

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 501

**SUSTAV ZA OTKRIVANJE ZARAZE MALARIJOM  
TEMELJEN NA SLIKAMA RAZMAZA PERIFERNE KRVI**

Domagoj Marić

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 501

**SUSTAV ZA OTKRIVANJE ZARAZE MALARIJOM  
TEMELJEN NA SLIKAMA RAZMAZA PERIFERNE KRVI**

Domagoj Marić

Zagreb, lipanj 2022.

## ZAVRŠNI ZADATAK br. 501

Pristupnik: **Domagoj Marić (0036527031)**  
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo  
Modul: Računarstvo  
Mentor: izv. prof. dr. sc. Alan Jović

Zadatak: **Sustav za otkrivanje zaraze malarijom temeljen na slikama razmaza periferne krvi**

### Opis zadatka:

Nacionalna knjižnica medicine SAD-a, kao vodeća medicinska knjižnica u svijetu, sadrži najveću svjetsku bibliografsku medicinsku bazu podataka MEDLINE koja sadrži skup podataka o zarazi malarijom ([https://data.lhncbc.nlm.nih.gov/public/Malaria/cell\\_images.zip](https://data.lhncbc.nlm.nih.gov/public/Malaria/cell_images.zip)). Skup podataka sadrži 27.558 slika razmaza periferne krvi ljudi jednako podijeljenih u dvije kategorije, zaražene malarijom i nezaražene. Cilj završnoga rada je izraditi automatski sustav za prepoznavanje zaraze malarijom kod ljudi. Sustav interno treba raditi s nekim algoritmom strojnog ili dubokog učenja koji će na temelju slika razmaza periferne krvi čovjeka otkriti koje su stanice zaražene nekim od parazita uzročnika malarije. Sustav je potrebno oblikovati i implementirati u programskom jeziku po vlastitom izboru te izvijestiti o ostvarenom modelu pri čemu je potrebno navesti brojčane vrijednosti vrednovanja modela na skupu za testiranje, uključujući konfuzijsku matricu (engl. confusion matrix).

Rok za predaju rada: 10. lipnja 2022.

*Koristim ovu prigodu da zahvalim izv. prof. dr. sc. Alanu Joviću na podučavanju i pomoći pri izradi rada. Također bih se želio zahvaliti svojoj obitelji i prijateljima koji mi tijekom čitavog studija pružaju bezuvjetnu podršku.*

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Umjetna inteligencija</b>	<b>2</b>
2.1. Turingov test . . . . .	2
2.2. Strojno učenje . . . . .	3
<b>3. Umjetne neuronske mreže</b>	<b>4</b>
3.1. Model umjetnoga neurona . . . . .	5
3.2. Pravila učenja perceptrona . . . . .	6
3.3. Prijenosne funkcije i arhitektura neuronskih mreža . . . . .	6
3.4. Učenje umjetnih neuronskih mreža . . . . .	7
3.5. Problem klasifikacije . . . . .	8
3.6. Primjena neuronskih mreža u medicini . . . . .	9
3.6.1. Detekcija malarije . . . . .	10
<b>4. Konvolucijske neuronske mreže</b>	<b>11</b>
4.1. Struktura konvolucijskih neuronskih mreža . . . . .	12
4.2. Konvolucijski sloj . . . . .	12
4.3. Sloj sažimanja . . . . .	13
4.4. Funkcije gubitka . . . . .	15
<b>5. Izrada sustava za detekciju malarije</b>	<b>17</b>
5.1. Skup podataka . . . . .	17
5.1.1. Predobrada podataka . . . . .	18
5.2. Složeni model neuronske mreže . . . . .	19
5.3. Jednostavni model neuronske mreže . . . . .	20
<b>6. Interpretacija rezultata učenja</b>	<b>21</b>
6.1. Prikaz funkcijama gubitka . . . . .	22

6.2. Prikaz konfuzijskom matricom . . . . .	24
<b>7. Zaključak</b>	<b>26</b>
<b>Literatura</b>	<b>27</b>

# 1. Uvod

Malarija je zarazna bolest, prenošena ženjkama komaraca roda lat. *Anopheles*, koju izazivaju protoktisti roda lat. *Plasmodium*. Postoji 5 parazitnih vrsta koje uzrokuju malariju kod ljudi, od kojih 2 vrste lat. *P. falciparum* i lat. *P. vivax*, predstavljaju najveću prijetnju za zdravlje ljudi.

Prema izvješću [12] Svjetske zdravstvene organizacije iz 2021. godine, u svijetu je 2020. godine postojao 241 milijun slučajeva zaraze malarijom (broj se smrtnih slučajeva procjenjuje na 627 000) od čega područje afričkoga kontinenta nosi najveće breme, čak 95% ukupnih slučajeva zaraze i 96% ukupnih smrtnih slučajeva. Četiri afričke zemlje činile su nešto više od polovice svih smrtnih slučajeva zaraze malarijom u svijetu: Nigerija (31,9%), Demokratska Republika Kongo (13,2%), Ujedinjena Republika Tanzanija (4,1%) i Mozambik (3,8%).

Tema je ovoga rada primjena načela umjetne inteligencije, odnosno algoritama strojnog učenja u području medicine. Strojno učenje grana je umjetne inteligencije koja se bave oblikovanjem algoritama koji svoju učinkovitost poboljšavaju na temelju empirijskih podataka. Ono je jedno od danas najaktivnijih i najuzbudljivijih područja računarске znanosti, ponajviše zbog brojnih mogućnosti primjene kao što je medicinska dijagnostika.

Rad će ponajprije dati osnovne uvide u umjetnu inteligenciju te strojno učenje, nakon čega slijedi opis umjetnih neuronskih mreža kao središnjega pojma rada. Nadalje, uz teoretska načela, u radu će biti prikazan rad jedne od vrsta umjetnih neuronskih mreža na jednoj od najvećih svjetskih medicinskih baza podataka *MEDLINE* koja sadrži skup podataka o zarazi malarijom.

## 2. Umjetna inteligencija

Umjetna inteligencija dio je računarske znanosti koji se bavi razvojem sposobnosti računala da obavljaju zadaće za koje je potreban oblik inteligencije sličan ljudskoj, tj. da se mogu snalaziti u novim prilikama, učiti nove koncepte i na kraju donositi zaključke. Iako mlada znanost, umjetna inteligencija naslijedila je mnoge ideje, pristupe i tehnike iz drugih disciplina, a posebno onih koje se bave istraživanjem načina ljudskoga razmišljanja: kognitivne znanosti, logike, psihologije, filozofije, matematike i dr. Ideje o umjetnoj inteligenciji su se već od pradavnih vremena pojavljivale u mitovima (npr. mit o Golemu) i literaturi. Primjer je riječ robot, koju češki pisac Karel Čapek upotrebljava da bi opisao uređaje za pomoć ljudima u svakodnevnom životu koji obavlja zadatke umjesto čovjeka. Nadalje, značajan je povijesni primjer bio Turčin, automat koji je 1770. godine Wolfgang von Kempelen konstruirao da igra šah i izvodi skakačev obilazak, što je u konačnici ispala vješto izvedena mehanička iluzionistička naprava. Međutim, razvoj je umjetne inteligencije započeo pojavom digitalnih računala u drugoj polovici 20. stoljeća. Među prve teoretske radove iz toga područja ubrajaju se razni radovi A. M. Turinga, dok se prvim računalnim programom umjetne inteligencije smatra logički teoretičar (engl. *The Logic Theorist*) [7]. Pojam umjetne inteligencije skovao je John McCarthy, kojega se danas smatra jednim od začetnika umjetne inteligencije, 1956. godine na Dartmutoj konferenciji (Dartmouth Conference, 1956.).

### 2.1. Turingov test

A. M. Turing je 1950. godine u svome radu engl. *Computing Machinery and Intelligence* [10] predložio operacionalizaciju pitanja "mogu li strojevi misliti?". Misaoni je to pokus koji uspoređuje performanse pretpostavljenog inteligentnog stroja i čovjeka na temelju nekog skupa pitanja.

Test se provodi s 3 sudionika od kojih svaki ima drugačiji cilj. Čovjek u ulozi ispitivača prirodnim jezikom komunicira s jednom osobom i jednim strojem, a pri tome se obojica ispitivaču predstavljaju ljudski. Ako ispitivač ne može sa sigurnošću



utvrditi koji je sugovornik stroj, smatra se da je stroj prošao test.

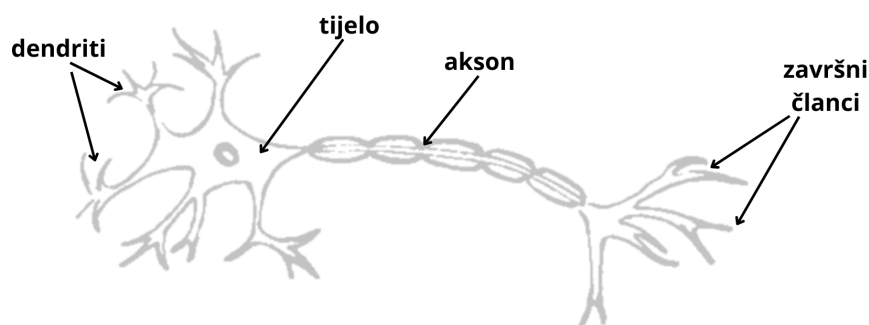
## 2.2. Strojno učenje

Strojno se učenje može definirati kao "programiranje računala tako da optimiraju neki kriterij uspješnosti temeljem podatkovnih primjera ili prethodnog iskustva. Raspoložemo modelom koji je definiran do na neke parametre, a učenje se svodi na izvođenje algoritma koji optimira parametre modela na temelju podataka ili prethodnog iskustva." [1]

Na temelju već viđenih podataka, model mora moći predvidjeti svojstva na do sada neviđene podatke, tj. mora generalizirati skup podataka. Povučemo li analogon s logičkim zaključivanjem, model radi induktivno zaključivanje, odnosno na temelju ograničenoga skupa podataka izvodi pravila za objašnjavanje određenih značajki. Stoga se kao cilj strojnoga učenja stavlja izrada modela koji dobro generaliziraju podatke. Danas se razlikuju tri vrste strojnoga učenja, nadzirano učenje (engl. *supervised learning*, u kojemu su nam podatci dani u paru ulaz i očekivani izlaz), nenadzirano učenje (engl. *unsupervised learning*, u kojemu su podatci dani bez ciljne vrijednosti te je potrebno pronaći pravilnosti u podacima) i podržano učenje (engl. *reinforcement learning*, koje se temelji na učenju optimalne strategije metodom pokušaja s određenom nagradom) [13].

### 3. Umjetne neuronske mreže

Umjetne se neuronske mreže se pojavljuju 40-ih godina 20. stoljeća, kao pokušaj imitacije biološkog mozga. Mozak je središte živčanoga sustava bića koji se sastoji od nakupina neurona. Neuroni, odnosno živčane stanice, građene su od tijela, dendrita, aksona te završnih članaka. Procjenjuje se da ljudski mozak sadrži 14 milijardi neurona, od kojih je svaki neuron sinapsama povezan s nekoliko tisuća drugih. Izgled neurona opisan je slikom 3.1.

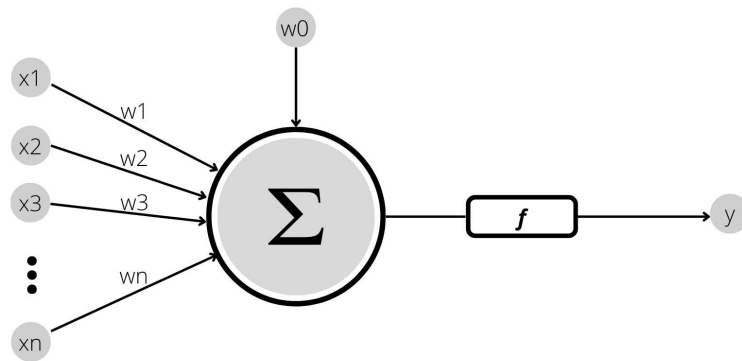


Slika 3.1: Biološki neuron

Kao što je već spomenuto, ideja o nastanku umjetnih neuronskih mreža pojavila se 1943. godine na engl. *Massachusetts Institute of Technology*, kada je grupa znanstvenika počela proučavati rad mozga i dala opis matematičkoga modela neuronske mreže. U nadolazećim su se godinama počele pojavljivati prve programske izvedbe, no zbog manjka snage tadašnje računalne opreme, nisu mogle rješavati kompleksnije probleme. Danas umjetne neuronske mreže imaju primjene u raznim sferama ljudskoga života kao što su autonomna vozila, analiza signala u biomedicini, meteorološko predviđanje i igranje igara. Neki od uspješnijih podviga u području igara svakako su AlphaGo i AlphaZero razvijeni od strane tvrtke DeepMind, dva računalna programa temeljena na dubokim neuronskim mrežama koje u igrama Go i šah bez problema pobjeđuju najvještije ljudske igrače.

### 3.1. Model umjetnoga neurona

Općenito govoreći, umjetni je neuron modeliran da prati građu biološkoga, što se najbolje može pokazati na modelu *perceptrona*, točnije TLU-perceptrona (engl. *Threshold Logic Unit*). Primjer jednog takvog dan je na slici 3.2.



Slika 3.2: Građa umjetnoga neurona

Vrijednosti sa svakog od ulaza TLU-perceptrona, označene vrijednostima  $x_1, x_2, x_3, \dots, x_n$ , množe se osjetljivošću specifičnoga ulaza, odnosno njegovom težinom  $w_i$  koja se zbraja u tijelu perceptrona. Težine umjetnoga neurona predstavljaju vrijednost koja govori koliko određeni ulaz doprinosi odluci o konačnome izlazu. Ukupnom se zbroju dodaje  $w_0$  (druga oznaka  $b$  od engl. *bias*) koja označava pomak. Zatim se akumulirana suma propušta kroz prijenosnu funkciju skoka  $f$  čime nastaje izlazna vrijednost  $y$ . Specifično za TLU-perceptrone izlaz će biti 0 ili 1, ovisno o tome je li akumulirana suma umnoška težina i ulaza veća od praga  $t$ . Realni parametar  $t$  predstavlja prag podražljivosti neurona, odnosno prag pri kojemu se neuron aktivira. Izlaz neurona dan je formulom (3.1.) u nastavku:

$$y = \begin{cases} 0, & \text{za } w_0 + \sum_{i=1}^n x_i w_i < t \\ 1, & \text{za } w_0 + \sum_{i=1}^n x_i w_i \geq t \end{cases} \quad (3.1)$$

## 3.2. Pravila učenja perceptrona

1949. godine Kanadski psiholog Donald Hebb u jednoj od svojih knjiga, *Organizacija ponašanja* navodi da se sinaptičke veze pojačavaju kada se dva ili više neurona aktiviraju kontinuirano u vremenu i prostoru [5]. Ta se spoznaja u kontekstu neuronskih mreža može prenamijeniti da učiti znači mijenjati jakosti veza. Ako dva skupa točaka trebaju biti linearno odvojiva perceptronom, moraju se pronaći odgovarajuće težine za računsku jedinicu. Proces pronalaska odgovarajućih težina bismo željeli automatizirati da se težine ne moraju ručno korigirati. 1958. godine Rosenblatt spaja Hebbovu ideju učenja s jednostavnim modelom perceptrona te nudi pravilo učenja perceptrona koje se bavi ovim problemom. Pseudokod je prikazan Algoritmom 1 [9].

---

### Algoritam 1 Pravilo učenja perceptrona

---

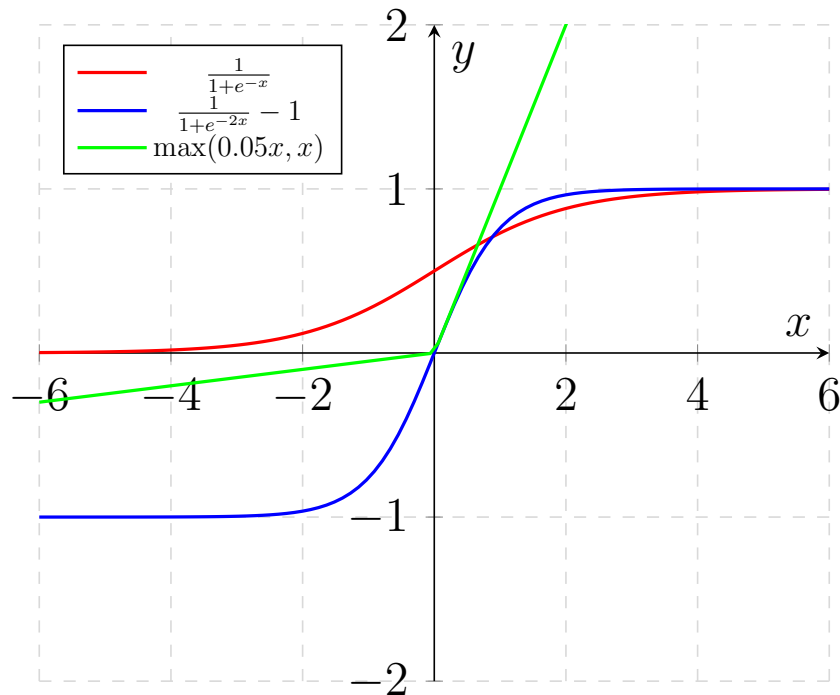
**Ulaz:**  $A \leftarrow$  parovi ulaz, izlaz  
Inicijaliziraj težine  $\mathbf{w}$  i  $b$   
**while** *not* dobro klasificira **do**  
  **for**  $i = 1, N$  **do**  
     $pogreska = y_i - \sigma(\mathbf{w}^T x_i + b)$   
    **if**  $pogreska \neq 0$  **then**  
       $\mathbf{w} = \mathbf{w} + pogreska \times x_i$   
       $b = b + pogreska$   
    **end if**  
  **end for**  
**end while**

---

## 3.3. Prijenosne funkcije i arhitektura neuronskih mreža

U prethodnome je poglavlju objašnjena osnovna ideja građe umjetnoga neurona, koja naizgled nema nikakvih mana. Međutim, u praksi umjetne neuronske mreže izgrađene od većeg broja takvih neurona nisu davale obećavajuće rezultate. Glavni je razlog tomu bio što je mala promjena neke od težina na ulazu rezultirala velikim promjenama na izlazu neurona. Pogledom na prethodno poglavlje vidi se da je izlaz neurona binaran, dakle može biti samo 0 ili 1, što bi značilo da upravo ta mala promjena na ulazu može promijeniti izlaz s 0 na 1 i obratno. Rješenje takvoga problema bilo je otkrivanje drugih prijenosnih funkcija od kojih su danas najkorištenije već spomenuta funkcija skoka (TLU-perceptroni), sigmoidalna funkcija (na slici 3.3 prikazana crvenom bojom), tan-

gens hiperbolni (na slici 3.3. prikazana plavom bojom), softmax, zglobnica i propusna zglobnica (na slici 3.3. prikazana zelenom bojom uz parametar  $\alpha = 0.05$ ).



**Slika 3.3:** Primjeri aktivacijskih funkcija

Kao što je biološki mozak građen od velikoga broja bioloških neurona, umjetne su neuronske mreže građene od niza umjetnih neurona povezanih međusobno. Obično su neuroni posloženi u slojeve. Različiti slojevi mogu izvoditi različite transformacije na svojim ulazima. Signali putuju od prvog (ulaznog) sloja do posljednjeg (izlaznog) sloja nakon višestrukog prelaska unutarnjih (skrivenih) slojeva.

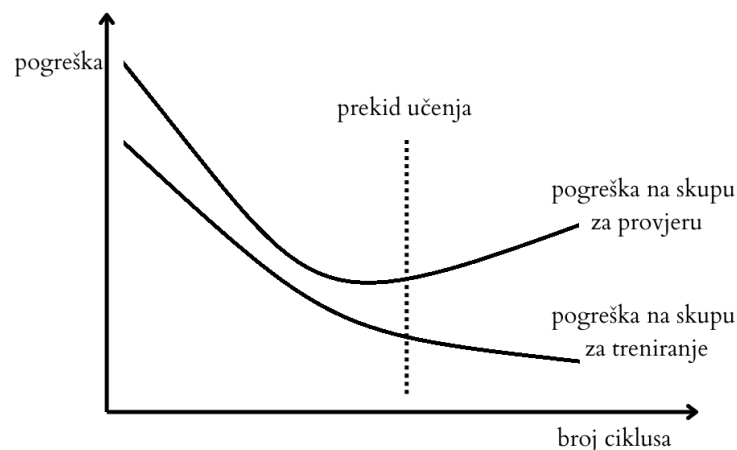
### 3.4. Učenje umjetnih neuronskih mreža

U stvarnome svijetu skup podataka nad kojim se provodi učenje nikad neće biti savršen, tj. neće sadržavati sve primjere. Iz toga razloga skup podatka za učenje dijeli na tri podskupa: skup za učenje (npr. 70% dostupnih primjeraka), skup za provjeru (npr. 15% dostupnih primjeraka) i skup za testiranje (npr. 15% dostupnih primjera nad kojima se provodi konačna provjera mreže).

Neuronske se mreže uče obradom primjera, od kojih svaki sadrži poznati ulaz i izlaz (ova je vrsta učenja poznata kao nadzirano učenje), tvoreći asocijacije ponderirane vjerojatnosti između njih dvoje, koje su pohranjene unutar strukture podataka same

mreže. Učenje neuronske mreže iz danog primjera obično se provodi određivanjem razlike između obrađenog izlaza mreže i ciljnog izlaza. Ovu razliku nazivamo pogreškom. Mreža zatim prilagođava svoje težine pravilima učenja koristeći vrijednost pogreške. Uzastopne prilagodbe uzrokovat će da neuronska mreža proizvodi izlaz koji je sve sličniji ciljnom izlazu. Nakon dovoljnog broja ovih prilagodbi, obuka se može prekinuti na temelju određenih kriterija. Jasnije objašnjenje bit će dato u poglavlju 4.

Jedan od problema vezanih za učenje neuronskih mreža svakako je prenaučenosť (engl. *overfitting*). U tome slučaju umjetna se neuronska mreža prilagođava podacima učeći ih napamet. Od posebnoga značaja je praćenje krivulje učenja. Mrežu se pusti da uči do trenutka dok se ne uoči gubitak generalizacije, te se daljnji postupak učenja prekida. Takvo se ponašanje može vidjeti na slici 3.4.



Slika 3.4: Učenje neuronske mreže

### 3.5. Problem klasifikacije

Klasifikacija je proces vezan uz kategorizaciju, proces pridjeljivanja oznaka objektima na temelju prepoznatljivih značajki, odnosno grupiranje povezanih činjenica u klase. Kategoriziramo li podatke u dvije klase govorimo o binarnoj klasifikaciji, dakle možemo koristiti sustav koji će imati samo jedan izlaz i na njemu dva jasno razlučiva stanja. Imamo li pak više razreda u koje se podatci klasificiraju, uobičajeno je koristiti jednojedinичno kodiranje (engl. *one-hot encoding*), dakle izlaza ima onoliko koliko

ima razreda, pri čemu je  $i$ -ti izlaz postavljen u 1 ako podatak pripada  $i$ -tom razredu, inače je postavljen u 0.

Kao najjednostavniji primjer binarnoga klasifikatora možemo uzeti onaj koji razvrstava slike na slike pasa ili mačaka. Neka nam bit 0 predstavlja razred pasa, dok bit 1 predstavlja razred mačka. Tada bismo za slike (slika 3.5.) u nastavku očekivali vrijednosti izlaza koje su prikazane ispod svake od slika.



(a) Macka, 1



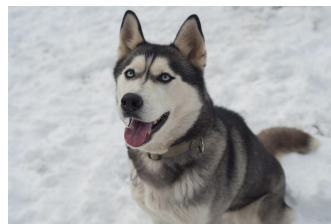
(b) Pas, 0

**Slika 3.5:** Primjer binarne klasifikacije

Uzmimo sada na primjer klasifikator koji slike klasificira u 3 razreda, psi, mačke i slonovi. Takve je razrede potrebno jednojedinčno zakodirati, pa se možemo odlučiti razred psa kodirati s 001, razred mačke s 010, te razred slona s 100. Provedemo li sada slike u nastavku kroz takav jedan klasifikator možemo očekivati izlaze kakvi su dani na slici 3.6. ispod svake od slika.



(a) Macka, 010



(b) Pas, 001



(c) Slon, 100

**Slika 3.6:** Primjer klasifikacije u više razreda

### 3.6. Primjena neuronskih mreža u medicini

Do kraja 90-ih godina prošloga stoljeća, nadzirano učenje postalo je popularno u medicinskoj obradi slika, u kojoj su slike korištene za dizajniranje modela. Međutim konvencionalne metode strojnog učenja zahtijevaju ključni proces izdvajanja značajki koji bi trebali provesti istraživači. Sljedeća razina razvoja je konstruirati sustav

koji može automatski naučiti značajke i može predstavljati slike na optimalan način. Umjetne neuronske mreže imaju potencijal učiniti zdravstvenu njegu humanom, smanjujući ogroman teret zdravstvenim radnicima koji se suočavaju sa sve starijom i mnogoljudnijom populacijom. Neuronske mreže mogle bi se uhvatiti u koštac sa sistemskim problemima zdravstvenih sustava, poput dugog čekanja, dijagnostičkih pogrešaka, neadekvatnog pristupa specijalističke skrbi u udaljenim područjima te mnogih drugih. Također postoje različiti modaliteti medicinskog snimanja uz pomoć kojih se hvataju kritične slike pacijentovih tjelesnih struktura i unose u model neuronske mreže za analizu. Jedno od takvih ostvarenja bit će izrada sustava za detekciju malarije klasifikacijom razmaza periferne krvi na zaražene i nezaražene.

### **3.6.1. Detekcija malarije**

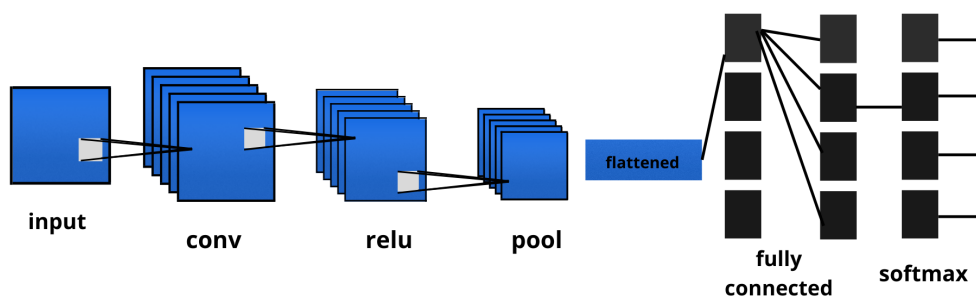
Na zarazu malarijom može se posumnjati na temelju pacijentove povijesti putovanja, simptoma i fizičkih nalaza na pregledu. Međutim, da bi se postavila konačna dijagnoza, laboratorijski testovi moraju pokazati parazite malarije ili njihove dijelove. U nekim endemskim područjima malarije prijenos je toliko intenzivan da je velik dio stanovništva zaražen, ali ne i obolio od parazita. Takvi su nositelji razvili dovoljno imuniteta da se zaštite od malarije, ali ne i od infekcije. Postoje razne metode testiranja zaraze od kojih ćemo istaknuti kliničko testiranje, mikroskopsku analizu razmaza krvi i testiranje antigena. Iako sve imaju svoje prednosti i nedostatke fokusirat ćemo se na mikroskopsku analizu.

Paraziti malarije mogu se identificirati ispitivanjem pod mikroskopom kapljice pacijentove krvi, raširene kao "krvni razmaz" na stakalcu mikroskopa. Prije pregleda uzorak se boji (najčešće bojanje je Romanowsky-Giemsa) kako bi se parazitima dao prepoznatljiv izgled. Ova tehnika ostaje zlatni standard za laboratorijsku potvrdu malarije. Međutim, to ovisi o kvaliteti reagensa, mikroskopa i iskustvu laboratorija [4].



## 4. Konvolucijske neuronske mreže

Konvolucijske neuronske mreže (engl. *convolutional neural networks*, skraćeno *CNN*) su vrsta neuronskih mreža posebno pogodne za obradu nestrukturiranih podataka, posebice vezanih za slike, tekst, zvuk i govor. Arhitektura se CNN često sastoji višeslojnih perceptronima u više dimenzija. Višeslojni perceptroni obično znače potpuno povezane mreže, odnosno svaki neuron u jednom sloju povezan je sa svim neuronima u sljedećem sloju. Upravo puna povezanost ovih mreža čini ih sklonim prenaučivosti podataka. Tipični načini regularizacije ili sprječavanja prenaučivosti uključuju: kažnjavanje parametara tijekom treninga (kao što je smanjenje težine) ili skraćivanje povezivanja (preskakanje veza, ispadanje itd.). Na slici 4.1. prikazan je model jednostavne konvolucijske mreže sa svim slojevima, čija će građa biti detaljnije opisana u nastavku.



Slika 4.1: Arhitektura jednostavne CNN

## 4.1. Struktura konvolucijskih neuronskih mreža

Konvolucijska se, kao i obična, neuronska mreža sastoji od jednog ulaznog, jednog izlaznog i jednog ili više skrivenih slojeva. Kao što je vidljivo na slici 4.1. kod konvolucijskih neuronskih mreža specifični su konvolucijski slojevi i slojevi sažimanja. Mreža najčešće završava s jednim ili više potpuno povezanih slojeva koji služe za klasifikaciju. Konvolucija ima tri vrlo značajna svojstva koja uvelike mogu poboljšati sustav učenja: raspršenu povezanost, dijeljenje parametara i translatornu ekvivarijantnost.

Svojstvo **raspršenosti** postiže se izborom filtera koji su manjih dimenzija od ulazne slike. Unatoč tome što ulazna slika može imati milijune piksela, možemo na njoj prepoznati male, nama bitne značajke, kao što su rubovi objekata izborom filtera malih dimenzija.

Pojam **dijeljenja parametara** predstavlja korištenje istih parametara modela više puta. Dok se kod klasičnih neuronskih mreža svaka težina matrice koristi točno jednom pri računanju izlaznog rezultata sloja, u konvolucijskim mrežama, svaka težina u filteru koristi se iznova na svakoj novoj poziciji ulazne aktivacijske mape.

**Translatorna ekvatorijalnost** predstavlja očuvanje, svojstvo uzrokovano vrstom dijeljenja parametara. Primjerice, ako je neka funkcija  $f$  takva da translata ulaznu sliku za određeni broj piksela, tada će i izlazni rezultat, nakon djelovanja funkcije  $f$  na transliranu sliku, također biti transliran za isti broj piksela [2].

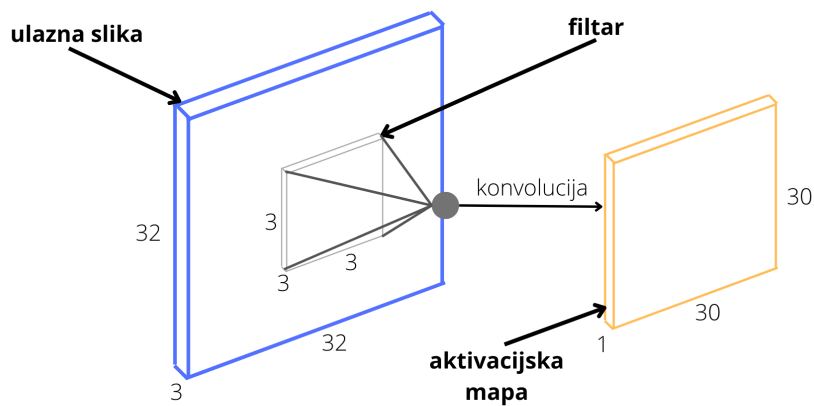
## 4.2. Konvolucijski sloj

Konvolucijski je sloj (engl. *convolutional layer*) osnovni dio svake konvolucijske neuronske mreže. Konvolucijski se slojevi najčešće sastoje od dva dijela, određenog broja filtera koji sadrže težine i sloja aktivacijske funkcije (što je vidljivo na slici 4.1.). Filtri su najčešće manjih prostornih dimenzija nego ulaz, ali uvijek imaju jednaku dimenziju izlaza. Uzmimo za primjer RGB (engl. *red green blue*) slike dimenzija  $64 \times 64$  piksela. Filter može biti dimenzije  $20 \times 20$  no potrebna su nam tri takva filtra, po jedan za svaki kanal (R, G i B) slike. Svaki sloj najčešće se sastoji od više filtera i svaki od njih će generirati jednu dvodimenzionalnu aktivacijsku mapu. Izlaz sloja bit će sve aktivacijske mape, odnosno izlaz će biti više 2D matrica gdje svaka predstavlja jednu dubinu izlaza. Kako bi bilo moguće definirati točnu veličinu izlaza konvolucijskog sloja potrebno je definirati i korak pomaka filtra (engl. *stride*). Korak predstavlja broj piksela za koji se pomičemo u horizontalnom i vertikalnom smjeru pri konvoluciji aktivacijske mape i filtera.

Neka je na ulazu konvolucijskog sloja slika dimenzija  $n \times n$ . Slika ulazi u konvoluciju s filterom dimenzije  $f \times f$  te proizvodi izlaznu aktivacijsku mapu dimenzija

$$(n - f + 1) \times (n - f + 1). \quad (4.1)$$

Uzmimo za primjer neku RGB sliku, koju skaliramo na dimenziju  $32 \times 32$ . Ta slika neka ulazi u konvoluciju s filtrom dimenzija  $3 \times 3 \times 3$ . Izlazna će aktivacijska mapa biti dimenzija  $(32 - 3 + 1) \times (32 - 3 + 1)$ , odnosno  $30 \times 30$ . Na slici 4.2. vidimo jednu takvu transformaciju.



Slika 4.2: Primjer konvolucije ulaza i filtra

### 4.3. Sloj sažimanja

Slojevi sažimanja (engl. *pooling layers*) se često koriste kod CNN nakon nekoliko konvolucijskih slojeva kako bi smanjile rezoluciju aktivacijskih mapa. Osim smanjivanja rezolucije, također povećavaju prostornu invarijantnost (neosjetljivost na manje pomake značajki u uzorku) mreže. Dakle, daljnje operacije se izvode na sažetim značajkama umjesto na precizno pozicioniranim značajkama koje generira sloj konvolucije. Operacija sažimanja na ulaznoj slici, uglavnom primjenjuje sumirane statističke podatke na lokalnom području zadanom veličinom filtra. Dva najčešća tipa sažimanja su sažimanje izborom maksimalnog elementa (engl. *max-pooling*) i sažimanje prosječnom vrijednošću (engl. *average-pooling*). Pogledajmo primjer sažimanja izborom najvećeg elementa na programskome primjeru (implementacija vanjskim bibliotekama *NumPy* i *TenstorFlow* u jeziku Python) prikazanom u isječku koda na slici 4.3.

---

```

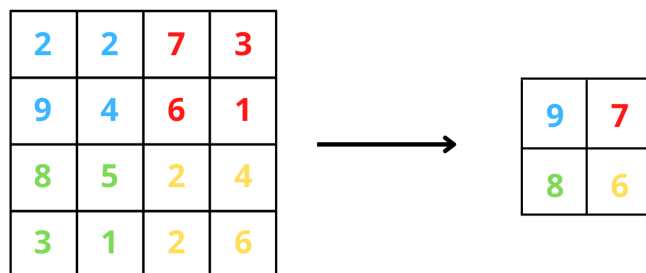
1 import numpy as np
2 from keras.models import Sequential
3 from keras.layers import MaxPooling2D
4
5 image = np.array([[2, 2, 7, 3],
6                  [9, 4, 6, 1],
7                  [8, 5, 2, 4],
8                  [3, 1, 2, 6]])
9 image = image.reshape(1, 4, 4, 1)
10
11 model = Sequential(
12     [MaxPooling2D(pool_size = 2, strides = 2)])
13
14 output = model.predict(image)
15 output = np.squeeze(output)
16 print(output)

```

---

**Slika 4.3:** Implementacija sažimanja izborom maksimalnog elementa

Programski isječak stvara neku monokromatsku sliku u obliku matrice koju zatim dijeli u 4 grupe što je vidljivo na slici 4.4 (lijeva matrica  $4 \times 4$ , grupe su označene različitim bojama). Zatim se definira jednostavan model konvolucijske mreže koja sadrži sloj sažimanja s jednim filtrom dimenzije  $2 \times 2$  s pomakom 2. Provedemo li sažimanje izborom maksimalnog elementa nad ovako definiranom aktivacijskom mapom dobit ćemo manju mapu dimenzija  $2 \times 2$  koja sadrži elementi koji su najveći iz svake pojedine grupe. Rezultat izvođenja programskog isječka vidljiv je na slici 4.4.



**Slika 4.4:** Grafički prikaz programskog primjera

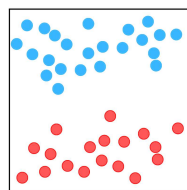
## 4.4. Funkcije gubitka

Funkcije gubitka (engl. *loss functions*) uglavnom su klasificirane u dvije različite kategorije, klasifikacijske i regresijske funkcije gubitka. Klasifikacijski gubitak je slučaj kada je cilj predvidjeti izlaz iz različitih kategoričkih vrijednosti, na primjer, ako imamo skup podataka rukom napisanih slika i treba predvidjeti napisanu znamenku koja se nalazi između  $[0, 9]$ . U ovakvim scenarijima koristi se klasifikacijska funkcija gubitka. Međutim ako je problem regresijske prirode kao što je predviđanje kontinuiranih vrijednosti na primjer (npr. treba predvidjeti vremenske uvjete ili predvidjeti cijenu kuća na temelju nekih karakteristika) koristi se regresijski gubitak. U ovome poglavlju fokusirat ćemo se na nekoliko funkcija koje su danas u širokoj upotrebi.

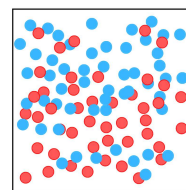
**Tablica 4.1:** Funkcije gubitka

Oznaka	Naziv	Jednadžba
$\mathcal{L}_1$	srednja apsolutna greška, $L_1$	$\frac{1}{N} \sum_{i=1}^N  y_i - \hat{y} $
$\mathcal{L}_2$	srednja kvadratna greška, $L_2$	$\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2$
$\log$	unakrsna entropija	$-\sum p(x) \log q(x)$

Posebnu ćemo pažnju dati unakrsnoj entropiji s obzirom na to da su srednja apsolutna i kvadratna pogreška vezane za problematiku regresije, a unakrsna entropija za problematiku klasifikacije [6]. Koncept unakrsne entropije seže u polje teorije informacija kada je Claude Shannon 1948. godine uveo koncept informacijske entropije. Entropiju možemo definirati kao mjeru neuređenosti sustava. Na slici 4.4. možemo vidjeti primjere sustava manje i veće entropije.



(a) Sustav male entropije



(b) Sustav veće entropije

**Slika 4.5:** Primjer entropije sustava

Kako unakrsna entropija služi za rješavanje klasifikacijskih problema, dijeli se na binarnu (engl. *binary cross-entropy*), kategoričku (engl. *categorical cross-entropy*) i rijetku (engl. *sparse categorical cross-entropy*) kategoričku unakrsnu entropiju. Kao

što i samo ime govori, binarna unakrsna entropija vrlo je pogodna za učenje modela za rješavanje mnogih klasifikacija problema u isto vrijeme, ako svaka klasifikacija može biti svedena na binaran izbor. Binarna unakrsna entropija se računa kao na formuli 4.2.

$$-\frac{1}{N} \sum_{i=1}^N [y_n \log \hat{y}_n + (1 - y_n) \log (1 - \hat{y}_n)] \quad (4.2)$$

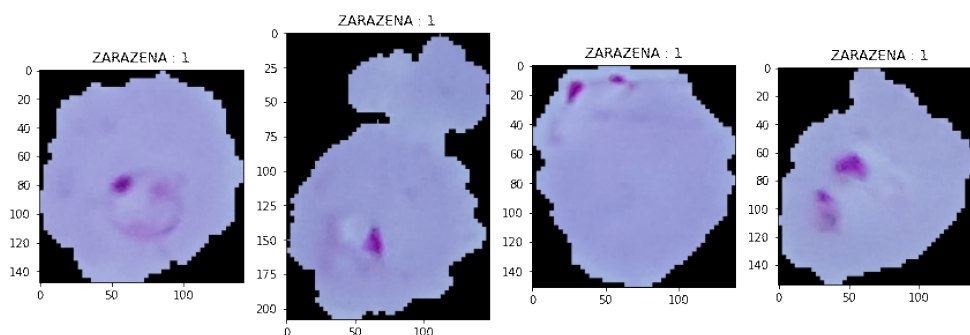
$y_n$  predstavlja vektor vrijednosti stvarnih klasa ulaznih primjera,  $\hat{y}_n$  vektor izlaznih vrijednosti klasa koje kao predviđanja daje neuronska mreža, dok  $N$  predstavlja veličinu skupa primjera.

## 5. Izrada sustava za detekciju malarije

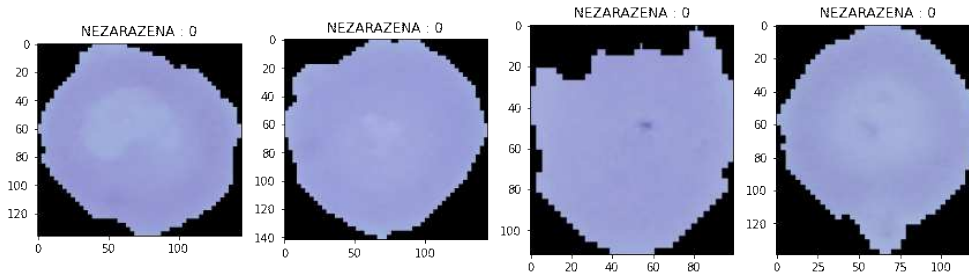
Kao što je već spomenuto, konvolucijske neuronske mreže postižu veliku točnost u radu sa slikovnim podacima, pa se nameću kao jedan od logičnih izbora za izradu sustava. U ovome će poglavlju biti objašnjena izrada mreže za klasifikaciju slika razmaza periferne krvi na one zaražene i nezaražene malarijom. Tekstni će dio opisa biti popraćen slikama implementacije sustava u programskome jeziku *Python* koristeći vanjske biblioteke za rad s algoritmima strojnoga učenja *TensorFlow*, *scikit-learn* i *NumPy* te *OpenCV* za predobradu ulaznog skupa podataka.

### 5.1. Skup podataka

Skup podataka preuzet je sa stranice Nacionalne knjižnice medicine SAD-a, iz najveće svjetske bibliografske baze podataka *MEDLINE* koja, između ostaloga, sadrži podatke o osobama zaraženim malarijom. Korišteni skup podataka sadrži 27 558 slika razmaza periferne krvi jednako podijeljenih u dva razreda, zaražene i nezaražene malarijom. Svaki je primjerak jedna RBG slika. Na slikama 5.1. i 5.2. su prikazani neki od primjera iz skupa.



Slika 5.1: Nasumično izabrani primjeri zaraženih



Slika 5.2: Nasumično izabrani primjeri nezaraženih

### 5.1.1. Predobrada podataka

Kao što je već objašnjeno u prethodnome poglavlju dobra je praksa skup podataka podijeliti na skup za učenje, validaciju i testiranje. Kako je ovaj skup podataka dovoljno velik, uzmimo 20% originalnog skupa za testiranje, te od ostatka uzmimo 70% u korist skupa za učenje te ostalih 30% za validaciju. Pogledajmo sada sliku 5.2.

```
def load_img_data(path):
    image_files = glob.glob(os.path.join(path, "Parasitized/*.png")) + glob.glob(os.path.join(path, "Uninfected/*.png"))
    X, y = [], []
    for image_file in image_files:
        # 0 za nezaražene i 1 za zaražene
        label = 0 if "Uninfected" in image_file else 1
        # učitavanje slike kao monokromatske
        img_arr = cv2.imread(image_file, cv2.IMREAD_GRAYSCALE)
        # skaliranje slike na dimenzije (96x96)
        img_resized = cv2.resize(img_arr, (img_size, img_size))
        X.append(img_resized)
        y.append(label)
    return X, y
```

Slika 5.3: Funkcija za učitavanje podataka

Na slici je dizajnirana funkcija `load_img_data` koja, koristeći podrazumijevanu Pythonovu biblioteku `glob`, pronalazi sve nazive puteva koji odgovaraju određenom uzorku prema pravilima koja koristi Unixova ljska, iako se rezultati vraćaju proizvoljnim redoslijedom. Nadalje, funkcija sadrži jednu petlju `for`, koja prolazi po svim slikama. Uz pomoć vanjske knjižnice `OpenCV` (zaslužne za obradu slika i računalni vid) čita sliku, međutim u monokromatskom obliku radi lakšega učenja mreže, koju zatim skalira da sve budu uniformne veličine npr.  $96 \times 96$  piksela. Zatim, sliku i njenu labelu, tj. klasu vraća kao dvije liste.

```
X, y = load_img_data(img_dir)
X = np.array(X).reshape(-1, img_size, img_size, 1)
X = X / 255
```

Slika 5.4: Skaliranje podataka

Nakon što učitamo naš skup podataka prethodno obrađen, proširujemo oblik polja slika (slika 5.4.) u  $(br\_uzoraka, 96, 96, 1)$  kako bi se uklopio u ulaz neuronske mreže.



Osim toga, kako bismo pomogli da mreža brže konvergira, trebali bismo izvršiti normalizaciju podataka. Postoje razne metode normalizacije podatka, no odabrat ćemo onu najjednostavniju. Kako se pikseli u monokromatskim slikama protežu u rasponu  $[0, 255]$ , a umjetne neuronske mreže najbolje rade sa skupom realnih brojeva, podijelit ćemo svaku vrijednost s 255 čime raspon brojeva smanjimo na  $[0, 1]$ .

## 5.2. Složeni model neuronske mreže

U ovome primjeru korištena je složenija arhitektura konvolucijske neuronske mreže. Mreža se sastoji od 12 slojeva. Unutar svih konvolucijskih i potpuno povezanih slojeva korištena je zglobnica kao aktivacijska funkcija. Ulazni sloj konvolucijske neuronske mreže je dimenzija  $96 \times 96$  s 32 filtra, no kako slike nisu tipa RGB nego su monokromatske, potrebna nam je samo jedna dodatna dimenzija za komponentu boje. Nakon toga slijede 3 skrivena konvolucijska sloja s rastućim brojem filtara (64, 128 i 256). Konvolucijski slojevi koriste filtre  $3 \times 3$  s korakom pomaka od 2 piksela. Funkcija ispadanja (engl. *dropout*) se koristi kako bi se spriječila prenaučenos modela. Funkcija ispadanja se također može tumačiti kao način reguliranja neuronske mreže dodavanjem šuma u skrivene jedinice. Ovakva je ideja prvobitno korištena u kontekstu (engl. *Denoising Autoencoders*) [8] gdje se šum dodaje na ulaze autoenkodera i izlaz se održava bez šuma.

Izlaz CNN je veličine 1 neurona, što nam je dovoljno s obzirom na to da se bavimo problemom binarne klasifikacije. Budući da je izlaz binaran koristili smo sigmoidu kao aktivacijsku funkciju izlaznog sloja. Pogledajmo sada programsku implementaciju ovih karakteristika na slici 5.5. Ovako definirana neuronska mreža sadrži 2 486 017 težina.

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=X_train.shape[1:]))
model.add(MaxPooling2D(2, 2))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))

model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))

model.add(Conv2D(256, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Flatten())
model.add(Dropout(0.3))
model.add(Dense(512, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

Slika 5.5: Implementacija složenoga modela

### 5.3. Jednostavni model neuronske mreže

Izvedimo sada neuronsku mrežu jednostavnije arhitekture sa sličnim karakteristikama kao kod prethodne. Neka se mreža sastoji od 8 slojeva. Ulazni sloj neka ostane istih dimenzija kao u prethodnoj mreži, te izbacimo 4 skrivena sloja. Također smanjimo dimenzionalnost prvoga sloja potpune povezanosti s 512 neurona na 64. Ostavimo aktivacijske funkcije kakve su bile, dakle zglobnica na svim slojevima osim izlaznome, gdje se koristi sigmoidalna funkcija. Ovako definirana neuronska mreža sadrži 1 000 929 težina. Na slici 5.6. je vidljiva programska implementacija modela.

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=X_train.shape[1:]))
model.add(MaxPooling2D(2, 2))

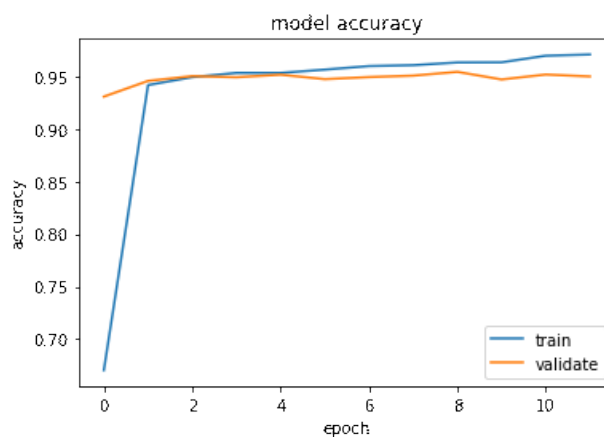
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))

model.add(Flatten())
model.add(Dropout(0.3))
model.add(Dense(64, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

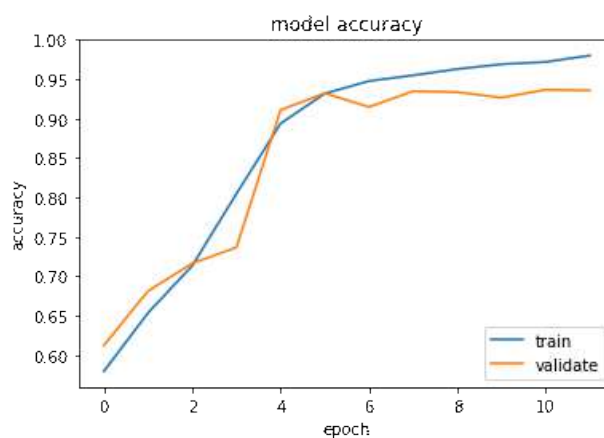
Slika 5.6: Implementacija jednostavnoga modela

## 6. Interpretacija rezultata učenja

Modeli su učeni na Windowsima 10 pokretanim procesorom Intel Core i7-8750H i CUDA 11.5/cuDNN 8.2.1 platformom potpomognutoj grafičkoj kartici Nvidia GTX 1060 6GB. Usporedimo sada rezultate učenja ovih dvaju neuronskih mreža. Jednostavni i složeni model su učeni na istome skupu podataka ranije navedenome, s jednakom podjelom skupa (slike 6.1. i 6.2.).



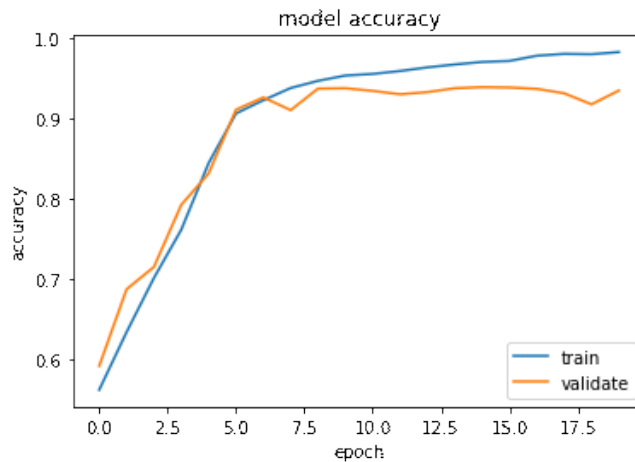
Slika 6.1: Točnost složenog modela



Slika 6.2: Točnost jednostavnog modela

Iz priloženih se slika može vidjeti da složeniji model brže uči. Osim što krivulja točnosti složenoga modela brže raste nego kod jednostavnog (točnost složenog modela se već na kraju druge epohe popne iznad 90%), također se vidi da svoju maksimalnu vrijednost postiže prije nego jednostavni (isto se može vidjeti da su oba modela s učenjem bila gotova u dvanaestoj epohi). Točnost složenoga modela na skupu za validaciju doseže 95%, a točnost jednostavnijeg nije ništa manja, s 93.57%. Nad oba je modela provedeno grupno (engl. *batch*) učenje. Veličina uzoraka nakon kojih se događa učenje postavljena je na 64.

U prethodne smo dvije slike mogli vidjeti da dvije različite arhitekture prekidaju učenje u različitim vremenskim trenucima. To se događa zbog implementacije zaustavljanja učenja (engl. *callback*) pri praćenju varijable *val\_loss*. Pogledajmo sada sliku 6.3.

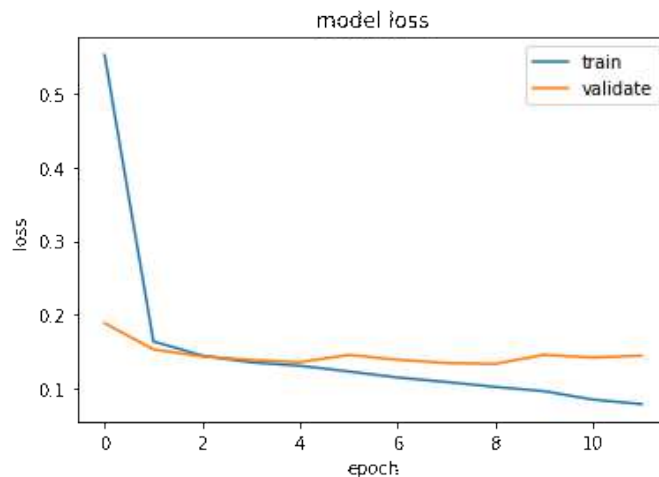


**Slika 6.3:** Točnost jednostavnog modela bez ranog zaustavljanja

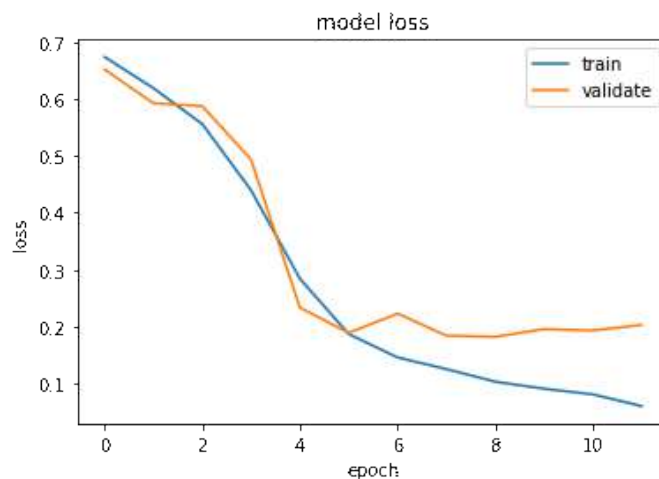
Slika prikazuje postupak učenja kroz 20 epoha. Vidi se da točnost na skupu za učenje raste skoro do 100%, no točnost predikcija na skupu za validaciju polako opada kako broj epoha teži ka 20. Konačna je vrijednost točnosti 92.7%. Ovdje se može primijetiti prenaučenosť podataka odnosno model postaje "štreber" i dobro klasificira primjere koje je samo do sada vidio.

## 6.1. Prikaz funkcijama gubitka

Kako radimo klasifikaciju razmaza krvi na zaražene i nezaražene malarijom, oba modela neuronske mreže koriste binarnu unakrsnu entropiju kao funkcija gubitka. Pogledajmo sada slike 6.4. i 6.5. Objašnjenje je slika dano u nastavku.



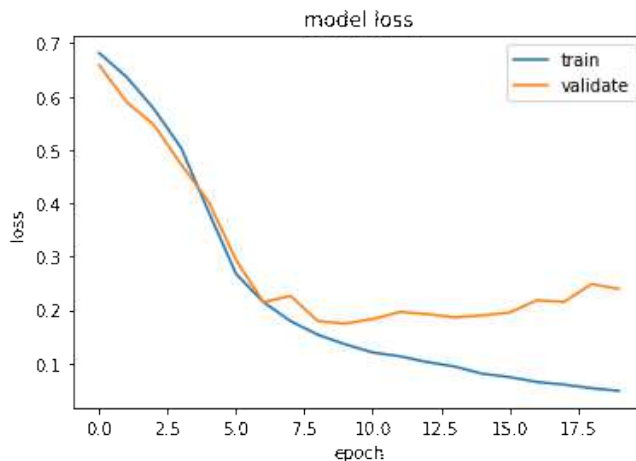
**Slika 6.4:** Greška složenog modela



**Slika 6.5:** Greška jednostavnog modela

Kao što se moglo pretpostaviti, složeni model ima strmiji grafički prikaz funkcije gubitka od jednostavnoga, što znači da gubitak uspijeva brže minimizirati kako na skupu za učenje tako i na skupu za validaciju. U konačnici složeni model postiže bolji rezultat u pogledu gubitka, kojemu je vrijednost gubitka 14.45% nasuprot jednostavnome modelu kojemu se nekoliko zadnjih epoha gubitak kreće oko 20% i u konačnici završi na 20.29%.

Vratimo se na prethodno provedeni eksperiment učenja jednostavnoga modela bez ranog zaustavljanja. Pogledamo li sliku 6.3. možemo vidjeti da se vrijednost parametra točnosti na skupu za validaciju s vremenom pogoršavala, dok se vrijednost na skupu za učenje poboljšavala. Zanima nas hoće li praćeni parametar pokazivati jednaka svojstva kao i točnost modela. Kretanje je vrijednosti *val\_loss* prikazano na slici 6.6.



**Slika 6.6:** Greška jednostavnog modela bez ranog zaustavljanja

Kao što se pretpostavlja, kako broj epoha teži ka 20 to se parametar  $val\_loss$  na skupu za validaciju povećava, dok u krajnjem slučaju poprimi vrijednost 23.92% (ne postiže dobre rezultate). S druge strane, vrijednost greške na skupu za učenje konstantno se smanjuje dok ne dosegne minimalnu vrijednost od 4.8% u dvadesetj epohi.

## 6.2. Prikaz konfuzijskom matricom

Konfuzijska matrica (engl. *confusion matrix*) je matrica dimenzije  $N \times N$  koja se koristi za ocjenjivanje izvedbe klasifikacijskog modela, gdje je  $N$  broj ciljnih klasa. Matrica uspoređuje stvarne ciljne vrijednosti s onima predviđenim modelom strojnog učenja.

**Tablica 6.1:** Konfuzijska matrica za binarnu klasifikaciju

	PREDICTED NEGATIVE	PREDICTED POSITIVE
ACTUAL NEGATIVE	a	b
ACTUAL POSITIVE	c	d

$a$  je broj točnih negativnih predviđanja,  $b$  netočnih pozitivnih predviđanja,  $c$  netočnih negativnih predviđanja, a  $d$  točnih pozitivnih predviđanja. Točnost predviđanja i pogreška klasifikacije mogu se dobiti iz ove matrice kako je zadano formulama 6.1 i 6.2 [11].

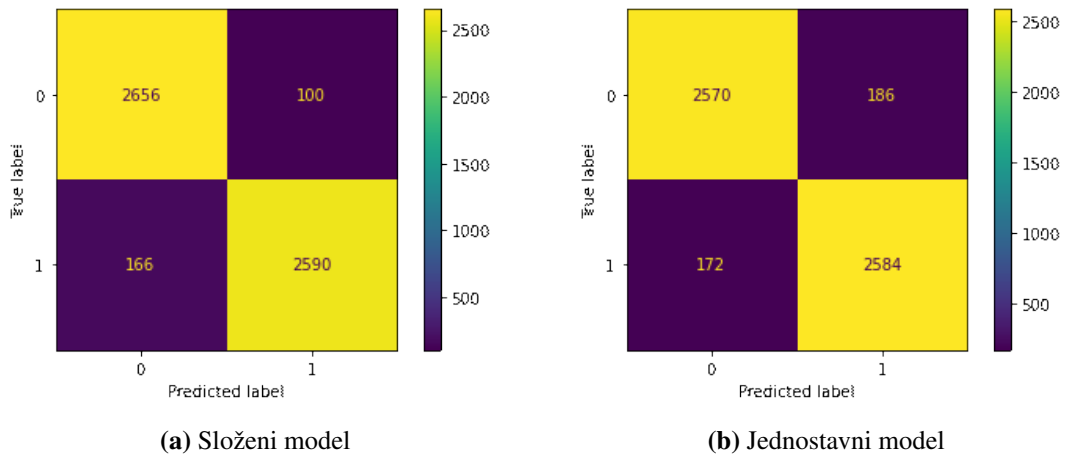
$$Accuracy = \frac{a + d}{a + b + c + d} \quad (6.1)$$

$$Error = \frac{b + c}{a + b + c + d} \quad (6.2)$$

Međutim pretpostavimo da klasifikator A ima veći odziv (engl. *recall*), a klasifikator B ima veću preciznost (engl. *precision*). Ovako definirano, točnost i pogreška ne bi davali valjanje rezultate u ocjenjivanju koji je klasifikator bolji. *F1-score* mjeri postotak točnih predviđanja koje je napravio model strojnog učenja. Vidjeli smo da je točnost loša metrika u slučaju neuravnoteženih podataka jer ne može razlikovati određene vrste pogrešaka (lažno pozitivne i lažno negativne). U ovom slučaju, *F1-score* za oba klasifikatora može se koristiti za određivanje koji daje bolje rezultate. *F1* računamo po formuli 6.3

$$F1 = \frac{2}{recall^{-1} + precision^{-1}}, \quad (6.3)$$

gdje se preciznost računa kao  $\frac{d}{b+d}$ , a odziv kao  $\frac{d}{c+d}$  [3]. Usporedimo sada konfuzijske matrice dvaju prethodno definiranih modela (slika 6.7.).



**Slika 6.7:** Prikaz konfuzijskih matrica

Uvrštavanjem zadanih vrijednosti iz slike 6.7. za složeni klasifikator dobivamo *F1* koji iznosi 95%, a za jednostavni klasifikator 93%. Iz ovih podataka se može zaključiti da složeniji klasifikator bolje klasificira primjere, te je pogodniji za implementaciju u sastav za detekciju malarije.

## 7. Zaključak

U prvome dijelu, uvodu, priložena je problematika kojom se bavi rad. Zatim su dani uvidi u umjetnu inteligenciju s naglaskom na područje strojnoga učenja, kao jednog od popularnijih područja istraživanja zadnjih nekoliko godina. U trećemu je poglavlju dan opis arhitekture umjetnih neuronskih mreža, prijenosnih funkcija i učenja samih mreža. Četvrto poglavlje naglasak stavlja na konvolucijske neuronske mreže zbog njihove primjene u području analize slika. U okviru konvolucijskih mreža navedene su i opisane razlike u odnosu na klasične umjetne neuronske mreže.

Peto se poglavlje osvrće na izradu sustava za detekciju malarije na temelju razmaza periferne krvi. U tome su dijelu priložene programske implementacije jednostavnijeg i složenijeg modela CNN. Sve su implementacije napisane u programskome jeziku Python uz korištenje vanjske biblioteke TensorFlow. Za potrebe učenja mreža skup podataka o zarazi malarijom preuzet je iz baze MEDLINE.

U šestome su poglavlju prikazani i međusobno uspoređeni rezultati učenja obaju modela. Vrednovanja obaju modela provedena su ispitivanjem točnosti, pogreške i prikazom konfuzijskom matricom. Iz konfuzijske je matrice izveden *F1-score*. U daljnjemu se radu preporučuje testiranje drugačije arhitekture konvolucijske mreže, aktivacijskih funkcija i funkcija pogreške.



# LITERATURA

- [1] Etheym Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2 izdanju, 2009.
- [2] Kristina Džomba. *Konvolucijske neuronske mreže*, 2018. URL <https://repositorij.pmf.unizg.hr/islandora/object/pmf\%3A5124/datastream/PDF/view>.
- [3] Tom Fawcett. *An introduction to ROC analysis*, 2006. URL <https://people.inf.elte.hu/kiss/11dwhdm/roc.pdf>.
- [4] Centers for Disease Control i Prevention. *Malaria Diagnostic Tests*, 2020. URL [https://www.cdc.gov/malaria/diagnosis\\_treatment/diagnostic\\_tools.html](https://www.cdc.gov/malaria/diagnosis_treatment/diagnostic_tools.html).
- [5] Donald O. Hebb. *Organization of Behavior*. Psychology Press, 1 izdanju, 1949.
- [6] J. Katarzyna i M. C. Wojciech. *On Loss Functions for Deep Neural Networks in Classification*, 2017. URL <https://arxiv.org/pdf/1702.05659.pdf>.
- [7] A. Newell, H. A. Simon, i C. Shaw. *Logic theorist*, 1956.
- [8] V. Pascal, L. Hugo, L. Isabelle, B. Yoshua, i M. Pierre-Antoine. *Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion*, 2010.
- [9] Frank Rosenblatt. *The Perceptron—a perceiving and recognizing automaton*. Cornell Aeronautical Laboratory, 1957.
- [10] Alan Mathison Turing. *Computing Machinery and Intelligence*, 1950.
- [11] S. Visa, B. Ramsay, A. Ralescu, i E van der Knaap. Confusion matrix-based feature selection. *Proceedings of the Twentysecond Midwest Artificial Intelligence and Cognitive Science Conference*, stranice 121–122, 2011.

- [12] World Health Organisation. *World malaria report 2021*, 2021. URL <https://www.who.int/publications/i/item/9789240040496>.
- [13] J. Šnajder, B. D. Bašić, i M. Čupić. *Umjetne neuronske mreže*. Fakultet elektrotehnike i računarstva, 2008.

## **Sustav za otkrivanje zaraze malarijom temeljen na slikama razmaza periferne krvi**

### **Sažetak**

U ovome je radu dan kratak uvod u područje umjetne inteligencije. Veći je naglasak stavljen na umjetne neuronske mreže, specifično na konvolucijske neuronske mreže. Uz rad su izrađena dva modela neuronskih mreža nad kojima je demonstrirano učenje. Za izradu sustava korišten je programski jezik Python. Na samome je kraju rada dana usporedba dvaju priloženih modela kroz testove točnosti, pogreške kao i prikaz konfuzijskom matricom.

**Ključne riječi:** Umjetna inteligencija, umjetne neuronske mreže, konvolucijske neuronske mreže, dijagnostika, klasifikacija slika.

## **Malaria infection detection system based on thin-blood smear images**

### **Abstract**

This paper gives a brief introduction to the field of artificial intelligence. Greater emphasis is placed on artificial neural networks, specifically on convolutional neural networks. Along with the work, two models of neural networks were developed, over which learning was demonstrated. The Python programming language was used to create the system. At the very end of the paper, a comparison of the two attached models is given through accuracy, error and confusion matrix tests.

**Keywords:** Artificial intelligence, artificial neural networks, convolutional neural networks, diagnostics, image classification.