**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені Ігоря Сікорського»**

**Факультет прикладної математики**

**Кафедра прикладної математики**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_ О. Р. Чертов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 р.

**Дипломна робота**

**на здобуття ступеня бакалавра**

з напряму підготовки 6.040301 «Прикладна математика»

на тему: Автоматизована система пошуку дублікатів зображень

Виконав: студент IV курсу, групи КМ-42

|  |  |  |
| --- | --- | --- |
|  | Волк Ілля Михайлович | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Керівник | асистент  Дрозденко О. М. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Консультант зі спеціальних питань | старший викладач  Мальчиков В.В. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Консультант із нормоконтролю | старший викладач Мальчиков В. В. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Рецензент | проф., д-р техн. наук, проф. Сидоренко С. С. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Засвідчую, що в цій дипломній роботі немає запозичень із праць інших авторів без відповідних посилань.

Волк І.М. \_\_\_\_\_\_\_\_\_\_\_\_

Київ — 2018**Національний технічний університет України**

**«Київський політехнічний інститут  
імені Ігоря Сікорського»**

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти — перший (бакалаврський)

Напрям підготовки 6.040301 «Прикладна математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_ О. Р. Чертов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

Волку Іллі Мизхайловичу

1. Тема роботи: «Автоматизована система пошуку дублікатів зображень»,

керівник роботи Дрозденко Олександр Миколайович, асистент,

затверджені наказом по університету від «06» квітня 2018 р. № 1108-С.

2. Термін подання студентом роботи: «15» червня 2018 р.

3. Вихідні дані до роботи: розроблювана система повинна працювати з нечіткими даними, мінімальна точність розпізнавання емоцій — 70%.

4. Зміст роботи: виконати аналіз існуючих методів розв’язання задачі, вибрати метод розпізнавання емоцій за зображенням людини, спроектувати автоматизовану систему розпізнавання емоцій, здійснити програмну реалізацію розробленої системи, провести тестування розробленої системи.

5. Перелік ілюстративного матеріалу: архітектурні графи нейронних мереж, блок-схеми розроблених алгоритмів, схема взаємодії модулів системи, знімки екранних форм.

6. Консультанти розділів роботи:

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
| завдання видав | завдання прийняв |
| Розділ 3. Математичне забезпечення | Бондаренко Б. Б., старший викладач |  |  |

7. Дата видачі завдання: «15» квітня 2016 р.

Календарний план

| № з/п | Назва етапів виконання дипломної роботи | Термін виконання етапів роботи | Примітка |
| --- | --- | --- | --- |
| 1 | Огляд літератури за тематикою та збір даних | 12.11.2015 |  |
| 2 | Проведення порівняльного аналізу математичних методів розпізнавання зображень | 14.12.2015 |  |
| 3 | Проведення порівняльного аналізу математичних методів розпізнавання емоційного стану за зображенням обличчя | 24.12.2015 |  |
| 4 | Підготовка матеріалів першого розділу роботи | 01.02.2016 |  |
| 5 | Розроблення математичного забезпечення для розпізнавання емоційного стану за статичним фронтальним зображенням обличчя | 01.03.2016 |  |
| 6 | Підготовка матеріалів другого розділу роботи | 15.03.2016 |  |
| 7 | Підготовка матеріалів третього розділу роботи | 05.04.2016 |  |
| 8 | Розроблення програмного забезпечення для розпізнавання емоційного стану за статичним фронтальним зображенням обличчя | 15.04.2016 |  |
| 9 | Підготовка матеріалів четвертого розділу роботи | 03.05.2016 |  |
| 10 | Оформлення пояснювальної записки | 01.06.2016 |  |

|  |  |  |
| --- | --- | --- |
| Студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Іваненко І. І. |
| Керівник роботи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Петренко П. П. | |

Анотація

Дипломну роботу виконано на 28 аркушах, вона містить 2 додатки та перелік посилань на використані джерела з 16 найменувань. У роботі наведено 4 рисунки та 2 таблиці.

Метою даної дипломної роботи є створення математичного та програмного забезпечення для розпізнавання базових людських емоцій за статичним фронтальним зображенням її обличчя.

У роботі проведено аналіз існуючих рішень указаної задачі — штучних нейронних мереж, систем адаптивного нейронечіткого виведення та прихованих марківських моделей. Виконано їх порівняння з погляду точності отримуваних розв’язків, ефективності алгоритмів та пристосованості методів до використання нечітких даних. Для розв’язання задачі в роботі вибрано метод адаптивного нейронечіткого виведення.

Для кожної розглянутої емоції сформовано нечіткі продукційні правила. Розроблено автоматизовану систему, що реалізує обраний метод. Виконано тестування розробленої системи.

Основні положення дипломної роботи опубліковано у вигляді тез доповіді на Міжнародній науково-технічній конференції SAIT 2016.

Ключові слова: емоція, система адаптивного нейронечіткого виведення, гібридний алгоритм навчання, перехресна перевірка, продукційні правила, вектор ознак.

ABSTRACT

The thesis is presented in 28 pages. It contains 2 appendixes and bibliography of 16 references. Four figures and 2 tables are given in the thesis.

The goal of the thesis is to develop mathematical and software tools for solving the problem of basic human emotion recognition by a static frontal image of her face.

In the thesis, existing solutions are analyzed, such as artificial neural networks, adaptive neuro-fuzzy inference systems, and hidden Markov models. They are compared in terms of the accuracy of obtained results, algorithm efficiency and method adaptation to fuzzy data. In the thesis, adaptive neuro-fuzzy inference approach is used to solve the task.

Fuzzy production rules are formulated for each discussed emotion. The automated system implementing the chosen method is developed. The developed system is tested.

Main ideas of the thesis were published in the Proceedings of the International Scientific and Technical Conference SAIT 2016.

Keywords: emotion, adaptive neuro-fuzzy inference system, hybrid learning algorithm, cross-validation, production rules, feature vector.

ЗМІСТ

[Перелік умовних позначень, скорочень і термінів 9](#_Toc514630115)

[Вступ 10](#_Toc514630116)

[1 Постановка задачі 11](#_Toc514630117)

[1.1.1 Метою даної дипломної роботи є створення автоматизованої системи для виявлення схожих зображень. 11](#_Toc514630118)

[2 Аналіз існуючих методів розпізнавання емоцій людини 12](#_Toc514630119)

[2.1 Виділення критеріїв оцінки методів 12](#_Toc514630120)

[2.2 Математичні методи розпізнавання емоцій людини 13](#_Toc514630121)

[2.2.1 Штучні нейронні мережі 13](#_Toc514630122)

[2.2.2 Системи адаптивного нейронечіткого виведення 14](#_Toc514630123)

[2.2.3 Приховані марківські моделі 14](#_Toc514630124)

[2.2.4 Порівняння математичних методів розв’язання задачі 15](#_Toc514630125)

[2.3 Огляд існуючих комерційних програмних рішень 15](#_Toc514630126)

[2.3.1 FaceReader 15](#_Toc514630127)

[2.3.2 eMotion 15](#_Toc514630128)

[2.3.3 MMER\_FEASy — the FacE Analysis System 16](#_Toc514630129)

[2.3.4 Порівняння програмних рішень для розв’язання задачі 16](#_Toc514630130)

[2.4 Висновки до розділу 17](#_Toc514630131)

[3 Аналіз існуючих методів розпізнавання емоцій людини 18](#_Toc514630132)

[3.1 Виділення критеріїв оцінки методів 18](#_Toc514630133)

[3.2 Математичні методи розпізнавання емоцій людини 19](#_Toc514630134)

[3.2.1 Штучні нейронні мережі 19](#_Toc514630135)

[3.2.2 Системи адаптивного нейронечіткого виведення 20](#_Toc514630136)

[3.2.3 Приховані марківські моделі 20](#_Toc514630137)

[3.2.4 Порівняння математичних методів розв’язання задачі 21](#_Toc514630138)

[3.3 Огляд існуючих комерційних програмних рішень 21](#_Toc514630139)

[3.3.1 FaceReader 21](#_Toc514630140)

[3.3.2 eMotion 21](#_Toc514630141)

[3.3.3 MMER\_FEASy — the FacE Analysis System 22](#_Toc514630142)

[3.3.4 Порівняння програмних рішень для розв’язання задачі 22](#_Toc514630143)

[3.4 Висновки до розділу 23](#_Toc514630144)

[4 Математичне забезпечення 24](#_Toc514630145)

[4.1 Архітектура ANFIS та її модифікація 24](#_Toc514630146)

[4.2 Алгоритм навчання ANFIS 26](#_Toc514630147)

[4.3 Формування вектора ознак 26](#_Toc514630148)

[4.4 Формулювання нечітких правил 27](#_Toc514630149)

[4.5 Висновки до розділу 27](#_Toc514630150)

[5 Програмне забезпечення 28](#_Toc514630151)

[5.1 Структура програми 28](#_Toc514630152)

[5.2 Опис розроблених алгоритмів 29](#_Toc514630153)

[5.3 Формат вихідних даних 30](#_Toc514630154)

[5.3.1 Вихідні дані для ініціалізації системи 30](#_Toc514630155)

[5.3.2 Вихідні дані для навчання системи 30](#_Toc514630156)

[5.4 Формат результуючих даних 31](#_Toc514630157)

[5.5 Результати випробування 31](#_Toc514630158)

[5.5.1 Навчання ANFIS 31](#_Toc514630159)

[5.5.2 Випробування системи 31](#_Toc514630160)

[5.6 Висновки до розділу 31](#_Toc514630161)

[Висновки 32](#_Toc514630162)

[Перелік посилань 33](#_Toc514630163)

[Додаток А Лістинги програм 35](#_Toc514630164)

[Додаток Б Ілюстративний матеріал 38](#_Toc514630165)

Перелік умовних позначень, скорочень і термінів

ANFIS — adaptive neuro-fuzzy inference system.

HMM — hidden Markov model.

MANFIS — modified adaptive neuro-fuzzy inference system.

ЕОМ — електронно-обчислювальна машина.

РЕСЛ — розпізнавання емоційного стану людини.

Вступ

Швидкий ріст обсягу даних які можна зберігати на персональних носіях, доступність засобів фотографування спричинили накопичення фотографій на персональних комп’ютерах. Це ускладнює пошук і перегляд зображень. Також, з’являється проблема коли не вистачає місця для нових зображень. Таке відбувається при фотографуванні одного і того ж місця декілька разів, при подальшій обробці зображення: використання фільтрів, підвищення контрастності, зміні розмірів.

Для вирішення цих проблем використовують програмне забезпечення для пошуку схожих зображень. Популярні методи, такі як метод відбитків, watermarking або метод ключових слів по-своєму визначають поняття «схожість». Watermarking використовується для пошуку підробок, а метод ключових слів визначає схожими зовсім різні зображення. Для пошуку схожих зображень на персональному комп’ютері найбільш підходить метод цифрових відбитків, адже інші методи вимагають деякої підготовки, а методи цифрових відбитків побудовані на завчасно визначених алгоритмах. Програми, побудовані на таких методах, порівнюють зображення лише поверхнево, або є хмарними комерційними рішеннями. Саме тому актуальною є проблема створення програмного забезпечення для виявлення дублікатів зображень на персональному комп’ютері.

# Постановка задачі

### Метою даної дипломної роботи є створення автоматизованої системи для виявлення схожих зображень.

Для досягнення поставленої мети необхідно виконати наступні завдання:

1. Огляд існуючих систем.
2. Вибір методу і його алгоритмізація.
3. Програмна реалізація вибраного методу.
4. Тестування створеного ПО.

Вимоги до вхідних даних:

1. Зображення:
2. Формат: JPEG, PNG.
3. Розмір: до 1440×2560 пікселів.

Розроблена система повинна виконувати наступні функції:

1. Працювати на персональному комп’ютері користувача
2. Показувати звіт про свою роботу

Розроблена система повинна вважати схожими зображення з такими відмінностями:

1. Використання фільтрів
2. Поворот
3. Зсув
4. Зміна розмірів
5. Додавання об’єктів

# АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОШУКУ СХОЖИХ ЗОБРАЖЕНЬ

## Огляд існуючих систем

### Tineye

Tineye – сервіс для пошуку візуально схожих зображень в колекціях. Має глобальну колекцію зображень з мережі, комерційна версія надає можливість створювати власні колекції.

Призначення сервісу – пошук дублікатів, модифікованих копій, перевірка зображення на ідентичність.

Технологія побудована на отриманні відбитка – невеликої кількості даних, що описують особливості, знайдені в зображенні. Відбитки зберігаються в їх внутрішніх базах даних. Для додавання зображення і пошуку використовуються виклики API. Технологія здатна скористатися схожістю частини відбитка.

Можливості системи:

1. Знаходить ідентичні зображення
2. Знаходить зображення, що мають схожу частину (Рис. 2.1.1.1)
3. Шукає зображення в мережі
4. Не вважає схожими унікальні зображення (Рис. 2.1.1.2)

Недоліки:

1. Посилання на публічне зображення не актуальне
2. Використання власних зображень для пошуку платне
3. Відсутність «агента» - для використання власних зображень спочатку їх потрібно завантажити в систему.

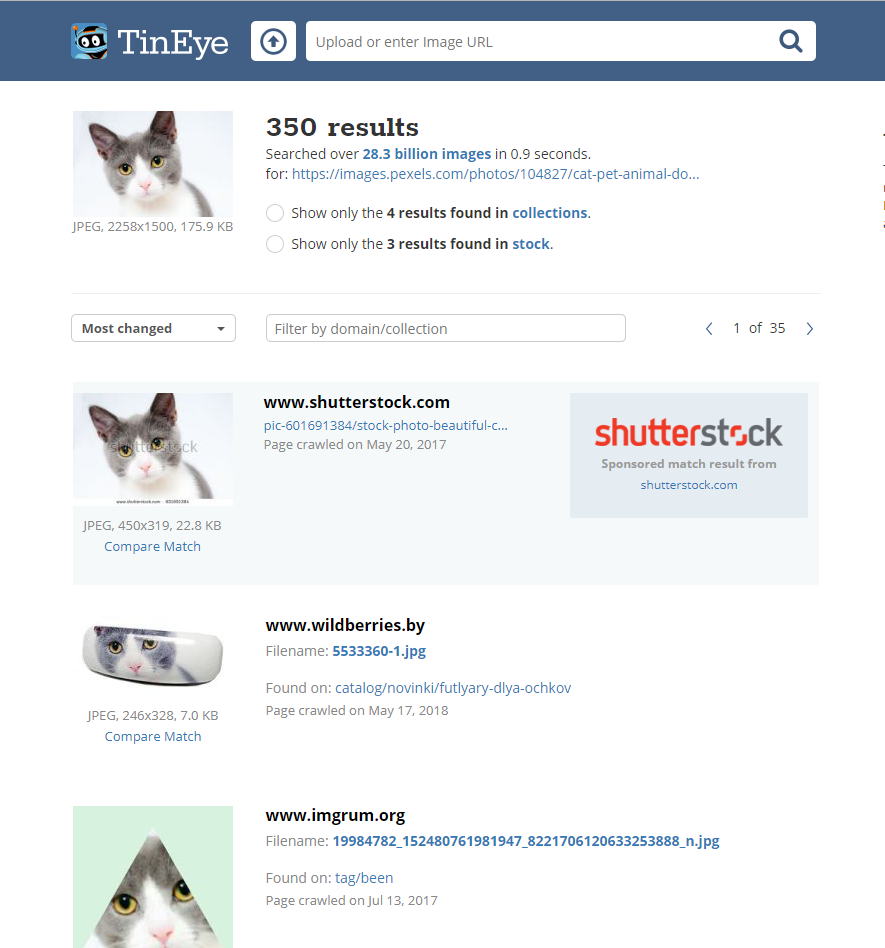


Рис. 2.1.1.1 – Результат пошуку Tineye – зображення з мережі

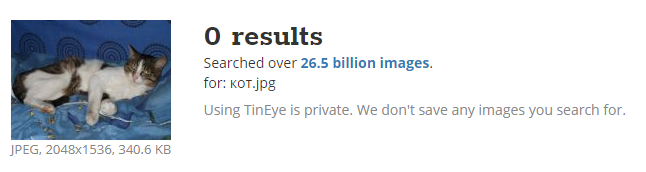


Рис. 2.1.1.2 – Результат пошуку Tineye – унікальне зображення

### Dup Detector (Keronsoft).

Keronsoft – команія, що спеціалізується на дослідженнях цифрових фотографій, відеозаписів, застосуваннях для створення медіаконтенту. Також вона розповсюджує мультимедійне програмне забезпечення.

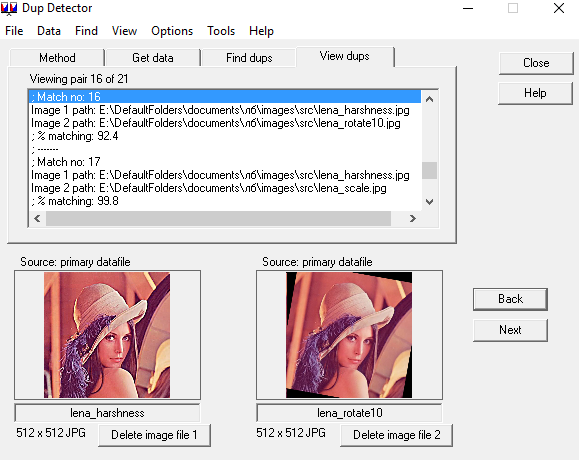


Рис. 2.1.3.1 – Інтерфейс програми Dup Detector

Програма змогла виявити схожість у випадках застосування фільтрів, додавання об’єктів, масштабування, повороту на 10, але не змогла виявити її у випадках зсуву, повороту на 90, 180.

### Image Comparer

Програма для windows для пошуку схожих зображень на машині користувача.

Судячи з результатів її роботи вона використовує ті ж методи що і tineye – на основі отриманні відбитка.

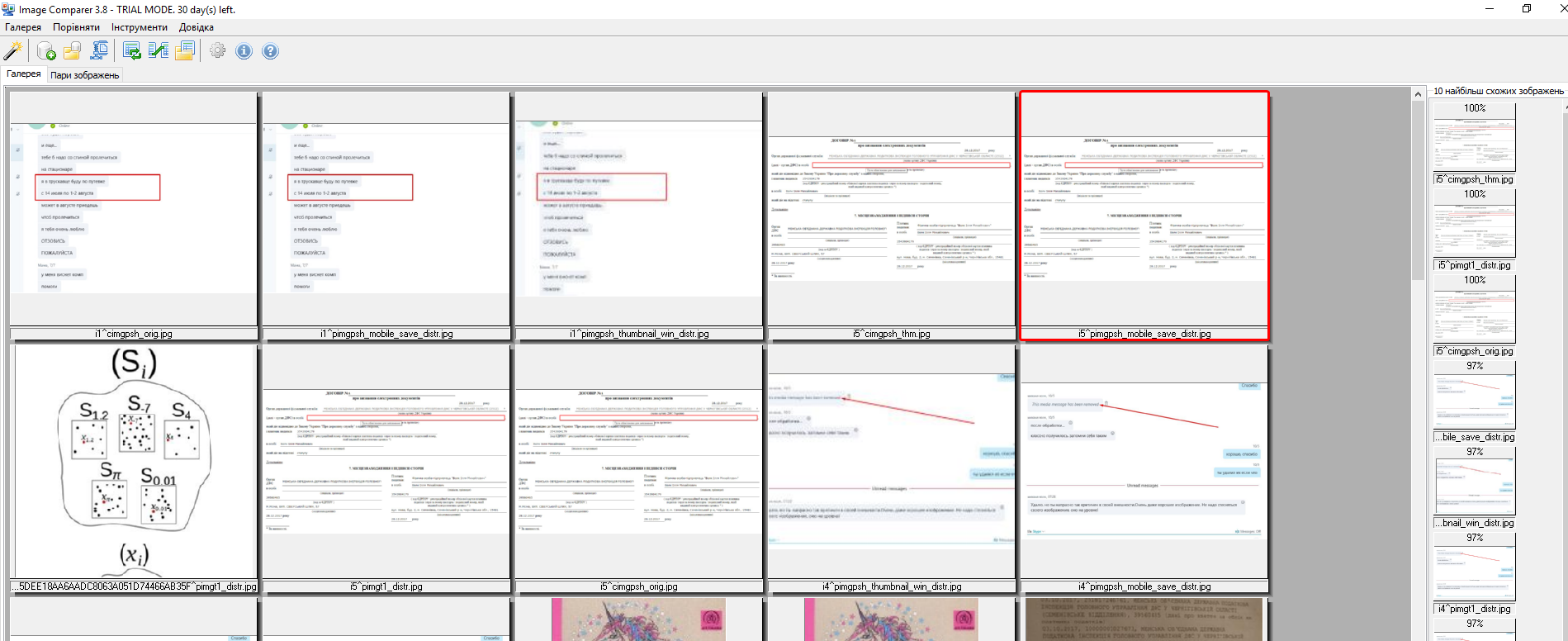


Рис. 2.1.3.1 – Інтерфейс програми Image Comparer

При тестуванні програми виявлено що вона добре визначає схожими масштабовані зображення, зображення з додаванням фільтрів, додаванням об’єктів, але не визначає схожими повернуті та зсунуті зображення.

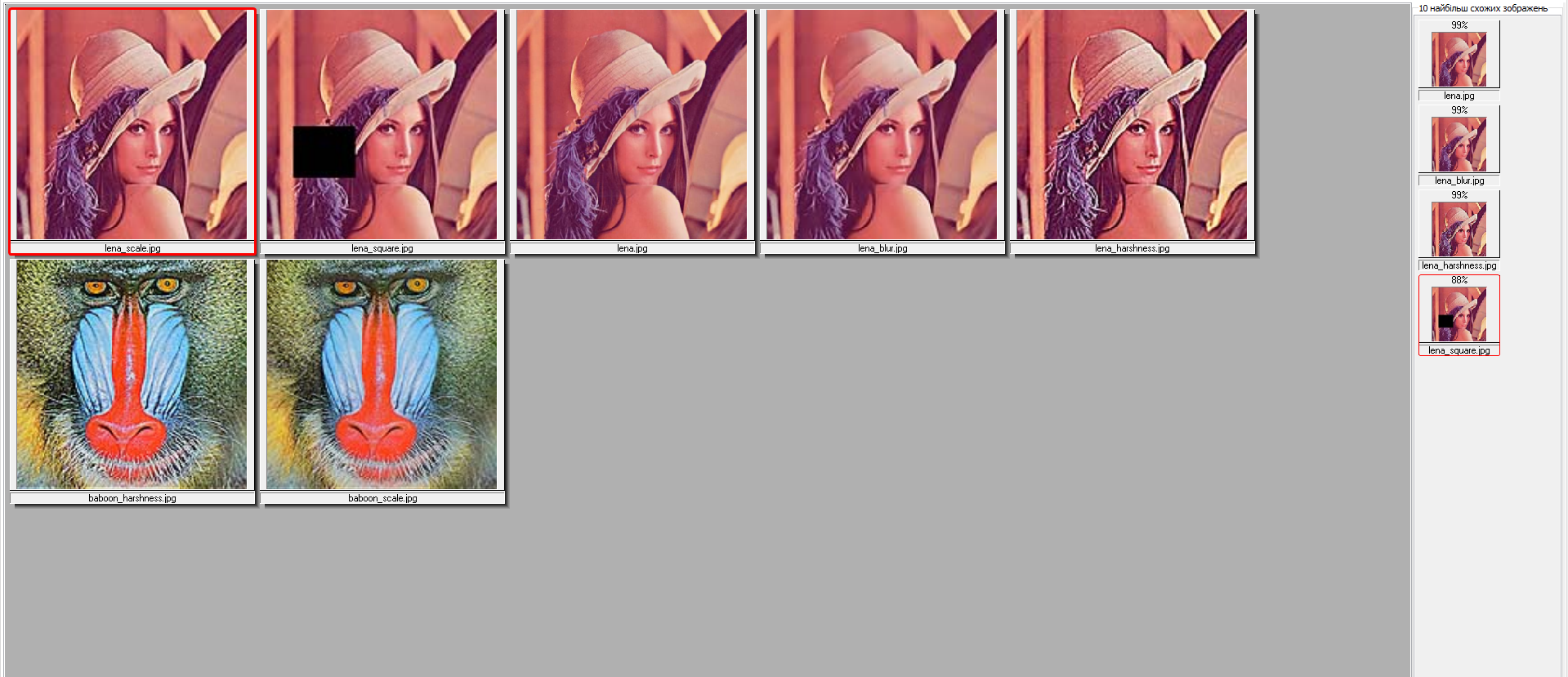


Рис. 2.1.3.2 – Тестування програми Image Comparer на зображеннях lena і baboon

### Google image search

Cервіс для пошуку схожих зображень в мережі. Він знаходить не лише візуально схожі зображення, а й логічно пов’язані. Система розпізнає об’єкти, що знаходяться на зображнні і повертає зображення з такими ж об’єктами.

Пошук побудований на ключових словах. Під час аналізу зображення система з’ясовує характеристики зображення і описує їх за допомогою слів. Результатами пошуку є зображення, опис яких є схожим.

На прикладі до tineye (зображення кота) google image search знаходить зображення, які б людина описала так само як і пошукове. В даному випадку це «Кіт на синьому фоні» (Напевно звіт чорно-білий, прошу повірити). Всі знайдені зображення містять котів і мають близький до синього (бодай не червоний) колір фону.

Рис. 2.1.4.1 – Зображення мого кота Рис. 2.1.4.2 – Результат пошуку Google

### Порівняння існуючих програмних рішень

Основні переваги і недоліки існуючих програмних рішень для пошуку схожих зображень наведено в таблиці 2.1.

Таблиця 2.1 – Порівняння існуючих програмних рішень

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Програмний продукт | Пошук в мережі | Пошук на персональному комп’ютері | Стійкий до афінних перетворень | Виведення всіх схожих зображень і їх місцезнаходження |
| Tineye | + | Платна можливість створення персональної хмарної колекції | + | + |
| Google image search | + | - | + | - |
| Dup Detector | - | + | + | + |
| Image Comparer | - | + | - | + |

## Огляд методів

Існує два протилежні підходи в пошуку схожих зображень – на основі водяного знаку і на основі вмісту. Підхід на основі водяного знака полягає в закарбовуванні підпису в зображенні перед його розповсюдженням. Згодом за цим підписом можна буде визначити чи є зображення дублікатами. З іншого боку, підхід на основі вмісту, як видно з його назви, полягає в аналізі зображення з метою визначення його особливостей.

### Пошук особливостей

Більшість таких методів побудовані на створенні короткого опису зображення – відбитку.

Відбиток – це короткий опис зображення. Аналогом є криптографічні хеш-функції, які для будь-якого об’єму даних визначають невелику завчасно визначену кількість бітів даних.[[2]](file:///E:\DefaultFolders\Documents\dyp\дипломнаяv2.doc#b2) Але криптографічні функції не можуть бути застосовані для створення відбитків зображень. Над зображеннями виконують різні маніпуляції, такі як стиснення, покращення, геометричні перетворення. Криптографічні функції дуже чутливі: зміна одного біта зображення може значно змінити отриманий хеш. Через це криптографічні функції не можуть бути застосовані для отримання цифрових відбитків. Методи створення таких відбитків повинні дозволяти незначні зміни в зображенні. Щоб усунути ці недоліки ми приходимо до поняття multimedia fingerprinting, його ще називають robust hash functions.[[3]](file:///E:\DefaultFolders\Documents\dyp\дипломнаяv2.doc#b3)

Стійкість (інваріантність за подібністю до сприйняття): відбитки, що виникають внаслідок модифікації зображення, повинні бути схожими на відбиток на оригінальному знімку. Стійкість - це здатність ідентифікувати два подібні об'єкти як схожі. [[1]](file:///E:\DefaultFolders\Documents\dyp\дипломнаяv2.doc#b1)

Відбитки можна порівнюються різними способами, наприклад, за допомогою норми Хемінга.

2, де f1, 72 – шістнадцяткові числа.



В деяких підходах кожен хеш описує конкретну частину зображення. Опис зображення стає багатшим. Кількість частин, їхнє положення і форма залежать від вмісту зображення. Таких частин може бути від кількох до кількох тисяч. Зображення вважаються схожими якщо кількість співпадаючих хешів є більшою за певний рівень. Такі методи є більш точними. Вони покладаються на конкретні особливості зображення і потребують великої кількості порівнянь.

Хоча ці методи є більш гнучкими ніж методи додавання водяного знака, і не вимагають попередньої модифікації зображення, вони роблять більше помилок ніж методи на основі водяного знака. В той же час вони будуть визначати два зображення як схожі якщо вони мають схожий зміст, навіть якщо вони є різними в сенсі авторства. Також особливості можуть бути визначені по-іншому після модифікації зображення, що спричинить створення різних відбитків.

### Пошук на основі водяного знака

Історично, дублікати шукалися за допомогою методів на основі водяного знаку.

Ідея водяного знака проста: автор зображення додає до нього стійкий непомітний секретний підпис перед його розповсюдженням. Прихований підпис виконує дві функції. По-перше, він дозволяє встановити власника зображення в суперечливих ситуаціях. По-друге, за допомогою підпису можна знайти копії зображення, наприклад в мережі, і водночас визначити чи є вони легальними. (3.2.1, EPEL\_TH3797)

Методи на основі водяного знака мають кілька недоліків.

1. Шахрай може видалити підпис з зображення, наприклад за допомогою зміни освітлення зображення, чи застосуванні того ж самого метода додавання водяного знака. Таке зображення можна законно розповсюджувати, метод на основі водяного знака виявиться марним.
2. Вони вимагають модифікації зображення перед його розповсюдженням. Це не завжди можливо, і деякі автори можуть негативно ставитися до модифікацій.

### Порівняння програмних рішень для розв’язання задачі

Результати аналізу розглянутих комерційних програмних рішень наведено в таблиці 2.2.

Таблиця 2.2 – Порівняння комерційних програмних рішень

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Вимагає підготовки | Стійкий до редагування | Кількість відмінностей, до яких метод є стійким |
| Fingerprinting | - | + | Середня |
| Watermarking | + | + | Середня |

# Математичне забезпечення

## Опис обраного методу

Для реалізації було обрано метод на основі перетворення Радона. [[1]](#b1)

Це метод на основі отримання відбитку.

Цей метод є стійким до афінних перетворень, шумів, розмиття, стиснення.

Перетворення Радона представляє зображення як колекцію проекцій по різних напрямках. Перетворення Радона зображення визначається як інтеграл по лінії, нахиленої на кут вісі на відстані від початку координат. ([Рис. 2.3.1](#im1)) Математично записується так:

,

де .

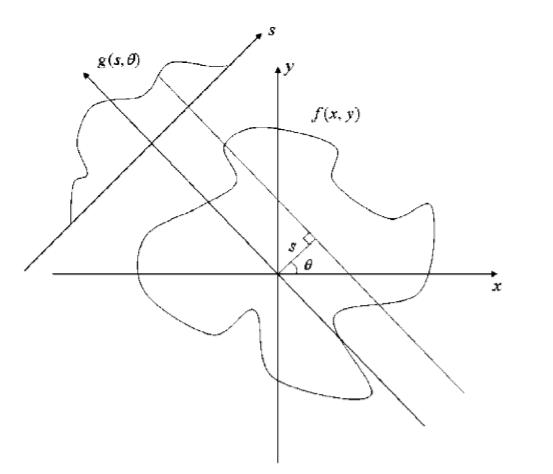


Рис. 2.3.1 – Інтеграл по напрямку

Перетворення Радона має такі корисні властивості, пов’язані з афінними перетвореннями зображення:

1. Зсув зображення на викликає зсув перетворення Радона на s:
2. Масштабування зображення в разів викликає масштабування перетворення в p разів.
3. Обертання зображення на кут викликає зсув перетворення:

На малюнку 2.3.2 зображено перетворення Радона відомого зображення Lena.

Рис. 2.3.2 – Перетворення Радона зображення Lena

Розглянемо афінні перетворення: зсув, масштабування (зі збереженням кутів) та поворот. З 1 властивості слідує що зсув зображення спричиняє зсув перетворення Радона, але розмір зсуву для кожної проекції є різним. Для забезпечення незмінності зсуву до кожної проекції перетворення застосовується нормалізована автокореляція.

Після використання автокореляції ми отримуємо сигнал , стійкий до зсувів. Серед афінних перетворень залишились масштабування і поворот. З 2 і 3 властивості перетворення Радона, автокореляція масштабованого і повернутого зображення представлена як де та – розмір масштабування та повороту відповідно. Для того щоб досягти стійкості до масштабування і повороту використовується перетворення log-mapping [[4]](#b4) та двомірне перетворення Фур’є. Перетворення log-mapping перетворює масштабування сигналу в зсув. Потім перетворення Фур’є відображає цей зсув в зміну фази. За допомогою log-mapping сигнал може бути записаний як

Потім використовується перетворення log mapping . Воно має перетворити масштабування в зсув.  
Потім використовується двомірне перетворення Фур’є.

Воно перетворює зсув зображення в зміну фази.

Потім потрібно вибрати перші 21х21 пікселя перетворення Фур’є. Найбільша енергія знаходиться в перших коефіцієнтах, решта є малими.

Потім потрібно розділити комплексні значення на модуль і аргумент.

До отриманих зображень потрібно застосувати 2Д фільтр .

В результуючих зображеннях замінити додатні значення на 1, а <=0 – на 0.

Об’єднати їх за допомогою двійкової функції виключаючого або. Отримані нулі і одиниці (біти) є відбитком.

Таким чином застосувавши перетворення Радона і log-mapping маштабування і обертання зображення буде перетворене в зсув, а за допомогою перетворення Фур’є зсув буде перетворено в зміну фази. Основна сила перетворення Фур’є зосереджена на початку, тому для створення відбитку буде обрано перші

## фівів

У цьому розділі розроблено математичне забезпечення системи розпізнавання емоцій. Для базових емоцій, які потрібно розпізнавати, спроектовано системи ANFIS, об’єднано їх за спільним шаром вхідних сигналів у MANFIS. Для вхідних сигналів у межах кожної емоції виділено множини нечітких значень, яких ці вхідні сигнали можуть набувати. На множині нечітких значень сформульовано правила продукції для виведення кожної емоції.

Установлення параметрів системи відбувається завдяки гібридному алгоритму навчання, застосованому до кожної підсистеми ANFIS виведення емоції. За критерій зупинки навчання взято погіршення ефективності узагальнення.

# Програмне забезпечення

## Структура програми

Програмне забезпечення для розв’язання поставленої задачі спроектовано з дотриманням парадигм об’єктно-орієнтованого програмування [16]. Використано шаблон проектування Модель-Подання-Пред’явник, відповідно до якого логіка роботи програми відділяється від візуального відображення даних у різні класи. На рисунку 4.1 представлено схему потоків даних між модулями використаного шаблону.

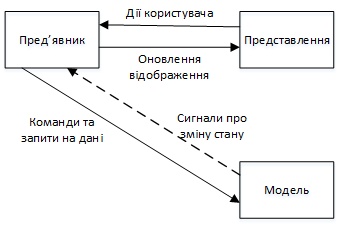


Рисунок 4.1 – Схема потоків даних між модулями MVP розробленого програмного забезпечення

Усю логіку роботи системи MANFIS зосереджено в модулі «Модель», у зв’язку з чим на нього покладаються такі функції:

* зчитування параметрів мережі з файлу;
* обчислення вихідних сигналів;
* здійснення однієї ітерації навчання;
* запис параметрів мережі у файл.

[…]

## Опис розроблених алгоритмів

До кожної підсистеми розробленої ANFIS застосовано гібридне навчання, описане в 3.2. Блок-схему алгоритму проведення такого навчання наведено на рисунку 4.2.

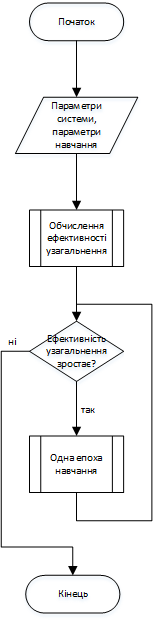


Рисунок 4.2 – Блок-схема алгоритму гібридного навчання ANFIS

[…]

## Формат вихідних даних

### Вихідні дані для ініціалізації системи

Вихідними даними для ініціалізації системи є файл із її параметрами weights.xml, що знаходиться в директорії sources.

Параметри у файлі відформатовано наступним чином:

1. у перших найглибших чотирьох тегах знаходяться передумовні параметри, у наступних найглибших тегах знаходяться наслідкові параметри;

передумовні параметри згруповано за елементами обличчя, ознаки яких вони описують;

кожний рядок передумовних параметрів містить параметри вузлової функції нейрона першого шару мережі, ,  та  відповідно;

наслідкові параметри згруповано за емоцією, яку вони описують:

1. здивування;
2. щастя;
3. сум;
4. відраза;
5. злість;

кожний рядок наслідкових параметрів складений із параметрів вузлової функції нейронів четвертого шару;

усі параметри — дійсні числа, ціла частина відділяється від дробової крапкою.

### Вихідні дані для навчання системи

[…]

## Формат результуючих даних

[…]

## Результати випробування

### Навчання ANFIS

[…]

### Випробування системи

[…]

## Висновки до розділу

У цьому розділі розроблено програмне забезпечення для автоматизованої системи розпізнання емоційного стану людини за статичним фронтальним зображенням її обличчя.

Реалізовану програмно систему навчено на вибірці зі 180 зображень.

У ході тестування обчислено показники ефективності навчання та тестування. Для навчання сукупно він склав 85%, для тестування сукупно — 73%.

Висновки

У роботі розглянуто основні підходи до розпізнавання емоційного стану людини за статичним фронтальним зображенням її обличчя: штучні нейронні мережі, приховані марківські моделі, порівняння з шаблоном та адаптивне нейронечітке виведення. У результаті проведеного порівняльного аналізу за наперед визначеними критеріями для вирішення поставленої задачі обрано модифікацію архітектури системи адаптивного нейронечіткого виведення — MANFIS.

Розглянуто й модифіковано методологію виділення вектора ознак із зображення обличчя. Для кожної з ознак уведено множину нечітких значень, яких вона може набувати. На множині цих нечітких значень побудовано правила продукції для ідентифікації кожної емоції.

Спроектоване математичне забезпечення реалізовано програмно.

Розроблену програмну систему навчено на вибірці зі 180 зображень гібридним алгоритмом із використанням перехресної перевірки для зупинки тренування.

У ході тестування виявлено показники ефективності: у середньому за всіма емоціями показник ефективності навчання склав 85%, а показник ефективності узагальнення 73%.

Перелік посилань

1. Экман П. Психология эмоций. Я знаю, что ты чувствуешь / П. Экман. — Питер, 2010. — 336 с.
2. Khandait S. P. Automatic Facial Feature Extraction and Expression Recognition based on Neural Network / S. P. Khandait, R. C Thool, P. D. Khandait // International Journal of Advanced Computer Science and Applications. — 2011. — Vol. 2, No. 1. — P. 113–118.
3. Khandait S. P. ANFIS and BPNN based Expression Recognition using HFGA for Feature Extraction / S. P. Khandait, R. C. Thool, P. D. Khandait // Bulletin of Electrical Engineering and Informatics. — 2013. — Vol. 2, No. 1. — P. 11–22.
4. Gomathi V. Human Facial Expression Recognition Using MANFIS Model / V. Gomathi, K. Ramar, A. S. Jeevakumar // Proceedings of World Academy of Science Engineering and Technology. — 2009. — 38. — P. 338–342.
5. I. Cohen. Emotion Recognition from Facial Expressions Using Multilevel HMM / I. Cohen, A. Garg, T. S. Huang // Neural Information Processing Systems. — 2000.
6. Єфімов Г. М. Технологія для моделювання та розпізнавання емоційної міміки на обличчі людини / Г. М. Єфімов // Системи обробки інформації. — 2012. — С. 36–39.
7. Jang J.-S. R. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence / J.-S. R. Jang, C.-T. Sun, E. Mizutani. — Prentice Hall, 1997. — 614 p.
8. Хайкин С. Нейронные сети: полный курс : пер. с англ. / C. Хайкин. — [2-е изд.]. — М. : Издательский дом «Вильямс». — 2006. — 1104 с.
9. N. Tsapatsoulis. A Fuzzy System for Emotion Classification Based on the MPEG-4 Facial Definition Parameter Set / N. Tsapatsoulis, K. Karpouzis, G. Stamou, F. Piat, S. Kollias // EURASIP Journal on Applied Signal Processing. — 2002. — P. 1021–1038.
10. Noldus Information Technology. Face Reader homepage [Electronic Resource]. — Mode of Access:

http://www.noldus.com/human-behavior-research/products/facereader

1. Visual Recognition. eMotion homepage [Electronic Resource]. — Mode of Access: http://www.visual-recognition.nl/eMotion.html
2. Face Analysis and Emotion Recognition [Electronic Resource]. — Mode of Access: http://www.amiproject.org/showcase/still-and-moving-image-processing/emotion-recognition
3. Affdex homepage [Electronic Resource]. — Mode of Access: http://www.affdex.com/
4. Ekman P. Manual of the Facial Action Coding System (FACS) / P. Ekman, W. V. Friesen. — Palo Alto : Consulting Psychologists Press, 1978.
5. Ekman P. Facial Action Coding System: Investigator’s Guide / P. Ekman, W. V. Friesen. — Palo Alto : Consulting Psychologists Press, 1978.
6. С. Прата. Язык программирования С++. Лекции и упражнения : пер. с англ. / C Прата. — 6-е изд. — М. : Издательский дом «Вильямс». — 2013. — 1248 с.
7. Лістинги програм

Лістинг файлу presenter.h — прототип пред’явника-вчителя

#ifndef PRESENTER\_H

#define PRESENTER\_H

#include <QObject>

#include "View/iView.h"

#include "Network/network.h"

//#include <vector>

class Presenter: public QObject

{

Q\_OBJECT

public:

Presenter();

Presenter(iView\* userInterface);

signals:

void result\_returned(QVector<qreal>);

void message\_showing(QString str);

void message\_sending(QString);

void message\_hiding();

private slots:

//from view

void testing(QVector<qreal> features);

void training(QVector<QPair<QVector<qreal>, QVector<qreal> > >& sample);

private:

QVector<qreal> \_normed(QVector<qreal>);

QVector<qreal> \_trunk(QVector<qreal>);

qreal \_effectiveness(Emotions::Emotions em);

QVector<QPair<QVector<qreal>, QVector<qreal> > > \_training\_data;

QVector<QPair<QVector<qreal>, QVector<qreal> > > \_checking\_data;

qreal \_eps;

Network ANFIS;

static long long int iteration;//

};

#endif // PRESENTER\_H

Лістинг файлу presenter.cpp — реалізація пред’явника-вчителя

#include "Presenter/presenter.h"

#include "Presenter/simplex.h"

#include <qmath.h>

#define PARAM\_NUMBER 519

long long int Presenter::iteration = 0;

Presenter::Presenter()

{

}

Presenter::Presenter(iView\* i\_userInterface)

{

QObject \*userInterface = dynamic\_cast<QObject\*>(i\_userInterface);

connect(userInterface, SIGNAL(test\_inputed(QVector<qreal>)),this, SLOT(testing(QVector<qreal>)));

connect(userInterface, SIGNAL(train\_sample\_inputed(QVector<QPair<QVector<qreal>,QVector<qreal> > >& )),this, SLOT(training(QVector<QPair<QVector<qreal>,QVector<qreal> > >&)));

connect(this, SIGNAL(result\_returned(QVector<qreal>)), userInterface, SLOT(show\_results(QVector<qreal>)));

connect(this, SIGNAL(message\_showing(QString)), userInterface, SLOT(show\_message(QString)));

connect(this, SIGNAL(message\_hiding()), userInterface, SLOT(hide\_message()));

connect(this, SIGNAL(message\_sending(QString)), userInterface, SLOT(show\_dlg\_message(QString)));

ANFIS.Init();

ANFIS.Deserialize("C:\\Users\\Sersajur\\Documents\\Qt-projects\\Emotion\_recognizer\\src\\sources\\weights.xml");

}

QVector<qreal> Presenter::\_trunk(QVector<qreal> result)

{

for(auto& it: result)

if (it<0) it=0;

else if(it>1) it=1;

return result;

}

QVector<qreal> Presenter::\_normed(QVector<qreal> features)

{

QVector<qreal> result = features;

qreal& Horizontal = features[13];

qreal& Vertical = features[19];

//eye\_brows\_size\_params:

result[0] /= Vertical;

result[1] /= Horizontal;

result[2] /= Vertical;

result[3] /= Horizontal;

//eyes\_size\_params:

result[4] /= Vertical;

result[5] /= Horizontal;

result[6] /= Vertical;

result[7] /= Horizontal;

//nose\_size\_params:

result[8] /= Vertical;

result[9] /= Horizontal;

//lips\_size\_params:

result[10] /= Vertical;

result[11] /= Horizontal;

//distances:

result[12] /= Horizontal;

result[14] /= Vertical;

result[15] /= Vertical;

result[16] /= Vertical;

result[17] /= Vertical;

result[18] /= Vertical;

//result[19] /= Vertical;

return result;

}

void Presenter::testing(QVector<qreal> features)

{

QVector<qreal> result;

result = ANFIS.Check(\_normed(features));

emit result\_returned(\_trunk(result));

}

QString makeLog(int V/\*, int N\*//\*, qreal eps\*/, QVector<QString> em, QVector<int> n, QVector<qreal> e)

{

QString log = QString("Об'єм вибірки: %1\n").arg(V);

log += QString("Критерій зупинки: перехресна перевірка \n ");

log += QString("------------------------------------\n");

log += QString("Емоція || Кількість епох || Ефективність \n");

log += QString("%1 %2 %3\n").arg(em[0],-14).arg(n[0],-19).arg(e[0],-9,'f');

log += QString("%1 %2 %3\n").arg(em[1],-14).arg(n[1],-19).arg(e[1],-9,'f');

log += QString("%1 %2 %3\n").arg(em[2],-14).arg(n[2],-19).arg(e[2],-9,'f');

log += QString("%1 %2 %3\n").arg(em[3],-14).arg(n[3],-19).arg(e[3],-9,'f');

log += QString("%1 %2 %3\n").arg(em[4],-14).arg(n[4],-19).arg(e[4],-9,'f');

return log;

}

qreal Presenter::\_effectiveness(Emotions::Emotions em)

{

qreal result = 0;

for(auto& it: \_checking\_data)

{

ANFIS.setInput(it.first);

result += qAbs(it.second[em]-ANFIS.Process(em))/\_checking\_data.size()/it.second[em];

}

return (1 - result);

}

void Presenter::training(QVector<QPair<QVector<qreal>, QVector<qreal> > > &sample)

{

for (auto& row: sample)

row.first = \_normed(row.first);

int k = qRound(sample.size()/qSqrt(PARAM\_NUMBER\*2));

for(int i=0; i < sample.size() - k; i++)

\_training\_data.append(sample[i]);

for(int i = \_training\_data.size(); i < sample.size(); i++)

\_checking\_data.append(sample[i]);

qreal l\_rate\_cons = 0.2;

qreal l\_rate\_prem\_a = 0.2;

qreal l\_rate\_prem\_b = 0.2;

qreal l\_rate\_prem\_c = 0.2;

int epoch\_counter;

qreal prev\_effect;

QVector<qreal> save\_param;

QVector<QString> str\_status = {"здивування", "щастя", "сум", "відраза", "злість"};

QVector<int> n\_epoch;

QVector<qreal> effectivenesses;

for(int i = 0; i < str\_status.size(); i++)

{

epoch\_counter = 0;

prev\_effect = \_effectiveness(Emotions::Emotions(i));

while((prev\_effect <= \_effectiveness(Emotions::Emotions(i))) || !epoch\_counter)

{

save\_param = ANFIS.getParams();

prev\_effect = \_effectiveness(Emotions::Emotions(i));

message\_showing("Засвоєння емоції: " + str\_status[i] + ", " + QString::number(epoch\_counter) + ", ефективність" + QString::number(prev\_effect,'f'));

for(auto& it: \_training\_data)

{

ANFIS.setInput(it.first);

ANFIS.One\_lesson(Emotions::Emotions(i), it.second[i], l\_rate\_cons, l\_rate\_prem\_a, l\_rate\_prem\_b, l\_rate\_prem\_c);

}

epoch\_counter++;

}

ANFIS.setParams(save\_param);

//prev\_effect = \_effectiveness(Emotions::Emotions(i));//

n\_epoch.append(epoch\_counter);

effectivenesses.append(prev\_effect);

}

emit message\_hiding();

emit message\_sending(makeLog(\_training\_data.size()/\*,max\_epoch\_number\*//\*,\_eps\*/,str\_status,n\_epoch,effectivenesses));

ANFIS.Serialize("C:\\Users\\Sersajur\\Documents\\Qt-projects\\Emotion\_recognizer\\src\\sources\\weights.xml", 6);

}

1. Ілюстративний матеріал

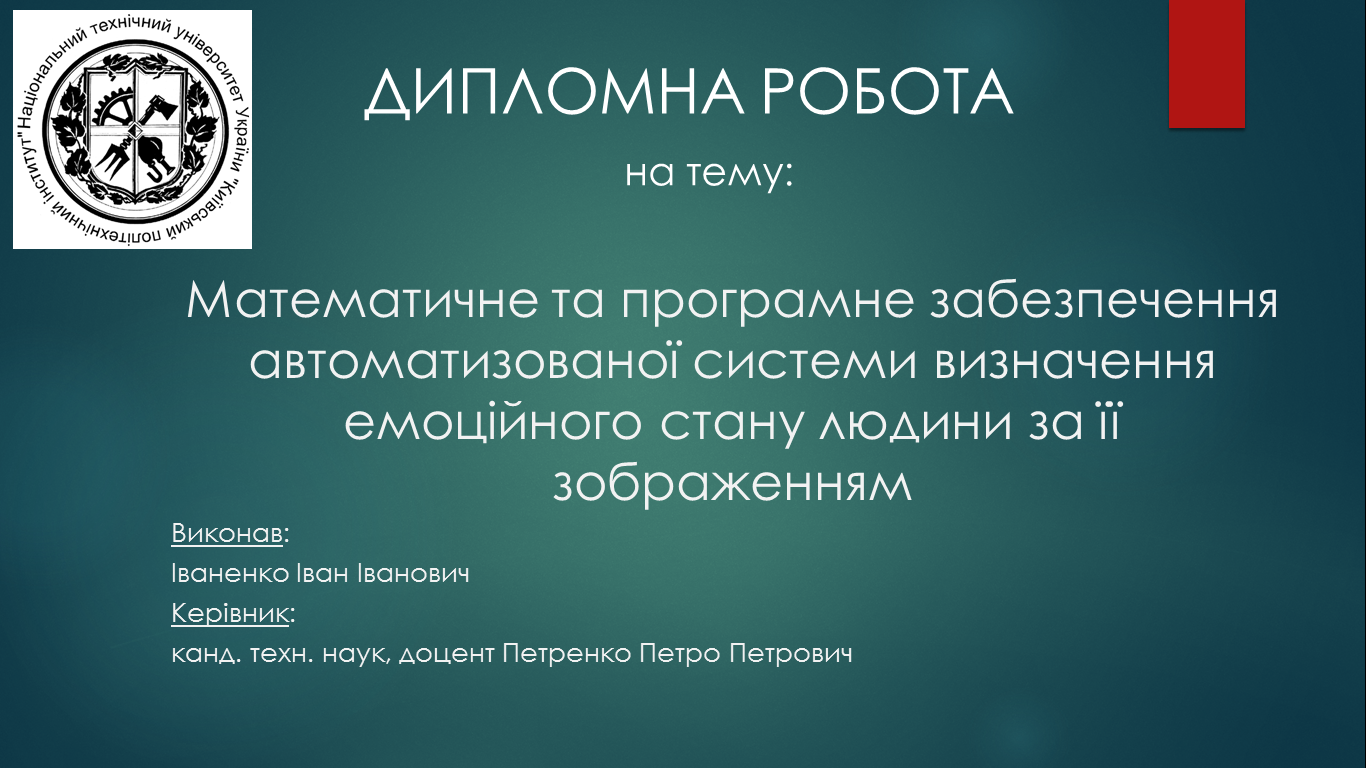


Рисунок Б.1 – Слайд 1



Рисунок Б.2 – Слайд 2