Ministry of Education and Science of Ukraine

Odessa National Academy of Food Technologies

International Competition of Student Scientific Works

BLACK SEA SCIENCE 2020

Information Technology, Automation and Robotics

Proceedings

Odessa, ONAFT 2020

UDC 004.01/08

Black Sea Science 2020: Proceedings of the International Competition of Student Scientific Works. Information Technology, Automation and Robotics. / Odessa National Academy of Food Technologies; B.Yegorov, M. Mardar, S.Kotlyk (editors-in-chief.) [*et al.*]. – Odessa: ONAFT, 2020. – 365 p.

These materials of International Competition of Student Scientific Works «Black Sea Science 2020» contain the works of the contest participants in the section «Information technologies, automation and robotics» (not winners).

The author of the work is responsible for the accuracy of the information.

Odessa National Academy of Food Technologies, 2020.

Organizing committee:

Prof. Bogdan Iegorov, D.Sc., Rector of Odessa National Academy of Food Technologies, Head of the Committee

Prof. Maryna Mardar, D.Sc., Vice-Rector for Scientific and Pedagogical Work and International Relations of Odessa National Academy of Food Technologies, Deputy Head of the Committee

Prof. Stefan Dragoev, D.Sc., Vice-Rector for Scientific Work and Business Partnerships of University of Food Technologies (Bulgaria)

Prof. Baurzhan Nurakhmetov, D.Sc., First Vice-Rector of Almaty Technological University (Kazakhstan)

Prof. Mircea Bernic, Dr. habil., Vice-Rector for Scientific Work of Technical University of Moldova (Moldova)

Prof. Jacek Wrobel, Dr. habil., Rector of West Pomeranian University of Technology (Poland)

Prof. Michael Zinigrad, D.Sc., Rector of Ariel University (Israel)

Dr. Mei Lehe, Ph.D., Vice-President of Ningbo Institute of Technology, Zhejiang University (China)

Prof. Plamen Kangalov, Ph.D., Vice-Rector for Academic Affairs of “Angel Kanchev” University of Ruse (Bulgaria)

Dr. Alexander Sychev, Ph.D., Assoc. Professor of Sukhoi State Technical University of Gomel (Belarus)

Dr. Hanna Lilishentseva, Ph.D., Assoc. Professor, Head of the Department of Merchandise of Foodstuff of Belarus State Economic University (Belarus)

Prof. Heinz Leuenberger, Ph.D., Professor of the Institute of Ecopreneurship of University of Applied Sciences and Arts (Switzerland)

Prof. Edward Pospiech, Dr. habil., Professor of the Institute of Meat Technology of Poznan University of Life Sciences (Poland)

Prof. Lali Elanidze, Ph.D., Professor of the Faculty of Agrarian Sciences of Iakob Gogebashvili Telavi State University (Georgia)

Dr. V. Kozhevnikova, Ph.D., Senior Lecturer of the Department of Hotel and Catering Business of Odessa National Academy of Food Technologies, Secretary of the Committee

The jury for the section «Information technologies, automation and robotics»

Head of the jury:

Serhiy Kotlyk – Ph.D., Associate Professor, Director of the P.M. Platonov Educational-Scientific Institute of Computer Systems and Technologies "Industry 4.0" of Odessa National Academy of Food Technologies

Members of the jury:

Francisco Augusto – Dr., International Relations Manager of Higher Institute of Information and Communication Technologies (Angola)

Andrey Kuprijanov – Ph.D., Associate Professor of the Department of Software for Computers and Automated Systems of Belarusian National Technical University (Belarus)

Simon Milbert – Vice-President of Xtra Information Management, Inc. (USA)

Ivan Palov – D.Sc., Professor of University of Ruse "Angel Kanchev" (Bulgaria)

Gerard H. Degla – Communications and Training Manager of "MAPCOM solutions informatiques" company group (Benin)

Viktor Khobin – D.Sc., Professor, Head of the Department of Technological Processes Automation and Robotic Systems of Odessa National Academy of Food Technologies

Valerii Levinskyi – Ph.D., Associate Professor of the Department of Technological Processes Automation and Robotic Systems of Odessa National Academy of Food Technologies

Viktor Yehorov – Ph.D., Supervisor of the Laboratory of Mechatronics and Robotics of Odessa National Academy of Food Technologies

Valeriy Plotnikov – D.Sc., Professor, Head of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies

Pavlo Lomovtsev – Ph.D., Associate Professor of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies

Yurii Kornienko – Ph.D., Associate Professor of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies

Sergii Artemenko – D.Sc., Professor, Head of the Department of Computer Engineering of Odessa National Academy of Food Technologies

Serhii Shestopalov – Ph.D., Associate Professor of the Department of Computer Engineering of Odessa National Academy of Food Technologies

Secretary of the jury:

Oksana Sokolova – Senior Lecturer of the Department of Advanced and Applied Mathematics of Odessa National Academy of Food Technologies

TABLE OF CONTENTS

AUTOMATED EMERGENCY CALL SYSTEM BASED ON SOUND INCIDENT RECOGNITION

Authors: Ruslan Kakatsiy, Valerii Stadnichuk, Kristina Tytarenko
Supervisor: Sergiy Krivenko
Mariupol State University (Ukraine)

Abstract. *An automated emergency call system based on the recognition of sound incidents has significant advantages over a video surveillance system. The review focuses on systems that use microphones to quickly respond to an emergency. The material of this research work is based on an analysis of publications on the topics of sound recognition, the use of WSN. Also, the program has been developed, which allows convenient and practical positioning microphones on the cards.*

The practical significance of the work is obvious. The results of the study can be used to create an automated emergency call system based on the recognition of sound incidents.

The objectives of the research work are: obtaining new results that are important for the implementation of the emergency call system based on the recognition of sound incidents; mastering the principles of sound recognition using neural networks.

Keywords: *emergency call, acoustic surveillance***,** *Wireless sensor network***,** *sampling, neural networks.*

1. Introduction

Why do we need an automated emergency call system based on the recognition of sound incidents? With the development of society, crime also develops. More and more, there is a need to protect people from attacks on their lives and property. The system allows you to identify violators in case of emergency.

The development of computer technology and artificial intelligence systems has led to the fact that now more and more information is analyzed automatically. Sound has many more advantages than even a good camcorder. The microphone does not have to be within line of sight, it does not have dead zones. Moreover, a good microphone is much cheaper than a good camera. In case of sound observation for many situations, just a simple analysis of the sound level and spectrum is enough, while video requires non-trivial computer vision algorithms.

The sound surveillance system is convenient because it does not take much time. If the system detects the sound of an emergency (screaming, shot, sudden braking), all information is sent to the police and to the company's servers for analysis, to reduce the number of false positives, for example, from firecrackers.

Acoustic sensors located at the intersection can provide accurate information about the number, speed of passing cars, collisions, accidents, etc.

Using microphones as surveillance will allow you to respond to emergencies much faster than using cameras.

2. Analytical review of the literature

The emergency call system is used to automatically alert emergency services of any incidents and provide timely medical assistance to victims. The use of an emergency call system based on the recognition of sound incidents can significantly reduce the level of injuries during incidents.

The system recognizes the sound of an emergency. Then she scans all available GSM networks and selects a channel for sending SMS messages about the incident. The system automatically contacts the emergency call center and provides information on the location of the received sound.

To obtain more accurate data, it is advisable to use WSN. Wireless sensor network (WSN) is a distributed, self-organizing network of many sensors (sensors) and actuators, interconnected via a radio channel [1]. According to work M. Brandstein and D. Ward "Microphone arrays: signal processing techniques and applications"[2], multi-microphone recordings enable to exploit spatial diversity, allowing to localize target sound sources and/or to cancel out interfering sound sources coming from certain directions.

A wireless sensor network (BSN) consists of sensor nodes that are tightly deployed, where each node has a sensor, processor, transmitter, and receiver. These nodes are inexpensive low-power and multi-functional devices for performing various probing tasks. Sensor nodes are deployed throughout the area to monitor certain events (for example, temperature) in real conditions. BSN mainly operate in an open and uncontrolled area. They are expected to play an important role in various areas, for example, military surveillance, forest fire monitoring, building security monitoring and process control. Most applications require a more accurate localization process for nodes in order to obtain their coordinates within the network. [3].

To write a program most practical disposing the microphones on the map, we used a scientific article. Alexander Bertrand clarifies in his work: in the case of signal estimation (also referred to as signal enhancement), the goal is to estimate a desired signal (e.g., a speech signal), while suppressing background noise and/or removing reverberant components. This usually relies on fusion of the recorded signals at different nodes, requiring transmission of audio signals. In the case of parameter estimation, the goal is to extract certain parameters from the recorded audio signal(s), such as the location or identity of speakers, the acoustic properties of a room, or speech features. In this case, the nodes may only exchange parameter vectors or energy measurements at a slow time-scale compared to the sampling rate of the microphones [4].

As for the sound recognition system itself. Theoretically, machine speech recognition, that is, its automatic presentation in the form of text, is an extreme degree of compression of the speech signal. Each sound has a complex wave structure, including various frequencies and vibrations. Sound is an air vibration, the frequency of which lies in the range of frequencies perceived by man. Sound vibrations lie in the range of 16-20000 Hz [5].

But how to turn sound waves into numbers? Sound waves are one-dimensional. At each moment in time, they have one value, depending on the amplitude of the wave. In order to turn a sound wave into numbers, we record the values of the wave amplitude at equally spaced points. This process is called sampling. We read the data thousands of times per second and write down the numbers corresponding to the amplitude of the sound wave at that moment in time. Uncompressed .wav audio files are obtained [6].

Thanks to Kotelnikov's theorem[7], we know that for a perfect reconstruction of the original sound wave, it's enough to use a sampling frequency twice the highest frequency of the recorded sound [6].

The formulation of the Kotelnikov theorem: if an analog signal has a compact (limited in width) spectrum, then it can be uniquely and losslessly reconstructed from its discrete samples taken with a frequency strictly exceeding twice the upper frequency [7].

The physical meaning of the theorem can be explained in the following words: if you need to transmit a certain signal, then it is not necessary to transmit it in its entirety. You can transmit its instant impulses.

To create a system that will learn from its mistakes, you need to use neural networks. In essence, a neural network is a learning system. It acts in accordance with not only a given algorithm and formulas, but also based on experience. Neural networks are models of biological neural networks of the brain in which neurons are imitated by relatively simple, often of the same type, elements (artificial neurons) [8]. A neural network is a collection of neurons that make up the layers. In each layer, neurons are not interconnected, but are connected with neurons of the previous and next layers. Information comes from the first to the second layer, from the second to the third, etc. To train a neural network, a large amount of information is needed: the more layers and neurons on each layer, the less errors and the higher the reliability of the network. However, if you build too large a network, you may encounter a decrease in productivity and an increase in the complexity of the model [9].

Le N.V. and Panchenko D.P. in their article on speech recognition based on artificial neural networks described a speech recognition model based on artificial neural networks[10].

3. Object, subject and methods of research

The object of this study is an emergency call system.

Subject of research: a method for creating an automated emergency call system based on recognition of sound incidents.

In the process of writing a research paper, we used research methods such as: analysis and generalization of specialized literature, publications in periodicals devoted to BSN, neural networks; creating a working program, capable, given the air resistance, the range of microphones, to place them on the map. A general description was also given of the emergency call system.

4. Work results

The first thing we managed to achieve was the creation of a program that would place the microphones on the card in the most practical way. Considering factors that can affect sound distortion. Such as air resistance, the reflection of sound from buildings, the attenuation of sound when propagating on the ground, and so on.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace SoundArea
{
   public partial class Form1 : Form
   {
      Graphics gr;
      Bitmap bitm;
      bool fl = true;
      public Random rnd = new Random();
      List<Figure> figuresList = new List<Figure>();
      public Form1()
      {
         InitializeComponent();
         textBox1.Text = "100";
      }
      public void ProceduralPixelCircleDraw(Graphics graphics, Pen pen, Rectangle rect, Bitmap bm)
      {
         Point Center = new Point(rect.X + rect.Width / 2, rect.Y + rect.Height / 2);
         int R = rect.Width / 2;
         for (int x = -R; x <= R; x++)
         {
            bool flag = true;
            int y = (int)(Math.Sqrt(R * R - x * x));
            if (x + Center.X > 0 && y + Center.Y > 0 && x + Center.X < bm.Width && y + Center.Y < bm.Height)
            {
               int x_r = x + Center.X;
               int y_r = y + Center.Y;
               int xn = x_r, xk = Center.X;
               if (x_r > Center.X)
               {
                  xn = Center.X;
                  xk = x_r;
               }
               for (int a = xn; (a < xk) && flag; a++)
               {
                  int b = ((a - Center.X) * (y_r - Center.Y) / (x_r - Center.X)) + Center.Y;
                  if (bm.GetPixel(a, Math.Abs(b)).R == 0 && bm.GetPixel(a, Math.Abs(b)).B == 0 && bm.GetPixel(a, Math.Abs(b)).G == 0)
                     flag = false;
               }
               if (flag)
                  bm.SetPixel(x + Center.X, y + Center.Y, pen.Color);
            }
         }
```

```
for (int x = R; x >= -R; x--)
{
    bool flag = true;
    int y = (int)(Math.Sqrt(R * R - x * x));
    if (x + Center.X > 0 && y + Center.Y > 0 && x + Center.X < bm.Width
&& y - Center.Y < bm.Height)
    {
        int x_r = x + Center.X;
        int y_r = Math.Abs(y - Center.Y);
        int xn = x_r, xk = Center.X;
        if (x_r < Center.X)
        {
            xn = Center.X;
            xk = x_r;
        }
        for (int a = xn; a > xk && flag; a--)
        {
            int b = ((a - Center.X) * (y_r - Center.Y) / (x_r - Center.X)) +
Center.Y;
            if (bm.GetPixel(a, Math.Abs(b)).R == 0 && bm.GetPixel(a,
Math.Abs(b)).B == 0 && bm.GetPixel(a, Math.Abs(b)).G == 0)
                flag = false;
        }
        if (flag)
            bm.SetPixel(x + Center.X, Math.Abs(y - Center.Y), pen.Color);
    }
}
for (int y = -R; y <= R; y++)
{
    bool flag = true;
    int x = (int)(Math.Sqrt(R * R - y * y));
    if (x + Center.X > 0 && y + Center.Y > 0 && x + Center.X < bm.Width
&& y + Center.Y < bm.Height)
    {
        int y_r = y + Center.Y;
        int x_r = x + Center.X;
        int yn = y_r, yk = Center.Y;
        if (y_r > Center.Y)
        {
            yn = Center.Y;
            yk = y_r;
        }
        for (int b = yn; b < yk && flag; b++)
        {
            int a = (b - Center.Y) * (x_r - Center.X) / (y_r - Center.Y) +
Center.X;
```

```
                        if (bm.GetPixel(a, Math.Abs(b)).R == 0 && bm.GetPixel(a, Math.Abs(b)).B == 0 && bm.GetPixel(a, Math.Abs(b)).G == 0)
                            flag = false;
                    }
                    if (bm.GetPixel(x + Center.X, y + Center.Y).R != 0 && bm.GetPixel(x + Center.X, y + Center.Y).B != 0 && bm.GetPixel(x + Center.X, y + Center.Y).G != 0)
                        if (flag)
                            bm.SetPixel(x + Center.X, y + Center.Y, pen.Color);
                }
            }
            for (int y = R; y >= -R; y--)
            {
                bool flag = true;
                int x = (int)(Math.Sqrt(R * R - y * y));
                if (x + Center.X > 0 && y + Center.Y > 0 && x + Center.X < bm.Width && y + Center.Y < bm.Height)
                {
                    int y_r = y + Center.Y;
                    int x_r = Math.Abs(x - Center.X);
                    int yn = y_r, yk = Center.Y;
                    if (y_r < Center.Y)
                    {
                        yn = Center.Y;
                        yk = y_r;
                    }
                    for (int b = yn; b > yk && flag; b--)
                    {
                        int a = (b - Center.Y) * (x_r - Center.X) / (y_r - Center.Y) + Center.X;
                        if (bm.GetPixel(a, Math.Abs(b)).R == 0 && bm.GetPixel(a, Math.Abs(b)).B == 0 && bm.GetPixel(a, Math.Abs(b)).G == 0)
                            flag = false;
                    }
                    if (flag)
                        bm.SetPixel(Math.Abs(x - Center.X), y + Center.Y, pen.Color);
                }
            }
        }
        private void ProceduralCircleDraw(float startingAngle, float step, Graphics graphics, Pen pen, Rectangle rect)
        {
            float nextAngle = startingAngle + step;
            Figure Arc = new Figure() { FillColor = pen.Color };
            Arc.Path.AddArc(rect, startingAngle, nextAngle);
            bool flag = false;
            for (int i = 0; i < figuresList.Count; i++)
```

```
        {
            if (figuresList.Count == 0)
                break;
            var r = Arc.Region;
            var rWithFigure = r;
            r.Intersect(figuresList[i].Region);
            if (r == null)
            {
                flag = true;
            } }
        if (flag)
        {
            ProceduralCircleDraw(nextAngle, step, graphics, pen, rect);
        }
        else
        {
            if (nextAngle >= 360)
            {
                return;
            }
            else
            {
                graphics.DrawArc(pen, rect, startingAngle, nextAngle);
                //System.Threading.Thread.Sleep(1);
                ProceduralCircleDraw(nextAngle, step, graphics, pen, rect);
            }
        }
    }
    private void DrawProceduralSoundArea(int xPoint, int yPoint, Graphics
graphics, double decibel, double attenuationCoefficient)
    {
        int attenuationCount = (int)(decibel * attenuationCoefficient);
        int colorCounter = (int)(255 / attenuationCount);
        float widthCounter = 5 / attenuationCount;
        Pen gradientPen = new Pen(Color.Red, 2);
        for (int i = 0; decibel - i * attenuationCoefficient > 0; i++)
        {
            int red = 255 - (i * colorCounter / 2);
            if (red < 0)
                red -= red;
            int blue = 0 + (i * colorCounter / 2);
            if (blue > 255)
                blue -= blue - 255;
            gradientPen.Color = Color.FromArgb(255, red, 0, blue);
            gradientPen.Width -= i * widthCounter;
            int range = 10 * (i + 1);
```

```
                Rectangle rect = new Rectangle(xPoint - range / 2, yPoint - range / 2,
range, range);
                ProceduralCircleDraw(0, 15, graphics, gradientPen, rect);
                decibel -= i * attenuationCoefficient;
            }
        }
        private void DrawProceduralSoundArea(int xPoint, int yPoint, Graphics
graphics, double decibel, double attenuationCoefficient, Bitmap bm)
        {
            int attenuationCount = (int)(decibel * attenuationCoefficient);
            int colorCounter = (int)(255 / attenuationCount);
            float widthCounter = 5 / attenuationCount;
            Pen gradientPen = new Pen(Color.Red, 2);
            for (int i = 0; decibel - i * attenuationCoefficient > 0; i++)
            {
                int red = 255 - (i * colorCounter / 2);
                if (red < 0)
                    red -= red;
                int blue = 0 + (i * colorCounter / 2);
                if (blue > 255)
                    blue -= blue - 255;
                gradientPen.Color = Color.FromArgb(255, red, 0, blue);
                gradientPen.Width -= i * widthCounter;
                int range = 10 * (i + 1);
                Rectangle rect = new Rectangle((int)(xPoint - range / 2), (int)(yPoint -
range / 2), range, range);
                ProceduralPixelCircleDraw(graphics, gradientPen, rect, bm);
                decibel -= i * attenuationCoefficient;
            }
        }
        private void DrawGradientSoundArea(int xPoint, int yPoint, Graphics
graphics, double decibel, double attenuationCoefficient)
        {
            int count = (int)(decibel / attenuationCoefficient);
            GraphicsPath path = new GraphicsPath();
            Rectangle rect = new Rectangle(xPoint - count / 2, yPoint - count / 2, count,
count);
            path.AddEllipse(rect);
            Color[] surroundColors = { Color.Blue };
            PathGradientBrush gradientBrush = new PathGradientBrush(path)
            {
                CenterColor = Color.Red,
                SurroundColors = surroundColors
            };
            graphics.FillEllipse(gradientBrush, rect);
        }
```

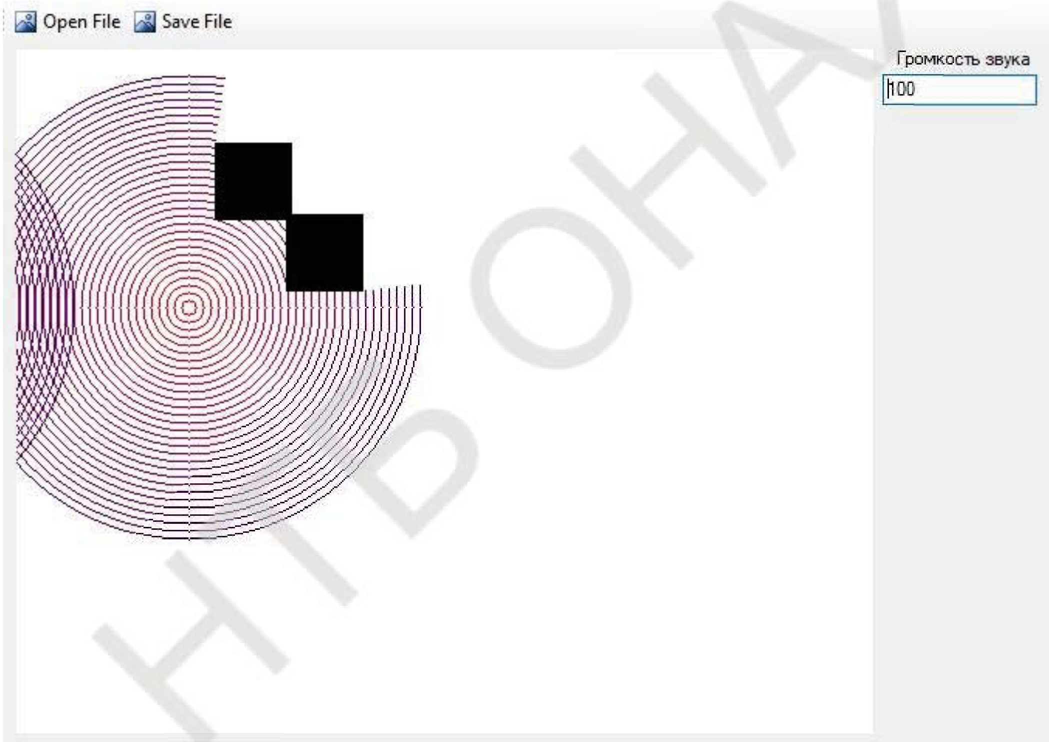
```
private void Form1_MouseDown(object sender, MouseEventArgs e)
{
    int x = e.X;
    int y = e.Y;
    Graphics graphics = CreateGraphics();
    if (e.Button == MouseButtons.Left)
        DrawProceduralSoundArea(x, y, graphics, 100, 0.226);
    else if (e.Button == MouseButtons.Right)
    {
        AddFigure(x, y);
    }
}
public void AddFigure(int xPoint, int yPoint)
{
    var figure = new Figure() { FillColor = Color.Black };
    int figureWidth = rnd.Next(50, 200);
    int figureHeight = rnd.Next(50, 200);
    figure.Path.AddRectangle(new Rectangle(xPoint - figureWidth / 2, yPoint - figureHeight / 2, figureWidth, figureHeight));
    figure.DrawFigure(CreateGraphics());
    figuresList.Add(figure);
}
public void pictureBox1_Paint(object sender, PaintEventArgs e)
{
    Bitmap bm = new Bitmap(pictureBox1.Width, pictureBox1.Height);
    pictureBox1.Image = (Image)(bm);
    Graphics gr = CreateGraphics();
    gr = e.Graphics;
    gr = Graphics.FromImage(pictureBox1.Image);
    gr.FillRectangle(new SolidBrush(Color.Black), 50, 50, 50, 50);
}
private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    int Decibel = Convert.ToInt32(textBox1.Text);
    int x = e.X;
    int y = e.Y;
    if (fl)
    {
        gr = CreateGraphics();
        bitm = new Bitmap(pictureBox1.Width, pictureBox1.Height);
        fl = false;
        for (int i = 1; i < bitm.Width; i++)
            for (int j = 1; j < bitm.Height; j++)
                bitm.SetPixel(i, j, Color.White);
    }
    pictureBox1.Image = (Image)(bitm);
```

```
        gr = Graphics.FromImage(pictureBox1.Image);
        if (e.Button == MouseButtons.Left)
            DrawProceduralSoundArea(x, y, gr, Decibel, 0.226, bitm);
        else
            gr.FillRectangle(new SolidBrush(Color.Black), x - 25, y - 25, 50, 50);
    }
    private void label1_Click(object sender, EventArgs e)
    {

    }
    private void toolStripButton1_Click(object sender, EventArgs e)
    {
        OpenFileDialog open = new OpenFileDialog();
        open.InitialDirectory = "C://";
        open.Filter = "Image file(*.jpg)|*.jpg|(*.png)|*.png|All Files(*.*)|*.*";
        open.FilterIndex = 1;
        if (open.ShowDialog() == DialogResult.OK)
        {
            string picpath = open.FileName.ToString();
            pictureBox1.ImageLocation = picpath;
        }
    }
    private void toolStripButton2_Click(object sender, EventArgs e)
    {
        SaveFileDialog save = new SaveFileDialog();
        save.FileName = "Image";
        save.DefaultExt = ".jpg";
        save.Filter = "Image (*.jpg)|*.jpg";
        save.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData);
        save.RestoreDirectory = true;
        if (save.ShowDialog() == DialogResult.OK)
        {
            string filename = save.FileName;
            bitm.Save(filename, System.Drawing.Imaging.ImageFormat.Jpeg);
        }
    }
}

class Figure
{
    public GraphicsPath Path = new GraphicsPath();
    public Color FillColor;
    public PointF Center
    {
        get
```

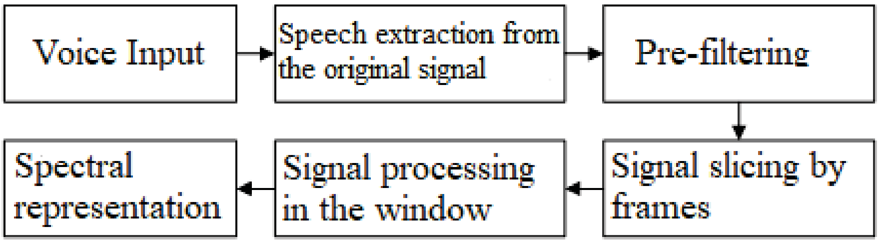
```
            {
                var rect = Path.GetBounds();
                return new PointF(rect.X + rect.Width / 2, rect.Y + rect.Height / 2);
            }
        }
        public Region Region
        {
            get { return new Region(Path); }
        }
        public void DrawFigure (Graphics graphics)
        {
            Random rnd = new Random();
            SolidBrush figureBrush = new SolidBrush(FillColor);
            graphics.FillPath(figureBrush, Path);
        }
    }
}
```



The practical arrangement of microphones will allow you to create the most effective emergency call system. Since there will be no areas that are not covered by the range of the microphones. Accordingly, the accuracy of the data is improved and the sound incident recognition system is improved.

We examined a method for recognizing sounds using neural networks. It is clear that the creation of an accurate neural network requires a lot of time and a large set of sounds. It will take more than one month, and not even one year, to create a system that will determine the sound of an emergency with almost no errors. Nevertheless, neural networks will help to improve this system, taking into account the experience of previous errors.

Picture 1 shows a diagram of the preprocessing of speech signals.

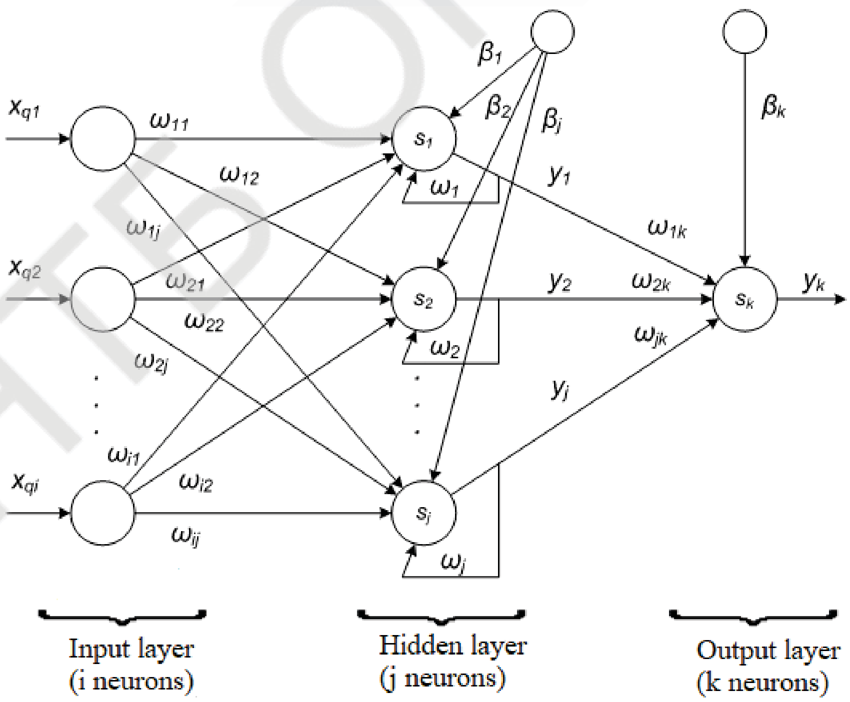


Picture 1. The scheme of preliminary processing of speech signals

After processing the audio data received, we get an array of signal segments. Each segment corresponds to a set of numbers characterizing the amplitude spectra of the signal. To prepare for the calculation for the output signal of the neural network, it is necessary to write all sets of numbers in the table, the line of which is the set of numbers of each frame.

Table 1. Description of a set of features of a speech signal

Frame	1st value	2nd value	…	ith value
1st frame	x_{11}	x_{12}	…	x_{1i}
2nd frame	x_{21}	x_{22}	…	x_{2i}
…	…	…	…	…
Nth frame	x_{N1}	x_{N2}	…	x_{Ni}



Picture 2. Single feedback neural network structure

- i – the number of values of one set of numbers
- N – number of sets of numbers (signal frame after slicing)
- x_{qi} - ith input value qth of a set of numbers;

– y_j – output j[th] neuron layer;
– ω_{ij} – coupling weight connecting i[th] neuron to j[th] neuron;
– ω_j – feedback weight j[th] neuron;
– β_j – displacement j[th] neuron layer.

– The number of input and output neurons is known. Each of the input neurons corresponds to one set of numbers. And on the output layer there is only one neuron, the output of which corresponds to the desired signal recognition value.

To calculate the output of a neural network, you must perform the following steps:

1. Initiate all contexts of all hidden layer neurons.
2. Submit the first set of numbers to the input of the neural network. Calculate the outputs of the hidden layer for it.

$y_j = f\left(\sum_{i=1}^{I} \omega_{ij} x_{1i} + \beta_j + \omega_j x_j\right)$[10],

where $f(x)$ - non-linear activation function $y_j = \frac{1}{1+\beta^{-aS_j}}$[10].

3. If the current set of numbers is not the last, then go to step 5, otherwise go to step
4. Record the outputs of hidden layer neurons to contexts where $x_j = y_j$. Go to step 2 for the next set of numbers.
5. Calculate the output neuron output layer.

Neural network learning algorithm: iterative adjustment of the weight matrix is necessary, gradually reducing the error in the output vectors.[10]

5. Conclusion

All of the above gives us the opportunity to draw the following conclusions. An automated emergency call system based on the recognition of sound incidents is quite a practical solution to many of the problems inherent in video surveillance. Such as the speed of response to an incident, the absence of "dead zones" in microphones. There are a sufficient number of difficulties that can be encountered when creating a sound recognition system using neutron networks. For example, a huge amount of information is necessary, since it is impossible to achieve high accuracy of the operation of artificial neural networks on a sufficiently small amount of data. There are a lot of sounds in our world. You cannot teach a system to analyze everything reliably in a couple of days. The first presented version of neural network-based speech recognition contained an error rate of about 25%, and only after 3 years the result was improved and already accounted for 8% of errors.

But despite the rather complicated implementation, with the help of an automated emergency call system based on the recognition of sound incidents, we get the opportunity to secure our city, our property, and our life.

6. References

1. Vorobienko P.P. Infocommunications: terms and definitions [Text] / Vorobienko PP, L.A. Nikityuk // East European Journal of Advanced Technologies. - 2012 .- 6/2 (54). - from. 4-6.
2. M. Brandstein and D. Ward, Microphone arrays: signal processing techniques and applications. Berlic Heidelberg New York: Springer-Verlag.

3. Efimenko M.S., Klymiv S.I., Satkenov R.B. Wireless sensor networks // Young Scientist. - 2018. - No. 51. - S. 40-42.
4. Bertrand A. Applications and trends in wireless acoustic sensor networks: A signal processing perspective/ Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
5. Romanova O. V. Speech recognition systems in the modern world [Electronic resourse]. - https://infourok.ru/issledovatelskaya-rabota-na-temu-sistemi-raspoznavaniya-rechi-v-sovremennom-mire-1459208.html
6. Geitgey A. Machine Learning is Fun! The world's easiest introduction to Machine Learning [Electronic resource]. - https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471
7. Kotelnikov V. A. About the bandwidth of ether and wire in telecommunications - All-Union Energy Committee. // Materials for the I All-Union Congress on the Technical Reconstruction of Communication and the Development of Low-Current Industry, 1933. Reprint of the article in UFN, 176: 7 (2006), 762-770.
8. Davis K.P. Forest Fire: Control and Use/ K.P. Davis, G.M. Byram, W.R.Krumm. – N.Y.: McGray-Hill, 1959. – 584 p.
9. Zhuravleva L. V., Strigulin K. A. Studies of the features of the development of neural networks in the modern world [Text] // Technical sciences: problems and prospects: materials of the IV Intern. scientific conf. (St. Petersburg, July 2016). - SPb .: Own publishing house, 2016. - S. 9-11.
10. Le N. V., Panchenko D. Speech recognition based on artificial neural networks [Text]//Technical sciences in Russia and abroad: materials of the Intern. scientific conf. (Moscow, May 2011).- M .: Your printing partner, 2011.-S.8-11.

AN APPLICATION FOR DEMONSTRATING AND COMPARING SORTING AND RETRIEVAL ALGORITHMS

Author: Oleksandr Sokolskyi
Supervisor: Oleksandr Melnykov
Donbass State Engineering Academy (Ukraine)

Abstract. *The use and information and communication of their technology and in education is constantly increasing, including - when teaching professional disciplines in high school. The study of algorithms for sorting and searching data is provided by many educational programs of the specialties of the field of knowledge "Information Technologies". Using the process of presenting these sections of the information and communication means learning how a demonstration is application allows to better understand the essence of each algorithm, cf. ivnyaty them on specific examples.*

The purpose of the work is to develop a program (application) in the visual programming environment, which would allow students studying the algorithms for sorting and searching data , to observe the process and to analyze the advantages and disadvantages of a number of methods to better understand the principles of their work.

The object of study - algorithms for sorting and searching data . The subject of the study is demonstration of a number of algorithms and their comparison.