

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

Кафедра комп'ютерних наук

**Теорія розпізнавання образів та класифікації в системах
штучного інтелекту**

Лабораторна робота №2

Виконав:

Студент групи КН-20002Б

Кропивка Анатолій Анатолійович

Київ 2023

Тема: Дослідження методу потенційних функцій у розпізнаванні образів.

Мета: Засвоєння базових знань щодо способів класифікації образів на основі методу потенційних функцій. Розробка програмної системи, що реалізує розпізнавання образів на основі зазначеного методу. Дослідження розробленої системи. Отримання практичних навичок з розробки програмних систем розпізнавання образів.

Підготовка до роботи: Вивчити й уявити теоретичні відомості щодо способів класифікації образів на основі методу потенційних функцій.

Хід роботи:

1. Розробити програмний застосунок на мові програмування C++, що реалізує розпізнавання образів на основі зазначеного методу.
2. Дослідити функціонування розробленого програмного застосунку на прикладах.
3. За результатами досліджень скласти звіт з описом отриманих результатів та обґрунтованими висновками.

Завдання: Класифікувати довільні реальні образи (об'єкти) – банки, фірми, інвестиційні фонди, кредитні спілки тощо за кількома показниками. Наприклад, власний капітал, прибуток і тому подібне.

Опис програми

Програма дозволяє провести дослідження та оцінку ефективності методу потенційних функцій у розпізнаванні образів. Цей метод використовується для класифікації образів на основі обчислення потенційних функцій, які відображають ступінь схожості між вхідним образом та кожним класом.

Програма обчислює потенційність функцій між кожним вхідним образом і еталоном за допомогою відповідного алгоритму порівняння. Результати розпізнавання відображаються користувачу. Для кожного вхідного образу вказується його клас, а також ступінь схожості з еталоном.

Основа програми будується на попередній лабораторній роботі.

Код програми

```
#include <Windows.h>

#include <chrono>
#include <cmath>
#include <cstdlib>
#include <format>
#include <iostream>
#include <stdexcept>
#include <string>
#include <thread>
#include <vector>

using namespace std;

double get_double()
{
    double value {};
    string str {};

    try {
        getline(cin, str);

        size_t position {};
        value = stod(str, &position);

        if (position != str.size()) {
            throw runtime_error("There are characters after the number!");
        }
        if (value < 0) {
            throw runtime_error("The entered value cannot be negative");
        }
    }
```

```

    }
} catch (const exception& e) {
    cout << format("Exception: {}. The return value is 0!", e.what()) << endl;

    value = 0;
}

return value;
}

```

```

class Animal {
protected:
    double height;
    double weight;

public:
    Animal()
        : height {}
        , weight {}
    {
    }

    Animal(double height, double weight)
        : height(height)
        , weight(weight)
    {
    }

    const double get_height() const
    {
        return height;
    }
}

```

```

const double get_weight() const
{
    return weight;
}

Animal& from()
{
    cout << "Enter the data about an animal." << endl;

    cout << "Height: ";
    height = get_double();

    cout << "Weight: ";
    weight = get_double();

    cout << endl;
    return *this;
}
};

```

```

class Animals {
private:
    vector<Animal> group;
    const char* name;
    double ave_hg;
    double ave_wg;

public:
    Animals(const char* name, vector<Animal> animals)
        : name(name)
        , group(animals)

```

```

        , ave_hg {}
        , ave_wg {}
    {
        for (auto& animal : group) {
            ave_hg += animal.get_height();
            ave_wg += animal.get_weight();
        }

        ave_hg /= group.size();
        ave_wg /= group.size();
    }

    void output() const
    {
        cout << format("Average value of {}. ", name) << endl;
        cout << format("Height: {}", ave_hg) << endl;
        cout << format("Weight: {}", ave_wg) << endl;
        cout << endl;
    }

    const double get_ave_hg() const
    {
        return ave_hg;
    }

    const double get_ave_wg() const
    {
        return ave_wg;
    }
};

double id(Animals& group, Animal& obj)

```

```

{
    double x { group.get_ave_wg() - obj.get_weight() };
    double y { group.get_ave_hg() - obj.get_height() };

    return sqrt(pow(x, 2) + pow(y, 2));
}

```

```

double potential(double id)
{
    return 1 / (1 + pow((0.01 * id), 2));
}

```

```

int main()
{
    thread task([]() {
        while (true) {
            this_thread::sleep_for(chrono::milliseconds(100));

            if (GetAsyncKeyState(VK_ESCAPE)) {
                exit(0);
            }
        }
    });
}

```

```

Animals cats("cats",
    std::vector<Animal> {
        Animal(20, 4),
        Animal(24, 4.2),
        Animal(22, 4.6),
        Animal(31, 5),
        Animal(27, 5.1),
        Animal(29, 4.4),
    }
);

```



```
});
```

```
Animals dogs("dogs",  
    std::vector<Animal> {  
        Animal(40, 6),  
        Animal(60, 12),  
        Animal(100, 20),  
        Animal(55, 9),  
        Animal(69, 15),  
        Animal(36, 7),  
    });
```

```
cats.output();  
dogs.output();
```

```
while (true) {  
    cout << endl;  
    cout << "The data of the test object." << endl;  
    Animal obj { Animal().from() };
```

```
    double id_cat { id(cats, obj) };  
    double id_dog { id(dogs, obj) };
```

```
    cout << format("Cat's id number is (similarity with the ethalon): {}", id_cat) << endl;  
    cout << format("Dog's id number is (similarity with the ethalon): {}", id_dog) << endl;
```

```
    double p_cat { potential(id_cat) };  
    double p_dog { potential(id_dog) };
```

```
    cout << format("The potential to be a cat is: {}", p_cat) << endl;  
    cout << format("The potential to be a dog is: {}", p_dog) << endl;
```

```
    if (p_cat > p_dog) {  
        cout << "The test object is cat" << endl;  
    } else {  
        cout << "The test object is dog" << endl;  
    }  
}  
  
return 0;  
}
```

Знімки екрану

```
PS D:\TheoryOfRecognizeImages\MyLabs\lab_2> ./bin/main.exe
Average value of cats.
Height: 25.5
Weight: 4.55

Average value of dogs.
Height: 60
Weight: 11.5

The data of the test object.
Enter the data about an animal.
Height: 13.334
Weight: 6.07

Cat's id number is (similarity with the ethalon): 12.260585467260526
Dog's id number is (similarity with the ethalon): 46.98085201441114
The potential to be a cat is: 0.985190424830698
The potential to be a dog is: 0.8191886449349547
The test object is cat

The data of the test object.
Enter the data about an animal.
Height: 100
Weight: 50

Cat's id number is (similarity with the ethalon): 87.26942477179507
Dog's id number is (similarity with the ethalon): 55.51801509420163
The potential to be a cat is: 0.5676672890665435
The potential to be a dog is: 0.7643945040035163
The test object is dog

The data of the test object.
Enter the data about an animal.
Height:
PS D:\TheoryOfRecognizeImages\MyLabs\lab_2> 
```

Висновок: Програма надає можливість дослідити та експериментувати з методом потенційних функцій у розпізнаванні образів. Вона дозволяє обчислювати потенційні функції, класифікувати образи та аналізувати результати.