

NEURONSKE MREŽE: PREPOZNAVANJE OBJEKTA SA SLIKE

ZADATAK: Korištenjem programskog okvira Encog potrebno je napisati program koji će naučiti prepoznavati emotikone na nekoj slici.

Na primjer, ako je zadana sljedeća **slika** niza smajlića, onda aplikacija treba vratiti odgovarajuće nazive/simbole svakog emotikona npr, „happyface“ „sadface“ ili „:)“, „:(“, i sl.



Slika 1 primjer slike

Studenti imaju mogućnost riješiti dvije verzije zadatka – lakšu i težu. Lakša verzija zadatka podrazumijeva raspoznavanje samo poznatih setova emotikona (onih na kojima je mreža trenirana) i/ili raspoznavanje manjeg broja različitih emotikona (do 6), a teža verzija zadatka podrazumijeva raspoznavanje seta emotikona na kojima mreža nije trenirana i setova koji sadrže više od 6 različitih vrsta emotikona.

Odabirom lakše opcije, student može dobiti maksimalno 70% bodova za taj tjedan.

Zadatak je potrebno riješiti korištenjem sljedećih koraka:

1. Sa sljedećih mrežnih stranica preuzeti posljednju inačicu programskog okvira Encog:

<http://www.heatonresearch.com/download>

2. Sa stranica kolegija ili iz preuzete arhive primjera potrebno je kopirati klasu „ImageNeuralNetwork“ koja se nalazi unutar mape:

„encog-examples-3.3.0\src\main\java\org\encog\examples\neural\image“, te u classpath projekta staviti JAR datoteku „encog-core-3.3.0.jar“.

Moguće je koristiti i starije verzije (3.1, 3.2)

3. Odaberite i pronađite slike nekoliko različitih setova emotikona po vlastitom izboru i složite od njih setove za treniranje.
Od nekoliko tih emotikona (ili nekih drugih, ovisno o verziji zadatka), složite proizvoljni niz emotikona kao jednu sliku (minimalno 3 u nizu) koju ćete dati mreži za raspoznavanje. Vaš program mora od te jedne slike napraviti (izrezati) više pojedinačnih sličica emotikona koje se zatim raspoznavaju.
4. Pomoću modificiranog programskog koda kakav je dan u nastavku, potrebno je sliku niza iz prethodnog koraka raščlaniti na odgovarajući broj sličica koje predstavljaju pojedini emotikon.

```
// Slika "word.png" sadrži cijelu riječ
File file = new File("word.jpg");
FileInputStream fis = new FileInputStream(file);
// Pozivanje metode za čitanje slike
BufferedImage image = ImageIO.read(fis);

// Definiranje broja redaka i stupaca slike
// 1 redak i 14 stupaca za broj slova u „neuralnetworks“
int rows = 1;
int cols = 14;
int chunks = rows * cols;

// određivanje veličine jednog dijela
int chunkWidth = image.getWidth() / cols;
int chunkHeight = image.getHeight() / rows;

// Definiranje polja koje sadrži dijelove slika
int count = 0;
BufferedImage imgs[] = new BufferedImage[chunks];

for (int x = 0; x < rows; x++) {
    for (int y = 0; y < cols; y++) {

        // Inicijalizacija polja s dijelovima slike
        imgs[count] = new BufferedImage(chunkWidth,
chunkHeight,
                                image.getType());

        // "Crtanje" pojedinih dijelova slike
        Graphics2D gr = imgs[count++].createGraphics();
        gr.drawImage(image, 0, 0, chunkWidth, chunkHeight,
                                chunkWidth * y, chunkHeight * x,
chunkWidth * y +
                                chunkWidth, chunkHeight * x + chunkHeight,
null);
        gr.dispose();
    }
}

System.out.println("Rezanje slike je završeno.");

// Zapisivanje dijelova slika u zasebne datoteke
```

Četvrta laboratorijska vježba
Nekonvencionalni računalni postupci

```
for (int i = 0; i < imgs.length; i++) {  
    ImageIO.write(imgs[i], "jpg",  
                  new File("slika" + i + ".jpg"));  
}  
  
System.out.println("Kreiranje slika je završeno.");
```

5. Korištenjem klase iz drugog koraka potrebno je konfigurirati neuronsku mrežu i „naučiti“ je da prepoznaje emotikone. Učenje je potrebno provesti nad unaprijed pripremljenim slikama iz koraka 3. Za svaki emotikon potrebno je koristiti više od jednog primjera za učenje, ali broj primjera po emotikonu ne mora biti isti za sve – za neke „teže“ će biti potrebno više primjeraka nego za „lakše“.
6. Nakon što je provedeno učenje neuronske mreže koje je završilo sa zadovoljavajućom razinom pogreške, potrebno je aplikaciji predati sliku niza iz 3. koraka, a aplikacija iz te slike mora prepoznati o kojem se nizu radi.
7. Prilikom analiziranja rada aplikacije potrebno je istestirati različite parametre neuronske mreže kao što su:
 - 1) Broj neurona u prvom skrivenom sloju
 - 2) Broj neurona u drugom skrivenom sloju
 - 3) Različite tipove propagacijskih metoda koje nasljeđuju apstraktnu klasu "Propagation" (u primjeru klase "ImageNeuralNetwork" koristi se "ResilientPropagation")
8. U izvještaj je potrebno staviti slike na kojima se mreža testirala i rezultate (izraze, opise, simbole, ovisno što ste odabrali) koje je mreža dala za te slike.
9. Dodatne bodove nosi proširivanje setova različitim bojama emotikona i pozadina, uspješno prepoznavanje niza sa slika izrezanih npr. s web stranica (a ne samostalno kreiranih za potrebe zadatka), kao i uspješno prepoznavanje složenih slika s razmacima, drugim simbolima koje aplikacija u tom slučaju zanemaruje i sl.