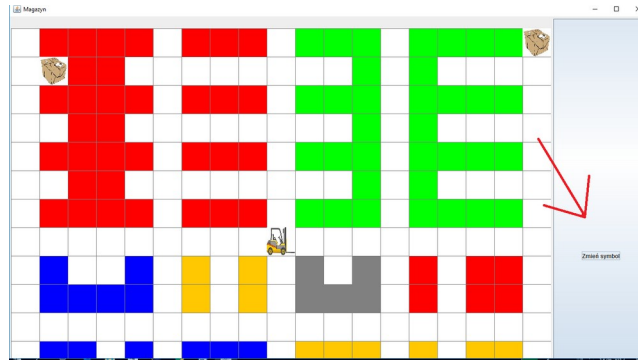
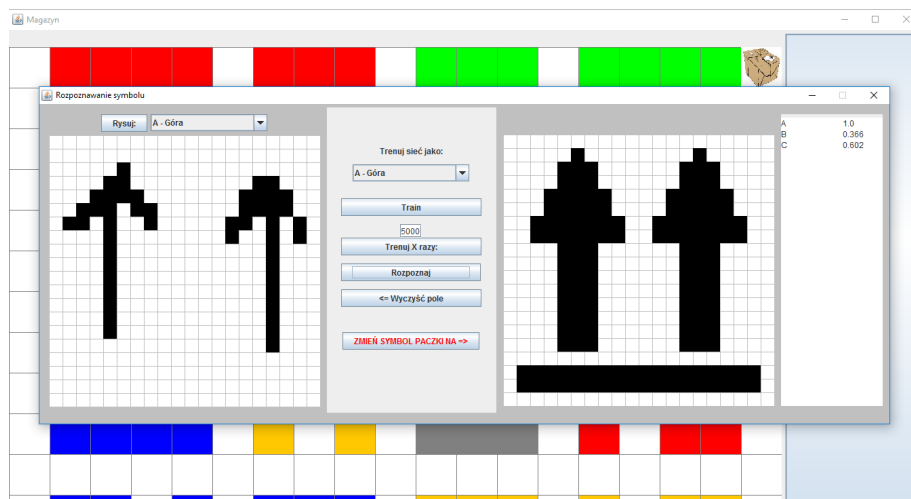


Raport podprojektu Sieci Neuronowe

autor:



Po otwarciu okna programu Magazyn należy wcisnąć przycisk „Zmień symbol” w celu uruchomienia okna odpowiedzialnego za rozpoznawanie obrazu poprzez zastosowanie Sieci Neuronowych.



Lewe okno służy do rysowania, prawe natomiast do wyświetlenia obrazu który rozpoznano. Na środku znajduje się panel w którym znajdują się elementy służące odpowiednio do Trenowania Sieci Neuronowej, przycisku do Rozpoznawania obrazu, czyszczenia pola do rysowania, oraz Przypisywania rozpoznanego symbolu do paczki.

W chwili otwarcia okna następuje utworzenie sieci neuronowej. Rysujemy grafikę i klikamy „Rozpoznaj”, który przekazuje parametry do sieci i otrzymane wyniki wyświetla w TextBoxie znajdującym się po prawej stronie. Następnie w zależności o naszego wyboru możemy zmienić parametr paczki albo wytrenować naszą sieć.

```
transformButton.addActionListener(e -> {
    networkTrainer.setInputs(drawingPanel.getPixels());
    ArrayList<Double> outputs = networkTrainer.getOutputs();
    int index = 0;
    for (int i = 0; i < outputs.size(); i++) {
        if (outputs.get(i) > outputs.get(index))
            index = i;
    }
    updateTextArea();
    trainAsCombo.setSelectedIndex(index);
    resultPanel.drawLetter(goodPixels.getInstance().getGoodPixels(index));
});

trainNetworkButton.addActionListener(e -> {
    int number = 0;
    try {
        number = Integer.parseInt(trainingSetsAmount.getText());
    } catch (Exception x) {
        JOptionPane.showMessageDialog(this, "Zła liczba albo występują puste znaki", "ERROR", JOptionPane.PLAIN_MESSAGE);
    }
    networkTrainer.train(number);
});
```

```
public class Train {
    private static final int NEURON_COUNT = 3;
    private Network network;
    private ArrayList<TrainingSet> trainingSets;

    public Train() {
        this.network = new Network();
        this.network.addNeurons(NEURON_COUNT);
        this.trainingSets = ReadWriteFile.readTrainingSets();
    }

    public void train(long count) {
        for (long i = 0; i < count; i++) {
            int index = ((int) (Math.random() * trainingSets.size()));
            TrainingSet set = trainingSets.get(index);
            network.setInputs(set.getInputs());
            network.adjustWeights(set.getGoodOutput());
        }
    }

    public void setInputs(ArrayList<Integer> inputs) {
        network.setInputs(inputs);
    }
}
```

Trenowanie sieci następuje po wciśnięciu przycisku „Trenuj” bądź „Trenuj X razy” gdzie X symbolizuje liczbę krotności trenowania pobierana z Textboxa znajdującego się w środkowym panelu. Następnie za pomocą wyliczam i przypisuję wagi.