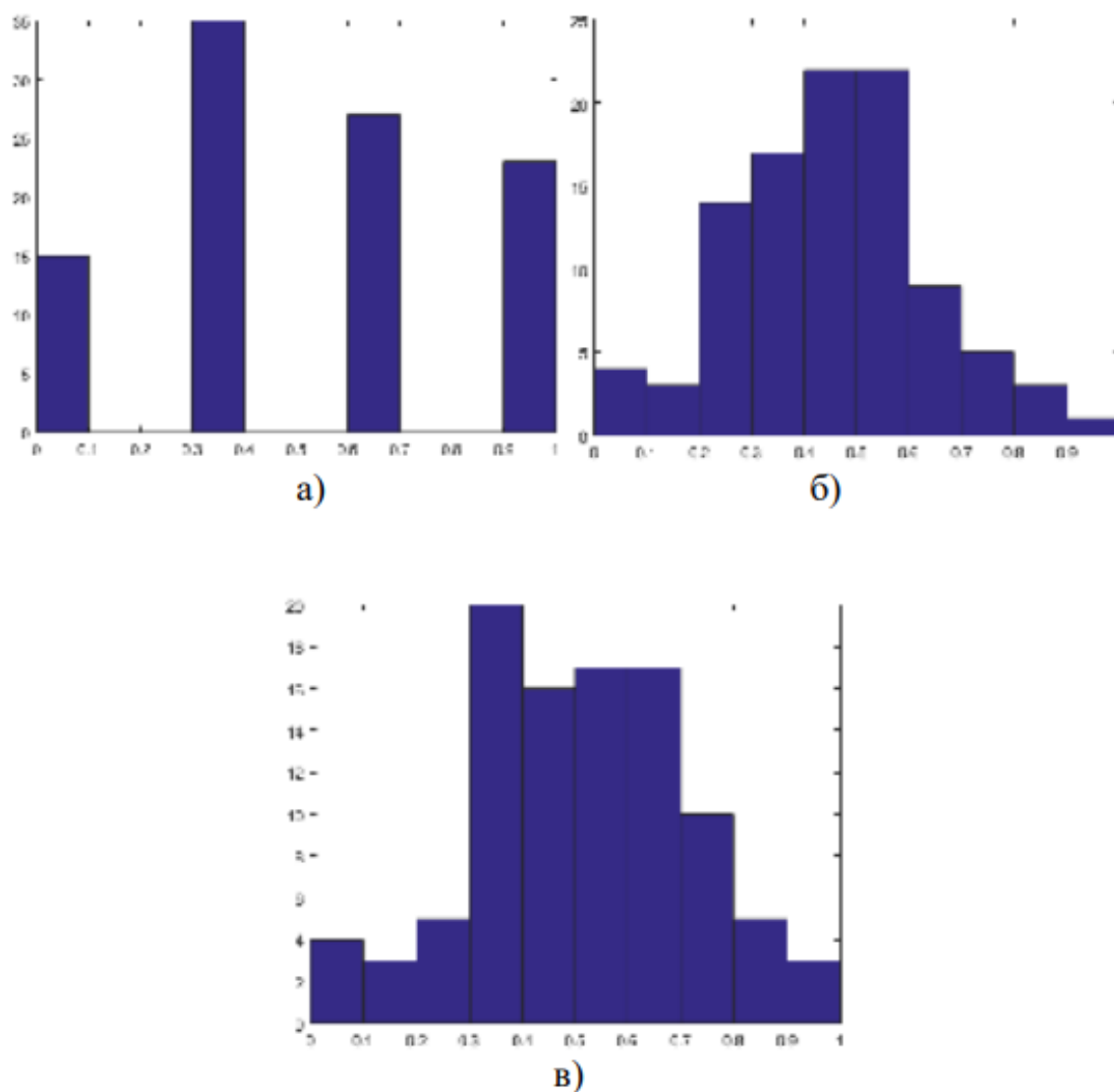


Рис. 3.10. Гістограми показів давачів

а) давач відстані; б) давач освітленості; в) давач частинок;

Матриця кореляції показів трьох давачів представлена в табл. 3.3.

Таблиця 3.3

Параметри експериментів

	Відстань	Освітленість	Частинки
Відстань	1.0000	0.0652	-0.0193
Освітленість	0.0652	1.0000	-0.6853
Частинки	-0.0193	-0.6853	1.0000

Вибірка ділиться на дві рівні частини по 100 елементів: навчальна та тестова вибірки. Перша використовується для генерації системи виявлення; друга – для аналізу якості роботи системи. При складанні кожної частини було взято по 50 зображень з мітками на колесах і по 50 зображень без мітки на 50 колесах.

3.4.2. Проведення експерименту та результати аналізу даних

З метою дослідження впливу вибору алгоритму виявлення об'єкта на зображенні та параметрів ФФ на отриману систему, було проведено 4 експерименти. Параметри, за яких були проведені експерименти, представлені в таблиці 3.4.

Таблиця 3.4

Параметри експериментів

№	Алгоритм виявлення	N_{max} глибини	ρ_{max}	γ_{in}	w_{in}	w_{xn}	w_{xn}
1	Гомографія точок	4	30	0.8	0.33	0.33	0.33
2		4			0.3	0.55	0.25
3		6					
4		2			0.33	0.33	0.33
5	Зіставлення шаблону	4					

Параметри ГА, що використовуються в експериментах, наведені в таблиці 3.5.

Таблиця 3.5

Параметри ГА

Параметр	Значення
Розмір покоління	30
Число поколінь	50
Генерація першого покоління	Алгоритм «створити дерево» запускається двічі для кожної особи: створення дерев T_1 і T_2 ; потім випадково генерується параметр c алгоритму виявлення об'єкта з діапазону $[0,1]$
Селекція	Виживають 50% найбільш пристосованих особин
Вибір батьків	Обидва батьки вибираються випадково, кожна особина популяції має рівні шанси бути обраною
Схрещування	З рівною ймовірністю витягуються один за одним відповідні елементи генотипів батьків А та В: $S_A = \{T_{A1}, T_{A2}, c_A\}$ та $S_B = \{T_{B1}, T_{B2}, c_B\}$.
Імовірність мутації	10%
Мутація	Для дерев T_1 та T_2 виконується алгоритм «Мутація_дерева»; до параметра додається випадкове число з нормальним розподілом $N(0, 0.2)$. Якщо результат виявиться меншим за 0, то c замінюється значенням 0; якщо результат перевищить 1, то c замінюється значенням 1.

Результати першого експерименту у вигляді найкращих особин показані на рис. 3.11 та 3.12.

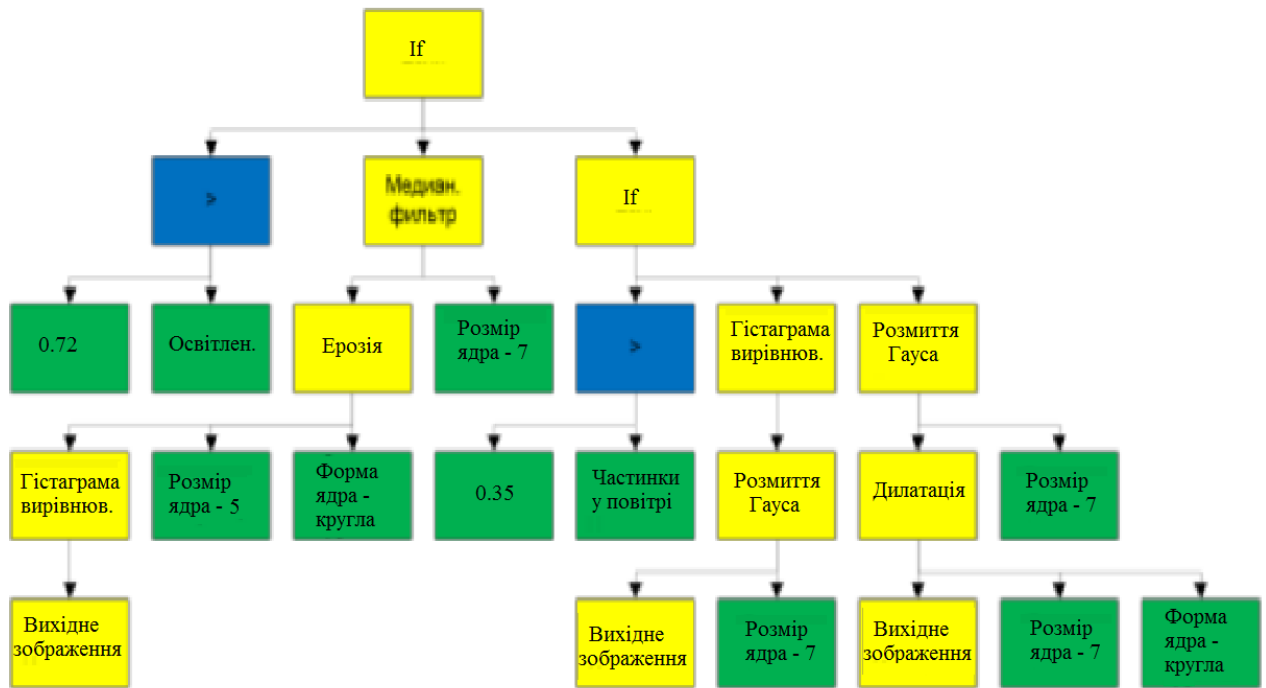


Рис. 3.11. Дерево T_1 для обробки зображення з камери

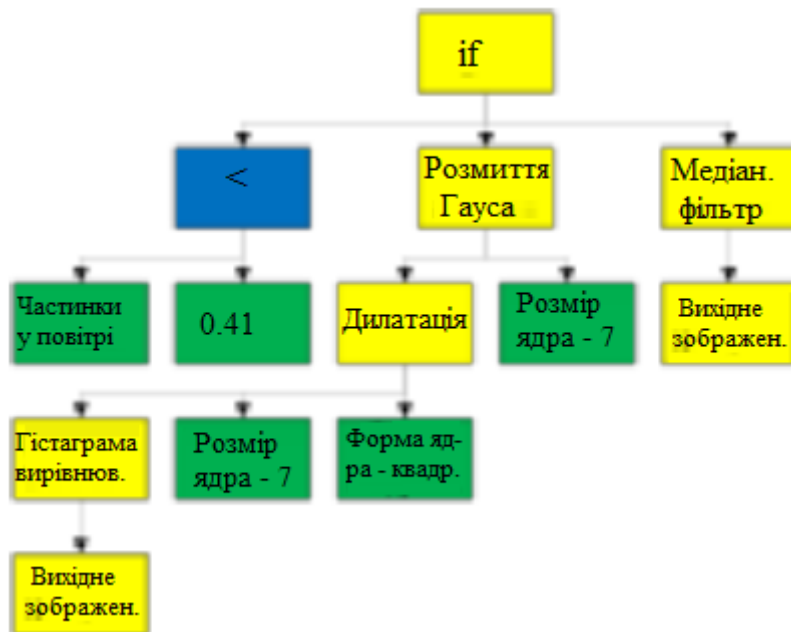


Рис. 3.12. Дерево T_2 попередньої обробки шаблонного зображення

Якість роботи системи оцінюється за трьома ознаками: відсоток хибно-позитивних випадків, відсоток хибно-негативних випадків і середня наведена похибка виявлення в істинно-позитивних випадках. Ці дані наведено у таблиці 3.6.

Якість роботи систем виявлення

№	Навчальна вибірка			Тестова вибірка			с	N _{зліб}
	% хибно-позитив.	% хибно-негатив.	Середн. привед. похибка	% хибно-позитив.	% хибно-негатив.	Середн. привед. похибка.		
1	0	1	0.4304	4	3	0.6352	0.8293	4
2	1	3	0.5032	0	7	0.6820	0.6934	4
3	2	1	0.3995	4	4	0.6104	0.8511	4
4	5	6	0.6413	11	13	0.7231	0.7533	2
5	3	2	0.5109	6	8	0.6689	0.4490	4

3.6. Висновки до розділу

Аналіз результатів показав, що в першому експерименті в навчальній вибірці помилково-негативні помилки детектування склали 1%. У другому експерименті на тестовій вибірці показано, що рахунок призначення різних ваг вдалося позбутися хибно-позитивних помилок, але відсоток хибно-негативних помилок збільшився. Третій експеримент показав, що збільшення максимальної глибини до шести рівнів дерева не покращило результату, однак, з четвертого експерименту видно, що максимальна глибина, котра дорівнює двом рівням є явно недостатньою. У п'ятому експерименті було повторно введено параметри першого експерименту, але разом з іншим алгоритмом детектування. З результатів експерименту випливає, що метод порівняння гомографії точок краще справляється із завданням. Варто також відзначити, що результати роботи системи на тестовій вибірці гірші, ніж на навчальній, що цілком очікувано.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Метою кваліфікаційної роботи магістра є підвищення ефективності контролю самоадаптивного керування вбудованими системам. Оскільки, проведення робіт з розробки та використання розробки передбачає використання комп'ютерної техніки, зокрема ПК та периферійних пристроїв, то обов'язковим є дотримання вимог з охорони праці і техніки безпеки.

Для ефективної і безпечної роботи колективу працівників з розробки ПЗ комп'ютерних систем, в тому числі і фахівців з підвищення ефективності контролю доступу в приміщення, необхідно організувати безпечні умови праці. При цьому керівник організації несе безпосередню відповідальність за порушення нормативно-правових актів з охорони праці [33]. Окрім цього, на робочих місцях працівників необхідно забезпечити дотримання вимог, затверджених Наказом Мінсоцполітики від 14.02.2018 за № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Згідно Вимог приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (сервер), мають бути оснащені системою автоматичної пожежної сигналізації відповідно до цих вимог;

– переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації, затвердженого наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи від 22.08.2005 N 161, зареєстрованого в Міністерстві юстиції України 05.09.2005 за N 990/11270 (НАПБ Б.06.004-2005);

– Державних будівельних норм "Інженерне обладнання будинків і споруд. Пожежна автоматика будинків і споруд", затверджених наказом Держбуду України від 28.10.98 N 247 (далі - ДБН В.2.5-56:2014, з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками.

В інших приміщеннях допускається встановлювати теплові пожежні сповіщувачі. Приміщення, де розміщені робочі місця операторів, мають бути оснащені вогнегасниками, кількість яких визначається згідно з вимогами ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників». Загальні технічні вимоги і з урахуванням граничнодопустимих концентрацій вогнегасної рідини відповідно до вимог НАПБ А.01.001-2014. Приміщення, в яких розміщуються робочі місця операторів сервера загального призначення, обладнуються системою автоматичної пожежної сигналізації та засобами пожежогасіння відповідно до вимог ДБН В.2.5-56:2014, ДБН В.2.5-56:2010, НАПБ А.01.001-2014 і вимог нормативно-технічної та експлуатаційної документації виробника. Проходи до засобів пожежогасіння мають бути вільними.

Лінія електромережі для живлення комп'ютера та периферійних пристроїв повинні бути виконаними як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення. Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники.

Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі має бути не менше площі перерізу фазового провідника. Усі провідники мають відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників,

температурному режиму та типам апаратури захисту, вимогам НПАОП 40.1-1.01-97.

У приміщенні, де одночасно експлуатуються понад п'ять комп'ютерів, на помітному, доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення. Комп'ютери повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення.

У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Не допускається підключати комп'ютери до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв. Електромережі штепсельних з'єднань та електророзеток для живлення комп'ютерної техніки повинні бути виконаними за магістральною схемою, по 3-6 з'єднань або електророзеток в одному колі. Штепсельні з'єднання та електророзетки для напруги 12 В та 42 В за своєю конструкцією мають відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В. Штепсельні з'єднання та електророзетки, розраховані на напругу 12 В та 42 В, мають візуально (за кольором) відрізнятися від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

При підвищенні ефективності контролю доступу в приміщення, де для забезпечення безпеки мешканців, співробітників і збереження майна використовуються ДС, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18. Організація робочого місця фахівця із дослідження методів та програмно-апаратних засобів оптимізаційних процесів на основі ГА повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування

ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги». Відстань від екрана до ока фахівців, які працюють за комп'ютером визначається згідно з вимогами ДСанПіН 3.3.2.007-98.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість екрана комп'ютера, зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля згідно з вимогами ДСанПіН 3.3.2.007-98.

Таким чином, у результаті аналізу вимог щодо охорони праці користувачів комп'ютерів, визначено особливості організації робочих місць, вимог з електробезпеки, природного та штучного освітлення для ефективної і безпечної роботи фахівців з дослідження апаратно-програмного забезпечення самоадаптивного ВС.

4.2. Ергономічні вимоги до робочого місця користувача ПК

Робоче місце — це зона простору, що оснащена необхідним устаткуванням, де відбувається трудова діяльність одного працівника чи групи працівників [33].

Раціональне планування робочого місця має забезпечувати: найкраще розміщення знарядь і предметів праці, не допускати загального дискомфорту, зменшувати втомлюваність працівника, підвищувати його продуктивність праці. Площа робочого місця має бути такою, щоб працівник не робив зайвих рухів і не відчував незручності під час виконання роботи. Важливо мати також можливість змінити робочу позу, тобто положення корпусу, рук, ніг. Проте доцільно виключати або мінімізувати всі фізіологічно неприродні і незручні положення тіла. Проведені дослідження показують, що при раціональній організації робочих місць продуктивність праці зростає на 15–25% [34]. Організація робочого місця користувача ПК має відповідати ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні

ергономічні вимоги», ДСан ПіН 3.3.2.007-98, характеру та особливостям трудової діяльності.

Площа одного робочого місця користувача ПК повинна складати не менше 6 м², а об'єм – не менше 20 м³. Конструкція робочого місця користувача ПК повинна відповідати сучасним вимогам ергономіки, характеру виконуваної роботи і забезпечити оптимальне розміщення на робочій поверхні документів та обладнання ПК (монітора, системного блоку, клавіатури, мишки та інших периферійних пристроїв. Монітор на робочому місці встановлюється так, щоб верхній край екрана знаходився на рівні очей.

Розташування монітора ПК має забезпечувати: безпечність роботи в цілому; зручність та ефективність зорової роботи з екраном в вертикальній площині під кутом 30° від лінії зору, площина екрана при цьому має бути перпендикулярною нормальній лінії зору користувача.

Клавіатура розміщується на поверхні столу або висувній полиці на відстані 100-300мм від краю, ближчого до користувача. Кут нахилу клавіатури має бути в межах 5-15°. Поверхня клавіатури повинна бути матовою з коефіцієнтом відбиття 0,4. Клавіші клавіатури мають бути зручними в роботі і м'якими при натисканні (хід всіх клавіш має бути однаковим з мінімальним опором натискання 0,25Н та максимальним – не більше 1,5Н) [35].

При розміщенні робочих місць з ПК слід дотримуватися вимог, зазначених в ДНАОП 0.00-1.31-99: робочі місця розміщуються на відстані не менше 1м від стін з світловими прорізами; відстань між бічними поверхнями моніторів ПК має бути не менше 1,2м; відстань між тильною поверхнею монітора одного ПК та екраном монітора іншого ПК має бути не меншою 2,5м.

Вимоги двох останніх пунктів враховуються також при розміщенні робочих місць з ПК в суміжних приміщеннях з урахуванням конструктивних особливостей стін та перегородок.

Загальні принципи організації робочого місця:

- на робочому місці не повинно бути нічого зайвого. Усі необхідні для роботи предмети мають бути поряд із працівником, але не заважати йому;
- ті предмети, якими користуються частіше, розташовуються ближче, ніж ті предмети, якими користуються рідше;
- предмети, які беруть лівою рукою, повинні бути зліва, а ті предмети, які беруть правою рукою – справа;
- якщо використовують обидві руки, то місце розташування пристосувань вибирається з урахуванням зручності захоплення його двома руками;
- робоче місце не повинно бути зашаржене;
- організація робочого місця повинна забезпечувати необхідну оглядовість.

Статичні напруження працівника в процесі праці пов'язані з підтриманням у нерухомому стані предметів і знарядь праці, а також підтриманням робочої пози.

Робоча поза – це основне положення працівника у просторі: зручна робоча поза має забезпечувати стійкість положення корпусу, ніг, рук, голови працівника під час роботи, мінімальні затрати енергії та максимальну результативність праці. Неправильна сидяча поза може викликати застій крові в ногах, а якщо виконується великий обсяг роботи для пальців рук – запалення суглобів.

Організація робочого місця користувача комп'ютера повинна забезпечувати відповідність усіх елементів робочого місця та їх взаємного розташування ергономічним вимогам (рисунок 4.1).

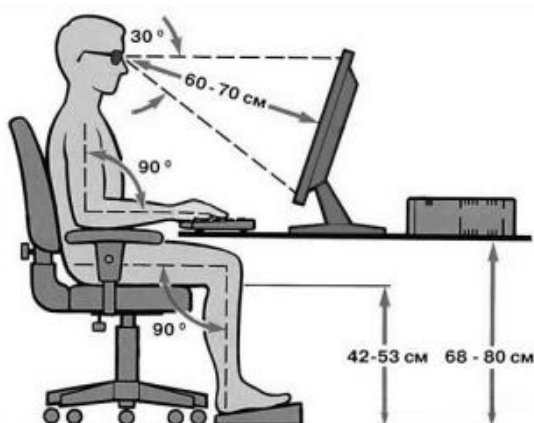


Рисунок 4.1 - Робоче місце і робоча поза користувача ПК

Найпоширенішими у процесі праці є пози сидячи і стоячи. Проектуючи робоче місце, потрібно враховувати, що при виконанні роботи з фізичним навантаженням бажана поза стоячи, а при малих зусиллях – сидячи.

Робоча поза стоячи втомлює людину більше, ніж сидяча. Вона вимагає на 10% більше енергії, спричиняє підвищення артеріального і венозного тиску крові, розширення вен на ногах, пошкодження ступень, викривлення хребта [35].

4.3. Висновки до розділу

В цьому розділі проаналізовано важливі питання охорони праці та безпеки в надзвичайних ситуаціях, висвітлено питання ергономічних вимог до робочого місця користувача ПК.

ВИСНОВКИ

В кваліфікаційній роботі досліджено методи та засоби самоадаптивного управління ВС, а також розроблено метод та ПЗ на основі еволюційних алгоритмів та ГП.

Основні результати, отримані в роботі:

- проведено огляд та аналіз існуючих методів розробки інтелектуальних систем для ПЗ ВС;
- на підставі проведеного аналізу було визначено основні завдання та шляхи їх вирішення;
- розроблено схему роботи ВС, для якої було обрано відповідне обладнання та розроблено алгоритм функціонування;
- розроблена система може бути була впроваджена в експлуатацію на діючому підприємстві під час виробництва автомобільних дисків на автоматизованій лінії механообробки.

Результати дисертаційної роботи будуть матеріалом для подальших досліджень застосування еволюційних методів для розробки самоадаптивних інтелектуальних систем управління ВС.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Програмування багатопроцесорних обчислювальних систем. URL: <http://https://uadoc.zavantag.com/text/18820/index-1.html/> (дата звертання: 30.11.2022).
2. Обчислювальні машини та важкорозв'язувані задачі. URL: http://https://elearning.sumdu.edu.ua/free_content/lectured:075b2e8a0bfe48bcef0ab3106c6d51679abc41f9/latest/117573/index.html/ (дата звертання: 30.11.2022).
3. Vahid F. Hardware/Software Codesign // *Proceedings of International Workshop*, Colorado : Estes Park, 2002. 2002. Pp. 1-6.
4. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій): навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська. Київ : КПІ ім. Ігоря Сікорського, 2020. 192с.
5. Обробка та аналіз цифрових зображень на LabVIEW IMAQ Vision . URL: <http://www.csun.edu/~rd436460/Labview/IMAQ-Manual.pdf> (дата звертання: 07.12.2022).
6. Кобилін О.А., Творошенко І.С. Методи цифрової обробки зображень: навч. посібник. Харків: ХНУРЕ, 2021. 124 с.
7. AForge.NET Framework. URL: <http://code.google.com/p/aforge/> (дата звертання: 29.11.2022).
8. VXL Framework. URL: <http://code.google.com/p/vxl/> (дата звертання: 29.11.2022).
9. Bradsky G., Kaebler A. Learning OpenCV. Computer Vision with the OpenCV Library. New York : O'RELLY, 2008. 253 с.
10. Евдокимов, Ю. К. Система технического зрения для геометрических // Вестник КГТУ. 2006. Выпуск 4(44). С. 21 – 23.
11. Davies E. R. Machine Vision: Theory, Algorithms, Practicalities : Academic Press. – San Diego : Elsevier, 1997. – 750 с.

12. Творошенко І.С. Конспект лекцій з дисципліни «Цифрова обробка зображень». Харків: ХНУМГ ім. О.М. Бекетова, 2015. 75 с.
13. Архітектурна модель цифрових обчислювальних систем. URL: http://www.informatuka.info/lesson_3.php (дата звертання: 29.11.2022).
14. Обчислювальні платформи інформаційних управляючих систем. URL: http://cloud-5.bitp.kiev.ua/?page_id=1637 (дата звертання: 29.11.2022).
15. Микитишин А.Г., Митник М.М., Стухляк П.Д., Пасічник В.В.. Комп'ютерні мережі. Львів, «Магнолія 2006». 256 с.
16. Blume H., Amer A. Parallel Predictive Motion Estimation using Object Recognition Methods // Proceedings of the European Workshop and Exhibition on Image Format Conversion and Transcoding. vol. 9. no. 2. PP. 813- 631.
17. Субботін С.О. Нейронні мережі : теорія та практика: навч. посіб. Житомир : Вид. О. О. Євенок, 2020. 184 с
18. Mari M. Interactive object detection using a fuzzy image segmentation approach. // Machine GRAPHICS & VISION 1998. vol. 7. no. 4. PP. 765-780.
19. Bishop C. M. Neural Networks and Pattern Recognition. Oxford : Press, 1995. 168 p.
20. Fogel L. J., Owens J. Artificial Intelligence throw Simulated Evolution. New York : Wiley, 1966. 265 p.
21. Mandavilli S., Patnaik L. Adaptation in genetic algorithms // In: *Genetic algorithms for pattern recognition*. CRC Press. 1996. P.45 – 64.
22. Скобцов Ю. А. Основы эволюционных вычислений. Донецк: ДонНТУ, 2008. 326с.
23. Eiben A. Adaptation in evolutionary computation: a survey // *Proceedings of IEEE international conference on evolutionary computation.*, New York : Piscataway, 1997. P.65 – 69.
24. Шамуратов О. Ю., Шаховська Н.Б. Алгоритми контурного аналізу зображень// Науковий вісник НЛТУ України. 2019. Т. 29, № 6. С. 123-127

25. Seeger U. R. Fast corner detection in grey-level images // *Pattern Recogn. Lett.*, 1994, №15(7), pp.669 – 675.
26. Максимов Н. А. Алгоритмы цифровой обработки изображений : Учеб. Москва : Изд-во МАИ, 1995, 31 с.
27. Lee C. H. Fast Motion Estimation Algorithm Based on the Block Sum Pyramid // *IEEE Trans. Image Processing*, vol. 6, pp. 1587 – 1591, Nov. 1997.
28. Николаев, М. И. Аппаратная реализация системы технического зрения для измерения геометрии микрообъектов // *Электронное приборостроение. Научно-практический сборник*. Выпуск 3(44). Казань : ЗАО «Новое знание», 2005. С. 19 – 29.
29. Koza, John R. Genetic Programming: *On Programming Computer by Means of Natural Selection and Genetics*. Cambridge, MA: The MIT Press, 1992.
30. Goldberg, D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, Massachusetts: Addison-Wesley, 1989.
31. Compact Optical Dust Sensor // SparkFun. URL: https://www.sparkfun.com/datasheets/Sensors/gp2y1010au_e.pdf (дата звертання: 12.12.2022).
32. Головатий І.Б. Обробка зображень із використанням генетичного алгоритму. *Інформаційні моделі, системи та технології: Праці X наук.-техн. конф.* (Тернопіль, 07-08 грудня 2022 р.), Тернопіль, 2022. – С. 73.
33. Толок А.О. Крюковська О.А. Безпека життєдіяльності: Навч. посібник. 2011. 215 с.
34. Яремко З. М. Безпека життєдіяльності: Навч. посіб. Львів., 2005.301 с.
35. Желібо Є. П. Заверуха Н.М., Зацарний В.В. Безпека життєдіяльності. Навчальний посібник. К.; Каравела, 2004. 328 с.

ДОДАТОК А

Тези конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

X НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



7-8 грудня 2022 року

ТЕРНОПІЛЬ
2022

А. Луцька, І. Барна АНАЛІЗ СЕРВІСНО-ОРІЄНТОВАНОЇ АРХІТЕКТУРИ В ПРОЦЕСІ ЗАСТОСУВАННЯ DEVOPS ПРАКТИК	
A. Lutskiv, I. Barna ANALYSIS OF SERVICE-ORIENTED ARCHITECTURE IN THE PROCESS OF APPLICATION OF DEVOPS PRACTICES	66
А. Луцька, М. Бондаренко ОСОБЛИВОСТІ ОПТИМІЗАЦІЇ СИСТЕМ ПІДТРИМКИ КОРИСТУВАЧІВ ІЗ ЗАСТОСУВАННЯМ ПІДХОДУ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ	
A. Lutskiv, M. Bondarenko FEATURES OF USER SUPPORT SYSTEMS OPTIMIZATION USING THE APPROACH OF MASS SERVICE SYSTEMS	67
А. Луцька, М. Бондаренко АРХІТЕКТУРА СИСТЕМИ ПІДТРИМКИ З ОПТИМІЗАЦІЄЮ ПРОЦЕСУ УПРАВЛІННЯ ЗВЕРНЕННЯМИ КОРИСТУВАЧІВ	
A. Lutskiv, M. Bondarenko SUPPORT SYSTEM ARCHITECTURE WITH OPTIMIZATION OF THE USER COMPLAINT MANAGEMENT PROCESS	68
В. Яцишин, Т. Кобець ТЕХНОЛОГІЯ MESH В КОМП'ЮТЕРНИХ СИСТЕМАХ ПЕРЕДАЧІ ДАНИХ	
V. Yatsyshyn, T. Kobets TECHNOLOGIES OF NON-INVASIVE GLUCOSE LEVEL MEASUREMENT IN BLOOD	69
В. Яцишин, Т. Кобець МЕТОДИ ВИБОРУ ОПТИМАЛЬНИХ КОМПОНЕНТІВ КОМП'ЮТЕРНИХ СИСТЕМ НА ОСНОВІ АНАЛІЗУ ІЄРАРХІЇ	
V. Yatsyshyn, T. Kobets METHODS OF SELECTING OPTIMUM COMPUTER SYSTEM COMPONENTS BASED ON HIERARCHY ANALYSIS	70
А. Луцька, М. Тимощук ВАЖЛИВІСТЬ ДОКУМЕНТУВАННЯ В ПРОЦЕСІ ВДОСКОНАЛЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ	
A. Lutskiv, M. Tymoshchuk THE IMPORTANCE OF THE DOCUMENTATION IN THE PROCESS OF IMPROVING COMPUTER SYSTEMS	71
А. Луцька, М. Тимощук МОДЕЛІ ПРЕДСТАВЛЕННЯ ДОКУМЕНТАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРИ ВДОСКОНАЛЕННІ КОМП'ЮТЕРНИХ СИСТЕМ	
A. Lutskiv, M. Tymoshchuk MODELS OF SOFTWARE DOCUMENTATION VIEW IN THE IMPROVEMENT OF COMPUTER SYSTEMS	72
І. Головатий ОБРОБКА ЗОБРАЖЕНЬ ІЗ ВИКОРИСТАННЯМ ГЕНЕТИЧНОГО АЛГОРИТМУ	
I. Holovatyi IMAGE PROCESSING USING GENETIC ALGORITHM	73
Ю. Гук, А. Паламар МЕТОД АДАПТИВНОГО РЕГУЛЮВАННЯ ДОРОЖНЬОГО РУХУ НА ПЕРЕХРЕСТІ НА ОСНОВІ ІНТЕРНЕТУ РЕЧЕЙ	
Y. Huk, A. Palamar METHOD OF ADAPTIVE TRAFFIC CONTROL AT AN INTERSECTION BASED ON INTERNET OF THINGS	74

УДК 004.89

І. Головатий

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ОБРОБКА ЗОБРАЖЕНЬ ІЗ ВИКОРИСТАННЯМ ГЕНЕТИЧНОГО АЛГОРИТМУ

UDC 004.89

I. Holovatyı

IMAGE PROCESSING USING GENETIC ALGORITHM

Принцип роботи генетичного алгоритму (ГА) полягає у створенні набору випадкових рішень, що називається населенням. На кожному етапі оптимізації цільова функція обчислюється для всієї популяції, у результаті виходить ранжований перелік рішень. Після ранжування рішень формується нова популяція, котра носить назву наступного покоління. Спочатку до нової популяції включаються найкращі рішення з поточної. Цей процес зветься елітизм [1].

Крім них, до наступної популяції входять абсолютно нові рішення, що отримуються шляхом модифікації найкращих за допомогою генетичних операцій, таких як схрещування та мутація. Конкретні реалізації способів уявлення особин і генетичних операцій породжують різні різновиди еволюційного програмування. Потім цей процес повторюється – нова популяція ранжується та створюється чергове покоління.

Так триває задане число разів або до тих пір, поки протягом кількох поколінь не спостерігається жодних покращень. Принцип методу показаний на рисунку 1.

Для аналізованої задачі генерації системи виявлення об'єкта на зображенні, як генотип був обраний вектор $S = (T_1, T_2, c)$. Тут T_1 – дерево рішень, відповідне до обробки зображення з камери, T_2 – дерево рішень, відповідне до обробки шаблонного зображення, c – параметр алгоритму виявлення об'єкта.

Важкається, що в результаті виконання ГА буде отримана система, здатна максимально точно встановлювати положення об'єкта, що цікавить, на зображенні або вивалити факт його відсутності.

Для перевірки якості роботи системи використовується вибірка зображень, елементи якої складаються з:

- зображення, отриманого з камери;
- значень датчиків зовнішнього середовища у момент отримання цього зображення;
- координат центральної точки об'єкта, що цікавить, на зображенні або індикатор факту його відсутності.

Система виявлення застосовується до кожного елемента вибірки.

Література

1. Goldberg, D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, Massachusetts: Addison-Wesley, 1989

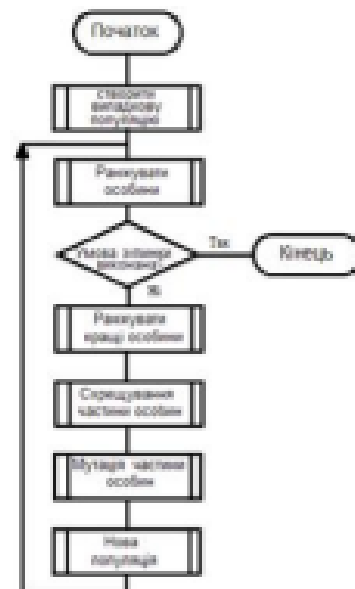


Рисунок 1. Блок-схема ГА

ДОДАТОК Б

Фрагмент лістингу програми генетичних операцій

```
import cv2
import numpy as np
import random
from copy import deepcopy
from deap import base, creator
from deap import tools

class FuncWrapper:
def init (self, function, childDataTypes, name):
self.function = функція
self.childCount =
len(childDataTypes)
self.childDataTypes =
childDataTypes self.name = name

class FuncNode:
def init (self, fw, children = None):
self.fw = fw
self.children = children
def evaluate(self, image, sensors):
results = [n.evaluate(image, sensors) for n in
self.children] return
self.fw.function(results)

class SensorNode:
def init (self, index):
self.index = index
def evaluate(self, image, sensors):
return sensors[self.index]

class ConstNode:
def init (self, value):
self.value = value
def evaluate(self, image, sensors):
return self.value

class ImageNode:
def evaluate(self, image, sensors):
return image

def createTree(maxDepth,
atatype = 2): if maxDepth ==
0:factories =
createTree.leafFactories[atatype]
probs =
createTree.leafProbs[atatype]
else:
```

```

factories = createTree.nodeFactories[atatype] probs =
createTree.nodeProbs[atatype] factory = np.random.choice(factories,
p = probs) node = factory.create() try:
children = [None] * node.fw.childCount for I in
range(node.fw.childCount):
children[i] = createTree(maxDepth - 1, node.fw.childDataTypes[i])
node.children = children except AttributeError: pass
return node

def mutateTree(tree, mutationProb, maxDepth, atatype = 2): if
np.random.random() < mutationProb: return createTree(maxDepth,
atatype) try:
for I in range(tree.fw.childCount):
mutatedChildren[i] = mutateTree(tree.fw.children[i], mutationProb,
maxDepth - 1,
tree.fw.childDataTypes[i])
return FuncNode(tree.fw, mutatedChildren)
except AttributeError:
pass
return deepcopy(tree)

ifWrapper = FuncWrapper(lambda args: args[1] if args[0] else
args[2], [0, 2, 2], 'if') greaterThanWrapper = FuncWrapper(lambda
args: args[0] > args[1], [1, 1], 'isGreater')
gaussianBlurWrapper = FuncWrapper(lambda args:
cv2.GaussianBlur(args[0], (20 * args[1], 20 * args[1]), 0), [2, 1],
'medianBlur')
medianBlurWrapper = FuncWrapper(lambda args: cv2.medianBlur(args[0],
20 * args[1]), [2, 1], 'medianBlur')
dilateWrapper = FuncWrapper(lambda args: cv2.dilate(args[0], 20 *
args[1], iterations = 1), [2, 1], 'dilate')
erodeWrapper = FuncWrapper(lambda args: cv2.erode(args[0], 20 *
args[1], iterations = 1), [2, 1], 'erode')
equalizeHistWrapper = FuncWrapper(lambda args:
cv2.equalizeHist(args[0]), [2], 'equalizeHist')

class FilterFactory: def create(self): return
FuncNode(np.random.choice(FilterFactory.filterWrapperList))

class FuncFactory:
def init (self, fw):
self.fw = fw
def create(self):
return FuncNode(self.fw)

class SensorFactory:
def init (self, sensorCount):
self.sensorCount = sensorCount def create(self):
return SensorNode(np.random.randint(0, self.sensorCount))

.....

```