

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

Кафедра комп'ютерних наук

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

Лабораторна робота №3

Виконав:

Студент групи КН-20002Б

Кропивка Анатолій Анатолійович

Київ 2023

Тема: Дослідження методу розподіляючих функцій у розпізнаванні образів.

Мета: Засвоєння базових знань щодо способів класифікації образів на основі методу розподіляючих 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 розподіляючих 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 розподіляючих функцій у розпізнаванні образів. Цей метод застосований на моделюванні розподілу вхідних образів для кожного класу його для класифікації нових образів.

Після моделювання розподілу, програма може використовувати метод класифікації для визначення класу до нових вхідних образів. За допомогою моделі розподілу, програма обчислює ймовірності належності образу до різних класів і вибирає найбільш ймовірний для класифікації

Основа програми будується на попередніх лабораторних роботах.

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:
```

```
    const char* name;
```

```
    double ave_hg;
```

```
    double ave_wg;
```

```
    vector<Animal> group;
```

```
public:
```

```
    Animals(const char* name, vector<Animal> animals)
```

```
        : name(name)
```

```
        , group(animals)
```

```

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

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

class Aim {

```

private:

double x;

double y;

double k;

void constant(Animals& cats, Animals& dogs)

{

$k = (\text{cats.get_ave_hg}() - \text{dogs.get_ave_hg}()) / (\text{cats.get_ave_wg}() - \text{dogs.get_ave_wg}());$

}

public:

Aim(Animals& cats, Animals& dogs)

{

 constant(cats, dogs);

$x = (\text{cats.get_ave_wg}() + \text{dogs.get_ave_wg}()) / 2;$

$y = (\text{cats.get_ave_hg}() + \text{dogs.get_ave_hg}()) / 2;$

}

const double distribution(Animal& obj)

{

$\text{return obj.get_height}() + (\text{obj.get_weight}() - x) / k - y;$

}

void output()

{

 cout << format("Constant: {}", k) << endl;

 cout << format("M(x0, y0): M({}, {})", x, y) << endl;

 cout << endl;

}

};


```

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

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

```

```

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

```

```
cats.output();  
dogs.output();  
  
Aim aim(cats, dogs);  
aim.output();  
  
while (true) {  
    cout << endl;  
    cout << "The data of the test object." << endl;  
    Animal obj { Animal().from() };  
  
    double value { aim.distribution(obj) };  
    cout << format("Distribution function: {}", value) << endl;  
  
    if (value < 0) {  
        cout << "The test object is cat" << endl;  
    } else {  
        cout << "The test object is dog" << endl;  
    }  
}  
  
return 0;  
}
```

Знімки екрану

```
PS D:\TheoryOfRecognizeImages\MyLabs\lab_3> ./bin/main.exe
Average value of cats.
Height: 25.5
Weight: 4.55

Average value of dogs.
Height: 60
Weight: 11.5

Constant: 4.9640287769784175
M(x0, y0): M(8.025, 42.75)

The data of the test object.
Enter the data about an animal.
Height: 12.89
Weight: 4.4

Distribution function: -30.590253623188403
The test object is cat

The data of the test object.
Enter the data about an animal.
Height: 88
Weight: 33.3

Distribution function: 50.341630434782616
The test object is dog

The data of the test object.
Enter the data about an animal.
Height: #10
Exception: stod. The return value is 0!
Weight: 15%
Exception: There are characters after the number!. The return value is 0!

Distribution function: -44.36663043478261
The test object is cat
```

Висновок: Програма надає можливість дослідити та експериментувати з методом розподіляючих функцій у розпізнаванні образів. Вона сприяє кращому розумінню теорії розпізнавання образів та надає можливість провести практичне дослідження методу розподіляючих функцій у реальних задачах класифікацій образів.