

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ, ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ЕЛЕКТРОНІКИ, РОБОТОТЕХНІКИ І ТЕХНОЛОГІЙ
МОНІТОРИНГУ ТА ІНТЕРНЕТУ РЕЧЕЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Шутко В.М.
« ____ » _____ 2020 р.

ДИПЛОМНА РОБОТА

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА
ЗІ СПЕЦІАЛЬНОСТІ 171 «ЕЛЕКТРОНІКА»
ОСВІТНЬО-ПРОФЕСІЙНОЇ ПРОГРАМИ «ЕЛЕКТРОННІ СИСТЕМИ»

Тема: «Пристрій для відеоспостереження із розпізнаванням облич за допомогою нейронних мереж»

Виконавець
студент групи ЕС-207М _____ Миронов Денис Олександрович

Керівник
професор, д.т.н. _____ Шутко Володимир Миколайович

Консультант розділу «Охорона праці»
старший викладач _____ Козлітін О.О.

Консультант розділу
«Охорона навколишнього середовища»
професор, д.т.н. _____ Маджд С.М.

Нормоконтролер
доцент, к.т.н. _____ Сініцин Р.Б.

КИЇВ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ, ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ

КАФЕДРА ЕЛЕКТРОНІКИ, РОБОТОТЕХНІКИ І ТЕХНОЛОГІЙ

МОНІТОРИНГУ ТА ІНТЕРНЕТУ РЕЧЕЙ

Спеціальність 171 «Електроніка»

Освітньо-професійної програми «Електронні системи»

Освітній ступінь «МАГІСТР»

ЗАТВЕРДЖУЮ

Завідуючий випускової кафедри

_____ Шутко В.М.

«___» _____ 2020 р.

ЗАВДАННЯ

на виконання дипломної роботи

Миронова Дениса Олександровича

- 1. Тема дипломної роботи :** «Пристрій для відеоспостереження із розпізнаванням обличчя за допомогою нейронних мереж»
затверджена наказом ректора від «02» жовтня 2020 р. № 1900/ст
- 2. Термін виконання роботи :** з 5 жовтня 2020 року по 27 грудня 2020 року.
- 3. Вихідні дані до роботи :** пристрій для розпізнавання обличчя.
- 4. Зміст пояснювальної записки:** реферат, список скорочень, вступ, 8 розділів, висновки, список використаних джерел.
- 5. Перелік обов'язкового ілюстративного матеріалу:** графічне представлення роботи пристрою.

6. Календарний план-графік

№	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1.	Пошук та обробка літератури за темою дипломної роботи	10.10.20 - 20.10.20	Виконано
2.	Огляд науково – технічної літератури за темою дипломної роботи	21.10.20 - 24.10.20	Виконано
3.	Опис систем відеоспостереження	25.10.20 - 29.10.20	Виконано
4.	Аналіз методів розпізнавання облич	30.10.19 - 03.11.20	Виконано
5.	Аналіз апаратного забезпечення	04.11.20 - 07.11.20	Виконано
6.	Реалізація алгоритму розпізнавання облич	08.11.20 - 19.11.20	Виконано
7.	Розробка програмного забезпечення	20.11.20 - 30.11.20	Виконано
8.	Оформлення електронного варіанту	01.12.20	Виконано
9.	Подання на кафедру. Усунення недоліків. Оформлення пояснювальної записки.	05.12.20	Виконано
10.	Електронна версія доповіді, ілюстративний матеріал доповіді	06.12.20	Виконано

7. Консультанти з окремих розділів:

Назва розділу	Консультант (посада, П.І.Б.)	Дата		Підпис
		Завдання видав	Завдання прийняв	
Охорона праці	Старший викладач Козлітін Олексій Олександрович			
Охорона навколишнього середовища	Професор, д.т.н. Маджд Світлана Михайлівна			

8. Дата видачі завдання: « ____ » _____ 2020 року

Керівник дипломної роботи: _____ Шутко В.М.

Завдання прийняв до виконання: _____ Миронов Д. О.

Реферат

Пояснювальна записка до дипломної роботи «Пристрій для відеоспостереження із розпізнаванням облич за допомогою нейронних мереж»: 104 с., 25 рис., 30 літературних джерел. Об'єкт дослідження: пристрій для відеоспостереження із розпізнаванням облич. Мета роботи: розробити програмне рішення для камери відеоспостереження, розпізнавання облич, та автоматичного сповіщення. Методи дослідження: побудова архітектури програмного забезпечення для камери відеоспостереження, дослідження та практичне використання технології розпізнавання облич, обробка літературних джерел.

КАМЕРА ВІДЕОСПОСТЕРЕЖЕННЯ, РОЗПІЗНАВАННЯ ОБЛИЧ,
ШТУЧНА НЕЙРОННА МЕРЕЖА, РОЗРОБКА ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ.

Зміст

Скорочення, та умовні позначення	6
Вступ.....	7
Розділ 1. Сучасні системи відеоспостереження. Огляд існуючих систем відеоспостереження.	9
1.1 Типи систем відеоспостереження.....	9
1.1.1 Система відеоспостереження на базі відеореєстратора	9
1.1.2 Система відеоспостереження на базі персонального комп'ютера і плат відеозахвату	10
1.1.3 IP відеоспостереження	10
1.1.4 Система відеоспостереження на базі серверу та мікропомп'ютерів з камерою.	11
Висновок. (Розділ 1)	11
Розділ 2. Штучні нейронні мережі. Теоретичні відомості.....	12
2.1 Мозок і біологічний нейрон.	12
2.2 Мережа нейроподібних елементів.	14
2.3 Функція активації	15
2.4 Архітектура нейронної мережі.....	18
2.4.1 Мережі прямого поширення.....	21
2.4.2 Нейронна мережа Хопфілда	23
2.4.3 Ланцюги Маркова.....	25
2.5. Робота нейронної мережі.	26
2.6 Навчання нейронної мережі.	27
2.6.1 Метод зворотного поширення помилки.....	29

2.7 Навчання ШНМ для класифікації.....	31
Висновок. (Розділ 2)	34
Розділ 3. Технологія розпізнавання облич.	36
3.1 Методи автоматичного розпізнавання осіб	36
3.2 Які завдання допомагає нам вирішувати Face recognition і як вони працюють	37
3.3 Face recognition - математичне обґрунтування.....	39
Висновок. (Розділ 3)	42
Розділ 4. Підбір апаратного та програмного забезпечення.	43
4.1 OpenCV	43
4.2 Raspberry Pi 3	43
4.3 Raspberry Pi Wide Camera	45
Висновок. (Розділ 4)	46
Розділ 5. Розпізнавання обличь людиною. Розпізнавання обличь машиною. Метод розпізнавання обличчя за ознаками Хаара. Метод розпізнавання Віюли-Джонса.	47
5.1 Розпізнавання обличь людиною	47
5.2 Розпізнавання обличь машиною	48
5.3 Методи розпізнавання обличь.....	51
5.3.1 Ознаки Хаара.....	51
5.3.2 Метод Віюли-Джонса.....	52
5.4 Використання ознак Хаара	54
Висновок. (Розділ 5)	55
Розділ 6. Розпізнавання обличь людиною за допомогою бібліотеки OpenCV.....	56

6.1	Можливі підходи реалізації.....	56
6.1.1	Короткий опис бібліотеки OpenCV	56
6.1.2	Опис класів і функцій.....	56
6.1.3	Спільний доступ функцій до ресурсів камери	57
6.1.4	Способи зберігання даних	57
6.1.5	Структура лог файлів	59
6.2	Схема реалізації	59
	Висновок. (Розділ 6)	60
	Розділ 7. Охорона праці.....	61
7.1	Перелік небезпечних та шкідливих факторів у робочій зоні.....	61
7.2	Технічні та організаційні заходи щодо зменшення впливу небезпечних та шкідливих виробничих факторів	62
7.2.1	Забезпечення освітлення робочої зони.....	63
7.3	Інструкція з техніки безпеки, пожежної та вибухової безпеки	66
7.3.1	Вимоги щодо забезпечення пожежної безпеки	66
7.3.2	Розробка інструкції з охорони праці	67
7.3.3	Вимоги безпеки перед початком роботи.....	68
7.3.3.1	Пересвідчитися, що прилад правильно підключений;	68
7.3.4.2	Пересвідчитися, що у робочій зоні немає зайвих предметів; .	68
7.3.3.3	Оглянути з'єднувальні ланцюги і кабелі, місця рознімання і пересвідчитися, що вони справні;.....	68
7.3.3.4	У разі виявлення несправності приладу чи інструмента, яким користуються під час роботи, повідомити керівника роботи.	68
7.3.4	Вимоги безпеки під час виконання роботи.....	68

7.3.4.1 Використовувати тільки справний термометр і тільки за призначенням;	69
7.3.4.2 Замінювати запобіжник тільки за номінальним струмом і тільки при вимкненому приладі;.....	69
7.3.4.3 При появі іскріння, короткого замикання, запаху гарі, диму, прилад негайно відключити та виявити причин можливого виникнення пожежі;.....	69
7.3.4.4 Періодично проводити перевірку електричної схеми приладу для виявлення причин, які можуть призвести до виникнення пожежі;.....	69
7.3.4.5 Прибрати з робочого місця зайві речі, які відволікають від роботи і можуть привести до ушкоджень інженера.	69
7.3.5. Вимоги безпеки після закінчення роботи	69
7.3.5.1 Вимкнути прилад, коли оператор чи інженерно-технічний склад працівників залишають свої робочі місця;	70
7.3.5.2 Прибрати своє робоче місце;.....	70
7.3.5.3 Дотримуватись санітарних норм і правил особистої гігієни; .	70
7.3.5.4 Повідомити керівника робіт про виявлені недоліки в роботі приладу.	70
7.3.6. Вимоги безпеки під час аварійних ситуацій:.....	70
7.3.6.1 В разі аварії негайно вимкнути всі електроспоживачі прилади від електромережі не допускати сторонніх осіб в небезпечну зону; повідомити керівника робіт про те, що сталося.	70
7.3.6.2 У випадку виникнення пожежі негайно викликати пожежну команду. До її приїзду приступити до тушіння пожежі своїми силами, а також спасінню людей та надання їм допомоги;.....	70
7.3.6.3 При виникненні пожежі слід знеструмити прилад;	70

7.3.6.4 В обов'язковому порядку в приміщені повинні бути засоби пожежегасіння;.....	70
Висновок. (Розділ 7)	70
Розділ 8. Охорона навколишнього середовища	71
8.1 Вступ.....	71
8.2 Рекомендації щодо зниження негативних чинників.....	72
8.3 Аналіз джерел впливу та їх наслідків та на її оточення	73
Висновок. (Розділ 8)	79
Висновок	80
Список літератури.....	81
Додаток 1. Файл програми	84
Файл 1. bot.py	84
Файл 2. face_recognition.py	90
Файл 3. face_training.py	93
Файл 4. camera_handler.py.....	95
Файл 5. cv_controller.py.....	95
Файл 6. current_names.py.....	100

Скорочення, та умовні позначення

API - Application Programming Interface.

USB - Universal Serial Bus.

JSON - JavaScript Object Notation.

XML - eXtensible Markup Language.

SQL - Structured Query Language.

ПЗ - Програмне забезпечення.

Вступ

В наші дні системи відеоспостереження виділилися в самостійну область засобів охорони, що має власні правила і особливості експлуатації. Системи відеоспостереження забезпечують надійний контроль території, дозволяють охороняти матеріальні цінності і запобігати розкрадання, оцінювати роботу персоналу фірми, контролювати можливі проникнення на територію, що охороняється ззовні. Крім усього іншого, системи відеоспостереження чисто психологічно впливають на потенційних порушників. Відеокамери спостереження відлякують порушників. Адже останні ясно собі уявляють, що всі їхні правопорушення можуть бути записані відеокамерою. І в разі потреби співробітники відділу безпеки без проблем відновлять хід подій, переглянувши зроблені раніше записи.

У більшості випадків система відеоспостереження дозволяє записувати зображення на носії інформації, а також виконувати інші функції. Наприклад, повністю управляти відеокамерами, повертати об'єктив, масштабувати зображення, створювати архіви записів, переглядати і управляти ними. Крім запису відео, сучасні системи відеоспостереження можуть також розпізнавати обличчя.

В даному проекті ми поставили за мету - розробити систему, за допомогою якої буде проводитися фіксація відвідування учнів в будь-якій зоні.

Система відеоспостереження повинна мати основними параметрами:

- можливість розпізнавання обличчя;
- невисока вартість.
- автоматичну систему сповіщення

Сервіс складається з двох частин - Telegram бот інтерфейс і Raspberry PI з камерою - пристрій для фіксації осіб.

У даній роботі описаний процес розробки модуля для роботи з камерою на мікроконтролері Raspberry PI [6]. Далі цей модуль буде використовуватися

при розробці Telegram бота, який виступить в якості інтерфейсу для користувача. Розробку такого модуля можна було пропустити, однак безпосередньо звертатися до низькорівневих функцій камери з бота не є доцільним. По-перше, бот повинен працювати стабільно, тому потрібно мінімізувати можливі помилки. По-друге, навіть якщо обробляти помилки усередині бота - це займає додатковий час, збільшуючи час очікування відповіді для користувача. Так само важливо передбачити можливість для масштабування. Набагато зручніше і наочніше додавати методи до класу (знаходиться всередині розробленого модуля), ніж дописувати нові функції в тілі бота.

Варто так само обґрунтувати вибір Telegram бота в якості інтерфейсу. Telegram - багатоплатформовий месенджер, що дозволяє обмінюватися повідомленнями і медіафайлів багатьох форматів. Так само Telegram надає API для написання ботів, яке детально задокументовано. Тому розробка Telegram бота - проста і дешева завдання. Telegram бот так само має перевагу перед класичними мобільними додатками в питанні впровадження. Користувач може користуватися ботом відразу як він буде запущений.

У свою чергу, впровадження самописного додатки - більш важке завдання. Наше додаток має бути попередньо додано в відповідний сервіс дистрибуції додатків (Google Play або App Store). Так само варто відзначити, що з великою ймовірністю Telegram вже встановлено на телефоні користувача, так як в даний момент кількість щомісячних використань Telegram досягло позначки в 200 млн.

У наступних розділах буде докладно описана сама технологія розпізнавання осіб і принципи її роботи. Так само будуть розглянуті можливі підходи до створення подібного сервісу.

Розділ 1. Сучасні системи відеоспостереження. Огляд існуючих систем відеоспостереження.

1.1 Типи систем відеоспостереження

Найбільш поширені на ринку безпеки такі системи відеоспостереження:

- Система відеоспостереження, побудована на базі відеореєстратора.
- Система відеоспостереження, побудована на базі персонального комп'ютера і плат відеозахоплення.
- Система відеоспостереження, побудована на базі серверу і IP-камер.
- Система відеоспостереження, побудована на базі серверу і мікрокомп'ютерів із камерним модулем

1.1.1 Система відеоспостереження на базі відеореєстратора

Відеореєстратор представляє собою закінчений пристрій з вигляду ідентичне звичайному DVD програвачу. За внутрішньою будовою відеореєстратор схожий на персональний комп'ютер.

Використовувані в даній системі відеоспостереження камери є аналоговими. Даний тип камер перетворює відеосигнал в формат, який може бути отриманий телевізійним або іншим приймачем, наприклад охоронним монітором. У той же час система відеоспостереження є цифровий, так як отримані дані зберігаються на цифровий носій.

В якості операційної системи в відеореєстраторах використовуються різні клони системи Linux, часто розроблені спеціально для даного пристрою.

Фахівці однозначно визначають операційну систему Linux, як найбільш стабільну при довгостроковій безперервній роботі і стійку до зовнішніх впливів - віруси, піратські зломи і т.п.

Відеореєстратори дають користувачеві обмежений доступ до програмної частини, наприклад, ви не можете встановити нові програми або

драйвера, видалити старі - тільки виконання тих команд, які передбачені в меню, як при роботі з будь-яким DVD-плеєром або телевізором. Це є запорукою безперебійної роботи і більш високої надійності системи.

1.1.2 Система відеоспостереження на базі персонального комп'ютера і плат відеозахвату

Система відеоспостереження на базі персонального комп'ютера і плат відеозахвату складається зі звичного всім персонального комп'ютера, зі спеціальною платою захоплення відеозображення. Кожна така плата, найчастіше, має вузьку спеціалізацію.

Тактова частота сучасного комп'ютера становить близько 3 ГГц, що перевищує тактову частоту відеореєстратора в 3 рази. Доречно буде зазначити, що сучасна архітектура процесора, що застосовує сучасну оперативну пам'ять, дає ще більш ніж дворазовий приріст продуктивності. Таким чином, продуктивність комп'ютера вище продуктивності відеореєстратора більш ніж в 6 разів. Це дає змогу підключити додаткові модулі для аналізу руху або розпізнавання обличь. Але нажаль такі системи коштують значно дорожче.

1.1.3 IP відеоспостереження

Система відеоспостереження, побудована на базі персонального комп'ютера і IP-камер багато в чому аналогічна попередній. Центром управління системою також є персональний комп'ютер.

Основною відмінністю від попередньої є спосіб передачі сигналу від камери до ПК і відсутність плат захоплення відеозображення. Сигнал передається по мережевому кабелю. Це зручно коли структура мережі добре організована, а робота стабільна. Замість плат тут використовується мережеве обладнання (маршрутизатори, роутери, мости) і програмне забезпечення, що вимагає ще більш високої кваліфікації інсталяторів, з глибоким знанням мережевих технологій. А, як відомо, праця професіоналів коштує дуже дорого.

За інтеграції і масштабованості дана система обходить всіх конкурентів, але не потрібно забувати, що і витрати на це будуть найбільшими в порівнянні з конкурентами.

1.1.4 Система відеоспостереження на базі серверу та мікропомп'ютерів з камерою.

Система відеоспостереження на базі серверу та мікропомп'ютерів з камерою складає сервер за управлінням ОС Windows або Linux-подібних систем, та 1го або більше мікрокомп'ютерів (наприклад на базі Raspberry PI) яка відрізняється від попередніх систем легкістю у модифікації та підключенням різноманітних модулів.

При використанні таких систем, як правило використовується відкрите ПЗ, або розроблено безпосередньо користувачем із використанням відкритих бібліотек. Тому підключення модуля розпізнавання обличчя коштує значно дешевше.

Опис та використання цього виду системи відеоспостереження буде описано та реалізовано у даному дипломному проекту.

Висновок. (Розділ 1)

Було розглянуто основні типи систем відеоспостереження та наведені їх основні характеристики. На основі проведеного аналізу можна сказати що система побудована на базі мікрокомп'ютеру із написанням власного коду дає можливість економити на створенні, але при створенні власного рішення не має можливості звернутися до сервісного центру, або викликати майстра для налаштування. Усі роботи з пристроєм повинні виконуватись власноруч.

Незважаючи на недоліки, автор ПЗ здатний модифікувати та додавати нові функції у камеру без додаткових коштових витрат, саме тому це єдиний можливий спосіб організувати розумне автоматичне відеоспостереження за малі кошти.

Розділ 2. Штучні нейронні мережі. Теоретичні відомості.

2.1 Мозок і біологічний нейрон.

Нейронні мережі виникли з досліджень в області штучного інтелекту, а саме, зі спроб відтворити здатність біологічних нервових систем навчатися і виправляти помилки, моделюючи структуру мозку. Основною областю досліджень з штучного інтелекту в 60-ті - 80-ті роки були експертні системи. Такі системи ґрунтувалися на моделі процесу мислення. Скоро стало ясно, що подібні системи, хоча і можуть принести користь в деяких областях, які не проникають у деякі ключові аспекти людського інтелекту. Згідно з однією з точок зору, причина цього полягає в тому, що вони не в змозі відтворити структуру мозку. Щоб створити штучний інтелект, необхідно побудувати систему з схожою архітектурою.

Мозок складається з дуже великого числа (приблизно 10,000,000,000) нейронів, з'єднаних численними зв'язками (в середньому кілька тисяч зв'язків на один нейрон, проте це число може сильно коливатися). Нейрони з'єднані між собою нейронними волокнами та утворюють нейронну мережу. Нервові волокна здатні передавати електрохімічні сигнали які завжди несуть корисну інформацію.

Кожний нейрон має відростки двох типів:

- Дендрити –приймачі імпульсів.
- Аксони – передавачі імпульсів.

Аксон поєднується с дендритом через синапси, які можуть збільшувати або зменшувати силу імпульсу. Коефіцієнт примноження сигналу у синапі називають – вагами.[24]

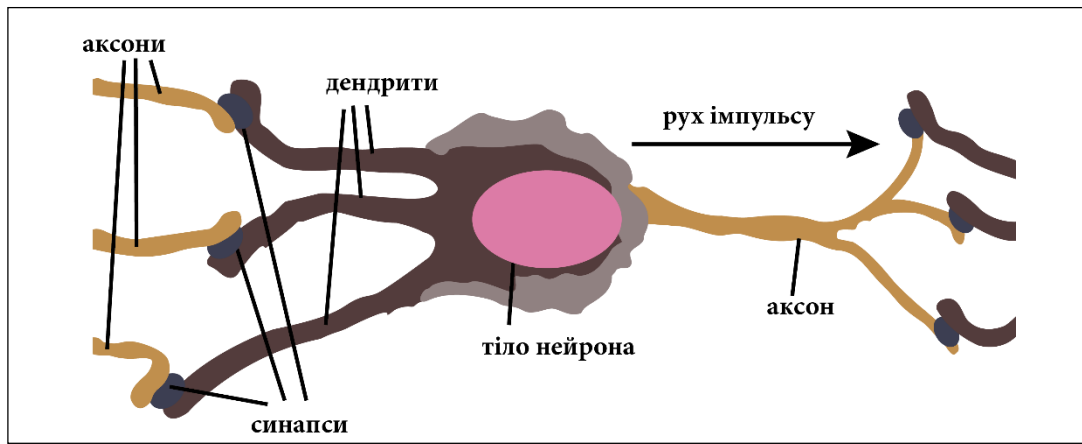


Рис 2.1.1

Коли нейрон приймає сигнали з одного або декількох дендритів, вони складаються, та якщо сумарний імпульс перевищує деякий поріг, нейрон збуджується, та передає новий імпульс далі по аксону.

Варто зазначити, синапси з часом можуть змінювати свої ваги, що впливає на поведінку нейрону та всієї мережі в цілому.

Штучна нейронна мережа:

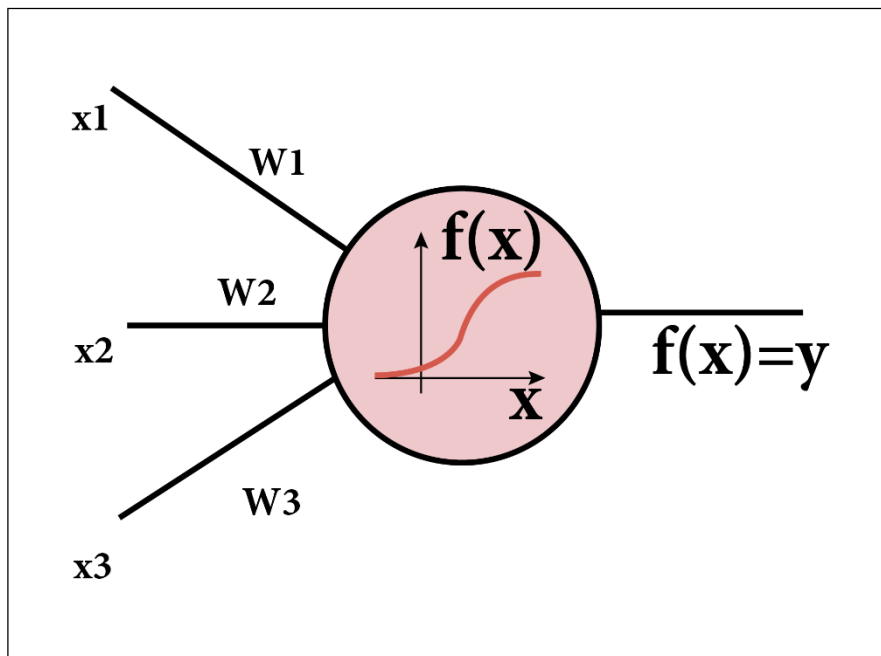


Рис 2.1.2

На рисунку зображена загальна математична модель нейрону з трьома входами.

$x_{1,2,3}$ – вхідні імпульси які подані аксонами інших нейронів

$w_{1,2,3}$ – ваги синапсів

$f(x) = y$ – функція активації нейрону.

Хай до синапсів поступають імпульси сили x_1, x_2, x_3 , тоді після прикладання значень ваг синапсів сигнали розраховуються як x_1w_1, x_2w_2, x_3w_3 відповідно. На вхід функції активації надходить сума:

$$y = f(x) = f(x_1w_1 + x_2w_2 + x_3w_3)$$

З формули маємо висновок, що головною характеристикою нейрону є конфігурація вагів на синапсах.

Таким чином мозок, який побудований з великої кількості дуже примітивних елементів, здатен вирішувати надзвичайно важкі задачі.

2.2 Мережа нейроподібних елементів.

Мережа нейроподібних елементів (штучна нейронна мережа) – це набір нейронів, які з'єднані між собою. У аналогії із біологічним варіантом можна відмітити головні відмінності: як правило функції активації фіксовані та не здатні змінюватися у процесі роботи мережі, також структура мережі не змінюється, не додаються нові нейрони, не зникають існуючі, а синапсові ваги змінюються тільки на деяких етапах користування.

Замість імпульсів використовуються числа, які подаються на вхід нейрона, та у свою чергу результатом кожного нейрона також буде число. Таким чином робота нейронної мережі складає множину однакових модулів для перетворення вхідного вектора у вихідний вектор.

У теорії, штучні нейронні мережі здатні вирішувати задачі аналогічно людському мозку, але на сьогоднішній день, не дивлячись на високий рівень інформаційного розвитку, для рішення подібних задач не вистачає досвіду та потужності.

Будь-яка архітектура ШНМ складається з штучних нейронів - елементів обробки, мають структуру трьох пов'язаних один з одним шарів: вхідним, прихованим і вихідним.

Вхідний шар складається з вхідних нейронів, які передають інформацію в прихований шар. Прихований шар в свою чергу передає інформацію в вихідний.

2.3 Функція активації

Зважена сума зі входів - активаційний сигнал - проходить через функцію активації для виведення даних з нейрона[24]. Є кілька видів функції активації:

- Лінійна функція

$$f(x) = ax$$

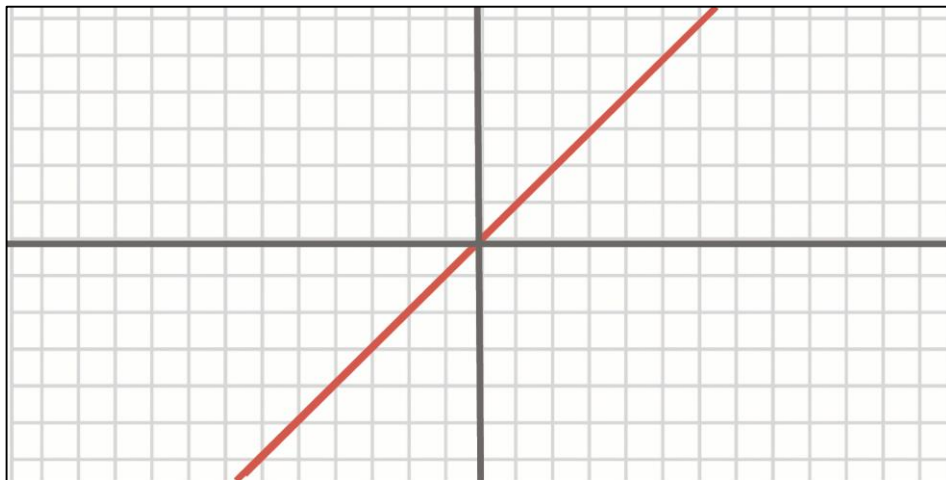


Рис. 2.3.1

- Ступінчата функція

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$



Рис. 2.3.2

- Сигмоїда

$$f(x) = \frac{1}{1 + e^{-x}}$$



Рис. 2.3.3

- Функція гіперболічного тангенсу

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

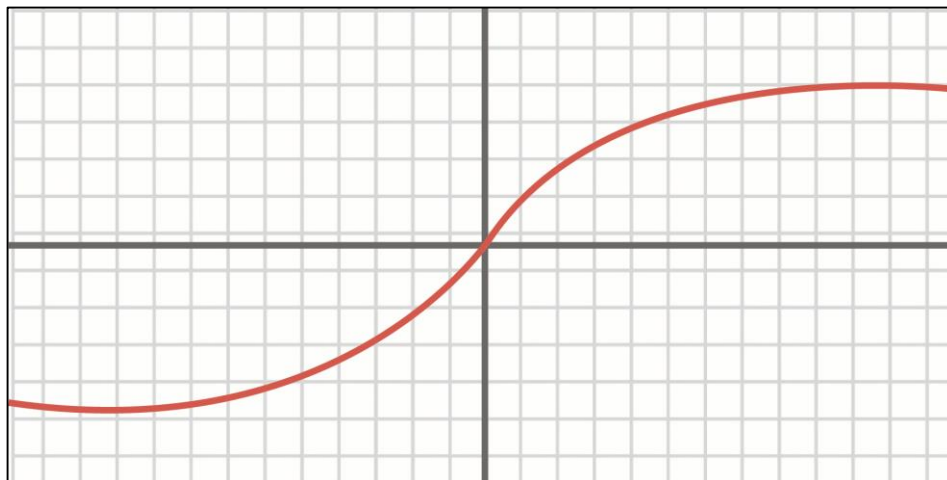


Рис. 2.3.4

- Функція лінійного випрямлення

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$



Рис. 2.3.5

2.4 Архітектура нейронної мережі.

Існує величезна кількість способів з'єднання нейронів, зростаюче зі збільшенням числа нейронів в мережі. Найбільш популярною є пошарова архітектура, в якій нейрони розташовуються «шарами». У найбільш загальному випадку аксони кожного нейрона одного шару спрямовані до нейронам наступного шару. Таким чином, нейрони першого шару є вхідними (приймають інформацію з зовнішнього світу), нейрони останнього шару - вихідними (видають інформацію в зовнішній світ).[26]

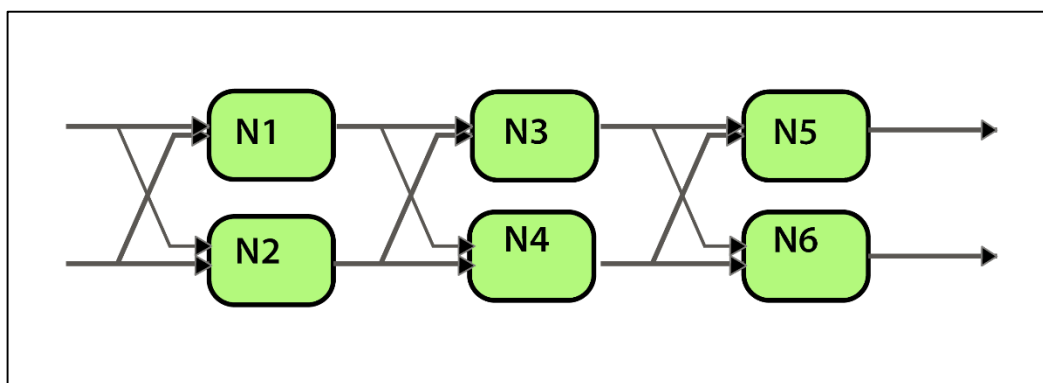


Рис. 2.4.1

Інший вид архітектури – повнозв'язна, коли кожен нейрон з'єднаний з кожним, в тому числі сам з собою. Приклад найпростішої нейромережі з трьох нейронів показаний на рисунку 2.4.2 . Для зручності зображення з кожного нейрона виходить не один, а кілька аксонів, спрямованих на інші нейрони або у зовнішній світ, що аналогічно приєднаним до одного аксону через синапси декільком дендритам.

Саме повнозв'язні нейромережі для побудови експертних систем вибираються виходячи з таких міркувань: по-перше, при однаковому числі нейронів повнозв'язні мережі мають більшу кількість міжнейронних зв'язків, що збільшує інформаційну ємність мережі; по-друге, повнозв'язна архітектура є набагато більш універсальною, що не вимагає експериментів з варіаціями схеми з'єднань для кожного завдання; по-третє, в разі емуляції мережі на

звичайному комп'ютері повнозв'язні мережі мають серйозні переваги - насамперед швидкістю функціонування та простотою програмної реалізації без шкоди для якості навченості.

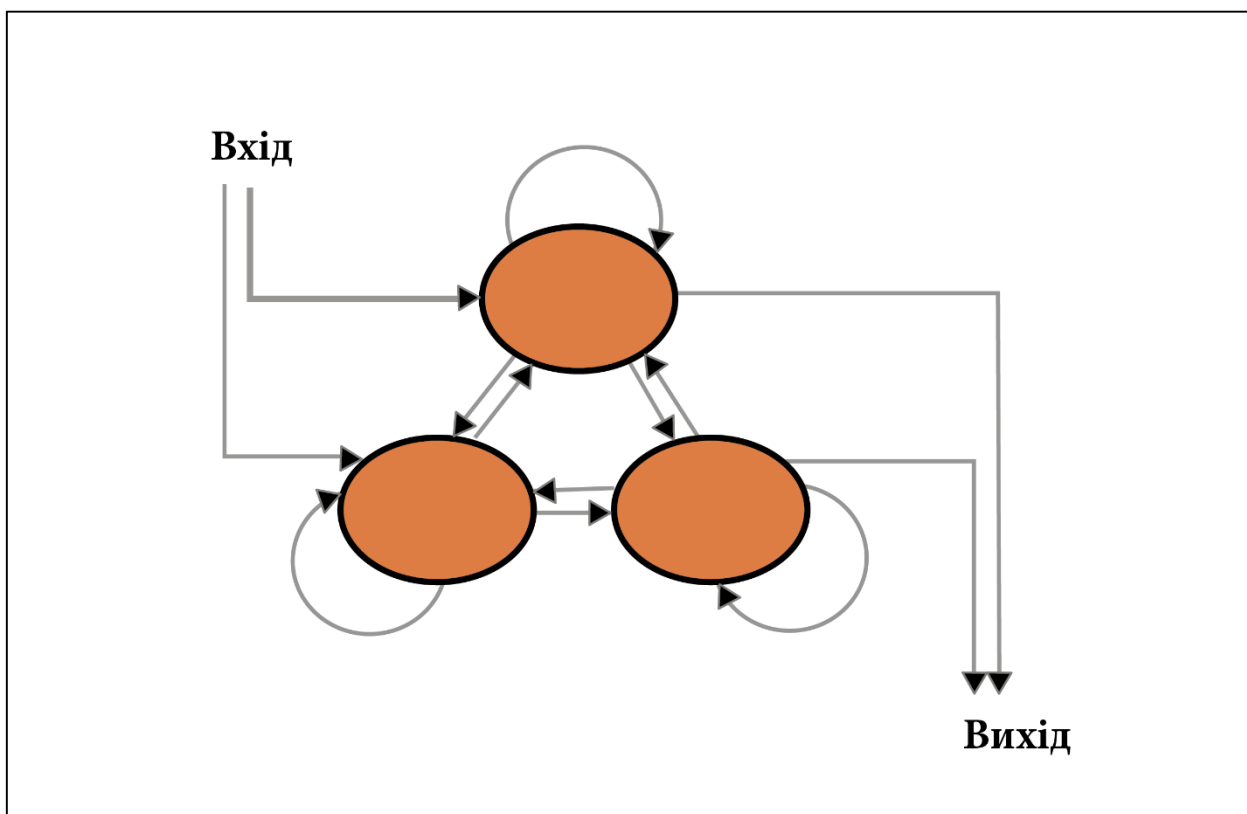















Рис. 2.4.2

Схема найпростішої нейронної мережі з трьох нейронів (мережа має тринадцять синапсів, чотири з яких служать для зв'язку із зовнішнім світом, а інші з'єднують нейрони між собою).[26]

Далі слідують перегляд найпопулярніших архітектур, та відзначити деякі з них зараз подаються як новітні і прогресивні. Оскільки наука про штучний інтелект зараз активно розвивається, деякі архітектури використовують не зовсім стандартні моделі нейронів (стандартні – це ті, які були описані у попередніх розділах), вони можуть запом'ятовувати інформацію, мати додаткові умови активації, або виконувати одночасно декілька ролей. Будуть використовуватися імена та скорочення на англійській мові (згідно з оригіналом). [25]

Позначення:

-  Нейрон зворотнього доступу
-  Вхідний нерон
-  Вхідний нейрон з додатковим шумом
-  Скритий нейрон
-  Імовірнісний скритий нейрон
-  Імпульсний скритий нейрон
-  Вихідний нейрон
-  Зрівнюючий нейрон входу/ виходу
-  Рекурентний нейрон
-  Нейрон з пам'яттю
-  Нейрон з пам'яттю
-  Ядро
-  Згортковий нейрон

Слід зазначити, що не всі (хоча і більшість) з використовуваних тут скорочень загальноприйняті. Під RNN іноді розуміють рекурсивні нейронні мережі (recursive neural networks), але зазвичай ця аббревіатура означає рекуррентну нейронну мережу (recurrent neural network). Але і це ще не все: у багатьох джерелах ви зустрінете RNN як позначення для будь-якої рекуррентної архітектури, включаючи LSTM, GRU і навіть двонаправлені варіанти. Іноді схожа плутанина відбувається з АЕ: VAE, DAE і їм подібні можуть називати просто АЕ. Багато скорочення містять різну кількість N в кінці: можна сказати "сверточное нейронна мережа" - CNN (Convolutional Neural Network), а можна і просто "сверточное мережу" - CN.

2.4.1 Мережі прямого поширення

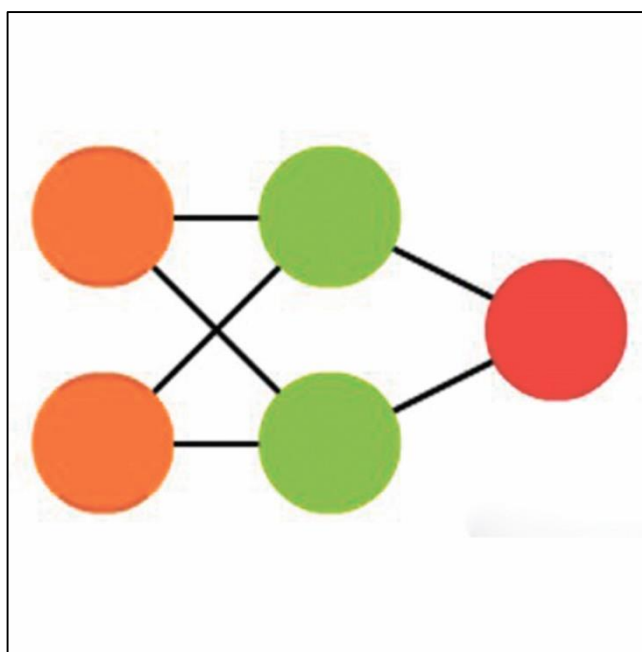


Рис. 2.4.1.1

Мережі прямого поширення (Feed forward neural networks, FF or FFNN). Вона дуже проста - нейрони передають інформацію від входу до виходу. Вважається, що у нейронних мереж є шари, кожен з яких складається з вхідних, прихованих або вихідних нейронів. Нейрони одного шару між собою

не пов'язані, при цьому кожен нейрон цього шару пов'язаний з кожним нейроном сусіднього шару.

Найпростіша мало-мальськи робоча мережа складається з двох вхідних і одного вихідного нейрона і може моделювати логічний вентиль - базовий елемент цифрової схеми, що виконує елементарну логічну операцію. FFNN зазвичай навчають методом зворотного поширення помилки, подаючи моделі на вхід пари вхідних і очікуваних вихідних даних. Під помилкою зазвичай розуміються різні ступені відхилення вихідних даних від вихідних (наприклад, середньоквадратичне відхилення або сума модулів різниць). За умови, що мережа має достатню кількість прихованих нейронів, теоретично вона завжди зможе встановити зв'язок між вхідними та вихідними даними. На практиці використання мереж прямого поширення обмежена, і частіше вони використовуються спільно з іншими мережами.

2.4.2 Нейронна мережа Хопфілда

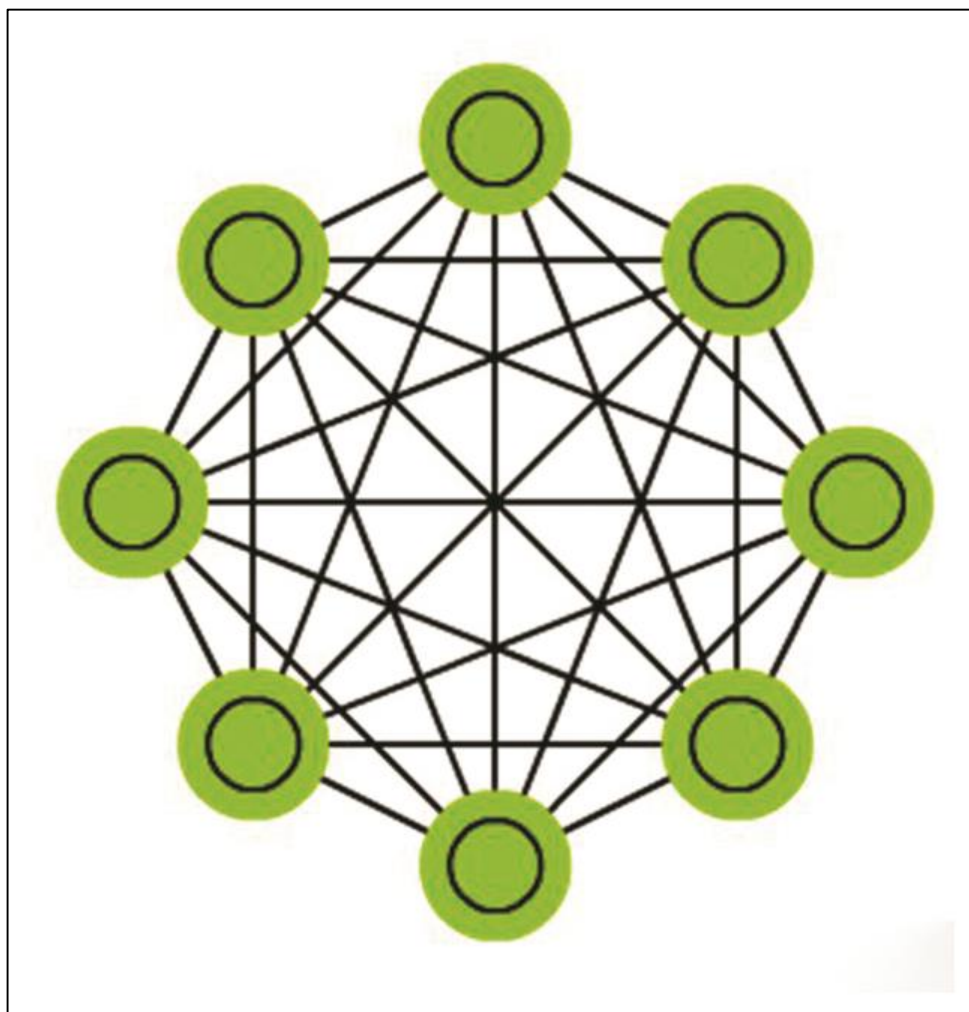


Рис. 2.4.2.1

Нейронна мережа Хопфілда – повнозв’язна мережа (кожен нейрон з’єднаний з кожним), де кожен нейрон виступає у всіх трьох типах. Кожен нейрон служить вхідним до навчання, прихованим під час нього і вихідним після. Матриця ваг підбирається таким чином, щоб всі «успішної реєстрації» вектора були б для неї власними. Одного разу навчена система буде сходиться до одного з відомих їй образів, тому що тільки одне з цих станів є стаціонарним. Відзначимо, що це не обов’язково відповідає бажаного стану. Система стабілізується тільки частково через те, що загальна "енергія" або "температура" мережі під час навчання поступово знижується.

Кожен нейрон має поріг активації, порівнянним з цієї температурою, і якщо сума вхідних даних перевищить цей поріг, нейрон може переходити в

одне з двох станів (зазвичай -1 або 1 , іноді 0 або 1). Вузли мережі можуть оновлюватися паралельно, але найчастіше це відбувається послідовно. В останньому випадку генерується випадкова послідовність, яка визначає порядок, в якому нейрони будуть публікувати оновлення стану. Коли кожен з нейронів оновився і їх стан більше не змінюється, мережа приходить в стаціонарний стан. Такі мережі часто називають асоціативною пам'яттю, так як вони сходяться з станом, найбільш близького до заданого: як людина, під час перегляду половини картинки, може домалювати відсутню половину, так і нейронна мережа, отримуючи на вході наполовину зашумлену картинку, добудовує її до цілої.

2.4.3 Ланцюги Маркова

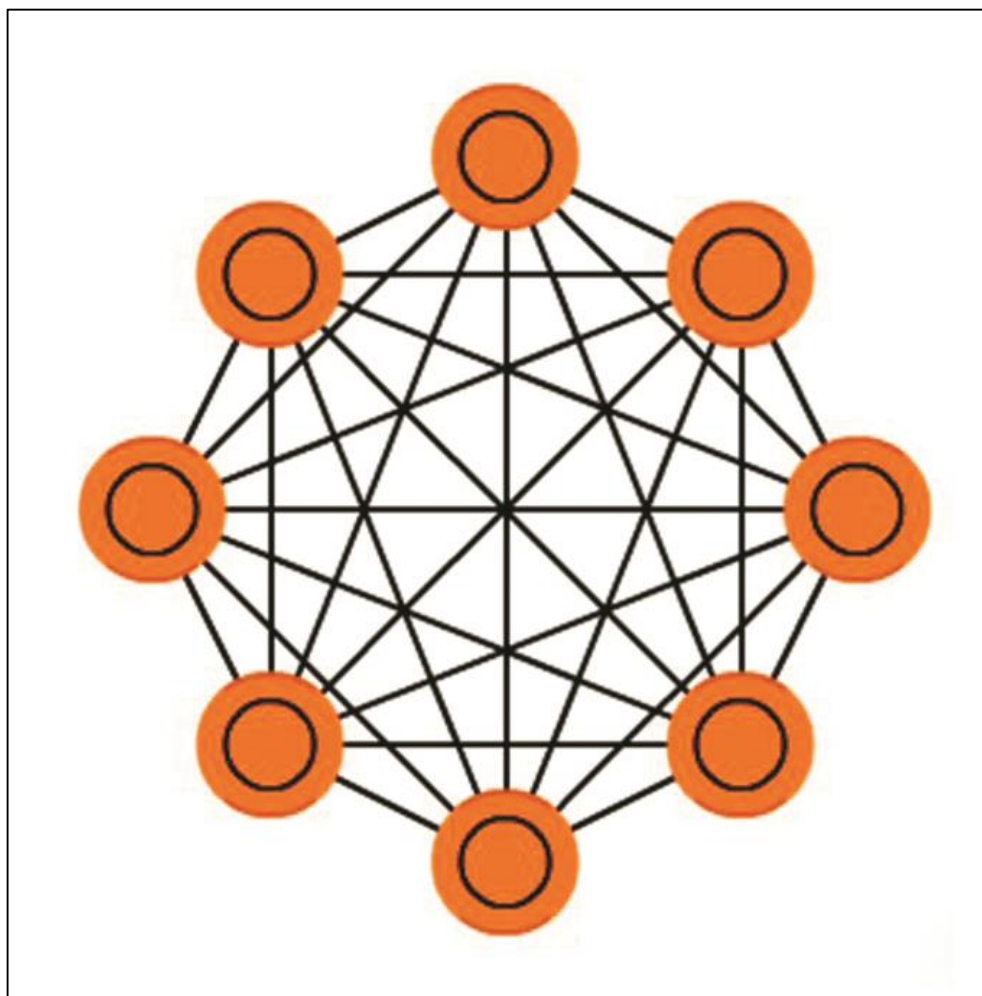


Рис. 2.4.3.1

Ланцюги Маркова (Markov Chains, MC або discrete time Markov Chain, DTMC) - свого роду попередники машин Больцмана (BM) і мереж Хопфілда (HN). У ланцюгах Маркова ми задаємо ймовірності переходу з поточного стану в сусідні. Крім того, ці ланцюга не мають пам'яті: подальший стан залежить тільки від поточного і не залежить від всіх минулих станів. Хоча ланцюги Маркова не можна назвати нейронною мережею, вона близька до них і формує теоретичну основу для BM і HN. Ланцюги Маркова також не завжди є повнозв'язними.

2.5. Робота нейронної мережі.

У разі емуляції ШНМ на звичайному комп'ютері все математичні операції здійснює програма. Нейронна мережа при цьому являє собою масив синаптичних ваг. Цей масив може перебувати або на диску ПК у вигляді файлу певного формату (при зберіганні ШНМ), або в оперативній пам'яті комп'ютера (коли нейросеть функціонує).

При створенні нової ШНМ в пам'яті комп'ютера відводиться місце під масив синаптичних ваг, званий картою. Цей масив заповнюється абсолютно випадковими числами з певного діапазону. Тому кожна створена мережа навіть при одних і тих же параметрах (число нейронів, їх характеристика) є унікальною. Унікальність мереж проявляється в тому, що мережі з однаковими параметрами, які навчаються на однакових завданнях, поведуться неоднаково. Це стосується часу навчання, якості навчання, впевненості в видаваних відповідях при тестуванні. У цьому ще одна схожість мереж з біоб'єктами.

Розгляньмо роботу мережі безвідносно до процесу навчання або видачі рішення. Після ініціалізації мережі ваги всіх синапсів випадкові. У початковий момент часу на вхідні нейрони через вхідні синапси (один або кілька) подається з зовнішнього світу вектор вхідних сигналів, що представляє набір чисел (або одне число). Далі цей сигнал починає поширюватися між нейронами, змінюючись при проходженні через кожен нейрон згідно зі схемою функціонування нейрона. В кінцевому підсумку, після одного проходу вихідні нейрони видадуть до зовнішнього світу будь-які сигнали. Вся процедура одноразового проходження сигналів по нейронній мережі є тактом функціонування мережі. Можна не зчитувати сигнали з вихідних нейронів після одного такту функціонування, а продовжити обмін сигналами ще кілька разів, не подаючи сигналів на вхід. Зазвичай кількість тактів функціонування між подачею сигналів на вхід і зняттям сигналів з виходу фіксовано для даної

мережі. Як правило, цей параметр задається при ініціалізації мережі і зберігається в файлі мережі разом з матрицею коефіцієнтів.

Таким чином, нейронна мережа, яка отримує на вході деякий сигнал, здатна після проходження його по нейронам видавати на виході чітку відповідь, який залежить від вагових коефіцієнтів всіх нейронів і від самого сигналу. Зрозуміло, що при здійсненні таких процедур на тільки що ініціалізованій мережі ми будемо отримувати на виході сигнали, позбавлені будь-якого сенсу (вагові коефіцієнти випадкові). Щоб домогтися видачі мережею необхідного результату, необхідно навчити її.[29]

2.6 Навчання нейронної мережі.

Для навчання нейронної мережі необхідна навчальна вибірка, що складається з прикладів та відповідей. Кожен приклад являє собою задачу одного і того ж типу з індивідуальним набором умов (вхідних параметрів) і заздалегідь відомим відповіддю.

Наприклад, в якості вхідних параметрів в одному прикладі можуть використовуватися дані обстеження одного хворого, тоді заздалегідь відомим відповіддю в цьому прикладі може бути діагноз. Кілька прикладів з різними відповідями утворюють задачник. Задачник розташовується в базі даних, кожен запис якої є прикладом.

Розглянемо загальну схему навчання ШНМ:

1. З навчальної вибірки береться поточний приклад (спочатку перший), і його вхідні параметри (що представляють в сукупності вектор вхідних сигналів) подаються його на вхідні синапси. Зазвичай кожен вхідний параметр прикладу подається на один відповідний вхідний синапс.

2. ШНМ виробляє задану кількість тактів функціонування, при цьому вектор вхідних сигналів поширюється по зв'язках між нейронами

3. Вимірюються сигнали, видані тими нейронами, які вважаються вихідними.

4. Проводиться інтерпретація виданих сигналів, і обчислюється оцінка, що характеризує відмінність між виданими мережею відповіддю і необхідним відповіддю, наявними в прикладі. Оцінка обчислюється за допомогою відповідної функції оцінки. Чим менше оцінка, тим краще розпізнано приклад, тим ближче виданий мережею відповідь до необхідного. Оцінка, що дорівнює нулю, слово, яке потрібно відповідність обчисленого і відомого відповідей досягнуто.

Зауважимо, що тільки що створена ШНМ може видати правильну відповідь абсолютно випадково.

5. Якщо оцінка прикладу дорівнює нулю, нічого не робиться. В іншому випадку на підставі оцінки обчислюються поправочні коефіцієнти для кожного синапсу матриці зв'язків, після чого проводиться підстроювання синаптичних ваг. У корекції ваг синапсів і полягає навчання.

6. Здійснюється перехід до наступного прикладу задачника, і перераховані вище операції повторюються. Прохід по всіх прикладів навчальної вибірки від першого до останнього вважається одним циклом навчання.

При проходженні циклу кожен приклад має свою оцінку. Обчислюється, крім того, сумарна оцінка безлічі всіх прикладів навчальної вибірки. Якщо після проходження декількох циклів вона дорівнює нулю, навчання вважається закінченим, в іншому випадку цикли повторюються.

Кількість циклів навчання, так само як і час, потрібний для повного навчання, знаходяться в залежності від багатьох факторів - величини навчальної вибірки, кількості вхідних параметрів, виду завдання, типу і параметрів нейромережі і навіть від випадкового розкладу ваг синапсів при ініціалізації мережі.[30]

2.6.1 Метод зворотного поширення помилки

Цей метод є основним і має ще одну назву - Backpropagation, так як використовує алгоритм градієнтного спуску. Тобто за допомогою руху уздовж градієнта розраховується локальний мінімум і максимум функції. Розуміння цього методу приведе до того, що наступний не викличе труднощів.

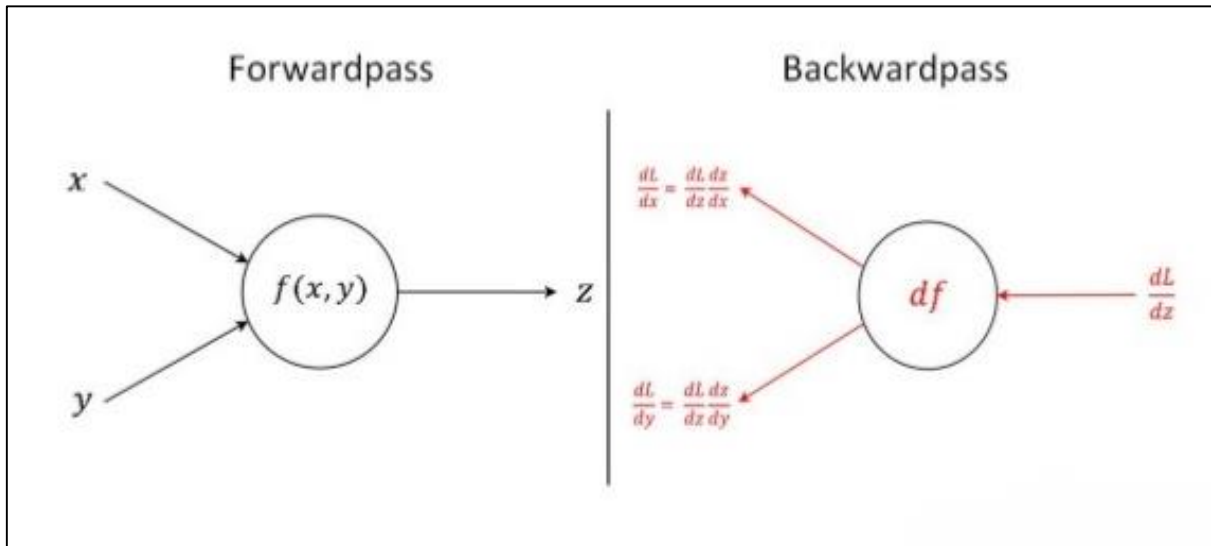


Рис. 2.6.1.1

Якщо уявити функцію у вигляді графіка, що визначає залежність помилки від ваги синапсу, то на цьому графіку необхідно знайти точку з мінімальним значенням помилки і з максимальним. При цьому кожен вага повинна мати своє графічне представлення і до кожного з них необхідно обчислити глобальний мінімум.

Сам градієнт є векторне значення визначає напрямок і крутизну схилу. Градієнт знаходиться за допомогою похідної від функції в потрібній точці. Опинившись в певній точці багатозначно ваги, який розподіляється у випадковому порядку, обчислюється градієнт і визначається напрямок руху спуску, і так в кожній наступній точці, поки не досягається локальний мінімум, який дозволить подальшого спуску. Щоб впоратися з цією неприємністю необхідно встановити потрібне значення моменту, яке дозволить подолати частину графіка і досягти потрібної точки. Якщо це значення буде

недостатнім, то подолати опуклість не вийде, у випадку з установкою більш високого значення, ніж потрібно їсти шанси проскочити глобальний мінімум. Крім моменту прискорення, є ще поняття, що визначає загальну швидкість навчання мережі. Це значення, як і попереднє являє собою гіперпараметр і підбирається методом проб і помилок. Оптимальний варіант заздалегідь ніколи не відомий, дізнатися його можна тільки провівши кілька навчань і коригуючи кожен раз значення в потрібному напрямку. Тепер можна перейти безпосередньо до методу навчання і визначити, що він із себе представляє. У процесі надходження інформації нейронна мережа послідовно передає її від одного нейрона до іншого за допомогою синапсів, до того моменту, поки інформація не опиниться на вихідному шарі і не буде видана як результат. Такий спосіб називається передачею вперед. Після того як результат отриманий обчислюється помилка і на її ґрунті виконуємо зворотний передачу. Суть якої - послідовно змінити вагу синапсів починаючи з вихідного і просуваючись до вхідного шару. При цьому значення ваги змінюється в бік кращого результату. Для використання такого методу навчання підійдуть тільки ті функції активації, які можна диференціювати. Так як зворотне поширення обчислюється за допомогою вирахування різниці результатів і множення його на похідну функції від вхідного значення. Для того щоб успішно провести навчання, необхідно поширити отриману помилку на весь вага мережі. Вирахувавши помилку на вихідному рівні, а також там можна обчислити дельту, яка буде послідовно передаватися між нейронами. Потім необхідно провести розрахунок градієнта для кожної вихідного зв'язку. Потім маючи всі необхідні дані потрібно оновити програмне забезпечення ваг і розрахувати завдяки функції МОР значення, яке стане величиною зміни. При цьому не варто забувати про момент і швидкість навчання. Одна ітерація МОР дає невеликий відсоток зменшення помилки, тому повторювати їх необхідно знову і знову поки показник помилки не буде наближений до 0.[30]

2.7 Навчання ШНМ для класифікації

При вирішенні класифікаційної задачі потрібно інтерпретувати відповіді вихідних нейронів, тому що остаточною відповіддю є клас, до якого належить приклад. Інтерпретація проводиться таким чином: тому що число вихідних нейронів збігається з кількістю класів в розв'язуваній задачі, кожен вихідний нейрон відповідає за клас з відповідним номером. Номер вихідного нейрона, яка видала найбільший вихідний сигнал, і є номер класу, до якого, на думку ШНМ, відноситься даний приклад. Однак при навчанні необхідно не тільки встановити факт збігу або розбіжності відповідей, а й обчислити оцінку, яка показує, наскільки обчислений відповідь неправильна. Це досягається обчисленням різниці між двома максимальними вихідними сигналами, виданими вихідними нейронами. При цьому виявляється не тільки вихідний нейрон, що видав максимальний сигнал, але і вихідний нейрон, що видав другий за величиною сигнал. Якщо число вихідних нейронів більше двох, відповіді інших нейронів не мають значення. Якщо різниця не перевищує заздалегідь встановленого значення (рівня надійності), оцінка прирівнюється до нуля, приклад вважається розпізнаним абсолютно правильно, а корекції синаптичних ваг на даному прикладі не потрібно.

Ситуацію розпізнавання прикладу в задачах класифікації можна представити у вигляді деяких дій, які виробляються в просторі, розмірність якого дорівнює числу класів в завданні. У разі бінарної класифікації робочий простір являє собою квадратний ділянку площині (двомірного простору), мінімальні і максимальні координати якого обмежені значеннями -1 і 1 .

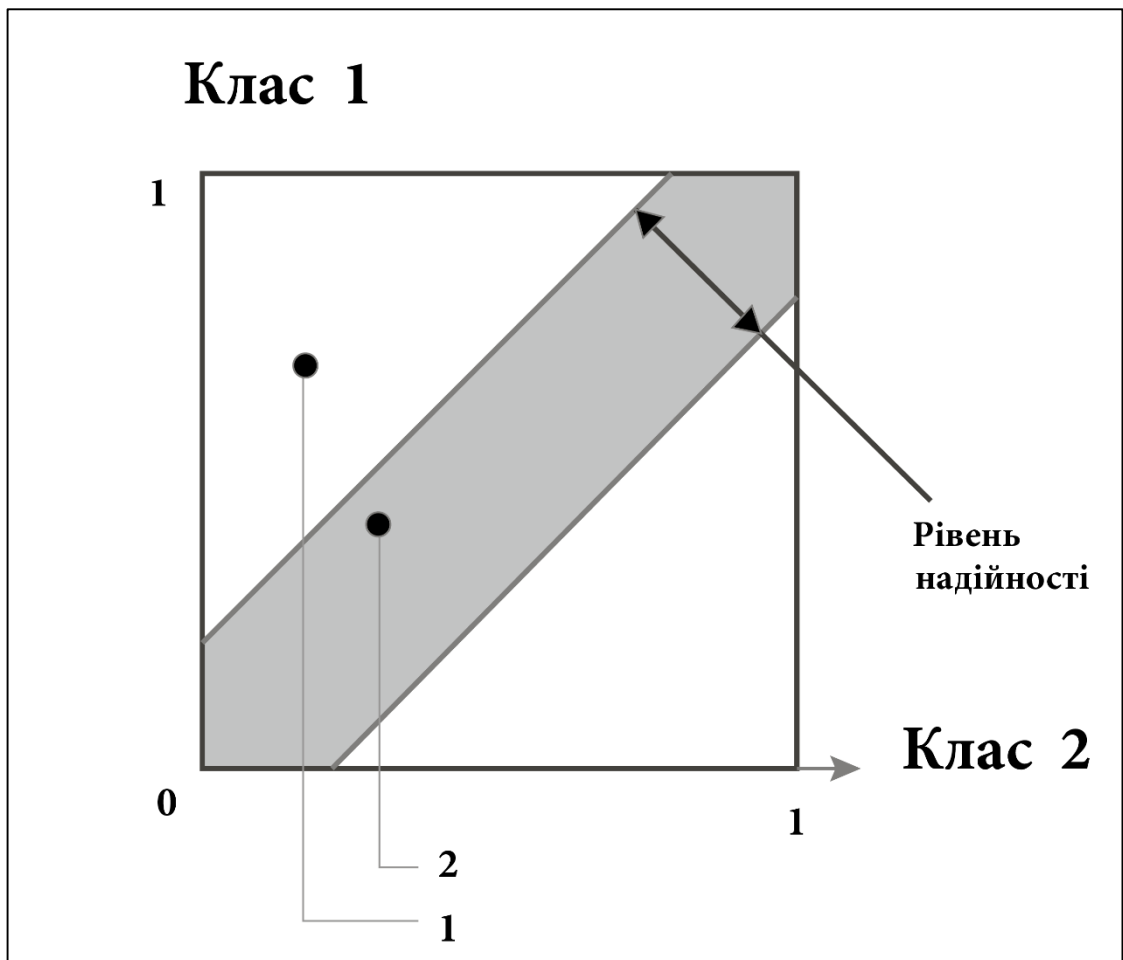


Рис. 2.7.1

На рисунку 2.7.1 Схема проєкції навчальних прикладів в двомірному просторі при бінарної класифікації: 1 - приклад першого класу, який потрапив в зону «надійності»; 2 - приклад першого класу, розпізнаний правильно, але з недостатнім рівнем надійності

Значення вертикальної координатної осі відповідають вихідним сигналам першого вихідного нейрона (що відповідає за перший клас), значення горизонтальної координатної осі - вихідним сигналам другого вихідного нейрона (другий клас). Простір поділяється на дві частини діагональною лінією. У поточний момент часу параметри прикладу подаються на вхідні нейрони ШНМ. Після всіх перетворень відповідь прикладу проєктується в певну точку простору і в залежності від значень, знятих з вихідних нейронів, потрапляє в одну з областей, розділених діагоналлю.

Область кожного класу в просторі відповідає якомога більших значень (1) «своїх» осі координат і якомога меншим значенням (-1) іншої осі координат.

Таким чином, чим більше буде значення, зняте з будь-якого вихідного нейрона, тим більше буде значення координати точки по осі, відповідної цьому нейрону, і навпаки. Ситуація, коли значення, зняті і з того, і з іншого нейронів, рівні, призведе до потрапляння точки на діагональ (незалежно від значень чисел). Пунктирною лінією відсічені зони «надійності», що знаходяться у кутів квадрата, відповідають класам завдання. Чим більше відстані між лініями, тим більше заданий рівень надійності і, відповідно, вище вимоги до мережі. Приклад, відповідь якого «потрапив» в свою область, і до того ж в зону надійності, вважається розпізнаним абсолютно вірно.

В процесі навчання відповіді прикладів, що подаються мережі багаторазово, поступово «розлучаються» якнайдалі від діагоналі і якомога ближче до кутів, кожен з яких відповідає певному класу. Після того, як всі приклади потрапили в зони «надійності», відповідні їх класам, мережа вважається навченої повністю. Поступово підвищуючи рівень надійності, можна домогтися більшої точності відповідей мережі і кращого розпізнавання прикладів навчальної вибірки.

Розглянемо питання: при якому мінімальному перевищенні сигналу одного вихідного нейрона над сигналом іншого слід вважати приклад правильно розпізнаним? Універсального рішення не існує, тому фахівець, який навчає ШНМ, сам задає рівень надійності виходячи з реальної завдання. У деяких випадках достатньо лише правильно визначити клас (з мінімальним перевищенням одного сигналу над іншими). Тоді задається невеликий рівень надійності. В інших випадках рівень надійності повинен бути збільшений. Максимальне значення рівня надійності дорівнює 2. Це ідеальне число, яке в

принципі недосяжно при навчанні. Очевидно, що при великих значеннях рівня надійності мережу буде навчатися довше. Крім того, для кожної реальної завдання існує свій граничний рівень надійності, з яким може навчитися нейросеть.[27]

В основі навчання ШНМ (підстроювання синаптичних ваг) лежить удосконалений метод подвійного функціонування. У наведеній схемі число вхідних нейронів точно відповідає кількості навчальних параметрів (розмірності вектора вхідних сигналів). Однак в наших експертних системах ми використовували більш гнучку і одночасно більш потужну подачу вхідних сигналів на вхід нейромережі. Суть її полягає в тому, що вектор вхідних сигналів перед подачею на мережу перетворюється множенням на адаптивну матрицю прийому сигналів, що містить підлаштовуватися, як і ваги синапсів, значення. Крім підвищення швидкості і якості навчання, це дає можливість задавати число нейронів в мережі довільним, незалежним від числа навчальних параметрів. Мережі з невеликим числом нейронів, очевидно, займають менше місця на диску і в пам'яті комп'ютера, швидше завантажуються з диска для видачі відповіді (іноді для тестування прикладів доводиться кожен раз завантажувати мережі з диска, особливо якщо мереж багато, а всі вони одночасно не поміщаються в оперативної пам'яті).

Висновок. (Розділ 2)

Людський мозок – це чорний ящик, який здатен розпізнавати, створювати, спілкуватися, контролювати життєдіяльність, і при цьому у сучасного людства нема повного розуміння як все ж таки він працює. Вчені відчуювають спокусу відтворити людський мозок програмно, і останні 50 років рухаються у цьому напрямку.

В даний час існує велика кількість варіантів архітектур ШНМ, кожна з яких добре вирішує певний тип задач, але як працює кожна з них – ніхто не

може відповісти. Напрямок вивчення роботи нейромереж побудований на досліджах та експериментах.

Розділ 3. Технологія розпізнавання облич.

Розпізнавання облич - це практичне застосування теорії розпізнавання образів, в завдання якого входить автоматична визначення особи на фотографії і, якщо нам необхідно, то ідентифікація персони по обличчю.

3.1 Методи автоматичного розпізнавання осіб

Завдання ідентифікації і розпізнавання осіб - це одна з перших практичних завдань, яка стимулювала становлення і розвиток теорії розпізнавання та ідентифікації об'єктів. Існує дев'ять категорій об'єктів, які відповідають гностичним областям і викликають зорові образи:

Інтерес до процедур, які лежать в основі процесу впізнавання і розпізнавання осіб, завжди був значним, особливо в зв'язку зі зростаючими практичними потребами: охоронні системи, верифікація, криміналістична експертиза, телеконференції і тоді. Незважаючи на ясність того життєвого факту, що людина добре ідентифікує обличчя людей, зовсім не очевидно, як навчити ЕОМ проводити цю процедуру, в тому числі як декодувати і зберігати цифрові зображення осіб. Ще менш ясними є оцінки схожості осіб, включаючи їх комплексну обробку. Можна виділити кілька напрямків досліджень проблеми розпізнавання осіб:

- нейрофізіологічні моделі;
- інформаційно - процесуальні моделі;
- комп'ютерні моделі розпізнавання.
- нейропсихологічні моделі;

Проблема розпізнавання осіб розглядалася ще на ранніх стадіях комп'ютерного зору. Ряд компаній протягом більше 40 років активно розробляють автоматизовані, а зараз і автоматичні системи розпізнавання людських обличь: ImageWare (system FaceID); Imagis, Epic

Solutions, Spillman, Miros (system Trueface); Vissage Technology (system Vissage Gallery); Visionics (system FaceIt), Smith & Wesson (ASID - Automated Suspect Identification System);. [4]

Технології розпізнавання осіб дозволяють виробляти автоматичний пошук і розпізнавання осіб в графічних файлах і відео потоці.

3.2 Які завдання допомагає нам вирішувати Face recognition і як вони працюють

Система розпізнавання осіб дозволяють робити і такі прості і складні речі:

- підтвердження і визначення особистості студента під час написання іспитів онлайн;
- розпізнавання людей в громадських місцях, які знаходяться в «чорному списку»;
- оплата різних товарів;
- в сучасних парках збереження вашого місця в черзі при відвідуванні будь-яких атракціонів;
- розблокування телефону.

У минулому системи розпізнавання осіб знаходили застосування в основному в сфері правоохорони, оскільки органи використовували їх для пошуку випадкових осіб в натовпі. Деякі урядові установи також використовували подібні системи для безпеки і для усунення шахрайства на виборах.

Однак є багато інших ситуацій, в яких таке програмне забезпечення стає популярним. Системи стають дешевшими, їх поширення зростає. Тепер вони сумісні з камерами і комп'ютерами, які використовуються банками і аеропортами. Туристичні агентства працюють над програмою «бувалого мандрівника»: з її допомогою вони проводять швидкий скринінг безпеки для пасажирів, які добровільно надають інформацію. Черги в аеропортах будуть

просуватися швидше, якщо люди будуть проходити через систему розпізнавання осіб, що зіставляє особи з внутрішньою базою даних.

Інші потенційні застосування включають банкомати і термінали видачі готівки. Програмне забезпечення може швидко перевірити особу клієнта. Після дозволу клієнта банкомат або термінал робить знімок особи. Програмне забезпечення створює відбиток особи, що захищає клієнта від крадіжки особистих даних і шахрайських транзакцій, - банкомат просто не видасть гроші людині з іншою особою.

При використанні 2D-зображень успішно розпізнати обличчя можна тільки при зйомці в анфас і при хорошому освітленні, що підходить для охоронних систем на підприємствах і в держустановах. Для роботи в громадських місцях застосовуються 3D-зображення. Кілька синхронізованих камер роблять ряд фотознімків з різних ракурсів, на основі яких формується тривимірна модель об'єкта, з якої і працює система, визначаючи контрольні точки. Після цього отримані дані порівнюються з наявними в базі даних, і, при збігу параметрів, людина ідентифікується.

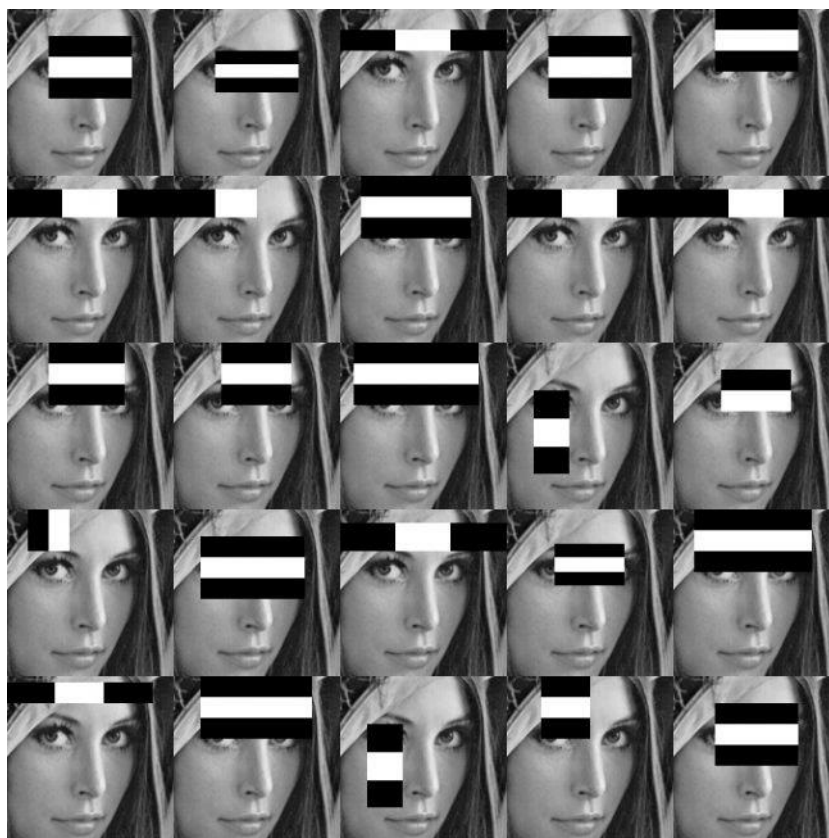


Рис. 3.2.1 як розпізнається 2D зображення

3.3 Face recognition - математичне обґрунтування

Опишемо процес розпізнавання і класифікації осіб камерою. В якості алгоритму розпізнавання осіб був обраний алгоритм «Каскади Хаара». Каскаду Хаара - алгоритм, який використовувався ще в старих фотоапаратах.

Для початку розберемося, як розпізнати обличчя на фотографії. Для того, щоб знайти особу ми повинні виділити його основні компоненти, такі як ніс, лоб, очі, губи і т.д. Для цього будемо використовувати шаблони (вони ж примітиви Хаара) (рис. 1.3.1).

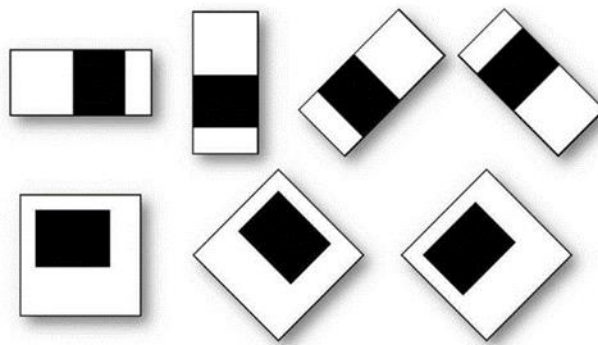


Рис. 3.3.1 Примітиви Хаара

Якщо шаблони відповідають конкретним областям на зображенні, будемо вважати, що на зображенні є людське обличчя. Для кожного з шаблонів вважається різниця між яскравістю білої і чорної областей. Це значення порівнюється з еталоном і приймається рішення про те, чи є тут частина людського обличчя чи ні.

Цей метод називається методом Віюлі-Джонса (так само відомий як каскади Хаара). Давайте уявимо, що у нас на фотографії не одне велике обличчя, а багато дрібних. Якщо застосувати шаблони до всієї зображенні ми не знайдемо там осіб, тому що вони будуть менше шаблонів. Для того щоб шукати на всіх фото особи різних розмірів використовується метод ковзного

вікна. Саме всередині цього вікна і вираховуються примітиви. Вікно як би ковзає по всьому зображенню. Після кожного проходження зображення вікно збільшується, щоб знайти особи більшого масштабу (рис. 1.3.2).

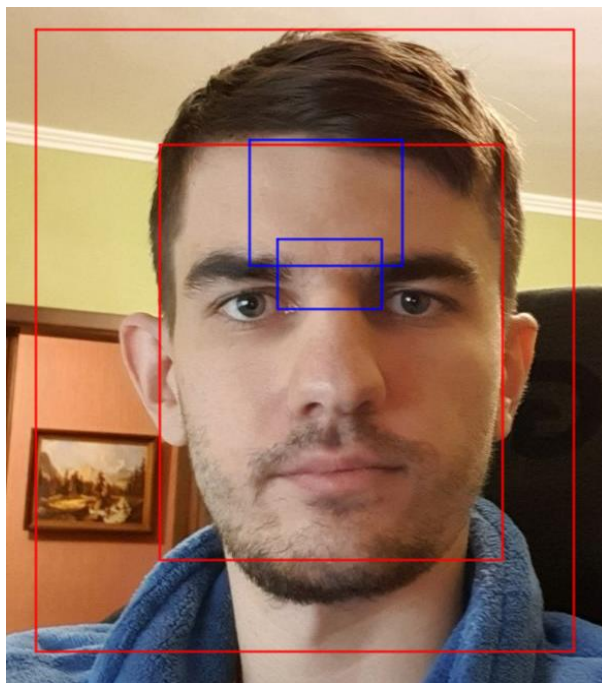


Рис. 3.3.2 Ковзаюче вікно

Тепер опишемо роботу класифікатора, який визначатиме чиє саме особа в даний момент захопила камера. Ознаки для моделі формуються за допомогою каскаду Хаара. Самі ознаки обробляються за допомогою алгоритму градієнтного бустінга. Метод найшвидшого бустінг буде класифікувати людей, а Каскади Хаара шукатимуть на фотографіях обличчя.

Переді тим як розповісти про «градієнтними бустінгу», необхідно розглянути більш простий алгоритм, який буде виступати в якості базового (градієнтний бустінг об'єднує кілька базових алгоритмів в композицію). Зазвичай, в якості базового алгоритму використовуються «Вирішальні дерева». Вирішальне дерево - бінарне дерево (в загальному випадку), яке в вузлах містить прості предикати, а в листі прогнози (рис. 1.3.3).

Кожне вирішальне дерево будується «автоматичні». На кожній ітерації алгоритм вибирає такий предикат, який максимально добре розділяє

класи. Якщо говорити простими словами - якщо ніс одну людину сильно відрізняється від носа іншої людини - алгоритм, швидше за все, вибере цю характеристику для першого розбиття. Далі алгоритм буде шукати вже менш явні залежності.

Процес побудови і, зокрема, вибору оптимального розбиття докладно описаний в книзі К. Бішопа «Pattern Recognition and Machine Learning».

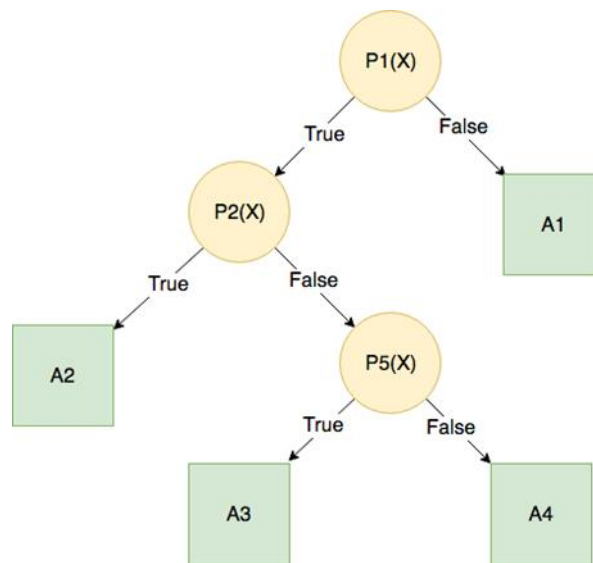


Рис. 3.3.3 Вирішальне дерево

Тепер необхідно описати один із способів об'єднання дерев в композиції. Метод найшвидшого бустінг - спрямоване побудова композиції. Іншими словами - ми будемо не один алгоритм (модель), а кілька. Побудова відбувається наступним чином - будуватися перша модель, обчислюється її помилка, потім відбувається побудова другої моделі таким чином, щоб вона виправляла помилки попереднього алгоритму. Таким чином ми отримуємо непогану якість класифікації, так як кожна нова модель в композиції виправляє помилку попередньої.

Розглянемо наочно процес виправлення помилки одного алгоритму іншими. Приклад, на якому буде розглянута робота градієнтного бустінгу не відноситься до нашого завдання, однак на ньому зручно демонструвати роботу алгоритму.

На рисунку 1.3.4 зображений процес виправлення помилки за допомогою спрямованого побудови композиції. Перший графік - справжня залежність. Другий графік (tree 1) - перший алгоритм, який намагається вловити залежність. Третій графік - алгоритм, який намагається виправити помилку попереднього. Четвертий графік - алгоритм, який намагається виправити помилку двох попередніх.

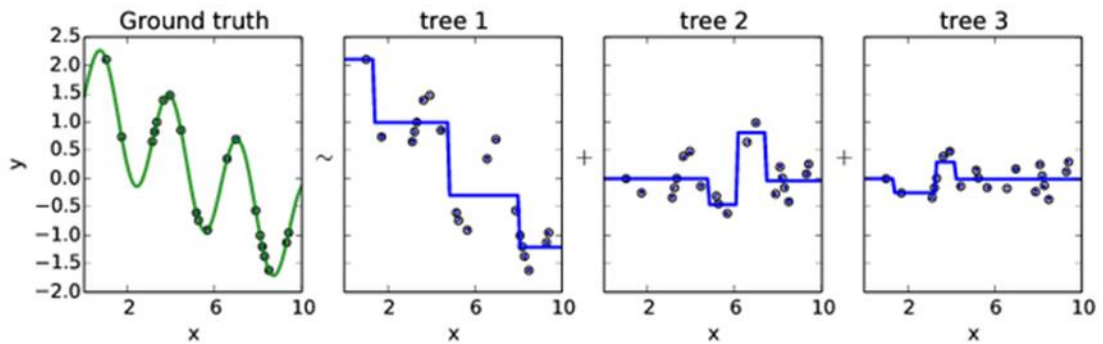


Рис. 3.3.4 Результат градієнтного бустінгу

Висновок. (Розділ 3)

В якості алгоритму пошуку осіб нами був обраний алгоритм «Каскади Хаара». В якості алгоритму класифікації був обраний «Метод найшвидшого бустінг». Використовуючи ці алгоритми в зв'язці, можна досягти прийнятної якості розпізнавання і класифікації. Крім того, ці алгоритми працюють досить швидко.

Розділ 4. Підбір апаратного та програмного забезпечення.

4.1 OpenCV

OpenCV (англ. Open Source Computer Vision Library, бібліотека комп'ютерного зору) - бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Реалізована на Python , також розробляється для C ++, Java, Ruby, Matlab, Lua та інших мов. (В нашому випадку використовується для Python) Її можна вільно використовувати як в академічних, так і в комерційних цілях - поширюється в умовах ліцензії BSD [5].

За допомогою даної бібліотеки ми можемо розпізнавати різні об'єкти в будь-якому потоці. (В нашому випадку ми розпізнаємо особи учнів)

4.2 Raspberry Pi 3

Raspberry Pi - одноплатний комп'ютер розміром з банківську карту, встановлений в місці контролю потоку викладачів. Він дозволяє виконувати багато функцій, доступні потужним настільним системам. Це пристрій виконує практично всі функції комп'ютера і навіть більше.

Raspberry володіє великими графічними можливостями і його можна використовувати, як платформу для розробки ігрових додатків. З іншого боку, його можна використовувати для розробки вимірювальних і робототехнічних систем, застосовуючи датчики і виконавчі механізми. Пристрій представлений на рисунку 4.1 .

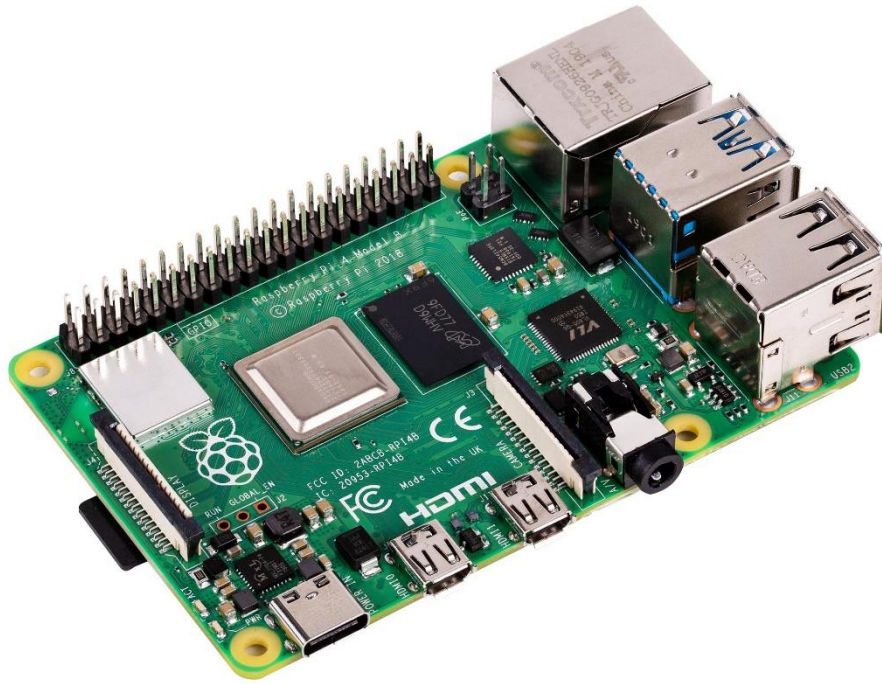


Рис. 4.1 Raspberry Pi 3

Для Raspberry Pi створюються спеціальні версії систем. Всі ОС засновані на дистрибутивах Linux. Є навіть спеціальні версії ArchLinux і Kali Linux для Raspberry Pi 2 [8]. Установка ОС на плату здійснюється за допомогою карти пам'яті microSD. В останній версії цього пристрою при бажанні цілком реально використовувати в якості системи ОС Ubuntu і навіть Microsoft Windows 10.

На Raspberry Pi 3 встановлений 64-х бітний чотирьохядерний процесор ARM Cortex-A53 з тактовою частотою 1,2 ГГц на ядро в складі однокристальної платформи Broadcom BCM2837.

Даний чіп забезпечує приріст продуктивності на 50-60% в порівнянні з Raspberry Pi 2 і майже десятикратне перевага перед першим Raspberry Pi. Завдяки цьому комп'ютер відкриває ще більше можливостей для вбудованих проєктів.

Raspberry Pi 3 Model B наділили 1 ГБ оперативної пам'яті, але ця пам'ять ділиться з графічною підсистемою.

Графічний двоядерний процесор VideoCore IV підтримує стандарти OpenGL ES 2.0, OpenVG, MPEG-2, VC-1 і здатний кодувати, декодувати і виводити Full HD-відео (1080p, 30 FPS, H.264 High-Profile).

Все вище перераховане дозволяє використовувати Raspberry Pi 3 як сервер, на якому будуть оброблятися всі зображення

4.3 Raspberry Pi Wide Camera .

До Raspberry Pi 3 підключається спеціальна камера Raspberry Pi Wide [8] camera (пристрій представлено на рисунку 2.3), за допомогою якої зчитується зображення потоку людей. Камера призначена для зйомки в освітлених приміщеннях.



Рис. 4.3 Raspberry Pi camera

Камера підключається до керуючої платі одним шлейфом - через нього йде обмін інформацією та подається харчування. Для підключення модуля використовується відеовхід CSI (Camera Serial Interface), що значно знижує навантаження на центральний процесор в порівнянні з підключенням камер по USB.

Велика частина проекту написана на мові Python , так як він має підтримку всіх необхідних бібліотек і дуже зручний для написання проектів, пов'язаних з використовуваними технологіями [6].

Висновок. (Розділ 4)

Для розробки системи відеоспостереження був обраний мікрокомп'ютер Raspberry PI 3B+ із камерним модулем. Для розпізнавання обличчя була обрана безкоштовна бібліотека OpenCV яка містить вбудовані класи та функція для розпізнавання обличчя.

Розділ 5. Розпізнавання обличь людиною. Розпізнавання обличь машиною. Метод розпізнавання обличчя за ознаками Хаара. Метод розпізнавання Віюли-Джонса.

5.1 Розпізнавання обличь людиною

Людина здатна ідентифікувати об'єкти навколишнього світу протягом десятків мілісекунд. Така висока швидкість розпізнавання предметів можлива тому, що наш мозок постійно створює припущення стосовно того, що знаходиться в полі зору, і порівнює ці прогнози з надходить ззовні.

Існує три основних етапи в процесі розпізнавання людських облич:

- 1) визначення фізичних характеристик особи, на яке ми дивимося;
- 2) визначення особистості людини, в ході якого ми розуміємо, чи знайомий нам та чи інша людина;

3) ми розпізнаємо людини, але все ще не знаємо, чи відомо нам його ім'я.

Дослідники виявили, що на кожному етапі активуються певні ділянки мозку.

Розпізнавання осіб, як відзначають психологи, - це процес, більше пов'язаний з когнітивною стороною сприйняття. Справа в тому, що, по суті, розпізнавання осіб, відбувається наступним чином: мозок постійно порівнює те, що бачить, з тим, що зберігається в його довготривалої пам'яті. Як не дивно, майже всі алгоритми розпізнавання конкретних осіб, вже закладених в їх основу, працюють саме так.

Наприклад, коли ми будемо дивитися на такий об'єкт, як годинник, ми визнаємо його за рахунок порівняння того, що ми бачимо, і тих характеристик,

які притаманні ментального образу годин. Хоча не всі годинники однакові і деякі копії цього об'єкта можуть відрізнятися від ментального прототипу, у кожного годинника є набір властивих тільки йому ключових характеристик, таких як хвилинна і годинна стрілка, циферблат, які і допомагають нам розпізнавати його. Потім образ об'єкта класифікується і

зберігається в пам'яті під певною категорією. Чим більше різновидів годин запам'ятав наш мозок, тим простіше йому буде розпізнати новий об'єкт.

Класифікація об'єкта - це етап, на якому, як правило, процес розпізнавання добігає кінця, але в разі розпізнавання осіб - це тільки початок. Якщо годинник досить визнати годинами, то людське обличчя недостатньо просто визнати людським обличчям. Практично відразу ж ми визначаємо стать і вік людини, його расу і навіть то, подобається він нам чи ні. Крім того, ми тут же визначаємо, чи знайоме нам це особа. Якщо людина нам знаком, ми тут же починаємо отримувати з цього інформацію, як і алгоритм розпізнавання осіб. Він може визначити конкретну людину, після чого видати доступ до інформації, відправити сигнал в кримінальні служби, в залежності від того, для чого створений.

5.2 Розпізнавання обличчя машиною

Завдання ідентифікації і розпізнавання осіб - це одна з перших практичних завдань, яка стимулювала становлення і розвиток теорії розпізнавання та ідентифікації об'єктів. Існує дев'ять категорій об'єктів, які відповідають гностичним областям і викликають зорові образи:

- 1) об'єкти, якими можна маніпулювати;
- 2) об'єкти, якими можна частково маніпулювати;
- 3) об'єкти не маніпульовані;
- 4) особи;
- 5) вирази обличчя;
- 6) живі істоти;
- 7) друковані знаки;
- 8) рукописні зображення;
- 9) характеристики і розташування джерел світла.

Інтерес до процедур, які лежать в основі процесу впізнавання і розпізнавання осіб, завжди був значним, особливо в зв'язку зі зростаючими

практичними потребами: охоронні системи, верифікація, криміналістична експертиза, телеконференції і т.д. Незважаючи на ясність того життєвого факту, що людина добре ідентифікує обличчя людей, зовсім не очевидно, як навчити ЕОМ проводити цю процедуру, в тому числі як декодувати і зберігати цифрові зображення осіб. Ще менш ясними є оцінки схожості осіб, включаючи їх комплексну обробку. Можна виділити кілька напрямків

досліджень проблеми розпізнавання осіб:

- 1) нейропсихологічні моделі;
- 2) нейрофізіологічні моделі;
- 3) інформаційно - процесуальні моделі;
- 4) комп'ютерні моделі розпізнавання.

Проблема розпізнавання осіб розглядалася ще на ранніх стадіях комп'ютерного зору. Ряд компаній протягом більше 40 років активно розробляють автоматизовані, а зараз і автоматичні системи розпізнавання людських облич: Smith & Wesson (система ASID - Automated Suspect Identification System); ImageWare (система FaceID); Imagis, Epic Solutions, Spillman, Miros (система Trueface); Vissage Technology (система Vissage Gallery); Visionics (система FaceIt).

Для вирішення задачі розпізнавання осіб були запропоновані різні методики, серед яких можна виділити підходи, засновані на нейронних мережах, на розкладанні Карунена - Лоева, на алгебраїчних моментах, лініях однакової інтенсивності, еластичних (деформуються) стандартах порівняння. У розробках алгоритмів розпізнавання особливі зусилля спрямовані на автоматичне виділення елементів особи (очі, ніс, рот, підборіддя і ін.) На його різних зображеннях: фас, профіль і довільний ракурс. Далі ці геометричні характеристики використовуються в рішенні задачі розпізнавання. Типовим при описі цих підходів є відсутність порівняння на статистично значущою базі даних осіб. Можна виділити два способи розпізнавання осіб:

- 1) порівняння типу відповідності між стимулами один проти одного;

2) порівняння між накопиченим, репрезентативним рядом осіб.

Геометричне порівняння, засноване на визначенні елементів особи - Елементи особи: очі, ніс, рот, підборіддя і ін. Особа може бути розпізнано, навіть коли індивідуальні Елементи особи видно недостатньо. Ідея підходу полягає в знаходженні відносного положення і власних характеристик окремих Елементи особи. Було показано, що

якщо навіть Елементи особи витягуються вручну, то комп'ютерне розпізнавання дає дуже хороші результати.

Еталонне порівняння побудовано на ідеї, що зображення, представлене у вигляді масиву байтів - величин інтенсивності, порівнюється в підходящій метриці з еталоном - цілим обличчям. Існують кілька шляхів підготовки еталонів і їх уявлення. Кілька еталонів використовуються для розпізнавання з різних ракурсів.

Заслуговує на увагу підхід, коли особа представляється у вигляді набору малих різних еталонів. Кращим і більш комплексним підходом є шлях у використанні єдиного зразка спільно з точною апріорної моделлю, яка дозволяє оцінити трансформацію основного особи, при зміні ракурсу спостереження. Деформируемая модель потім використовується в побудові метрики порівняння еталонних осіб. Дана ідея є основою методики деформуються еталонів.

Схема еталонного порівняння в роботі Bruce V. досить сильно модифікована, щоб називати її кореляційно - екстремальною. Вона використовує нормалізацію зображення, яке переводить його в карту величин градієнтів і є вільною від карти країв. Одним з успішних знахідок є використання декількох дозволів і малих за розмірами еталонів для очей, рота і носа. На цих підходах побудовані детектори елементів особи. Важливо відзначити, що наступний крок є конструктивним: спочатку детектувати очі (шляхом еталонного порівняння), потім автоматично нормалізувати зображення за масштабом і орієнтації.

Можна помітити, що такий підхід містить елементи розпізнавання на основі еталона всього особи: ЕЛ (очі) використовуються для нормалізації зображення, і еталонне порівняння проводиться окремо по окремим характерним рисам обличчя (очі, ніс, рот). Однак як показали експерименти, найуспішніше розпізнавання особи відбувається на архітектурі, комбінує підхід розпізнавання всього особи з підходом на основі еталонного порівняння його елементів.

5.3 Методи розпізнавання обличчя.

Для початку давайте розберемося, як розпізнати обличчя на фотографії. По-перше, потрібно знайти, де на фото розташовано особу людини і не сплутати його з годинником на стіні і кактусом на підвіконні. Здавалося б, просте завдання для людини, виявляється не такою простою для комп'ютера. Для того, щоб знайти особу ми повинні виділити його основні компоненти, такі як ніс, лоб, очі, губи. Для цього будемо використовувати шаблони, зображені на рисунку 5.1.

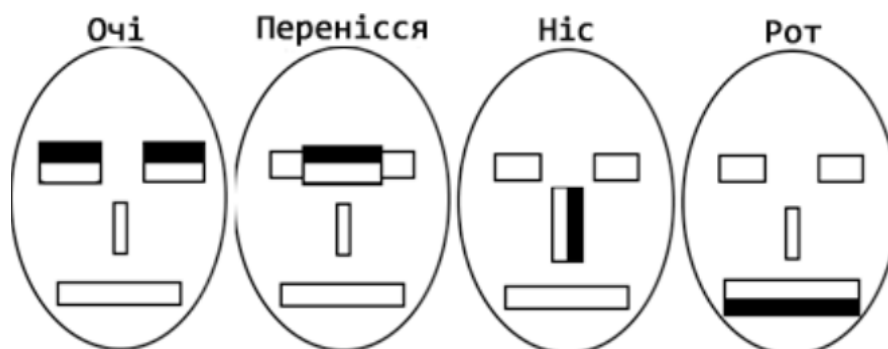


Рис. 5.1 – Основні шаблони.

5.3.1 Ознаки Хаара

Для початку я б хотів розглянути ознаки Хаара і їх види. Прямокутні ознаки Хаара. Найпростіший прямокутний ознака Хаара можна визначити як різниця між сумою пікселів двох суміжний областей всередині прямокутника,

який може займати різні положення і масштаби на зображенні. Такий вид ознак називається 2-прямокутним. Віола і Джонс також визначили 3-прямокутні і 4-прямокутні ознаки. Кожна ознака може показати наявність (або відсутність) будь-якої конкретної характеристики зображення, такий як

кордону або зміна текстур. Наприклад, 2-прямокутний ознака може показати, де знаходиться межа між темним і світлим регіонами.

Похилі ознаки Хаара. Лінхарт і Майд представили ідею нахилених під 45° ознак Хаара. Це було зроблено для збільшення розмірності простору ознак. Спосіб виявився вдалим і деякі похилі ознаки були здатні краще описувати об'єкт. Наприклад, 2-прямокутний похилий ознака Хаара може показати наявність краю, нахиленого на 45 градусів. Мессі і Барзаков доповнили концепцію похилих ознак Хаара. Хоч ідея і є математично вірною, на практиці при використанні ознак під різними кутами виникають проблеми. Для прискорення обчислень, детектор використовує зображення низького дозволу, що призводить до помилки округлення. Виходячи з цього, похилі ознаки Хаара зазвичай не використовуються.

5.3.2 Метод Віоли-Джонса

Зазвичай у кожного методу є основа, то, без чого цей метод не міг би існувати в принципі, а вже над цією основою будується вся інша частина. У методі Віоли-Джонса цю основу складають примітиви Хаара, що представляють собою розбивку заданої прямокутної області на набори різнотипних прямокутних підобластей, зображених на рисунку 2:

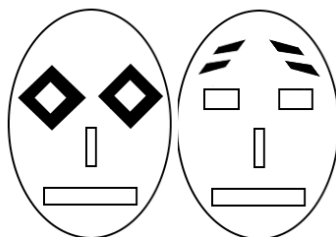


Рис 5.2. Шаблони з поворотом

В оригінальній версії алгоритму Віоли-Джонса використовувалися тільки примітиви без поворотів, а для обчислення значення ознаки сума яскравостей пікселів одній підобласті віднімалася з суми яркостей іншої підобласті. У розвитку методу були запропоновані примітиви з нахилом на 45 градусів і несиметричних конфігурацій. Також замість обчислення звичайної різниці, було запропоновано приписувати кожній підобласті певне політичне значення і значення ознаки обчислювати як зважену суму пікселів різнотипних областей, використовуючи формулу 5.1:

$$feature = \sum_{i \in I=1, \dots, N} w_i \cdot RectSum(r_i) \quad (5.1)$$

Складність обчислення ознаки так само як і отримання значення пікселя залишається $O(1)$: значення кожної підобласті можна обчислити скомбінувавши 4 значення інтегрального уявлення (Summed Area Table - SAT), яке в свою чергу можна побудувати заздалегідь один раз для всього зображення за $O(n)$, тут n - число пікселів в зображенні, використовуючи формулу 5.2.

$$\begin{aligned} SAT(x, y) &= SAT(x, y-1) + SAT(x-1, y) + I(x, y) - SAT(x-1, y-1) \\ SAT(-1, y) &= SAT(x, -1) = SAT(-1, -1) = 0 \end{aligned} \quad (5.2)$$

Це дозволило створити швидкий алгоритм пошуку об'єктів, який користується успіхом вже більше десятиліття. Але повернемося до наших ознак. Для визначення приналежності до класу в кожному каскаді, перебувати сума значень слабких класифікаторів цього каскаду. Кожен слабкий класифікатор видає два значення в залежності від того більше або менше заданого порогу значення ознаки, що належить цьому класифікатором. В кінці сума значень слабких класифікаторів порівнюється з порогом каскаду і виноситься рішення знайдений об'єкт чи ні даними каскадом.

5.4 Використання ознак Хаара

Якщо шаблони, вони ж примітиви, відповідають конкретним областям на зображенні, будемо вважати, що на зображенні є людське обличчя. Для кожного з них вважається різниця між яскравістю білої і чорної областей. Це значення порівнюється з еталоном і приймається рішення про те, чи є тут частина людського обличчя чи ні. Цей метод називається методом Віюлі-Джонса (так само відомий як каскади Хаара). Давайте уявимо, що у нас на фотографії не одне велике обличчя, а багато дрібних. Якщо застосувати шаблони до всієї зображенні ми не знайдемо там осіб, тому що вони будуть менше шаблонів. Для того щоб шукати на всіх фото особи різних розмірів використовується метод ковзного вікна. Саме всередині цього вікна і вираховуються примітиви. Вікно як би ковзає по всьому зображенню. Після кожного проходження зображення вікно збільшується, щоб знайти особи більшого масштабу.

Знайдено особа на фотографії, але для визначення конкретної людини потрібно виконати ще кілька дій. Для вирішення цього завдання будемо використовувати алгоритм Local Binary Patterns. Суть його полягає в тому, що ми розбиваємо зображення на частини і в кожній такій частині кожен піксель порівнюється з сусідніми 8 пікселями. Якщо значення центрального пікселя більше сусіднього, то пишемо 0, в іншому випадку. І так для кожного пікселя у нас виходить деяке число. Далі на основі цих чисел для всіх частин, на які ми розбивали фотографію, вважається гістограма. Все гістограми з усіх частин об'єднуються в один вектор характеризує зображення в цілому. Якщо ми хочемо дізнатися наскільки схожі дві особи, нам доведеться обчислити для кожного з них такий вектор і порівняти їх, обчислення вектора зображено на рисунку 3.

5	8	1		1	1	0
3	4	6		0		1
7	1	3		1	0	0

Рис 5.3. Розрахунок важелів LBP

Висновок. (Розділ 5)

Були розглянуті алгоритми розпізнавання обличчя за ознаками Хаара та методом Віюлі-Джонса, який базується на пошуку незмінних ознаках обличчя, таких як позиція рота, носу, очей, тощо. та будуванні матричної сітки для задання значення важелів. За допомогою порівняння основних ознак, можна з великою точністю порівняти обличчя із двох різних фотографій, та отримати ймовірність тотожності.

Також цей спосіб не залежить від віку людини, тому порівняння однієї людини із самою собою з різницею 10 років буде давати результат аналогічної точності.

Розпізнавання за ознаками Хаара не потребує великих апаратних витрат, та може бути використаний навіть на мікроконтролері. Саме тому він і був обраний у даній дипломній роботі.

Розділ 6. Розпізнавання обличчя людиною за допомогою бібліотеки OpenCV.

6.1 Можливі підходи реалізації

Ми розглядали 2 варіанти виконання:

1. Варіант з використання бібліотеки OpenCV .
2. Реалізація бібліотеки самостійно.

Нами був обраний 1 варіант через декілька причин. По-перше бібліотека OpenCV вільно поширюється [4]. По-друге, крім розпізнавання осіб, бібліотека надає побічні функції, корисні для налагодження - отрисовка вікон, отрисовка геометричних примітивів навколо осіб і т.д.

Важливо сказати, що класифікатори, які вбудовані в OpenCV добре оптимізовані для роботи з неякісними даними (погане освітлення і інший шум в даних). Тому реалізовувати таку ж бібліотеку з нуля своїми силами недоцільно. OpenCV - бібліотека, яку виправляють і налагоджують вже багато років, написати схожу власну бібліотеку - трудомістка і дорога процедура. Немає сенсу витратити додатковий час і фінанси, якщо в OpenCV вже реалізовані всі необхідні функції.

6.1.1 Короткий опис бібліотеки OpenCV

Наведемо що таке OpenCV і які функції і класи з цієї бібліотеки були використані. OpenCV - бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом [5].

6.1.2 Опис класів і функцій

Розглянемо функції і класи з бібліотеки, які були використані.

1. Додавання нового обличчя
 - а. для захоплення зображення з камери був створений екземпляр класу VideoCapture .

- b. для виявлення осіб на захопленому зображенні був використаний метод екземпляра класу `CascadeClassifier.detectMultiScale`.
- c. для тренування класифікатора була використана метод `train` екземпляра класу `LBPFFaceRecognizer`.
- d. для відтворення геометричних примітивів навколо осіб (під час налагодження) була використана функція `rectangle`.
- e. для звільнення камери була використана функція `release`.

2. Розпізнавання відомих / невідомих осіб

- a. для використання класифікатора були використані методи `read` і `predict` екземпляра класу `LBPFFaceRecognizer`.

6.1.3 Спільний доступ функцій до ресурсів камери

Так само було необхідно вирішити проблему доступу до камери. Перед тим як записати нове обличчя в класифікатор, необхідно звільнити камеру, тому що вона безперервно працює в режимі спостереження. Ця проблема була вирішена наступним чином: був створений окремий модуль для постійної роботи камери (запускається паралельно з ботом) і, якщо необхідно, отримати доступ до камери з іншої функції - паралельно запущений скрипт припиняє свою роботу. Якщо всі функції, запит доступ до камери, відпрацювали, скрипт постійного спостереження запускається заново.

6.1.4 Способи зберігання даних

Раніше було сказано, що створений сервіс повинен "запам'ятовувати" імена людей, особи яких були зафіксовані. Були розглянуті наступні способи зберігання даних:

- SQL база даних
- NoSQL база даних
- Текстовий файл (.txt)

Було прийнято рішення не використовувати SQL / NoSQL бази даних. Так само було прийнято рішення не використовувати популярні мови розмітки (XML , JSON). Структура даних, які необхідно зберігати, максимальна проста - це просто список імен без будь-яких параметрів. У такій ситуації текстовий файл - найпростіше рішення.

Наведемо ряд аргументів на користь тестових файлів. По-перше, немає необхідності налаштовувати і проектувати SQL / NoSQL бази даних для зберігання малої кількості імен без будь-яких параметрів. По-друге, немає сенсу використовувати популярні мови розмітки (XML, JSON), так як імена зберігаються без будь-яких параметрів.

В результаті було прийнято рішення, що текстовий файл - кращий з варіантів в поточній ситуації.

Тепер необхідно описати структуру текстового файлу, в якому зберігаються імена. Всього в програмі використовується два текстові файли:

- Файл з іменами всіх студентів - user _ names . txt
- Файл з іменами зафіксованих студентів - current _ names . txt

Файл « user _ names . txt »зберігає імена всіх відомих класифікатором студентів.

Файл « current _ names . txt »зберігає імена студентів, особи яких були зафіксовані камерою.

Обидва файли мають однакову структуру, схема якої приведена на рисунку 6.1.4.1.

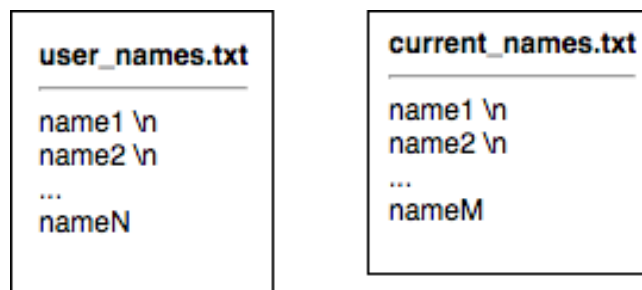


Рис. 6.1.4.1 Структура використовуваних txt -файлів

6.1.5. Структура лог файлів

Під час роботи програми, всі дії з камерою логіруються в консоль. Це необхідно для відстежуючи коректності роботи програми.

Кожен лог починається з ключового префікса [INFO]. Після префікса нахоодітся саме лог повідомлення.

Наведемо список лог повідомлень:

При запуску постійної роботи камери
- *Camera handler subprocess is running* .

Переривання постійної роботи камери - *Camera handler subprocess terminating for a while* .

Також існують лог повідомлення, що повідомляють про додавання / видалення нового обличчя і про фіксацію камерою будь-якого людини. Опустимо опис структури подібних лог файлів, тому що вони являють собою просте речення англійською мовою.

6.2 Схема реалізації

Система повинна робити вибірку даних за певними критеріями з бази даних, компонувати дані в певній формі та відправляти дані користувачеві. Вибірка і компоновка відбуваються на Raspberry Pi 3, а дані відправляються за коштами додатки Telegram , за допомогою бота, налаштованого спеціально для даного проекту. У базі даних повинні зберігатися: прізвища всіх учнів і прізвища учнів, які перебувають в тій чи іншій контрольованій області. Загальна схема проекту представлена на рисунку 4.2.1

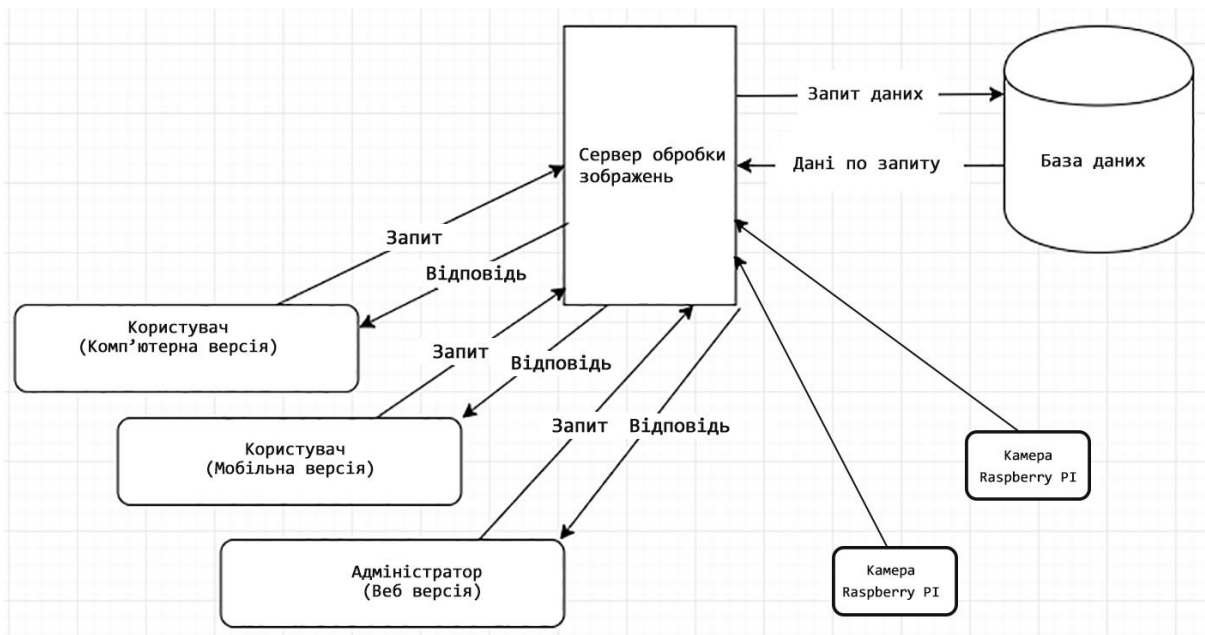


Рис. 6.2.1 Спрощена схема реалізації

Система виконана у вигляді клієнт-серверної архітектури. Під сервером розуміється Raspberry Pi 3 з приєднаною камерою, а під клієнтом всі користувачі, які звертаються до нашого сервера для отримання різного роду інформації. База даних розташовується на Raspberry Pi 3. На схемі видно, що всі користувачі можуть взаємодіяти з сервером через запити, що в нашому проекті реалізовано через додаток Telegram (всі доступні його реалізації, включаючи комп'ютерну, мобільну і веб-версії).

Висновок. (Розділ 6)

OpenCV - Бібліотека комп'ютерного зору і машинного навчання з відкритим вихідним кодом. У неї входять понад 2500 алгоритмів, в яких є як класичні, так і сучасні алгоритми для комп'ютерного зору і машинного навчання. Ця бібліотека має інтерфейси на різних мовах, серед яких є Python.

Був описаний підхід до реалізації пристрою відеоспостереження та організації структури класів та функцій. Для зберігання даних буде використовуватись звичайний файл для економії апаратних ресурсів.

Розділ 7. Охорона праці

Даний дипломний проект описує проектування та програмування системи відеоспостереження із можливістю розпізнавання обличчя на основі мікрокомп'ютеру Raspberry Pi. Пропонований пристрій буде використовуватись в приміщенні або на вулиці спроектованих відповідно вимогам СНІП 2.09.04-87 "Адміністративні й побутові будинки".

Так як, в нинішній час схеми електричні принципів розробляються й моделюються за допомогою мікрокомп'ютеру, приміщення, де проводилася розробка проекту, оснащена чотирма персональними комп'ютерами, офісним устаткуванням, шафами для зберігання паперів. Воно являє собою окрему кімнату загальною площею 32м² і дозволяє розмістити в ній 4 інженера (по 8 м² на одного чоловіка), обсяг 27 м³ на одного інженера-розробника. Аналіз умов праці і розробка заходів з охорони праці здійснюються для інженера електронної техніки.

7.1 Перелік небезпечних та шкідливих факторів у робочій зоні

В процесі праці, на інженера згідно ГОСТу 12.0.003-74 впливають наступні небезпечні та шкідливі фактори [9]:

несприятливий мікроклімат;

Згідно ГОСТу 12.1.005–88 про «Загальні санітарно-гігієнічні вимоги до повітря робочої зони» в виробничих приміщеннях при виконанні робіт пов'язаних з нервово-емоційною напругою, повинні дотримуватися оптимальні величини температури повітря 22-24 ° С, при його відносній вологості 60-40% і

швидкості руху (не більше 0,1 м / с) [21].

запиленість робочої зони;

Основним параметром складу повітряного середовища є запиленість. Пил у робочому приміщенні являє собою аерозоль дезінтеграції, утворену в

результаті механічного здрібнювання часток ґрунту, що заносять у приміщення на взутті й одязі. Запиленість повітря в приміщенні не перевищує 0.1 мг/м² при розмірах часток не більше 3 мкм, що відповідає нормі. Зменшувати рівень запиленості потрібно вологим прибиранням приміщення.

- виробничі випромінювання;

При роботі з ПЕОМ інфрачервоним, ультрафіолетовим й іншим видами випромінювань, опромінюється до 50% поверхні тіла користувача ПЕОМ, згідно ГОСТу 12.1.005-88 інтенсивність теплового випромінювання повинна становити не більше 70 Вт/ м². [21]

- підвищений рівень шуму, що створюється технічним оснащенням;

Основні характеристики і допустимі рівні шуму на робочих місцях визначаються згідно з ДСН 3.3.6.037-99 "ССБТ. Шум. Загальні вимоги безпеки.". Допустимі рівні звукового тиску звуку для аналітичних та вимірювальних робіт – 60 ДБА [9].

- недостатня освітленість робочої зони .

7.2 Технічні та організаційні заходи щодо зменшення впливу небезпечних та шкідливих виробничих факторів

Перераховані вище небезпечні та шкідливі фактори в тій чи іншій мірі впливають на самопочуття людини, що призводить до зниження працездатності і продуктивності праці. Заходи, що до зменшення впливу шкідливих факторів:

дотримання норм і правил згідно ГОСТу 12.1.005-88 про «Загальні санітарно-гігієнічні вимоги до повітря робочої зони»;

вологе прибиранням приміщення;

забезпечення від інфрачервоних випромінювань: огорожувальними, герметизуючими, теплоізолюючими пристроями, знаками безпеки, дистанційним управлінням;

раціоналізація режимів праці та відпочинку (додаткові перерви);

зменшення шуму на шляху поширення (за рахунок місцевої та загальної звукоізоляції, шумовловлюючих екранів, поглинаючих фільтрів).

7.2.1 Забезпечення освітлення робочої зони

Вимоги до освітлення наведені у ДБН В.2.5-28-2006 «Природне і штучне освітлення». Збереження зору людини, стан його центральної нервової системи значною мірою залежить від освітлення [10].

Розрахунок освітленості робочого місця зводиться до вибору системи освітлення, визначенню необхідного числа світильників, їхнього типу й розміщення. У приміщенні використовується система загального штучного освітлення з використанням люмінесцентних джерел світла у світильниках загального освітлення. Як світильники загального освітлення використовуються люмінесцентні лампи типу ЛПО71-4x18-571/Мілано із індексом передачі кольору не менш 70 ($R \geq 70$), як світильники - установки типу УСП-35-4x18 з можливістю плавного регулювання яскравістю.

Нормативними документами встановлено вимоги освітлення робочого місця інженера, так як рекомендована освітленість на поверхні столу в зоні розміщення робочого документу 300 - 500 лк. Місцеве освітлення не повинно створювати відблисків на поверхні екрану і збільшувати освітленість екрана понад 300 лк.

Джерелами штучного світла є люмінесцентні лампи. Їх перевагами є: економічність, великий термін служби, рівномірне освітлення в полі зору, спектр випромінювання близький до спектру природного кольору.

У приміщенні, де знаходиться робоче місце інженера використовується змішане освітлення, тобто сполучення природного і штучного освітлення. У

якості природного бічне освітлення через вікна. Штучне освітлення використовується при недостатньому природному освітленні. Тип світильника: підвісний растровий світильник для громадських приміщень ЛП071-4x18-571/Мілано з умовним номером групи 1.

Для проектування системи штучного освітлення скористаємося методом світлового потоку, оскільки будемо розраховувати рівномірне загальне освітлення приміщення при освітленості тільки в горизонтальній площині.

Висота приміщення:

$$H=3\text{ м};$$

Висота робочої поверхні столу:

$$h_p=0.8\text{ м};$$

Висота звісу світильника від стелі:

$$h_z=0.1\text{ м};$$

Визначаємо висоту звісу світильника над робочою поверхнею:

$$b=H-h_p-h_z$$

$$b=3-0,8-0,1=2,1\text{ м}$$

Ширина приміщення:

$$A=4\text{ м};$$

Довжина приміщення:

$$B=8\text{ м};$$

Визначаємо освітлювану площу приміщення:

$$S=A\cdot B$$

$$S=4\cdot 8=32\text{ м}^2$$

Визначаємо індекс приміщення:

$$i=\frac{S}{h\cdot(A+B)}$$

$$i=\frac{32}{2.1\cdot(4+8)}=1.26$$

Коефіцієнти відбиття від стелі 70%, від стін – 50%, від підлоги – 10%.

Вибираємо характеристики, що відповідають типу лампи – ЛЕЦ(754) напруга 110 +/- 11В, тривалість горіння – 5200 годин, світловий потік після 100 годин горіння 4000 лм.

Характеристики вибраного світильника:

довжина $D_{\text{св}} = 625 \text{ мм}$

ширина $Ш_{\text{св}} = 625 \text{ мм}$

висота $B_{\text{св}} = 100 \text{ мм}$

чотири лампи типу ЛЕЦ(754) потужністю по 18 Вт кожна. Таким чином, світловий потік:

$\Phi = 4000 \text{ лм};$

З урахуванням індексу приміщення, коефіцієнтів відбиття стелі, стін, підлоги і типу обраного світильника визначаємо коефіцієнт світлового потоку:

$\eta = 40;$

Коефіцієнт затінення:

$K_3 = 1,4;$

Коефіцієнт затінення для даного приміщення з фіксованим положенням інженера-програміста:

$K_{\gamma} = 0,9;$

Коефіцієнт нерівномірності освітлення:

$Z = 1,3;$

Кількість люмінесцентних ламп в світильнику:

$n = 4;$

Освітленість:

$E = 300 \text{ лк};$

Розраховуємо потрібну кількість світильників:

$$N_c = \frac{100 \cdot E \cdot S \cdot K_3 \cdot Z}{n_i \cdot \Phi \cdot \eta \cdot K_{\gamma}}$$
$$N_c = \frac{100 \cdot 300 \cdot 32 \cdot 1,4 \cdot 1,3}{4 \cdot 4000 \cdot 40 \cdot 0,9} = 3,97$$

З розрахунків видно, що для даного приміщення з ПЕОМ потрібно N=4 світильників типу ЛГ1071-4x18-57.

При розміщенні світильників з люмінесцентними лампами, останні розміщуються рядами – паралельно рядам обладнання або віконним рамам. Перед проектуванням системи штучного освітлення потрібно перевірити фактичну освітленість у приміщенні при числі світильників 4:

$$E_{\phi} = \frac{N_c \cdot n_i \cdot \Phi \cdot \eta \cdot K_y}{100 \cdot S \cdot K_s \cdot Z}$$

$$E_{\phi} = \frac{4 \cdot 2 \cdot 3000 \cdot 40 \cdot 0.9}{100 \cdot 32 \cdot 1.4 \cdot 1.3} \approx 340 \text{ лк}$$

отримана фактична освітленість приміщення забезпечує рекомендовану освітленість приміщення.

7.3 Інструкція з техніки безпеки, пожежної та вибухової безпеки

7.3.1 Вимоги щодо забезпечення пожежної безпеки

Загальні вимоги до систем відвертання пожеж і пожежного захисту регламентуються спеціальною нормативно-технічною документацією.

По класифікації приміщень по пожежній небезпеці робоче місце інженера відноситься до категорії В, що характеризується наявністю твердих горючих і слабо горючих речовин і матеріалів, а також легкозаймистих матеріалів.

В зв'язку з цим можна виділити ряд мір що до пожежного захисту:

- 1) не палити і не використовувати нагрівальні прилади в приміщенні з ПЕОМ;
- 2) не від'єднувати і приєднувати кабелі, усувати несправності за наявності напруги в мережі;
- 3) не визначати наявність напруги в ланцюзі, замиканням клем.

В електронно-обчислювальній техніці пожежну небезпеку створюють прилади, що нагріваються, електро- і радіотехнічні елементи. Вони нагрівають навколишнє повітря і близько розташовані деталі і провідники. Все це може призвести до займання означених елементів, руйнування ізоляції і короткого замикання.

На випадок виникнення пожежі повинно бути передбачений один (з розрахунку не менш 2-х на кожні 20 м² площі) вуглекислотний вогнегасник ВВК-5.

Технологічні об'ємні поли виконуються з негорючих або слабо горючих матеріалів з межею вогнестійкості не менше 0.5 ч. Підпільні простори під об'ємними полами поділяють негорючими перегородками з межею вогнестійкості не менше 0.75 г на ділянки площею не більш 250 м².

7.3.2 Розробка інструкції з охорони праці

Інструкція з техніки пожежної та вибухової безпеки згідно з вимогами НПАОП 0.00-6.03-93 „Порядок опрацювання і затвердження власником нормативних актів про охорону праці, що діють на підприємстві”, затвердженим Держнаглядом охорони праці (наказ від 21 грудня 1993 р. № 132) та НПАОП 0.00-4.15-98 „Положення про розробку інструкцій з охорони праці”, затвердженим Держнаглядом охорони праці (наказ від 29 січня 1998 р. № 9) [23].
Загальні вимоги:

1) До роботи з приладом допускаються інженери, що вивчили проектний відділ, інструкцію з технічної експлуатації, діючу інструкцію, а також склали залік з техніки безпеки та пожежної безпеки;

2) Налагоджувально-ремонтні роботи виконує інженер – електронщик. Інструмент повинен бути справним, джерело живлення відключеним;

3) Для забезпечення заходів безпеки і виключення можливості ураження електричним струмом оператор повинен:

а) не порушувати порядок роботи з приладом, встановлений інструкцією з технічної експлуатації приладу;

б) підключати прилад до джерела живлення тільки при надітому корпусі;

в) перед підключенням приладу до джерела живлення перевірити візуально стан шнурів та кабелів;

г) з'єднувати і роз'єднувати кабельні та інші роз'єми тільки при виключеному живленні [22].

7.3.3 Вимоги безпеки перед початком роботи

До роботи з приладом допускається оператор, що вивчив об'єкт, інструкцію з технічної експлуатації, діючу інструкцію, а також склав залік з техніки безпеки та пожежної безпеки. Перед початком роботи необхідно:

7.3.3.1 Пересвідчитися, що прилад правильно підключений;

7.3.4.2 Пересвідчитися, що у робочій зоні немає зайвих предметів;

7.3.3.3 Оглянути з'єднувальні ланцюги і кабелі, місця рознімання і пересвідчитися, що вони справні;

7.3.3.4 У разі виявлення несправності приладу чи інструмента, яким користуються під час роботи, повідомити керівника роботи.

7.3.4. Вимоги безпеки під час виконання роботи

Під час виконання роботи необхідно:

7.3.4.1 Використовувати тільки справний термометр і тільки за призначенням;

7.3.4.2 Замінювати запобіжник тільки за номінальним струмом і тільки при вимкненому приладі;

7.3.4.3 При появі іскріння, короткого замикання, запаху гарі, диму, прилад негайно відключити та виявити причин можливого виникнення пожежі;

7.3.4.4 Періодично проводити перевірку електричної схеми приладу для виявлення причин, які можуть призвести до виникнення пожежі;

7.3.4.5 Прибрати з робочого місця зайві речі, які відволікають від роботи і можуть привести до ушкоджень інженера.

7.3.5. Вимоги безпеки після закінчення роботи

Після закінчення роботи необхідно:

7.3.5.1 Вимкнути прилад, коли оператор чи інженерно-технічний склад працівників залишають свої робочі місця;

7.3.5.2 Прибрати своє робоче місце;

7.3.5.3 Дотримуватись санітарних норм і правил особистої гігієни;

7.3.5.4 Повідомити керівника робіт про виявлені недоліки в роботі приладу.

7.3.6. Вимоги безпеки під час аварійних ситуацій:

7.3.6.1 В разі аварії негайно вимкнути всі електроспоживаючі прилади від електромережі не допускати сторонніх осіб в небезпечну зону; повідомити керівника робіт про те, що сталося.

7.3.6.2 У випадку виникнення пожежі негайно викликати пожежну команду. До її приїзду приступити до тушіння пожежі своїми силами, а також спасінню людей та надання їм допомоги;

7.3.6.3 При виникненні пожежі слід знеструмити прилад;

7.3.6.4 В обов'язковому порядку в приміщенні повинні бути засоби пожежегасіння;

Висновок. (Розділ 7)

Даний дипломний проект описує проектування та програмування системи відеоспостереження із можливістю розпізнавання обличчя на основі мікрокомп'ютеру Raspberry Pi. Пропонований пристрій буде використовуватись в приміщенні або на вулиці спроектованих відповідно вимогам СНІП 2.09.04-87 "Адміністративні й побутові будинки".

Також були проаналізовані усі небезпечні фактори при роботі з обладнанням, та описані правила та вимоги безпеки при роботі із пристроєм.

Розділ 8. Охорона навколишнього середовища

8.1 Вступ

Охорона навколишнього середовища регламентується законом «Про охорону навколишнього природного середовища України». На сучасному етапі щорічно в Україні утворюється близько 1 млрд. тонн твердих відходів виробництва та споживання. Тільки десята частина із них застосовується як вторинні матеріальні ресурси, а решта попадає в сховища. Тверді промислові відходи на сучасному етапі займають площу 1600 км², а загальний їх обсяг досяг 25 млрд. тонн, в тому числі 4,5 млрд. тонн високотоксичних. За обсягами утворення домінують токсичні відходи, які містять важкі метали (хром, свинець, нікель, кадмій, ртуть). За статистикою кожна людина в середньому за одну добу створює від 2 до 3 кг твердих побутових відходів. Тому вони являються одними з найбільш вагомих факторів забруднення довкілля і негативно впливають на самопочуття і здоров'я людей.

Ефективне вирішення цієї екологічної проблеми, яке пов'язане з негативним впливом твердих відходів на довкілля та здоров'я людей можливе тільки на основі послідовної реалізації Законів України «Про охорону навколишнього природного середовища», «Про відходи», «Про забезпечення санітарного та епідеміологічного благополуччя населення», Постанови Кабінету Міністрів України «Про затвердження порядку розроблення, затвердження і перегляду лімітів на утворення та розміщення відходів» та інших нормативно-правових актів, державних стандартів України з охорони навколишнього природного середовища, санітарних норм і правил.

Даний дипломний проект описує проектування та програмування системи відеоспостереження із можливістю розпізнавання обличчя на основі мікрокомп'ютеру Raspberry Pi.

8.2 Рекомендації щодо зниження негативних чинників

Для забезпечення зменшення шкідливого впливу на навколишнє середовища при виробництві, експлуатації, утилізації однієї електронної апаратури, ми повинні ще на етапі проектування віднести до основних критеріїв вибору конструкції – екологічну доцільність. До екологічної доцільності вибору технології виробництва можна віднести зменшення: відходів виробництва, використання природних ресурсів, електроенергії, шляхом застосування новітніх технологій.

Основним елементом живлення мікроконтролера являється акумуляторна батарея. Вона і є основним потенційним забруднювачем навколишнього середовища, оскільки при порушенні цілісності її корпусу на зовні можливий витік електроліту (у випадку використання кислотної акумуляторної батареї). Для запобігання забруднення навколишнього середовища електролітом, блок акумуляторних батарей виконаний в формі герметичного контейнеру.

Оскільки в цифровий мікроконтролер живиться напругою 5В то ми будемо використовувати акумуляторну батарею згідно статті № 17 ЗАКОНУ УКРАЇНИ «Про хімічні джерела струму» (цей закон визначає правові, організаційні та економічні засади у сфері поводження з хімічними джерелами струму і спрямований на суттєве покращення екологічної ситуації в Україні за рахунок зменшення впливу відпрацьованих хімічних джерел струму шляхом утилізації відпрацьованих хімічних джерел струму).

Хімічними джерелами струму прийнято називати пристрої, що виробляють електричний струм за рахунок енергії окислювально-відновних реакцій хімічних реагентів. Відповідно до експлуатаційної схеми і здатністю віддавати енергію в електричну мережу хімічні підрозділяються на первинні, вторинні і резервні, а також електрохімічні генератори. Первинні джерела струму (гальванічні елементи і батареї) допускають, як правило, однократне використання енергії хімічних реагентів. Окремі конструкції гальванічних

елементів і батарей вирішують короточасне повторне використання енергії реагентів після електричного заряджання. Позитивний (катод) і негативний (анод) електроди, розділені електролітом в рідкому або пастоподібному стані або ж пористою мембраною-сепаратором з поглиненням в ній електролітом, електрично зв'язані (гальванічний зв'язок) протягом всього терміну служби..

Вторинні джерела струму (окремі аккумулятори і аккумуляторні батареї) допускають багатократне (сотні і тисячі розрядних для заряду циклів) використання енергії складових хімічних реагентів. Електроди і електроліт весь термін служби аккумуляторів знаходяться в електричному контакті один з одним. Для збільшення ресурсу аккумуляторів в деяких специфічних умовах експлуатації розроблені способи сухозарядженого зберігання аккумуляторів. Такі аккумулятори перед включенням заздалегідь заливають електролітом.

Резервні джерела струму допускають лише однократне використання енергії хімічних реагентів. На відміну від гальванічних елементів і аккумуляторів, в резервних джерелах струму електроліт при зберіганні ніколи гальванічно не пов'язаний з електродами. Він зберігається в рідкому стані (у скляних, пластмасових або металевих ампулах) або в твердому (але неелектропровідному) поляганні в міжелектродних зазорах. При підготовці до роботи резервних джерел ампули руйнують стислим повітрям, вибухом, а кристали твердого електроліту розплавляють за допомогою електричного або піротехнічного розігрівання. Резервні джерела застосовують для живлення електричної апаратури, яка довгий час може (вимушена) знаходитися в резервному (непрацюючому) стані. Термін зберігання сучасних резервних джерел струму перевищує 10 – 15 років.

8.3 Аналіз джерел впливу та їх наслідків та на її оточення

Електричні джерела енергії за останнє століття поступово стали найбільш універсальними серед всіх винайдених людством джерел енергії. Головна перевага електричної енергії перед іншими формами (перш за все

теплової) – зручність доставки споживачеві, простота дозування і масштабування споживаючих пристроїв.

«Батарейки» в більшості випадків правильніше називати елементами, так як «електрична батарея» – це кілька електрохімічних елементів, з'єднаних між собою. Наприклад, пальчикова «батарейка» типорозміру AA або AAA – це елемент, а типорозміру «Крона» – справді батарея. Але в розмовній мові давно заведено називати і те і інше батарейками. Серед одноразових батарейок в останні роки безумовно домінують лужні (alkaline) елементи. Вони за всіма статтями обходять традиційні, які прийнято називати сольовими (їх можна відрізнити від лужних ще за маркуванням General Purpose – «загального застосування»), за винятком ціни: лужні того ж типорозміру рази в два-три дорожчі. Але разова економія коштів обертається збитками у перспективі: лужні мають в середньому в три рази більшу енергоємність (тобто в перерахунку на кожен отриманий джоуль енергії вони виявляються навіть трохи дешевшими), і значно довше зберігаються – до п'яти-семи років без принципової втрати ємності, що важливо в таких пристроях, наприклад, як телевізійні пульти, де основну частину часу батарейки просто лежать. Нарешті, вони можуть віддати більший одноразовий струм, і є ряд пристроїв, де звичайні сольові батарейки просто не «потягнуть».

Інший широко використовуваний тип одноразових елементів – літієві. Вони помітно дорожчі від лужних, але мають ще менший саморозряд (гарантія зазвичай становить близько 10 років), що й обумовлює основну область їх застосування: у якості резервних джерел живлення (це, наприклад, всім відомі «таблетки» типорозміру 2032 для комп'ютерних материнських плат, де вони забезпечують безперервний хід годинника і календаря, а також збереження налаштувань BIOS) і в якості елементів живлення дуже малоспоживаючих пристроїв, таких як наручний годинник.

Багаторазові акумулятори, на жаль, повністю замінити одноразові елементи не в змозі, перш за все через високий саморозряд. Будь-який

акумулятор (крім літій-іонних) має саморозряд в середньому не менше 5-10% на місяць, і якщо ви спробуєте поставити пальчикові акумулятори в телевізійний пульт, то, швидше за все, він не пропрацює і півроку, як їх доведеться перезаряджати.

По електрохімічній системі акумулятори діляться на декілька видів:

- свинцево-кислотні (Sealed Lead Acid, SLA);
- нікель-кадмієві (Ni-Cd);
- нікель--металогідридних (Ni-MH);
- літій-іонні (Li-Ion);
- литий-полимерные (Li-Pol).

Як не дивно, але одним з найпоширеніших типів акумуляторів досі є запущені у виробництво ще в позаминулому столітті свинцево-кислотні (СКА). СКА, хоч і стали в останні десятиліття герметизовані і необслуговувані, але все-таки досить незручні в застосуванні – наприклад вони «бояться» глибокого розряду («в нуль»), перезарядження, зберігання в незарядженому стані. Крім того, вони мають серед всіх поширених типів найнижчу енергоємність в розрахунку на одиницю маси – не більше 20-30 Вт-год/кг.

Тим не менше СКА дешеві, невибагливі, не бояться низьких температур і здатні віддати досить великий струм за короткий час. Тому вони лідирують у тих пристосуваннях, де потрібна значна кількість запасеної енергії в умовах великих перевантажень: у стартерних пристроях для автомобілів, в джерелах безперебійного живлення. До того ж СКА без належної утилізації вкрай шкідливі з точки зору забруднення навколишнього середовища.

В останні десятиліття з'явився удосконалений тип – SLA-акумуляторів (Sealed Lead Acid – «герметизовані свинцево-кислотні») з гелевим електролітом, які не бояться перевертання догори ногами і мають поліпшені характеристики: на деякі їх різновиди дають гарантію на 12 років безвідмовної роботи.

Нікель-кадмієві (Ni-Cd) акумулятори все менше використовують, в першу чергу через природоохоронні міркування. Але вони також дешеві, володіють досить високою енергоємністю (удвічі більшою ніж СКА в розрахунку на одиницю маси), перевантажувальною здатністю, і до того ж, на відміну від СКА, не бояться глибокого розряду. Великий недолік Ni-Cd - вони володіють шкідливим «ефектом пам'яті», при якому регулярний заряд недорозрядженого до кінця акумулятора веде до швидкого зниження характеристик, а потім і виходу з ладу.

Область застосування нікель-металогідридних (Ni-Mh) різновидів в побуті практично обмежується пальчиковими акумуляторами, що можуть замінити одноразові батарейки там, де, як говорилося, не потрібно довготривалого зберігання. Повністю замінити ні нікель-кадмієві акумулятори, ні батарейки вони не в змозі, оскільки володіють високим саморозрядом (до 10 і більше відсотків на місяць) і помітно гірше працюють при перевантаженнях - чому їх не можна застосовувати, наприклад, в деяких старих конструкціях цифрових камер. Є ще одна область, де Ni-Mh домінують в даний час: це електромобілі, у тому числі популярні гібридні конструкції (наприклад, знаменитий Toyota Prius), хоча в цій галузі вони поступово витісняються літій-іонними.

Літій-іонні (Li-ion), які незрівнянно вигідніші від усіх інших типів перш за все за енергоємністю, яка досягає 130-150 Вт-год/кг, а також за невеликим саморозрядом (не більше 3% на місяць), найбільш поширені в області харчування високотехнологічних пристроїв – мобільних телефонів, ноутбуків всіх класів, цифрових камер. Це обумовлено тим, що Li-ion відносно дорогі і вимагають досить обережного поводження – зокрема, всі вони мають вбудований контроллер і не підлягають зарядці від «самодіяльних» зарядних пристроїв. Зауважимо, що часто згадувані в пресі літій-полімерні (Li-pol) є просто різновидом літій-іонних, де рідкий електроліт замінений на спеціальний полімер. Головний недолік Li-ion – окрім дорожнечі – у тому, що

при недотриманні режиму заряджання (як, на жаль, і при несправності вбудованого контролера) вони просто вибухають, що сильно обмежує їх застосування. Скандали з відкликанням мільйонів батарей по всьому світу відбуваються з лякаючою регулярністю.

Тим не менше за Li-ion майбутнє. Слід відзначити досягнення невеликої компанії A123Systems, інженери якої навчилися робити літій-іонні акумулятори із залізофосфатним катодом, який відрізняється винятковою стійкістю, повністю усуваючи в тому числі і проблему самозаймання. Причому застосування нанотехнологій дозволило збільшити корисну площу електродів приблизно на чотири порядки, що підвищило питому енергоємність, а, головне, значно збільшило струм, що віддається при перевантаженнях. Ці акумулятори можна навіть використовувати як стартерні в звичайних автомобілях, з чим до цього часу не справлялися нікелеві. Судячи за укладеними угодами A123Systems з деякими провідними виробниками світу (General Motors, General Electric, китайською BAK Battery), ми можемо очікувати таких акумуляторів на прилавках і у складі різних пристроїв у найближчому майбутньому.

Юридичні та фізичні особи (суб'єкти підприємницької діяльності), які використовують хімічні джерела струму, зобов'язані:

- Забезпечувати експлуатацію хімічних джерел струму з додержанням вимог супровідної технічної або експлуатаційної документації та інформації, розміщеної на корпусах хімічних джерел струму;
- Здійснювати облік придбання та експлуатації хімічних джерел струму, обсягів накопичення відпрацьованих хімічних джерел струму та передачі їх на утилізацію;
- Забезпечувати зберігання відпрацьованих хімічних джерел струму в спеціально відведених місцях;

- Здавати відпрацьовані небезпечні хімічні джерела струму ємністю 7 А/год та більше для утилізації на спеціалізовані виробництва з утилізації, спеціалізованим підприємствам з утилізації чи на їх приймальні пункти протягом трьох місяців з дня закінчення строку експлуатації хімічних джерел струму або з дня оформлення документа про непридатність хімічних джерел струму для експлуатації внаслідок фізичного, морального зносу, непоправного браку, пошкодження, з інших причин;
- Надавати в установленому порядку інформацію щодо обсягів придбання та експлуатації нових хімічних нових джеонл струму, обсягів накопичення відпрацьованих хімічних джерел струму та передачі їх на утилізацію.

Фізичні особи які не зареєстровані як суб'єкти підприємницької діяльності та використовують хімічні джерела струму для забезпечення власних потреб зобов'язані здавати відпрацьовані небезпечні джерела струму ємністю 7 А/год та більше для утилізації на спеціалізовані виробництва з утилізації, спеціалізованим підприємствам з утилізації чи на їх приймальні пункти.

Після відпрацювання мікроконтролером свого терміну експлуатації, або несправності ремонту, його потрібно утилізувати. Основний процес утилізації скерований на можливість повторного використання матеріалів з яких був зроблений трекер.

Утилізація включає такі пункти:

- Відокремлення на окремі частини, що надалі переробляються і використовуються для виготовлення нової продукції.
- Те що не підлягало відокремленню надходить в подрібнюючий станок. Дрібні частини відокремлюються і потрапляють на рухливий транспортер. Більш великі фракції проходять повторне дробіння. За допомогою всмоктування повітря через подріблену

фракцію, з неї відокремлюють пластмасовий пил, а метал, що залишився переплавляється.

Розрахунок витрат на утилізацію відпрацьованих джерел струму

В даному трекері використовуються літій-іонні батарейки, тому що їх можна викинути на свалку.

Для розрахунку ми скористаємося формулою вартості утилізації:

$$B_{\text{відх}} = B_m + (B_{\text{утр}} + E_n \cdot K_c) \cdot A_{\text{від}}$$

B_m – втрати на видалення (транспортування, завантаження розвантаження) відходів, $B_m = 8,18 \text{ грн./т}$;

$B_{\text{утр}}$ - експлуатаційні витрати, що пов'язані з обслуговуванням звалища знезараженням відходів, $B_{\text{утр}} = 0,75 \text{ грн./т}$;

E_n – нормативний коефіцієнт ефективності капітальних вкладень, беруть $E = 0,16 \text{ рік}^{-1}$;

K_c – питомі капітальні витрати на будівництво систем видалення, знешкодження (знищення) відходів у спеціальних спорудах, $K_c = 0,3 \text{ грн./т}$;

$A_{\text{від}}$ – кількість відходів, т/рік.

$$B_{\text{відх}} = 8,18 + (0,75 + 0,16 \cdot 0,3) \cdot 1,05 = 9,43 \text{ грн./т}$$

Висновок. (Розділ 8)

В даному дипломному проекті було розглянуто умови забезпечення екологічної безпеки експлуатації системи відеоспостереження на базі мікрокомп'ютеру Raspberry Pi, перераховано правові засади придбання, зберігання і утилізації акумуляторів, описані основні етапи утилізації мікрокомп'ютера.

Висновок

У цьому дипломному проєкті було наведено докладний опис системи з функцією розпізнавання осіб. Були описані різні підходи до реалізації програмних компонентів. Так само була приведена інформація про реалізовані класах.

В ході роботи над даним проєктом були вивчені:

- Методи розпізнавання осіб, такі як каскади Хаара (метод Віолі-Джонса)
- Системи відеоспостереження
- Нейронні мережі
- Бібліотека OpenCV
- Алгоритм градієнтного бустінга
- Мова Python

За даним проєктом була проведена розробка ПО. Так само було проведено тестування. Даний проєкт був створений виключно в наукових цілях і не несе ніякої комерційної вигоди. Надалі цю роботу можна буде модернізувати і використовувати в інших сферах, таких як магазини, банки, ресторани і т.д.

Список літератури

1. Bishop C. Pattern Recognition and Machine Learning / Bishop C. - Singapore .: Springer, 2006 - 738 с .
2. Інтернет-ресурс - <https://hi-news.ru/eto-interesno/kak-eto-rabotaet-sistema-raspoznavaniya-lic.html>
3. Інтернет-ресурс - <https://habr.com/post/198338/>
4. Інтернет-ресурс - https://ru.wikipedia.org/wiki/Распознавание_лиц
5. OpenCV. Python Tutorials [Електронний ресурс] URL : https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html.
6. Робота з Raspberry Pi 3 [Електронний ресурс] URL : <http://dmitrysnotes.ru/raspberry-pi-3-obzor-i-nachalo-raboty>.
7. Документація Python [Електронний ресурс] URL : <https://docs.python.org/3.7/tutorial/>.
8. Raspberry Pi 3. [Електронний ресурс] URL : <http://raspberrypi3.ru>
9. ГОСТ 12.0.003-74. "ССБТ. Опасные и вредные производственные факторы. Классификация".
10. ГОСТ 12.1.003–76 “Шум. Загальні вимоги безпеки”.
11. ГОСТ 12.1.029–80 “Засоби та методи захисту від шуму”.
12. СНІП II-4-79 "Природне і штучне освітлення. Норми проектування".
13. ГОСТ 12.1.005–88 про «Загальні санітарно-гігієнічні вимоги до повітря робочої зони».
14. НПА ОП 0.00-4.15-98 „Положення про розробку інструкцій з охорони праці”.
15. НПА ОП 0.00-6.03-93 „Порядок опрацювання і затвердження власником нормативних актів про охорону праці, що діють на підприємстві”.
16. Савицький В.М., Хільчевський В.К., Чунарьов О.В., Яцюк М.В. Відходи виробництва і споживання та їх вплив на ґрунти і

- природні води: Навчальний посібник / За ред. В.К. Хільчевського.
– К.: Видавничо-поліграфічний центр "Київський університет",
2007. – 152 с.
- 17.ІСАЄНКО В. М., КРИВОРОТЬКО В. М., ФРАНЧУК Г. М.
Екологія та охорона навколишнього середовища. Дипломне
пректування: Навч.посіб.– К.: Книжкове видавництво НАУ, 2005.
– 192с.
- 18.Постанова Кабінету Міністрів України від 28 червня 1997р. №668
«Про програму використання відходів виробництва і споживання
на період до 2005 року».
- 19.Закон України «Про охорону навколишнього природного
середовища» ВВР, 1991, № 41.
- 20.Закон України «Про відходи» ВВР, 1998, № 36-37.
- 21.Постанова Кабінету Міністрів України від 31 серпня 1998 р. №
1360 «Про затвердження Порядку ведення реєстру об'єктів
утворення, оброблення та утилізації відходів».
- 22.Законів України «Про забезпечення санітарного та
епідеміологічного благополуччя населення».
- 23.Законів України «Про хімічні джерела струму».
- 24.Саймон Хайкин, «Нейронные сети. Полный курс. Второе издание»
- 25.Kohonen, Teuvo. “Self-organized formation of topologically correct
feature maps.” Biological cybernetics
- 26.<https://medium.com/@datamonsters/artificial-neural-networks-for-natural-language-processing-part-1-64ca9ebfa3b2> (7 types of Artificial Neural Networks for Natural Language Processing)
- 27.<https://arxiv.org/pdf/1510.06168.pdf> (Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network)

28. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/CNN_ASLPTrans2-14.pdf (Convolutional Neural Networks for Speech Recognition)
29. <http://www.meanotek.ru/files/TarasovDS%282%292015-Dialogue.pdf> (Генерация естественного языка, парафраз и автоматическое обобщение отзывов пользователей с помощью рекуррентных нейронных сетей)
30. Christopher Bishop, «Pattern Recognition and Machine Learning»

Додаток 1. Файл програми

Файл 1. bot.py

```
#!/usr/bin/env python3
import telebot
import parser
from telebot import types
from telebot import apihelper
import cv_controller as cvc
import subprocess
from tokens import TOKEN, proxydata, admins

bot = telebot.TeleBot(TOKEN)

# Setting proxy
apihelper.proxy =
{'https':'socks5h://{}:{}_@{}:{}'.format(proxydata['username'],proxydata['pass'],pr
oxydata['ip'],proxydata['port'])}

# Camera vars
fo = cvc.FaceOperator()
camera_running = False
process = None

@bot.message_handler(commands=['start', 'go'])
def start_handler(message):
    global camera_running
    global process
```



```

clockworker = subprocess.Popen(['python', 'clockworker.py'])

if not camera_running:
# Start camera handler subprocess
    print('[INFO] Camera handler subprocess is running...')
    print('>>>>>> Face detection active')
    process = subprocess.Popen(['python', 'camera_handler.py'])

    camera_running = True

itr = message.chat.id
# Check if user is admin
if itr in admins:
    keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.add(*[types.KeyboardButton(name) for name in ['Прошедшие
сотрудники', 'Узнать по фамилии', 'Добавить', 'Удалить', 'База сотрудников']])
    bot.send_message(message.chat.id, 'Здравствуйте!',reply_markup=keyboard)

else:
    keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.add(*[types.KeyboardButton(name) for name in ['Прошедшие
сотрудники', 'Узнать по фамилии']])
    bot.send_message(message.chat.id, 'Здравствуйте!')
    bot.send_message(message.chat.id, 'Для получения необходимой
информации используйте клавиатуру.',reply_markup=keyboard)

@bot.message_handler(content_types=['text'])
def name(message):
    global camera_running

```

global process

```
itr = message.chat.id
```

```
if itr in admins:
```

```
    if message.text == 'Прошедшие сотрудники':
```

```
        names = fo.get_current_names()
```

```
        outstring = 'Все прошедшие сотрудники:\n'
```

```
        for name in names:
```

```
            outstring = outstring + name + '\n'
```

```
        bot.send_message(message.chat.id, outstring)
```

```
    elif message.text == 'Узнать по фамилии':
```

```
        bot.send_message(message.chat.id, 'Введите фамилию сотрудника через  
спец символ ?, например: \n?Смолянинова\n')
```

```
    elif message.text[0] == '?':
```

```
        name = message.text;
```

```
        name = name.replace('?', '')
```

```
        names = fo.get_current_names()
```

```
        if name in names:
```

```
            bot.send_message(message.chat.id, 'Сотрудник '+name+' на месте!')
```

```
        else:
```

```
            bot.send_message(message.chat.id, 'Сотрудника '+name+' нет!')
```

```
    elif 'Добавить' == message.text:
```

```
        bot.send_message(message.chat.id, 'Чтобы добавить, начни писать с +,  
например: +Марков 1 1\n Не забывай, первая цифра - логи\n Вторая видео')
```

```
    elif message.text[0] == '+':
```

```
        name = message.text;
```

```
        name = name.replace('+', '')
```

```

parts = [i for i in name.split()]

# Terminate camera handling
process.terminate()
print('[INFO] >Camera handler subprocess terminating for a while...')
print('>>>>> Face detection inactive')

try:
    fo.add_face(parts[0],int(parts[1]),int(parts[2]))
except IndexError:
    fo.add_face(parts[0],1,0)
bot.send_message(message.chat.id, 'succesfull')

# Back to camera handling
print('[INFO] Camera handler subprocess is running...')
print('>>>>> Face detection active')
process = subprocess.Popen(['python','camera_handler.py'])
camera_running = True

elif 'Удалить' == message.text:
    bot.send_message(message.chat.id, 'Чтобы удалить, начни писать с -,
например: -Марков')
elif message.text[0] == '-':
    name = message.text;
    name = name.replace('-',")

# Terminate camera handling
process.terminate()
print('[INFO] >Camera handler subprocess terminating for a while...')

```

```

print('>>>>> Face detection inactive')

res, _ = fo.del_face(name)

# Back to camera handling
print('[INFO] Camera handler subprocess is running...')
print('>>>>> Face detection active')
process = subprocess.Popen(['python', 'camera_handler.py'])
camera_running = True

if res:
    bot.send_message(message.chat.id, "\"+name+\" \u201cуспешно удалено!\u201c")
else:
    bot.send_message(message.chat.id, "\"+name+\" \u201cне найдено.\u201c")

elif 'База сотрудников' == message.text:
    lnames = fo.get_names()
    outstring = ""

    for i in lnames:
        outstring = outstring + i + '\n'
    if outstring == "":
        outstring = 'Empty'
    bot.send_message(message.chat.id, outstring)
else:
    bot.send_message(message.chat.id, 'Такой команды нет')

else:
    if message.text == 'Прошедшие сотрудники':

```

```

names = fo.get_current_names()
outstring = 'Все прошедшие сотрудники:\n'

for name in names:
    outstring = outstring + name + '\n'
    bot.send_message(message.chat.id, outstring)
elif message.text == 'Узнать по фамилии':
    bot.send_message(message.chat.id, 'Введите фамилию сотрудника через
спец символ ?, например: \n?Смольянинова\n')
elif(message.text[0] == '?'):
    name = message.text;
    name = name.replace('?', '')
    names = fo.get_current_names()
    if name in names:
        bot.send_message(message.chat.id, 'Сотрудник '+name+' на месте!')
    else:
        bot.send_message(message.chat.id, 'Сотрудника '+name+' нет!')
else:
    bot.send_message(message.chat.id, 'Такой команды нет')

if __name__ == "__main__":
    bot.polling()

```

Файл 2. face_recognition.py

```
#!/usr/bin/env python3

# Execute real-time face detection

import cv2
import numpy as np
import os
from . import user_names

def run(video=False):
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('./face_operations/trainer/trainer.yml')
    cascadePath = "./face_operations/haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(cascadePath);

    font = cv2.FONT_HERSHEY_SIMPLEX

    # Iniciate id counter
    id = 0

    # Names related to ids: example ==> Alexander: id=1, etc
    names = user_names.get_names()
    #print('[INFO] Known names: ',names)

    # Initialize and start realtime video capture
    cam = cv2.VideoCapture(-1)
    cam.set(3, 640) # set video width
    cam.set(4, 480) # set video height
```

```

# Define min window size to be recognized as a face
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)

# Variable for first detected name
detected_name = ""

while True:

    ret, img =cam.read()
    #img = cv2.flip(img, -1) # Flip vertically
    gray = None
    if ret is True:
        gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    else:
        continue

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
        )

    for(x,y,w,h) in faces:

        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)

```

```

id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

# Check if confidence is less them 100 ==> "0" is perfect match
if (confidence < 100) and (id < len(names)):
    id = names[id]
    confidence = " {0}%".format(round(100 - confidence))
    detected_name = id
else:
    id = "unknown"
    confidence = " {0}%".format(round(100 - confidence))
    detected_name = id

cv2.putText(img, str(id), (x+5,y-5), font, 0.7, (255,255,255), 2)
cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

if video:
    cv2.imshow('camera',img)

# If some face detected
if detected_name != "":
    break

k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
if k == 27:
    break

# Cleanup
print("[INFO] Cleanup stuff... (camera afteruse trash)")
cam.release()

```



```
cv2.destroyAllWindows()
```

```
return detected_name
```

Файл 3. face_training.py

```
#!/usr/bin/env python3
```

```
# Train classifier
```

```
import cv2
```

```
import numpy as np
```

```
from PIL import Image
```

```
import os
```

```
def train_clf():
```

```
    # Path for face image database
```

```
    path = './face_operations/dataset'
```

```
    recognizer = cv2.face.LBPHFaceRecognizer_create()
```

```
    detector
```

```
cv2.CascadeClassifier("./face_operations/haarcascade_frontalface_default.xml");
```

```
# Function to get the images and label data
```

```
def getImagesAndLabels(path):
```

```
    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
```

```
    faceSamples=[]
```

```

ids = []

for imagePath in imagePaths:

    PIL_img = Image.open(imagePath).convert('L') # convert it to grayscale
    img_numpy = np.array(PIL_img,'uint8')

    id = int(os.path.split(imagePath)[-1].split(".")[1])
    faces = detector.detectMultiScale(img_numpy)

    for (x,y,w,h) in faces:
        faceSamples.append(img_numpy[y:y+h,x:x+w])
        ids.append(id)

return faceSamples,ids

print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))

# Save the model into trainer/trainer.yml
recognizer.write('./face_operations/trainer/trainer.yml') # recognizer.save()
worked on Mac, but not on Pi

# Print the numer of faces trained and end program
print("\n      [INFO]      {0}      faces      trained.      Exiting
Program".format(len(np.unique(ids))))

```

Файл 4. camera_handler.py

```
#!/usr/bin/env python3

from face_operations import current_names, face_recognition

while True:
    detected_name = face_recognition.run()

    if detected_name == 'unknown' or detected_name == 'None':
        print('[INFO] Detected unknoww face. Continue...')
        continue

    # names in current_names file
    names_in_file = current_names.get_names()

    # change indicators
    if detected_name in names_in_file:
        current_names.del_name(detected_name)
        print('[INFO] ',detected_name,' status changed to 0')

    else:
        current_names.add_name(detected_name)
        print('[INFO] ',detected_name,' status changed to 1')
```

Файл 5. cv_controller.py

```
#!/usr/bin/env python3
```

```
from face_operations import user_names, current_names, face_recognition,  
face_add, face_training  
import face_operations as fo
```

```
"""
```

This file is wrapper for face_operations package

face_operations package contains:

- face_add.py - add new user face with id
- face_training.py - train classifier with new user faces
- face_recognition.py - executable program

Ones from face_operations will not used directly (only imports)

```
"""
```

```
# Face op class
```

```
class FaceOperator:
```

```
    """
```

```
    - add_face(string new_name, bool verobose, bool with_video) -
```

```
        [Params]
```

```
        :new_name: - string - user name
```

```
        :verbose (default=True): - bool - if true - print text logs
```

```
        :with_video (default=False): - bool - if true - init video logs
```

```
        [Description]
```

```
        create id for new user,
```

take photos,
train classifier,
add new_name into names file,
add name into dict current_names,
print logs (if verbose == True)

[Returns]

no return

- del_face(string name_to_del) -

[Params]

:name_to_del: - string - user name

[Description]

replace name with 'None',

remove photos from dataset for user with name_to_del

[Returns]

True if successfull (face found -> deleted)

False if unsuccessful (face not found -> terminated)

- get_names() -

[Params]

no params

[Description]

get names list from names file (all known names)

[Returns]

List of names (ex: ['Alex', 'Artem'])

- get_current_names() -

[Params]

no params

[Description]

get names of people with indicator = 1

[Returns]

List of names (ex: ['Alex', 'Artem'])

"""

@staticmethod

def add_face(new_name, verbose=True, with_video=False):

Use len to get new id

Last id was len - 1

new_id = len(user_names.get_names())

Add face to base

face_add.user_face_add(new_id, with_video, verbose)

Train clf

face_training.train_clf()

```

# Add new name
user_names.add_name(new_name)
current_names.add_name(new_name)

# Report
names = user_names.get_names()
if verbose:
    print('[INFO] User added with id = ',new_id,'\n[INFO] Known
name list = ',names,'\n[INFO] Length = ',len(names))

```

```

@staticmethod
def del_face(name_to_del):
    good_del_names = user_names.del_name(name_to_del)
    good_del_curr_names = current_names.del_name(name_to_del)

    return good_del_names, good_del_curr_names

```

```

# Get all known names list
@staticmethod
def get_names():
    names = user_names.get_names()
    while 'None' in names:
        names.remove('None')
    return names

```

```

# Get current names list
@staticmethod

```

```
def get_current_names():
    return current_names.get_names()
```

Файл 6. current_names.py

```
#!/usr/bin/env python3#!/usr/bin/env python3
import os

# Get the list of existing names
def get_names():
    with open('./face_operations/current_names.txt', 'r') as f:
        names = map(lambda s: s.strip(), f.readlines())

    return list(names)

# Add the name
def add_name(new_name):
    with open('./face_operations/current_names.txt', 'a') as f:
        if os.stat('./face_operations/current_names.txt').st_size == 0:
            f.write(new_name)
        else:
            f.write('\n' + new_name)

# Delete name by replacing it with None
def del_name(name_to_del):
    with open('./face_operations/current_names.txt', 'r+') as f:
        names = f.readlines()
        names = list(map(lambda s: s.strip(), names))
        success = False
```



```

try:
    ind = names.index(name_to_del)
    del names[ind]
    f.seek(0)
    f.truncate()
    for idx, name in enumerate(names):
        if idx != (len(names) - 1):
            f.write(name+'\n')
        else:
            f.write(name)

    success = True

except ValueError:
    success = False

return success

```

Файлы user_names.py

```
#!/usr/bin/env python3
```

```
import os, glob
```

```
# Get the list of existing names
```

```
def get_names():
```

```
    with open('./face_operations/user_names.txt', 'r') as f:
```

```
        names = map(lambda s: s.strip(), f.readlines())
```

```
    return list(names)
```

```
# Add the name
```

```

def add_name(new_name):
    with open('./face_operations/user_names.txt', 'a') as f:
        if os.stat('./face_operations/user_names.txt').st_size == 0:
            f.write(new_name)
        else:
            f.write('\n' + new_name)

# Delete name by replacing it with None
def del_name(name_to_del):
    with open('./face_operations/user_names.txt', 'r+') as f:
        names = f.readlines()
        names = list(map(lambda s: s.strip(), names))
        success = False
        try:
            ind = names.index(name_to_del)
            names[names.index(name_to_del)] = 'None'
            f.seek(0)
            f.truncate()
            for idx, name in enumerate(names):
                if idx != (len(names) - 1):
                    f.write(name+'\n')
                else:
                    f.write(name)

            success = True
            for filename in glob.glob('./face_operations/dataset/User.'+str(ind)+'.*'):
                os.remove(filename)

        except ValueError:

```

```
success = False
```

```
return success
```