Tamara Stanković i Luka Bulatović

Optičko prepoznavanje hemijskih formula ugljovodonika

U ovom radu opisan je postupak prepoznavanja hemijske formule ugljovodonika sa date fotografije pomoću računara. Taj postupak se sastoji iz nekoliko koraka. Na početku, izvršena je obrada fotografije i izdvajanje karaktera i linija. Oni se potom prepoznaju uz pomoć algoritma mašinskog učenja, neuralne mreže. Određuje se koji karakteri su međusobno povezani i konstruiše se matrica susedstva grafa hemijskog jedinjenja. Na osnovu dobijene matrice susedstva, uz pomoć vrednosti Wienerovog indeksa, zaključuje se o kom jedinjenju se radi. Ovaj algoritam je implementiran u vidu aplikacije u programskom jeziku Java i dobijena tačnost iznosi 90%.

1 Uvod

Hemijska jedinjenja se zapisuju uz pomoć hemijskih formula – molekulskih ili strukturnih. Dok molekulske formule govore samo o tome koji atomi se nalaze u molekulu, strukturne formule daju informaciju o tome kako su i kojim vezama ti atomi međusobno povezani. Poznato je da je broj mogućih hemijskih jedinjenja toliko veliki da je nemoguće zamisliti da će ikada sva biti sintetisana i ispitana. Do sada je potpuno identifikovano i registrovano preko 128 miliona hemijskih jedinjenja, a svakog dana se dodaje oko 12 hiljada novih. Poznato je da od strukture molekula zavise fizičke osobine (boja, gustina, tačka topljenja…), hemijska reaktivnost i biološka aktivnost jedinjenja, kao i da se slična jedinjenja slično ponašaju.

Hemijska teorija grafova predstavlja granu matematike kojom se modeliraju molekuli u cilju sticanja uvida u fizičke, hemijske i biološke osobine jedinjenja i njihove što bolje aproksimacije. Vezu između strukturne formule i grafova prvi je uočio britanski matematičar Artur Kejli. On je uveo definiciju molekulskog grafa u svom radu iz 1874. godine. Naime, hemijska formula se može posmatrati kao graf, koji je u matematici poznat pod terminom „molekularni graf" ili „hemijski graf". On se definiše na sledeći način:

Definicija 1. Molekularni graf je povezan, neusmeren graf, koji odgovara strukturnoj formuli hemijskog jedinjenja tako da čvorovi grafa

Tamara Stanković (1995), Niš, Branka Krsmanovića 15/14, učenica 3. razreda Gimnazije „Svetozar Marković" u Nišu

Luka Bulatović (1995), Pančevo, Karađorđeva 2/33, učenik 3.razreda Matematičke gimnazije u Beogradu

MENTOR: Andreja Ilić, MDCS, Beograd

odgovaraju atomima u molekulu, a grane grafa hemijskim vezama između tih atoma.

U ovom radu korišćeni su neki od pojmova iz teorije grafova (Živković 2008).

Ugljovodonici su organska jedinjenja koja sadrže samo atome ugljenika (C) i vodonika (H). Dele se na ciklične (zatvorene) i aciklične (otvorene), gde je podela izvršena na osnovu toga da li molekularni graf tog ugljovodonika sadrži ciklus. Takođe, podela može biti izvršena i na osnovu broja hemijskih veza između atoma u molekulu. Tako se aciklični ugljovodonici dele na: alkane (jednostruka veza), alkene (dvostruka veza) i alkine (trostruka veza) (Stojiljković 2012). Neka hemijska svojstva ovih jedinjenja iskorišćena su u radu radi lakše obrade molekularnih grafova ugljovodonika.

Digitalizacija podataka je proces pretvaranja signala iz analognog u digitalni oblik. Moguće je digitalizovati sve vrste podataka, od teksta, preko audio i video zapisa, sve do trodimenzionalnih objekata. Razlozi za digitalizaciju su mnogobrojni: od zaštite podataka, preko povećanja dostupnosti podataka, sve do novih usluga nad podacima (ubrzano pretraživanje teksta, lakša analiza podataka, virtualno spajanje fizički udaljenih podataka i sl.). Međutim, digitalizacija podataka može biti dug i spor proces u kome postoji mogućnost greške. Da bi se taj proces ubrzao i sprečila mogućnost greške, razvijeni su različiti algoritmi za optičko prepoznavanje karaktera. To je relativno nova oblast koja se razvija od kraja 20. veka.

Jedan od algoritama za optičko prepoznavanje karaktera korišćen je i u ovom radu. Cilj ovog rada je pronalaženje i opisivanje postupka da od date fotografije hemijske formule računar zaključi o kojoj formuli se radi. Naime, potrebno je sa fotografije na kojoj se nalazi molekularni graf odrediti o kom grafu se radi. Zato je ovo problem optičkog prepoznavanja grafova uz pomoć računara.

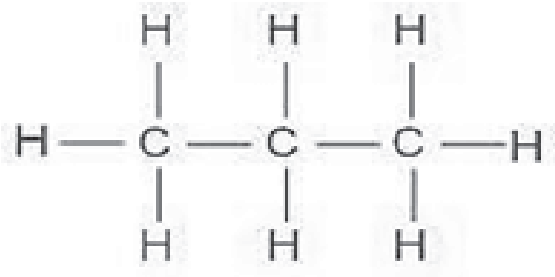
Postupak optičkog prepoznavanja molekularnih grafova koji je korišćen u ovom radu sastoji se iz sledećih koraka:

- Obrada fotografije. Neophodno je obraditi fotografiju i prevesti je u oblik kojim računar može da manipuliše.
- Izdvajanje i obrada karaktera i linija iz fotografije. Potrebno je da se iz slike izdvoje karakteri i linije da bi se kasnije dalje obrađivali.
- Prepoznavanje karaktera. Karakteri na slici predstavljaju čvorove molekularnog grafa, dakle, neophodno je otkriti koji su karakteri na slici. Ovo se postiže algoritmom mašinskog učenja, neuronskom mrežom.
- Prepoznavanje linija na slici. Linije na slici predstavljaju grane u molekularnom grafu.
- Određivanje povezanosti karaktera. Neophodno je odrediti koji karakteri su međusobno povezani i kojim linijama. Na taj način dobija se matrica susedstva molekularnog grafa sa fotografije.
- Određivanje molekula na osnovu veza između karaktera. Na osnovu dobijene matrice susedstva, molekularni graf je na jedinstven način određen, pa se na taj način zaključuje o kom grafu, odnosno, o kojoj hemijskoj formuli se radi.

Za potrebe ovog rada napisana je aplikacija HemijskeFormule, u programskom jeziku Java i svi rezultati koji su prikazani u radu dobijeni su korišćenjem te aplikacije.

2 Obrada fotografije

Računar ne može da radi sa pojavama iz prirode (slika, zvuk, pokret) u njihovom analognom obliku, već ih mora prevesti na sebi razumljiv jezik. Da bi se fotografija predstavila na računaru, ona se mora digitalizovati. Jedan od načina predstavljanja fotografije u računaru je pomoću matrice (mreže) kvadratića, koji se nazivaju pikseli. Proces digitalizacije zapravo podrazumeva prevođenje fotografije u odgovarajuću matricu piksela. Svaki piksel ima svoju boju. Boja piksela predstavljena je u računaru određenim brojem bitova; broj bitova za opis boje jednak je za sve piksele na slici. Fotografije u punoj boji mogu se na ekranu prikazati sa 24 bita po pikselu. U RGB modelu boja, sa po 8 bitova predstavljaju se komponente crvene, zelene i plave boje, a njihovom kombinacijom određuje se boja piksela.



Slika 1.
Fotografija hemijske formule koja se prepoznaje

Figure 1.
A photo of chemical formula that should be recognized

2.1 Dobijanje slike u nijansama sive boje

Prvi korak je prevođenje RGB modela fotografije u fotografiju u nijansama sive boje (grayscale), koja se prikazuje sa 8 bitova po pikselu. Moguće je za svaki piksel odrediti vrednosti crvenog, zelenog i plavog kanala. Nijansa sive za određeni piksel dobija se primenom jednostavne formule:

$$\text{gray} = 0.21 \cdot \text{red} + 0.71 \cdot \text{green} + 0.007 \cdot \text{blue}$$

Primenjujući formulu za svaki piksel, dobija se matrica fotografije u nijansama sive boje.

2.2 Binarizacija

Fotografiju, koja je dobijena prethodno opisanim postupkom, potrebno je prevesti u matricu nula i jedinica, jer su algoritmi za rad sa binarnom matricom značajno brži i lakše dolaze do rešenja. Vrednosti piksela su prirodni brojevi iz intervala [0, 255]. U ovom trenutku, potrebno je odrediti granicu za vrednost piksela, da li je on 0 (beo) ili 255 (crn). Može se odrediti konstanta koja će biti granica, npr. 120. Međutim, to nije

dovoljno precizno, te je potrebno granicu računati u zavisnosti od konkretne slike. U tu svrhu koristi se jedan od najpoznatijih algoritama (Sezgin i Sankur 2004) za konverziju fotografije u nijansama sive boje u crno-belu fotografiju, Otsu metod (Otsu 1979), čiji je opis dat u daljem tekstu.

Na početku se cela fotografija može podeliti na dve klase: pozadina i slika. Otsu metod ima za cilj da pronađe takvu granicu kojom se minimizuje razlika unutar jedne klase. To znači da bi trebalo maksimizovati razliku između suprotnih klasa.

Sledeći korak je konstruisanje histograma P fotografije. Histogram je niz dužine 256 koji za svaku nijansu sive boje, sadrži informaciju o tome koliko se puta ta nijansa javlja na slici. Potom se za svaku boju t ($t = \{0,\ldots,255\}$) izračunaju procenjene verovatnoće q da su klase $q_1(t)$ i $q_2(t)$ razdvojene granicom t i procenjene srednje vrednosti klasa pri takvoj podeli $\mu_1(t)$ i $\mu_2(t)$. Procenjene verovatnoće definisane su kao:

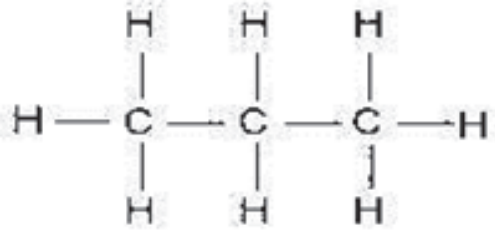
$$q_1(t) = \sum_{i=0}^{t} P(i) \text{ i } q_2(t) = \sum_{i=t+1}^{255} P(i),$$

a procenjene srednje vrednosti klasa kao:

$$\mu_1(t) = \sum_{i=0}^{t} \frac{i \cdot P(i)}{q_1(t)} \text{ i } \mu_2(t) = \sum_{i=t+1}^{255} \frac{i \cdot P(i)}{q_2(t)}.$$

Za granicu se uzima ono za koje važi da je razlika između suprotnih klasa maksimalna, odnosno da je izraz $q_1(t) \cdot q_2(t) \cdot (\mu_1(t) - \mu_2(t))^2$ maksimalan.

Pošto je granica određena, vrednost svakog piksela slike upoređuje se sa tom granicom. Ako je vrednost piksela veća od granice taj piksel postaje crn i dobija vrednost 255, a ako je manja od granice onda je piksel beo i dobija vrednost 0. Na ovaj način dobijena je binarizovana matrica slike, koja može dalje da se koristi za izdvajanje komponenata.



Slika 2.
Fotografija dobijena binarizacijom Slike 1.

Figure 2.
Photo received after the binarization of the Figure 1.

3 Izdvajanje i obrada karaktera i linija sa slike

U ovom delu, neophodno je dati formalnu definiciju povezanih komponenata grafa.

Definicija 2. Povezana komponenta grafa G je podgraf H grafa G u kome su svaka dva čvora međusobno povezana putem, i nisu povezana ni sa jednim čvorom grafa G koji nije u H.

U konkretnom slučaju, komponente predstavljaju sve crne površine na slici – dakle, slova, brojevi i linije. Prvi zadatak je izdvojiti iz binarizovane matrice te komponente, a potom ih sve svesti na istu veličinu.

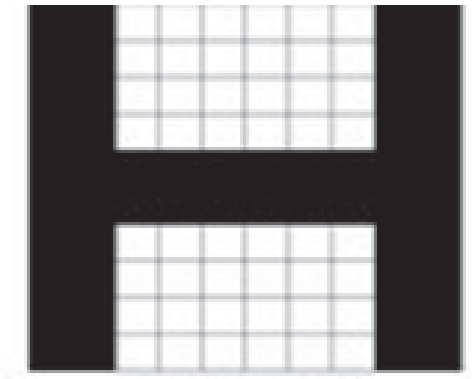
3.1 Prepoznavanje i izdvajanje komponenata

U matrici slike cilj je pronaći spojene crne površine, koje tako grupisane predstavljaju neki karakter ili liniju. U tu svrhu, iskorišćen je algoritam za pretragu grafa DFS (depth first search).

DFS algoritam radi tako što obilazi graf „po dubini". Ako je dat neusmeren graf u kojem su svi čvorovi na početku obeleženi kao ne posećeni, pretraga počinje izborom proizvoljnog čvora u grafu. Ovaj čvor se odmah obeležava kao posećen, a zatim se svaki ne posećen, njemu susedan čvor, rekurzivno posećuje, prateći granu između njih. Kada se posete svi čvorovi do kojih se može stići iz početnog čvora, prelazi se na sledeći ne posećeni čvor. Pretraga se obustavlja kada svi čvorovi budu posećeni.

Primena ovog algoritma na konkretan problem sastoji se u sledećem: čvorovima grafa pridružuju se pikseli (elementi matrice) slike. Dva čvora biće povezana ako su ta dva polja u matrici slike susedna i obojena su istom bojom. Dalje, za svako polje (i, j) u matrici proverava se da li je neko od susednih (u opštem slučaju) osam polja, $(i-1, j-1)$, $(i-1, j)$, $(i-1, j+1)$, $(i, j-1)$, $(i, j+1)$, $(i+1, j-1)$, $(i+1, j)$ i $(i+1, j+1)$ već posećeno. Ako jeste, prelazi se na sledeće ne posećeno polje. Ako nije, onda se ono markira kao posećeno i pridružuje se trenutnoj komponenti. Polje se markira kao posećeno tako što se u njega upiše broj komponente kojoj se pridružuje. Ako su svi susedi trenutnog polja posećeni, prelazi se na sledeće ne posećeno polje, koje se pridružuje novoj komponenti.

Na kraju ovog postupka, dobija se ukupan broj komponenti na slici, kao i matrica slike podeljena na komponente. Sada je moguće od svake komponente napraviti nezavisnu matricu, iz koje su izbačeni, tzv. prazni redovi i kolone, odnosno oni redovi i kolone u kojima nema crnih polja te komponente. Rezultat primene ovog postupka dat je na slici 3.



Slika 3.
Jedna od izdvojenih komponenti sa fotografije

Figure 3.
One of the extracted components from the photo

Pri pravljenju nezavisnih matrica komponenata, radi lakše manipulacije, crna polja se obeležavaju brojem 1, a ne više sa 255.

3.2 Obrada komponenata

Kako su, zahvaljujući prethodno opisanom postupku, dobijene matrice komponenata, neophodno je svesti ih na istu veličinu, dakle, smanjiti ih ili povećati.

Neka su p i q polazne dimenzije komponente prikazane matricom A, a l*engthfixed* i *widthfiksed* dimenzije na koje je potrebno svesti komponentu (matrica B). Važi sledeća relacija:

$$B(i, j) = A\left(\left\lceil \frac{i \cdot p}{lengthfixed} \right\rceil , \left\lceil \frac{j \cdot q}{widthfixed} \right\rceil\right),$$

gde je $i \in [0, lengthfixed]$ i $j \in [0, widthfixed]$

Primenom ovog postupka, prikazani su rezultati u zavisnosti od toga da li se komponenta smanjuje ili povećava.

Slika 4.
Povećana (levo) i smanjena komponenta (desno)

Figure 4.
Increased (left) and reduced component (right)

Za konkretan slučaj izdvajanja karaktera i linija sa slike, izabrano je *lengthfixwd* = 20 i *widthfixed* = 20 i primenjen je prethodno opisani postupak na svaku od komponenata.

4 Prepoznavanje karaktera

Za prepoznavanje karaktera korišćen je algoritam mašinskog učenja, neuralna mreža. Pre nego što je detaljno opisan algoritam neuralne mreže koji je iskorišćen, dato je objašnjenje pojma i svrhe mašinskog učenja i neuralne mreže (Ng 2013).

4.1Mašinsko učenje

Mašinsko učenje je sposobnost računara da daje tačne odgovore, a da nije prethodno eksplicitno programiran za to. Naime, ako je računaru dat određen broj ulaznih podataka i za svaki od ulaznih podataka tačno rešenje za taj ulaz, njegov zadatak je da za nove ulazne podatke „pronađe" tačne odgovore. Problem može biti izračunavanje neke konkretne vrednosti na osnovu ulaznih podataka ili određivanje kojoj od mogućih klasa pripada ulazni podatak.

Training set-om se naziva skup ulaznih vrednosti (matrica *X*) za koje se znaju izlazne vrednosti (vektor *Y*). Pri tome, broj ulaznih vrednosti, odnosno veličina training set-a obeležena je sa *m*, tako da je prva dimenzija i matrice i vektora upravo *m*. Jedan training primer obeležava se kao (x, y), gde je *x* jedna vrsta matrice .

Jedan objekat ili pojava može da zavisi od jedne osobine, ali često zavisi od više njih. Te osobine se nazivaju atributima. Broj atributa biće obeležen sa *n*. Neophodno je da atributi budu istog reda veličine, pa je ponekad potrebno izvršiti skaliranje atributa. Vrednosti atributa se mogu smestiti u matricu $x = [x_0 x_1 \ldots x_n]$ veličine $1 \times n$, gde je $x_0 = 1$. Potrebno je napomenuti da je training set, matrica *X* veličine $m \times n$.

Hipoteza (*h*) je funkcija koja povezuje matricu *X* sa vektorom *Y*. U zavisnosti od vrste hipoteze mogu postojati potpuno različiti postupci pronalaženja rezultata. Vrednost te funkcije za neku novu ulaznu vrednost trebalo bi da predstavlja rešenje problema za taj ulaz. Tačan oblik hipoteze određuje se na osnovu training set-a.

Ako se hipoteza definiše kao funkcija *h*: $R^n \rightarrow R$, oblika:

$$h(x) = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_n x_0 n,$$

gde su $\theta_0, \theta_1, \ldots, \theta_n$ realni parametri, takva hipoteza se naziva linearna. Ona se, dalje, može zapisati kao proizvod $h(x) = x \cdot \theta^T$, gde je $[\theta_0, \theta_1, \ldots, \theta_n]$ matrica parametara hipoteze veličine $1 \times n$. Pitanje koje se postavlja je kako izabrati parametre $\theta_0, \theta_1, \ldots, \theta_n$? Potrebno je izabrati parametre $\theta_0, \theta_1, \ldots, \theta_n$ tako da vrednost hipoteze bude što bliža *y*, za neki training primer (x, y). Odnosno, matematički definisano, cilj je minimizovati *cost funkciju* $J(\theta)$ po parametrima $\theta_0, \theta_1, \ldots, \theta_n$, gde je cost funkcija $J(\theta)$, koja je definisana kao:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h(x) - y)^2$$

Ovo je najčešći oblik hipoteze koji se koristi za rešavanje problema mašinskog učenja. On nije korišćen u ovom radu, ali je dat kao očigledan primer koncepta mašinskog učenja.

Za probleme u kojima je potrebno izvršiti neku vrstu klasifikacije, obično nije moguće koristiti linearne ili polinomske hipoteze. Ako je potrebno izvršiti binarnu klasifikaciju, dakle kada objekat može pripadati jednoj od dve moguće klase $y \in [0,1]$, hipoteza bi trebalo da vrati vrednost $h(x) \geq 0.5$, ako je $y = 1$ i $h(x) < 0.5$, ako je $y = 0$. Odnosno, hipoteza $h(x)$ se interpretira kao verovatnoća da je $y = 1$ izlaz za ulaz *x*. Ako objekat može biti u više od dve klase, $y \in \{1, 2, \ldots, k\}$, onda se za svaku klasu, ta klasa posmatra kao jedna, a ostale kao druga klasa. Na taj način se dobija *k* hipoteza, gde je $h^{(i)}(x) = P(y = i | x)$, za svako $i = 1, 2, \ldots, k$ i kao rešenje bira se ona klasa za koju je $h^{(i)}(x)$ najveće.

4.2 Neuronska mreža

Algoritam neuronske mreže je algoritam mašinskog učenja sa velikom primenom. Njegova osnovna prednost u odnosu na ostale algoritme je, što značajno brže radi, kod velike ulazne matrice *X*. Algoritam je nastao na osnovu ideje o simuliranju nervnog sistema čoveka, prevashodno zato što čovečiji mozak, bez obzira na to kakve mu informacije dolaze, uspešno uspeva da se prilagodi i obradi ih.

Neuralna mreža se sastoji iz nekoliko slojeva, gde postoje ulazni sloj, izlazni sloj i nekoliko skrivenih slojeva. Svaki sloj sadrži određen broj neurona, osnovnih jedinica u neuronskoj mreži. Takođe, svakom sloju odgovara neka hipoteza. Vrednosti u svakom sledećem sloju dobijaju se primenom hipoteza na vrednosti iz tog sloja. Zato, svaki sledeći sloj neuralne mreže može da izračuna sve složenije funkcije. Hipoteze, u ovom radu, imaju oblik logističke ili sigmoid funkcije:

$$g(z) = \frac{1}{1+e^{-x}}$$

Neuroni u različitim slojevima, povezani su težinskim granama (weigths), gde težine grana odgovaraju parametrima $\theta^{(j)}$ hipoteze koja povezuje sloj *j* neuronske mreže sa slojem $j+1$. Pri tome, broj neurona u *j*-tom sloju obeležen je sa S_j, a *i*-ti neuron u *j*-tom sloju sa $a_i^{(j)}$. Vrednost svakog neurona može se izračunati korišćenjem hipoteze prethodnog sloja, odnosno računanjem izraza:

$$a_i^{(j)} = g\left(\sum_{k=0}^{S_j} \theta_{ik}^{(j-1)} a_k^{(j-1)} \right).$$

Na ovaj način moguće je izračunati vrednosti svih neurona i tako dobiti krajnje rešenje.

Ako se neuralna mreža koristi za određivanje kojoj od *K* klasa pripada neki element (što je ovde slučaj), hipoteza *h* je vektor čiji je broj elemenata jednak broju klasa, odnosno *K*. Ako je ukupan broj slojeva u mreži obeležen sa *L*, odgovarajuća cost funkcija za neuronsku mrežu ima sledeći oblik:

$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} \sum_{k=1}^{K} y_k^{(i)} \cdot \log(h(x^{(i)}))_k \cdot (1-y_k^{(i)}) \cdot \log(1-h(x^{(i)}))_k \right] + \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{S_l} \sum_{j=1}^{S_{l+1}} \Theta_{ji}^{(l)}$$

Za učenje neuronske mreže, odnosno pronalaženje parametara Θ koji minimizuju ovu cost funkciju, postoji više algoritama. Jedan od najpoznatijih je *backpropagation* algoritam, koji je korišćen u ovom radu (Ng 2013).

4.3 Izbor arhitekture mreže

Arhitektura mreže predstavlja način na koji su različiti neuroni međusobno povezani u mreži. Postavlja se pitanje na koji način izabrati odgovarajuću arhitekturu neuronske mreže. Broj ulaznih neurona jednak je

broju atributa, dok je broj izlaznih neurona jednak broju klasa. Broj skrivenih slojeva je najčešće 1, ali ako ih ima više, obično je isti broj neurona u svakom sloju.

Pošto ne postoji način da se automatski odredi optimalan broj neurona u skrivenom sloju, uz pomoć programa Neuroph, rađena je provera rezultata dobijenih korišćenjem različitih arhitektura mreže i izabrana je ona arhitektura za koju su dobijeni najbolji rezultati.

Za konkretan slučaj prepoznavanja hemijskih formula ugljovodonika, korišćene su dve neuralne mreže – jedna, koja prepoznaje molekulske, a druga koja prepoznaje strukturne formule ugljovodonika. U tabelama koje slede, posmatrana je tačnost učenja i izabrana je ona arhitektura neuralne mreže za koju je ta tačnost najveća.

Arhitektura neuralne mreže za prepoznavanje molekulskih formula ugljovodonika, u zavisnosti od broja slojeva u neuralnoj mreži i broja neurona po svakom sloju, izabrana je na osnovu sledećih rezultata (tabela 1):

Tabela 1. Analiza tačnosti neuralne mreže za prepoznavanje molekulskih formula za različite arhitekture

	Broj slojeva	Broj neurona	Tačnost
1.	2	20, 10	70%
2.	2	30, 10	82%
3.	2	40, 10	93%
4.	2	50, 10	75%
5.	2	45, 10	69%
6.	2	40, 20	67%

Zaključeno je da je najpovoljnija arhitektura neuralne mreže, arhitektura pod rednim brojem 3. i ona je korišćena u svrhu prepoznavanja karaktera u molekularnom grafu.

Arhitektura neuralne mreže za prepoznavanje strukturnih formula ugljovodonika, u zavisnosti od broja slojeva u neuralnoj mreži i broja neurona po svakom sloju, izabrana je na osnovu sledećih rezultata:

Tabela 2. Analiza tačnosti neuralne mreže za prepoznavanje strukturnih formula za različite arhitekture

	Broj slojeva	Broj neurona	Tačnost
1.	2	15, 10	82%
2.	2	20, 10	85%
3.	2	25, 10	87%
4.	2	30, 10	88%
5.	2	35, 10	74%
6.	2	40, 10	85%
7.	2	50, 10	41%

Zaključeno je da se najbolji rezultati dobijaju za arhitekturu pod rednim brojem 4 i ona je korišćena u svrhu prepoznavanja karaktera u molekularnom grafu.

4.4 Implementacija neuralne mreže

Implementacija prethodno opisanih neuralnih mreža u programskom jeziku Java izvršena je uz pomoć Neuroph-a, aplikativnog softvera koji se koristi za razvijanje algoritama neuralnih mreža. Ovaj softver sadrži Java-biblioteku sa malim brojem osnovnih klasa koje odgovaraju osnovnim operacijama i postupcima za korišćenje neuralnih mreža. Neuroph ima tri osnovne biblioteke:

- org.neuroph.core, koja sadrži osnovne klase,
- org.neuroph.util, koja sadrži neke korisne klase,
- org.neuroph.nnet, koja sadrži klase za implementaciju neuralne mreže i algoritma za učenje.

4.5 Opis Training set-a

Training set, korišćen za učenje prve neuralne mreže (za prepoznavanje molekulskih formula), sastoji se iz nizova dužine u kojima je dat opis slova C i H i cifara 0 – 9.

Svaki karakter napisan je uz pomoć 10 standardnih fontova (Ariel, Ariel Narrow, Calibri, Comic Sans MS, Garamand, Georgia, Rockwell, Times New Roman, Tahoma, Vedrana) i u svim veličinama (8 – 72). To čini 160 primera po karakteru, odnosno ukupno 1920 primera.

Training set, korišćen za učenje druge neuralne mreže (za prepoznavanje strukturnih formula), sastoji se iz nizova dužine u kojima je dat opis slova C i H. Svaki karakter je napisan na isti način, kao i u prvom training set-u, čineći, na taj način, ukupno 320 primera za ovu neuralnu mrežu.

5 Prepoznavanje linija

Linije sa slike se mogu posmatrati kao komponente koje nisu slova i brojevi. Prethodno opisana neuralna mreža je za svaku od komponenti izračunala verovatnoću da je ta komponenta neki od ponuđenih karaktera i izabrala najveću. Tako je dobijen odgovor kojoj komponenti odgovara koji karakter. Međutim, poznato je da se na fotografiji ne nalaze samo karakteri već i linije. Najveća verovatnoća neke komponente za koju se zna da je linija, manja je od najveće verovatnoće komponente za koju se zna da je karakter. Na osnovu ovog razmatranja, uzeta je konstantna vrednost *verConst*, za koju važi:

- ako je najveća verovatnoća komponente veća od *verConst*, onda se radi o karakteru, a neuralna mreža daje informaciju o kom karakteru.
- ako je najveća verovatnoća manja od *verConst* onda je u pitanju linija.

Vrednost *verConst* uzeta je na osnovu sledećeg razmatranja. U tabeli 3 dati su rezultati za različite vrednosti i tačnost neuralane mreže testirane na 100 primera hemijskih formula. Posmatra se procenat tačno određene formule.

Tabela 3. Analiza tačnosti određivanja linija

	verConst	Tačnost
1.	0.90	88%
2.	0.92	86%
3.	0.94	85%
4.	0.96	68%

Na osnovu ovih rezultata za vrednost *verConst* izabrano je 0.90. Kada je određeno koje komponente su linije, a koje karakteri, potrebno je to istaći u matrici slike. Matrica slike je, na kraju ovog postupka, tako obređena, da su linije obeležene negativnim brojevima, a slova i brojevi pozitivnim. To je urađeno uz pomoć DFS algoritma za pretragu grafa, koji je ranije opisan, uz jednu izmenu. Naime, potrebno je posećivati samo ona polja koja sadrže redne brojeve onih komponenti, za koje je prethodno utvrđeno da ne predstavljaju karaktere. Na taj način linije su numerisane negativnim brojevima, počevši od –1, dok karakteri imaju istu numeraciju kao u početnoj matrici.

6 Određivanje povezanosti karaktera

Kako je dobijena matrica slike, u kojoj su slova i brojevi obeleženi pozitivnim brojevima, a linije negativnim, potrebno je odrediti koja su slova međusobno povezana (i kojim linijama). Mogu se primetiti sledeće činjenice:

- jedna linija povezuje tačno dva slova.
- jedno slovo može biti povezano sa drugim slovima proizvoljnim brojem linija.
- dva slova mogu biti povezana sa jednom, dve ili tri linije, što odgovara mogućem broju hemijskih veza u molekulu.
- jedno slovo je povezano sa nekim drugim slovom onom linijom koja mu je fizički najbliža.

Na osnovu ovog razmatranja, moguće je realizovati postupak za pronalaženje veza između karaktera. Za svaku komponentu *i*, za koju je utvrđeno da predstavlja karakter (u matrici slike obeležena je pozitivnim brojem), posmatra se kvadrat sa središtem u središtu te komponente i duplo većim dimenzijama od dimenzija komponente u matrici slike. Svi negativni brojevi koji se nalaze u tom kvadratu predstavljaju redne brojeve linija koje povezuju karakter *i* sa nekim drugim karakterom. To znači da, ako se u tom kvadratu, između ostalog, nalazi i negativan broj *p*, jedan kraj linije *p* sigurno će biti karakter *i*. Kada se ovaj postupak realizuje za svaki karakter *i*,

dobija se informacija o tome koja dva karaktera povezuje linija *p*, za svaku liniju *p*.

Sada je moguće konstruisati matricu susedstva ovog molekularnog grafa, s tim što bi trebalo obratiti pažnju na vrstu hemijske veze u molekulu.

7 Određivanje molekula na osnovu veza između karaktera

Pošto je prethodno opisanim postupkom dobijena matrica susedstva u molekularnom grafu sa slike, potrebno je odrediti koji je graf, odnosno jedinjenje, u pitanju. Ovo je, u suštini, problem određivanja izomorfnosti dva grafa, koji je poznat kao NP-problem, čija složenost još uvek nije precizno određena, ali je to prevaziđeno, u ovom radu, na sledeći način: iskorišćena su hemijska svojstva ugljovodonika i vrednost Wienerovog indeksa koja je konstantna za svako hemijsko jedinjenje. Dobijena vrednost Wienerovog indeksa upoređuje se sa vrednostima iz tabele koja sadrži vrednosti Wienerovog indeksa za sva hemijska jedinjenja i, na osnovu toga, utvrđuje se o kom hemijskom jedinjenju se radi.

7.1 Hemijska svojstva ugljovodonika

Iz hemije je poznato da je opšta formula alkana C_nH_{2n+2}. Ova formula se može interpretirati kao: u svakom molekulu alkana sa n atoma ugljenika postoji tačno $2n+2$ atoma vodonika. Opšta formula alkena je C_nH_{2n}, dok je opšta formula alkina C_nH_{2n-2}. Iz razloga što postoji opšta formula ugljovodonika, moguće je na osnovu broja atoma vodonika i ugljenika otkriti neka svojstva jedinjenja na slici.

Prvo što se može odrediti je vrsta hemijske veze u molekulu. Veza može biti jednostruka (alkani), dvostruka (alkeni) ili trostruka (alkini). Na osnovu broja komponenata na slici, kao i broja atoma vodonika i ugljenika, moguće je odrediti vrstu hemijske veze molekula na slici. Neka je sa br obeležen broj komponenata na slici, a sa c i h broj prepoznatih atoma ugljenika i vodonika, redom.

Za alkane važi da je $br = 2h + 2c - 1$. Kako je poznata opšta formula alkana, takođe važi $h = 2c + 2$. Na osnovu ove dve jednakosti može se zaključiti da važi:

$$br = 6c + 3 \tag{1}$$

Slično se može primetiti da za alkene važi $br = 2h + 2c + 1$, a na osnovu opšte formule važi $h = 2c - 2$, pa se može zaključiti da je:

$$br = 6c \tag{2}$$

Analogno, za alkene važi da je $br = 2h + 2c$ dok je, po opštoj formuli $h = 2c$. Na osnovu toga važi:

$$br = 6c - 3 \tag{3}$$

Proveravanjem koja od jednakosti (1), (2) i (3), važi za posmatrani molekul, određuje se vrsta veze u ovom ugljovodoniku.

Može se pretpostaviti da bi se moglo, na sličan način, brojanjem C i H atoma, odrediti o kom molekulu se radi. Međutim, postoje izomeri - molekuli koji imaju istu molekulsku, a različitu strukturnu formulu, pa je nemoguće samo brojanjem ugljenikovih i vodonikovih atoma uvek jednoznačno odrediti o kom molekulu se radi. Iz tog razloga, za preciznije određivanje molekula iskorišćen je Wienerov indeks (Mohar i Pisanski 1988).

7.2 Wienerov indeks

U hemijskoj teoriji grafova, Wienerov indeks je topološki indeks molekula, definisan kao suma dužina najkraćih puteva između svaka dva čvora u hemijskom grafu. On se formalno može definisati na sledeći način:

Definicija 3. Wienerov indeks grafa $G(V, E)$ je zbir najkraćih puteva između svih parova čvorova u grafu G:

$$W(G) = \sum_{(u,v)\in E} d(u, v)$$

gde je sa $d(u, v)$ obeležena dužina najkraćeg puta između čvorova u i v.

Harry Wiener je osmislio 1947. godine ovaj indeks, po kome je on i dobio ime. Wiener je pokazao da je vrednost Wienerovog indeksa u direktnoj vezi sa nekim osobinama hemijskog jedinjenja, kao što su tačke ključanja, gustina i viskozitet jedinjenja. U ovom radu, upravo je vrednost Winerovog indeksa poslužila da se, na osnovu dobijene matrice susedstva hemijske formule, odredi o kom molekulu je reč. Naime, mnoga hemijska jedinjenja se na jedinstven način mogu odrediti uz pomoću vrednosti Wienerovog indeksa, ali ne i sva. Kada je reč o ugljovodonicima, ovaj način zaključivanja o kom jedinjenju se radi je pouzdan, osim u slučaju malog broja jedinjenja sa velikim brojem C-atoma koja imaju istu vrednost Wienerovog indeksa.

Tabela 4. Vrednosti Wienerovog indeksa za neka hemijska jedinjenja

	Wienerov indeks	Ime jedinjenja
1.	0	metan
2.	1	etan
3.	4	propan
4.	10	Butan
5.	9	2-metil propan
6.	20	pentan
7.	18	2-metil butan
8.	16	2,2-dimetil propan
9.	35	heksan
10.	56	heptan

Potrebno je reći da se pri računanju Wienerovog indeksa nekog organskog jedinjenja, kao čvorovi grafa, posmatraju samo ugljenikovi atomi i računaju samo putevi između njih. U tabeli 4 date su vrednosti Wienerovog indeksa za neka hemijska jedinjenja.

Važno je napomenuti da se pri računanju Wienerovog indeksa ne vodi računa o tipu hemijske veze – da li je ona jednostruka, dvostruka ili trostruka. Već je ranije opisano kako se može odrediti vrsta hemijske veze u molekulu, tako da se pri konačnom određivanju molekula i ovaj parametar mora uzeti u obzir.

Postavlja se pitanje na koji način izračunati vrednost Wienerovog indeksa za neki hemijski graf. Za to je, u ovom radu, korišćen Floyd-Warshallov algoritam za pronalaženje najkraćih puteva između svih parova čvorova.

7.3 Floyd-Warshallov algoritam

Težinski graf $G(V, E)$, sa n čvorova i m grana, je graf čijim su granama pridruženi celi brojevi, koji se nazivaju težinama ili dužinama grana. U vezi sa tim, ako je $e = (u, v)$, grana grafa G, njena dužina se označava sa $w(e)$ ili $w(u, v)$. Dužina puta se definiše kao zbir dužina grana na tom putu. Rastojanje najkraćeg puta od čvora u do čvora v u grafu G, u oznaci $d(u, v)$, je dužina puta najmanje dužine od u do v, ako takav put uopšte postoji. Put dužine $d(u, v)$ od u do v naziva se najkraći put od u do v.

Problem pronalaženja najkraćeg rastojanja za svaki par čvorova u težinskom grafu G naziva se problem najkraćih puteva između svih parova čvorova. Za rešavanje ovog problema postoji veći broj algoritama, a u ovom radu korišćen je Floyd-Warshallov algoritam, koji pripada kategoriji algoritama dinamičkog programiranja.

Floyd-Warshallov algoritam se zasniva na ideji da se određivanje najkraćeg puta između dva čvora v_i i v_j podeli na određivanje dva najkraća puta, jednog od v_i do v_k i drugog od v_k do v_j, gde je v_k neki posredan čvor na putu između v_i i v_j. Preciznije, najkraće rastojanje od v_i do v_j izračunava se koristeći samo posredne čvorove iz skupa $V_k = \{v_1, v_2, \ldots, v_k\}$, ali za svako $k = 1, 2, \ldots, n$.

Najkraće rastojanje $D(i, j, k)$ od čvora v_i do čvora v_j, koristeći posredne čvorove na putevima od v_i do v_j samo iz skupa V_k, za $k = 1, 2, \ldots, n$, računa se na sledeći način.

Početno, za $k = 0$,

$$D(i, j, 0) = \begin{cases} 0 & \text{ako je } i = j \\ w(v_i, v_j) & \text{ako je } (v_i, v_j) \in E \\ +\infty & \text{inače} \end{cases}$$

a za $k = 1, 2, \ldots, n$:

$$D(i, j, k) = \min\{D(i, j, k-1), D(i, k, k-1) + D(k, j, k-1)\}.$$

Drugim rečima, najkraći put koji ide od v_i do v_j i prolazi samo kroz čvorove iz skupa V_k jednak je kraćem od dva moguća puta. Prva mogućnost

je najkraći put koji uopšte ne prolazi kroz čvor v_k, tj. to je dužina najkraćeg puta od v_i do v_j i prolazi kroz čvorove iz skupa V_{k-1}. Druga mogućnost je najkraći put koji prolazi kroz čvor v_k, tj. to je zbir dužina najkraćeg puta od v_i do v_k koji prolazi samo kroz čvorove iz skupa V_{k-1} i najkraćeg puta od v_k do v_j i prolazi samo kroz čvorove iz skupa V_{k-1}.

Po završetku ovog postupka, u matrici $D(i, j, n)$ nalaze se dužine najkraćih puteva između svaka dva čvora u grafu.

Slično svakom algoritmu dinamičkog programiranja, Floyd-Warshallov algoritam je jednostavna iterativna implementacija prethodno opisane rekurzivne formule za izračunavanje najkraćeg puta između svaka dva čvora grafa G.

Floyd-Warshallov algoritam za pronalaženje najkraćeg puta između svaka dva čvora u grafu:

Input: G
for $i = 1$ to n do
  for $j = 1$ to n do
    if $i = j$ then $D(i, j, 0) = 0$;
    else if $(v_i, v_j) \in E$ then $D(i, j, 0) = l(v_i, v_j)$;
      else $D(i, j, 0) = +\infty$;

for $k = 1$ to n do
  for $i = 1$ to n do
    for $j = 1$ to n do
      $D(i, j, k) = \min\{D(i, j, k-1), D(i, k, k-1), D(k, j, k-1)\}$;
return D;

7.4 Pronalaženje Wienerovog indeksa jedinjenja

Za pronalaženje Wienerovog indeksa hemijskog jedinjenja iskorišćen je prethodno opisani Floyd-Warshallov algoritam. Iako hemijski graf nije po definiciji težinski, on se može posmatrati kao težinski graf čije su težine grana međusobno jednake i imaju vrednost 1. Takođe, pošto se kao čvorovi grafa posmatraju samo ugljenikovi atomi, broj čvorova n molekulskog grafa koji se nalazi na slici, jednak je broju komponenata za koje je zaključeno da predstavljaju slovo C. Uz ove napomene, Floyd-Warshallov algoritam se može jednostavno primeniti na problem određivanja Wienerovog indeksa datog molekularnog grafa.

Ako je S suma elemenata matrice $D(i, j, n)$, vrednost Wienerovog indeksa za traženo jedinjenje izračunava se kao $wiener = \frac{S}{2}$. Suma S se deli sa 2 zato što su svi putevi u matrici $D(i, j, n)$ dva puta računati (i put (u, v) i put (v, u), za neka dva čvora u i v). Upoređivanjem dobijenog broja sa tabličnim vrednostima, zaključuje se koje hemijsko jedinjenje je prikazano na slici.

8 Testiranje aplikacije

Aplikacija je testirana na 200 fotografija hemijskih formula. Tačnost rezultata je posmatrana u odnosu na neke parametre, kao što su vrsta hemijske formule, broj C-atoma i vrsta hemijske veze u molekulu.

Zavisnost od vrste hemijske formule. Posmatrajući tačnost optičkog prepoznavanja hemijskih formula ugljovodonika u zavisnosti od vrste hemijske formule kojom je jedinjenje zapisano, dobijeni su sledeći rezultati (tabela 5):

Tabela 5. Tačnost u zavisnosti od vrste hemijske formule

Vrsta formule		Tačnost
1.	strukturna	88%
2.	molekulska	93%

Ovi rezultati su pokazali da se veća tačnost postiže za molekulske, a manja za strukturne formule. To se može objasniti činjenicom da u slučaju strukturnih formula neuralna mreža, osim što prepoznaje karaktere C i H, mora i tačno prepoznavati linije, koje nisu u training setu te neuralne mreže, pa iz tog razloga prepoznavanje ima manju preciznost.

Zavisnost od broja C atoma u molekulu. Posmatrajući tačnost u zavisnosti od broja C atoma u molekulu, dobijeni su sledeći rezultati (tabela 6).

Tabela 6. Tačnost u zavisnosti od broja C-atoma

Broj C-atoma	Molekulska formula	Strukturna formula	Ukupna tačnost
1	100%	100%	100%
2	90%	83%	86%
3	97%	87%	92%
4	95%	90%	92%
5	80%	90%	85%

Tačnost prepoznavanja molekulskih formula ne zavisi od broja C--atoma u molekulu. Sa druge strane, primećeno je da se veća tačnost prepoznavanja strukturnih formula dobija za jedinjenja sa manjim brojem C-atoma. To je u potpunoj saglasnosti sa činjenicom da se verovatnoće tačnog određivanja atoma u molekulu smanjuju sa porastom broja atoma. Naime, ako je verovatnoća da je tačno određen jedan atom, a verovatnoća da je tačno određeno n atoma u molekulu, važi da je $p_n = p_1^n$. Kako je za svako n $0 < p_n < 1$, važi da je uvek $p_{n+1} < p_n$. Ako se uzme u obzir i tačnost prepoznavanja linija, ovakvi rezultati su očekivani.

Zavisnost od vrste hemijske veze u molekulu. Ako se posmatra tačnost u zavisnosti od vrste hemijske veze u molekulu, dobijaju se sledeći rezultati (tabela 7).

Tabela 7. Tačnost u zavisnosti od vrste hemijske veze

Vrsta hemijske veze	Molekulska formula	Strukturna formula	Ukupna tačnost
1	90%	93%	91%
2	93%	70%	81%
3	100%	90%	95%

Primećeno je da tačnost prepoznavanja molekulskih i strukturnih formula ugljovodonika ne zavisi od vrste hemijske veze u molekulu.

Ukupna tačnost. Ukupna tačnost ovako konstruisane aplikacije za prepoznavanje hemijskih formula ugljovodonika je 90%.

9 Zaključak

U ovom radu je opisan jedan od prvih metoda za automatsko prepoznavanje baš hemijskih formula i zato predstavlja napredak u oblasti optičkog prepoznavanja grafova. Korišćenjem ovog metoda, proces digitalizacije bi se ubrzao i omogućilo bi se lakše prepoznavanje komplikovanih hemijskih formula ugljovodonika. Za realizaciju svakog od koraka algoritma: obrada fotografije, izdvajanje i obrada karaktera i linija sa slike, prepoznavanje karaktera, prepoznavanje linija, određivanje povezanosti karaktera, određivanje molekula na osnovu veza između karaktera, moguće je koristiti i druge postupke, pored onih koji su upotrebljeni i opisani u ovom radu, što bi dovelo do drugačijih rezultata.

Kao dalje istraživanje, moguće je proširiti rad na prepoznavanje svih vrsta hemijskih jedinjenja, a ne samo ugljovodonika. Osim toga, moguće je izvršiti poboljšanje, u smislu prepoznavanja formule sa izostavljenim C-atomima, koje sadrže samo linije. Preporuka je da se realizacija izvede uz pomoć Sobel operatora (Vairalkar i Nimbhorkar 2012).

Literatura

Mohar B., Pisanski T. 1988. How To Compute the Wiener Index of the Graph. *Journal of Mathematical Chemistry*, **2**: 267.

Ng A. 2013. *Machine Learning*. Stanford University, Coursera

Otsu N. 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transaction on Systems, Man and Cybernetics*, **9** (1): 62.

Sezgin M., Sankur B. 2004. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, **13** (1): 146.

Stojiljković A. 2012. *Hemija za III razred gimnazije prirodno-matematičkog smera, medicinske, veterinarske i škole za negu lepote*. Beograd: Zavod za udžbenike i nastavna sredstva

Vairalkar M. K., Nimbhorkar S. U. 2012. Edge Detection of Images Using Sobel Operator. *International Journal of Emerging Technology and Advanced Engineering*, **2** (1): 291.

Živković D. 2007. *Osnove dizajna i analize algoritama*. Beograd: Računarski fakultet Beograd i CET Beograd

---

Tamara Stanković and Luka Bulatović

Optical Recognition of Hydrocarbons' Chemical Formula

In this paper, an algorithm for optical Hydrocarbons' chemical formula recognition of given photo, has been described. This algorithm has a few of stages. In the beginning, the photo which is in RGB model is converted to grayscale photo and then to binary image, using Otsu method. Secondly, characters and lines have been extracted using depth-first-search algorithm. They have been recognized using machine learning algorithm neural network, which was implemented using applicative software Neuroph. Furthermore, it has been determined which characters are connected and adjacency matrix has been made. Calculating the value of Wiener index, it has been concluded which molecule is in the photo. This algorithm has been implemented in programming language Java and the precision of that application is 90%.