

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 937

**KLASIFIKACIJA MOŽDANIH TUMORA METODAMA
DUBOKOG UČENJA NA TEMELJU SNIMKI MAGNETSKE
REZONANCIJE**

Katarina Šabić

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 937

**KLASIFIKACIJA MOŽDANIH TUMORA METODAMA
DUBOKOG UČENJA NA TEMELJU SNIMKI MAGNETSKE
REZONANCIJE**

Katarina Šabić

Zagreb, lipanj 2023.

ZAVRŠNI ZADATAK br. 937

Pristupnica: **Katarina Šabić (0036532311)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: izv. prof. dr. sc. Alan Jović

Zadatak: **Klasifikacija moždanih tumora metodama dubokog učenja na temelju snimki magnetske rezonancije**

Opis zadatka:

Rana detekcija moždanih tumora važna je za pravovremeno liječenje. Pri automatskom raspoznavanju vrsti tumora na temelju snimki magnetske rezonancije danas su sve značajnije metode dubokog učenja. U ovom završnom radu potrebno je opisati i primijeniti metode dubokog učenja, a posebice višeslojni perceptron i konvolucijsku neuronsku mrežu za izgradnju modela klasifikacije tumora na temelju snimki magnetske rezonancije. Pritom je potrebno koristiti prikladne knjižnice za izgradnju modela (npr. Keras, PyTorch). Za izgradnju i vrednovanje modela potrebno je preuzeti i pripremiti javno dostupni skup podataka Brain Tumor MRI Dataset (<https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>) u kojem se nalaze snimke tumora klasificiranih u tri klase: gliomi, meningeomi i tumori hipofize te snimke bez prisutnih tumora. U radu je potrebno prikazati rezultate modela na izdvojenom testnom podskupu podataka.

Rok za predaju rada: 9. lipnja 2023.

SADRŽAJ

1. Uvod	1
2. Opis medicinskog problema i skupa podataka	2
2.1. Građa mozga	2
2.2. Tumori na mozgu i njihova podjela	3
2.2.1. Primarni tumori	3
2.3. Opis skupa podataka	4
3. Duboke neuronske mreže	5
3.1. Duboko učenje i neuronske mreže	5
3.2. Problem klasifikacije	6
3.3. Konvolucijske neuronske mreže	7
3.3.1. Konvolucijski sloj	8
3.3.2. Sloj sažimanja	9
3.3.3. Potpuno povezani sloj	10
3.4. Primjena konvolucijskih neuronskih mreža u medicini	11
4. Izrada neuronske mreže	12
4.1. Obrada podataka	12
4.2. Izrada jednostavnijeg modela	15
4.3. Izrada složenijeg modela	17
4.4. Evaluacija rezultata	17
4.4.1. Učenje modela	17
4.4.2. Funkcija preciznosti i funkcija gubitka	18
4.4.3. Matrica konfuzije	20
5. Zaključak	24
Literatura	25

1. Uvod

Tumore centralnog živčanog sustava čine oboljenja mozga i leđne moždine koji su posljedica nakupljanja i prekomjernog nekontroliranog širenja abnormalnih stanica. Tumor na mozgu čini abnormalno tkivo, odnosno izraslina koja može nastati u bilo kojoj životnoj dobi, te mogu biti vrlo opasni po život. Procjenjuje se da je 2020. godine tumor na mozgu dijagnosticiran kod nešto više od 300 000 osoba diljem svijeta [5].

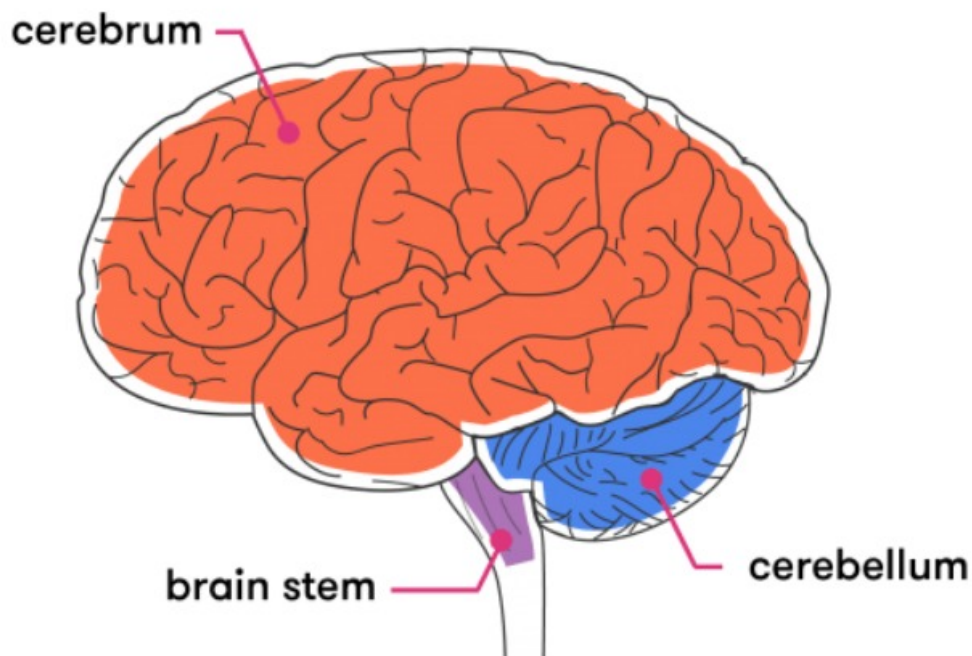
Tema ovog završnog rada je primjena algoritama strojnog učenja pri izradi modela pomoću kojeg je moguće klasificirati snimke magnetske resonance mozga u kategorije ovisno o vrsti tumora. Strojno učenje poddisciplina je umjetne inteligencije i računar-ske znanosti čiji je cilj izgraditi algoritam koji uporabom podataka pokušava oponašati način na koji ljudi uče i pritom maksimizirati njegovu točnost. U ovom radu je pri izradi modela korišteno duboko učenje, koje je jedna od tehnika strojnog učenja čiji se algoritmi zasnivaju na izgradnji i uporabi dubokih neuronskih mreža. Duboke neuronske mreže pokušavaju imitirati ponašanje ljudskog mozga, te tako što točnije prepoznavati, klasificirati i opisivati objekte unutar podataka. Podaci koje neuronske mreže koriste najčešće su slike, audio zapisi ili tekstni zapisi.

Struktura rada sastoji se od pet poglavlja nakon kojih slijedi popis literature. U drugom poglavlju proučavaju se moždana oboljenja, anatomija mozga, opis i podjela tumora, te je opisan skup podataka koji je korišten u svrhu izrade modela neuronske mreže. Treće poglavlje opisuje duboke neuronske mreže te njihovu primjenu u medicini. U četvrtom poglavlju opisuje se izrada i učenje modela za klasifikaciju moždanih tumora, te se iznose dobiveni rezultati u obliku grafova funkcija preciznosti i gubitka i matrice konfuzije. U posljednjem poglavlju rada iznosi se zaključak.

2. Opis medicinskog problema i skupa podataka

2.1. Građa mozga

Mozak je uz leđnu moždinu središnji organ centralnog živčanog sustava. On kontrolira mnoge funkcionalnosti ljudskog tijela kao što su razmišljanje, motoričke sposobnosti, pamćenje, emocije, govor i mnoge druge procese koji se odvijaju u našem tijelu. Drugim riječima, odgovoran je za sve karakteristike i ponašanja ljudskog organizma. Njegovi glavni dijelovi su veliki mozak, mali mozak i moždano deblo [10], slika 2.1.



Slika 2.1: Anatomija mozga, preuzeto iz [8]

Dva tkiva koja grade mozak su siva i bijela tvar. Siva tvar nalazi se na površini mozga i sastoji se uglavnom od neurona, dok se bijela tvar nalazi u njegovoj unutraš-

njosti i nju čine aksoni. Uloge tih dvaju tkiva su različite: siva tvar je odgovorna za obradu i interpretaciju informacija dobivenih podražajima, dok bijela tvar ima zadatak te informacije prenijeti drugim dijelovima živčanog sustava [10].

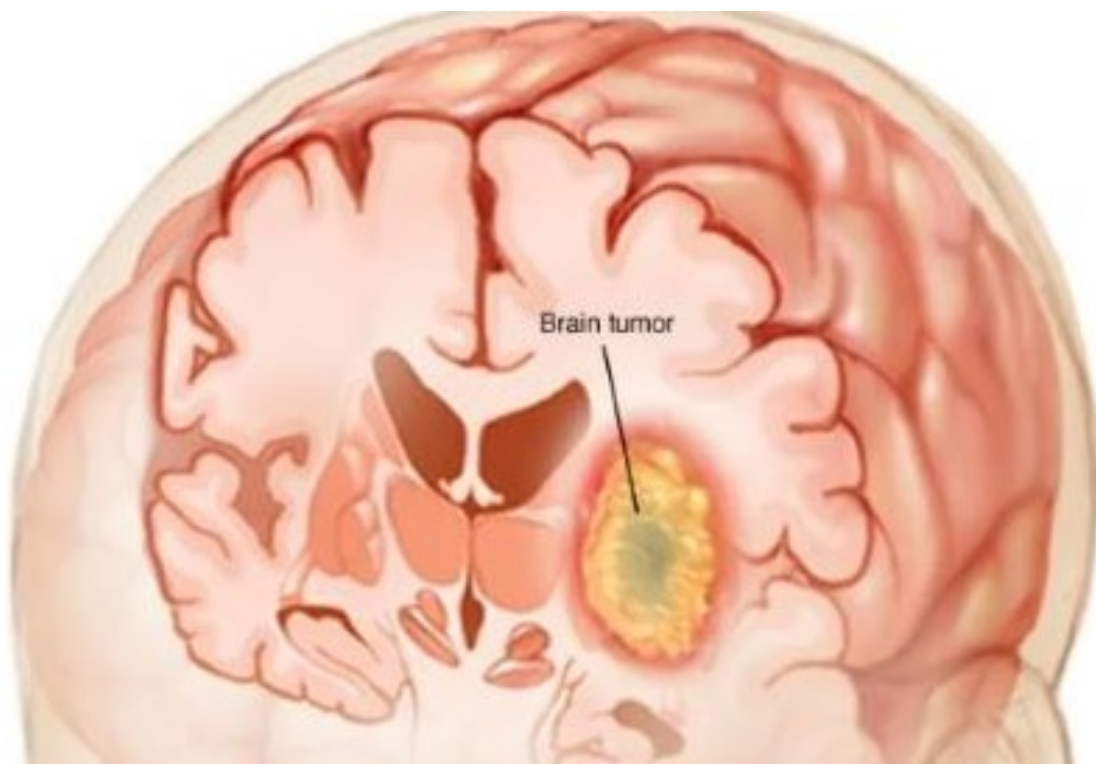
2.2. Tumori na mozgu i njihova podjela

Tumori na mozgu nastaju kao posljedica prekomjernog nastajanja i širenja abnormalnih stanica, odnosno abnormalnog tkiva. Mogu biti nekancerogeni (benigni tumori) ili kancerogeni (maligni tumori). Općenita podjela tumora na mozgu prema mjestu nastanka je na primarne i sekundarne. Primarni tumori nastaju u samom mozgu, dok sekundarni nastaju kao posljedica metastaze tumora s nekog drugog organa na mozak [13].

2.2.1. Primarni tumori

Rast stanica primarnog tumora započinje u samom mozgu ili u obližnjem tkivu. Glavna podjela primarnih tumora je na gliomske i negliomske tumore. Gliomi su jedna od najčešćih vrsta tumora mozga, i većinom su maligni. Smatra se da nastaju iz vrste stanica u mozgu nazvanih glija stanice. Dodjeljuje im se stupanj koji služi kao pokazatelj koliko je tumor agresivan, te koliko brzo raste.

Negliomske tumore čine meningeom, tumori pinealne žlijezde i hipofize, meduloblastom, kraniofaringioma, Schwannoma i primarni limfom srednjeg živčanog sustava [7].



Slika 2.2: Gliomski tumor, preuzeto iz [6]

2.3. Opis skupa podataka

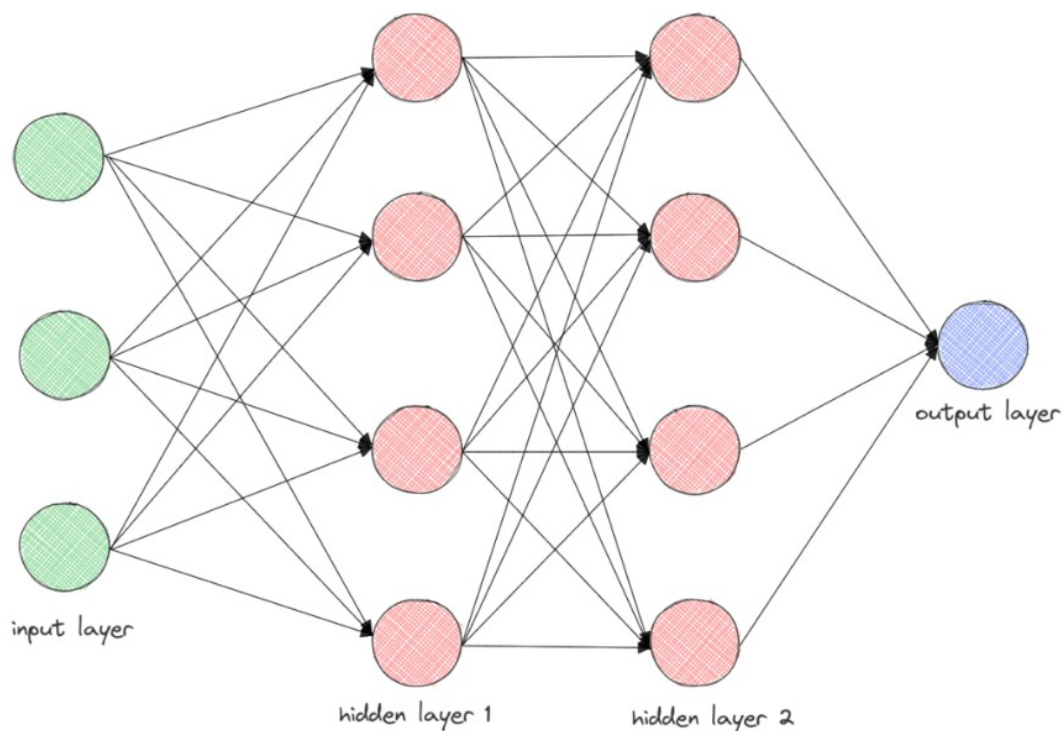
Skup podataka korišten za klasifikaciju tumora čini snimke magnetske rezonance koje se smatraju najpreciznijim prikazom patološki promjenjivog tkiva, te se koriste za otkrivanje i prevenciju različitih oboljenja. Preuzet je sa stranice Kaggle, a sačinjavaju ga 3 skupa podataka: figshare, SARTAJ dataset i Br35H. Sastoji se od ukupno 7023 snimke magnetnskih rezonanci podjeljenjih u četiri klase: snimke na kojima je prikaz glioma, snimke na kojima je prikaz meningioma, snimke zdravog mozga i snimke mozga na kojima je prikaz tumora hipofize. Skup podataka moguće je preuzeti u zip formatu s poveznice: <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>.

3. Duboke neuronske mreže

3.1. Duboko učenje i neuronske mreže

Duboko učenje grana je strojnog učenja koja se temelji na algoritmima čiji je cilj donositi odluke oponašajući ljudski mozak korištenjem dostupnih podataka i posebne višeslojne logičke strukture poznatije pod nazivom neuronska mreža. Neuronske mreže građene su od međusobno povezanih čvorova odnosno umjetnih neurona, koji oponašaju ponašanje neurona koji grade ljudski mozak. Drugim riječima, u kontekstu neuronske mreže, neuron je najosnovnija jedinica obrade podataka jer prima podatke, obrađuje ih i potom dobiveni rezultat prosljeđuje idućem sloju neurona [3]. Slojevi neurona međusobno su povezani i svakoj vezi pridjeljena je određena težina, koja je mjera značajnosti veze između neurona pojedinih slojeva mreže.

Neuronske mreže mogu učiti iz podataka i s vremenom poboljšavati svoju točnost, te se tako osposobiti za prepoznavanje uzoraka, klasifikaciju podataka i predviđanje budućih događaja [12]. Svaki idući sloj mreže nadograđuje se na prethodni u svrhu poboljšanja i optimizacije predviđanja odnosno kategorizacije. Ulazni sloj mreže je sloj koji dobiva podatke za daljnju obradu dok se u izlaznom sloju vrši konačno predviđanje ili klasifikacija. Oba sloja spadaju u vidljive slojeve mreže [11]. Ostali slojevi koji se nalaze između ulaznog i izlaznog sloja nazivaju se skriveni slojevi, te je njihova uloga transformacija podataka primjenom složenih nelinearnih funkcija. Neke od nelinearnih funkcija su funkcija sigmoida koja skalira realnu vrijednost broja na vrijednosti iz intervala 0 i 1, tangens hiperbolni (tanh) koja skalira realnu vrijednost broja na vrijednosti iz intervala -1 i 1, te zglobnica ReLu koja ukoliko joj je poslana pozitivna vrijednost vraća upravo tu vrijednost [16]. Skriveni slojevi nužna su komponenta neuronske mreže kako bi se omogućilo provođenje složenijih funkcija nad skupom podataka i postiglo bolje performanse [1]. Na slici 3.1 prikazana je arhitektura neuronske mreže u kojoj su vidljivi jedan ulazni, jedan izlazni i više skrivenih slojeva.



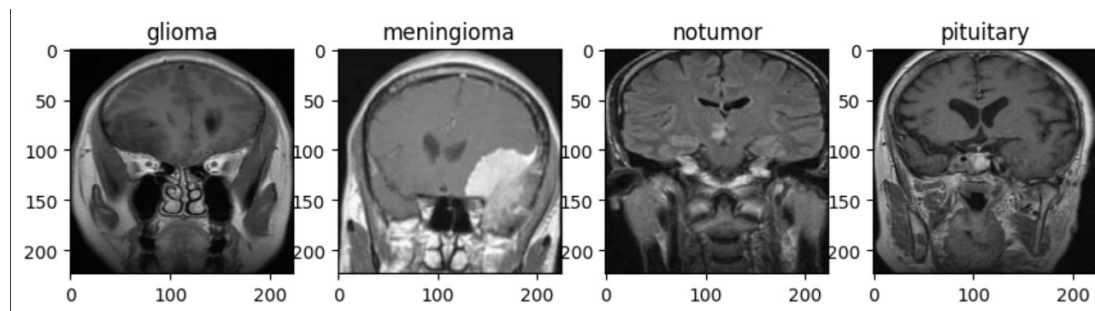
Slika 3.1: Prikaz arhitekture mreže, preuzeto iz [1]

Duboke neuronske mreže uče na temelju danih podataka koji su podijeljeni u nekoliko skupova: skup za učenje, skup za validaciju i skup za testiranje. Skup za učenje je ujedno i najveći od tri navedena skupa, te se koristi pri treniranju modela. Model vidi i uči iz tih podataka [18]. Skup za validaciju je skup koji neizravno utječe na model nego utječe na hiperparametre modela kao što su broj skrivenih slojeva modela, veličina sloja, tip aktivacijske funkcije, itd. Naposljetku, skup za testiranje sadrži podatke koje model do tad nije vidio, a koristi se za procjenu točnosti modela. Svrha razdvajanja podataka u tri navedena skupa je osiguravanje svojstva generalizacije i istovremeno sprječavanje prenaučnosti modela. Svojstvo generalizacije podrazumjeva osiguravanje velike točnosti modela na dotad neviđenim podacima, odnosno na skupu za testiranje. Do prenaučnosti modela dolazi zbog pretjerane prilagodbe modela na podatke iz skupa za učenje što za posljedicu ima lošu generalizaciju čime se poništava svrha modela.

3.2. Problem klasifikacije

Klasifikacija je definirana kao problem prepoznavanja, pridjeljivanja odgovarajuće oznake objektima i određivanje pripadnosti pojedinoj kategoriji. Klasifikacijski algo-

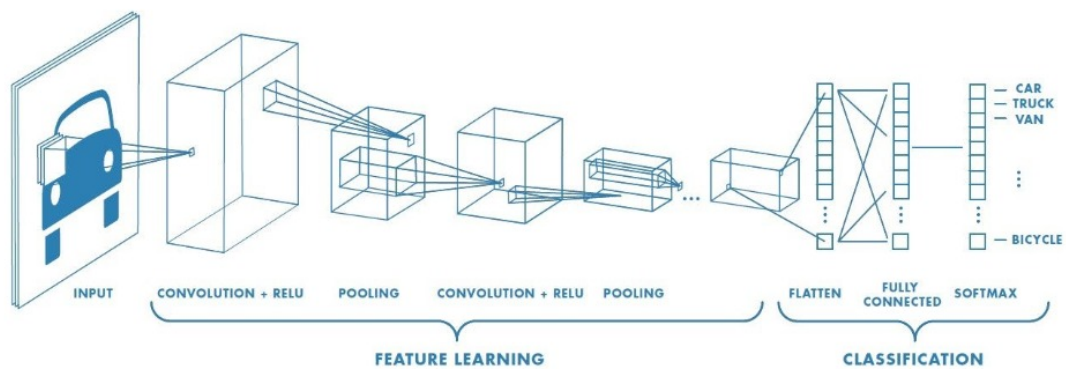
ritmi koji se koriste u strojnom, odnosno dubokom učenju koriste ulazne podatke u svrhu predviđanja pripadnosti nekoj od unaprijed određenih kategorija [2]. Postoji nekoliko vrsta klasifikacijskih problema: binarna klasifikacija koja daje jedan izlaz i svrstava podatke u točno dvije klase, klasifikacija s više izlaza (engl. *multi-label classification*) gdje se svakom objektu može dodijeliti više neekskluzivnih oznaka [22] i višeklasna klasifikacija (engl. *multiclass classification*) koja sadrži više od dvije klase i svaki objekt smješta u njegovu pripadnu klasu. Za rješavanje klasifikacijskih problema koriste se različiti algoritmi strojnog učenja koji uče kako dodijeliti oznaku klase primjerima iz domene problema odnosno danog skupa podataka [4]. Problem klasifikacije snimki magnetskih rezonanci kojim se bavim u ovom radu spada u višeklasnu klasifikaciju jer svaka snimka spada u jednu od kategorija odnosno klasa: *glioma*, *meningioma*, *notumor* i *pituitary*. Jedan od pogodnih algoritama za učinkovitu obradu, povezivanje i razumijevanje velike količine podataka u slikama su konvolucijske neuronske mreže. Na slici 3.2 je prikaz četiri klase koje su mogući izlazi iz sustava.



Slika 3.2: Prikaz mogućih izlaza iz sustava

3.3. Konvolucijske neuronske mreže

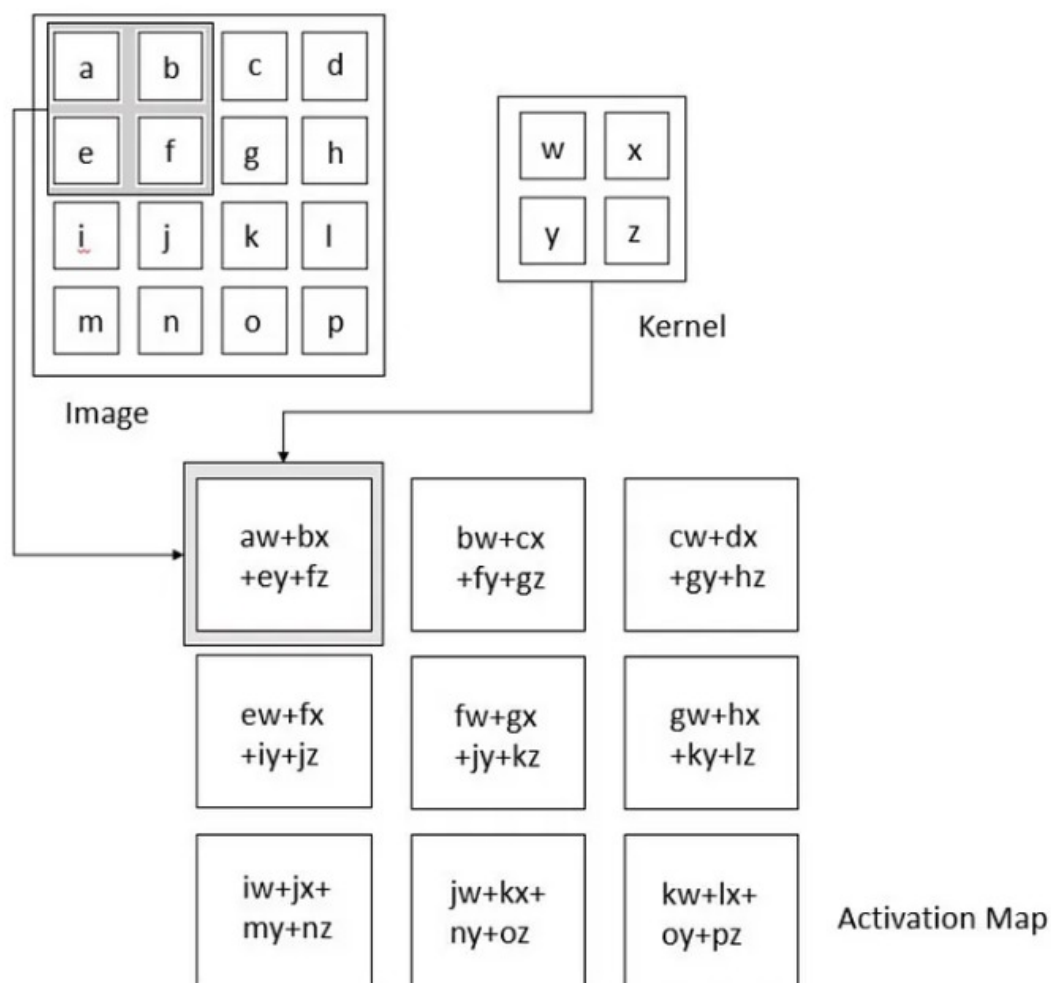
Konvolucijska neuronska mreža (engl. *convolutional neural network*, skraćeno *CNN*) je vrsta neuronske mreže koja je pogodna za analizu i obradu podataka kao što su slike, zvuk i tekst jer pomoću konvolucijskih struktura može postupno izvući značajke iz danih podataka [15]. Slika koju mreža dobiva kao ulaz sadrži niz piksela raspoređenih u obliku mreže koja sadrži vrijednosti piksela koje označavaju koliko bi svaki piksel trebao biti svijetao i koje boje. CNN je izgrađena od više slojeva od kojih su glavni konvolucijski sloj, sloj sažimanja i potpuno povezani sloj. Slojevi su raspoređeni na takav način da najprije otkrivaju jednostavnije uzorke (linije, krivulje, itd.), a kasnije složenije uzorke (objekte), slika 3.3 [16].



Slika 3.3: Prikaz slojeva u konvolucijskoj neuronskoj mreži, preuzeto iz [17]

3.3.1. Konvolucijski sloj

Konvolucijski sloj je temeljni element svake konvolucijske neuronske mreže. Zadaća ovog sloja je izdvajanje raznolikih značajki iz slika koje su dane kao ulaz u mrežu. On provodi množenje matrica pri čemu je jedan faktor filtar (engl. *kernel*) odnosno skup parametara koji se mogu naučiti određene $N \times N$ veličine, a drugi faktor je dio ulazne slike početne veličine $M \times M$ odnosno receptivnog područja. Tijekom učenja filtar prolazi određenim kliznim pomakom filtra (engl. *stride*) koji je zapravo broj piksela za koji se filtar pomiče po visini i širini slike i tako stvara slikovnu reprezentaciju tog receptivnog područja [16]. Rezultat prolaska je izdvajanje značajki iz slike u mape značajki (engl. *feature maps* ili *activation maps*) koja sadrži informacije o slici kao što su na primjer rubovi i kutovi slike. Većim brojem prolazaka filtra po receptivnom polju odnosno slici stvara se više mapi značajki koje osim osnovnih informacija o slici sadrže i neke složenije informacije, primjerice prostorne i vremenske ovisnosti u slici [17].



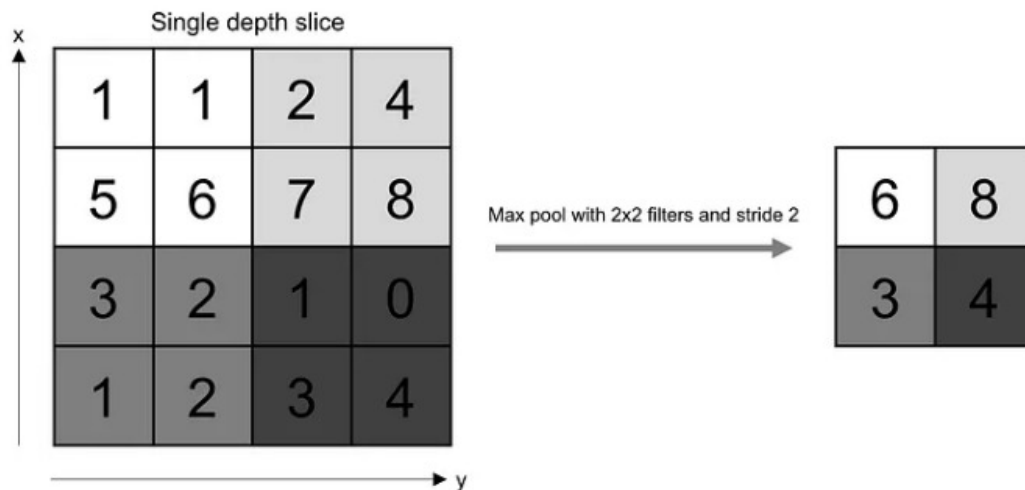
Slika 3.4: Stvaranje aktivacijske mape (mape značajki), preuzeto iz [16]

Na slici 3.4 prikazano je već spomenuto množenje matrica, kao što je vidljivo slika je veće veličine od filtra. Množenje se odvija tako da se filter veličine 2×2 pomiče po slici s lijeva na desno i obavlja operaciju množenja sa dijelom slike na kojem se trenutno nalazi dok ne obiđe cijelu veličinu slike. Veličina mape značajki određena je veličinama filtra i ulazne slike, te se izračunava formulom $(m - f + 1) \times (m - f + 1)$.

3.3.2. Sloj sažimanja

Sloj sažimanja dolazi nakon pojedinog konvolucijskog sloja, a odgovoran je za smanjivanje prostora veličine kovulacijske značajke (mape značajki) što za posljedicu ima smanjenje potrebne količine daljnjih izračuna zbog smanjenog broja parametara. Sažimanje se provodi nad svakom mapom značajki. Cilj je početnu mapu značajki podijeliti na više dijelova koji su dimenzije filtra, te iz svakog dijela, primjenom nekom

od funkcija sažimanja, uzeti najznačajniji njegov dio i tako smanjiti dimenzije početne mape značajki. Postoji nekoliko funkcija sažimanja a najčešće korištene su funkcija maksimalnog sažimanja i funkcija prosječnog sažimanja. Funkcija maksimalnog udruživanja uzima najveću vrijednost iz svakog dijela mape značajki, dok funkcija prosječnog udruživanja uzima prosječnu vrijednost svih elemenata dijela slike koje pokriva filter [17]. Na slici 3.5 je dan primjer funkcije maksimalnog sažimanja dobivene mape značajki.

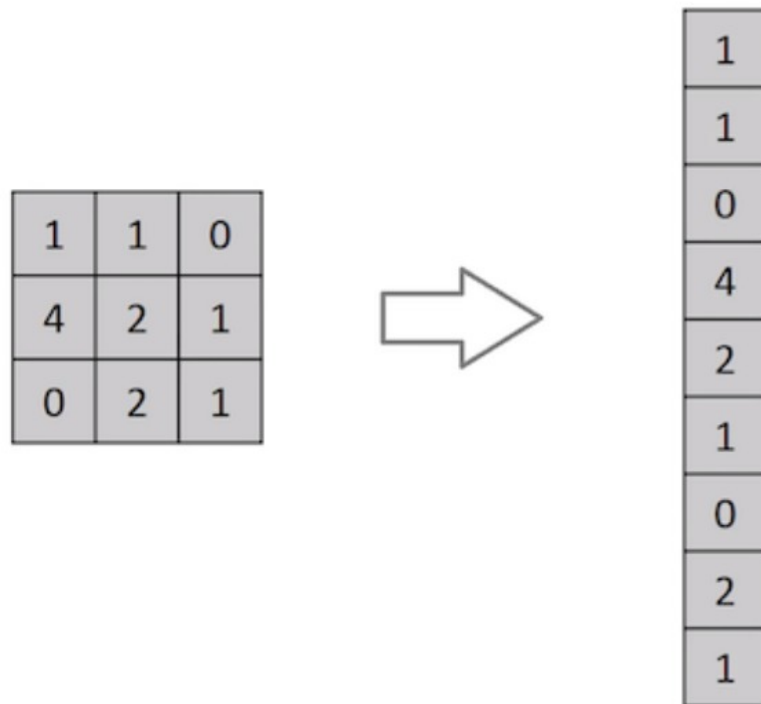


Slika 3.5: Sloj sažimanja i funkcija maksimalnog udruživanja, preuzeto iz [16]

Sloj sažimanja maksimalnom vrijednošću osim smanjenja dimenzionalnosti također ima još neke pogodnosti za neuronsku mrežu: on uklanja šum među podacima u mapi značajki.

3.3.3. Potpuno povezani sloj

Potpuno povezani slojevi konvolucijske neuronske mreže dolaze na samom kraju, a karakterizira ih potpuna povezanost neurona sa svim drugim neuronima u prethodnom i sljedećem sloju. Kao ulaz dobivaju stupac-vektor koji nastaje dodavanjem sloja za izravnavanje (engl. *Flatten layer*) između sloja koji prethodni potpuno povezanom sloju i samog potpuno povezanog sloja, koji izlaznu matricu pretvara u jednodimenzionalni stupac-vektor. Slika 3.6 daje prikaz sloja izravnavanja.



Slika 3.6: Sloj izravnavanja, preuzeto iz [16]

3.4. Primjena konvolucijskih neuronskih mreža u medicini

Konvolucijske neuronske mreže svoju primjenu su pronašle u mnogim ljudskim djelatnostima pa tako i u medicinskim analizama slike, odnosno u medicinskoj dijagnostici. Koriste se za klasifikaciju slika, segmentaciju, detekciju i druge zadatke kao što su registracija, dohvaćanje slika temeljeno na sadržaju, generiranje i poboljšanje slika, u nekim tipičnim područjima medicinske dijagnoze kao što su primjerice otkrivanje tumora na mozgu, dojci, koži, ili nekom drugom organu. Medicinske slike mogu biti veoma složene, te je liječnicima i radiolozima katkad teško dati učinkovitu i točnu dijagnozu, te im stoga pomažu računalne tehnologije koje provode analizu slika tehnikama dubokog učenja i pomažu donijeti ispravnu dijagnozu.

4. Izrada neuronske mreže

Cilj ovog rada je bio izgraditi učinkoviti model duboke neuronske mreže koji će raditi ispravnu klasifikaciju snimki magnetske rezonance. Za potrebe izgradnje modela korišteni su programski jezik Python zajedno sa bibliotekama Keras, NumPy i Seaborn za obradu podataka i izgradnju modela. Biblioteka Keras je aplikacijsko programsko sučelje (engl. *Application programming interface* ili *API*) razvijeno od strane tvrtke Google koje se koristi za implementaciju neuronskih mreža. Napisano je u Pythonu i koristi se za jednostavnu implementaciju neuronskih mreža [19]. Osim Kerasa, za potrebe implementacije neuronske mreže korištene su i dvije biblioteke za obradu podataka: NumPy koja ubrzava rad sa velikim višedimenzionalnim poljima ili matricama što ubrzava rad s velikom količinom podataka, i Seaborn, čiji je zadatak vizualizacija podataka i crtanje grafova.

4.1. Obrada podataka

Prvi korak pri izgradnji konvolucijske neuronske mreže jest prilagoditi podatke. Preuzeti podaci odnosno slike nisu iste veličine, te je ih je potrebno preobraditi. Nakon preuzimanja podataka, potrebno je ukloniti nepotrebne praznine u slikama, promijeniti im veličinu i postaviti ju na jedinstvenu vrijednost 224×224 piksela. Na slici 4.1 je prikazana funkcija koja smanjuje šum u slici, zaglađuje konture, pretvara piksele u vrijednosti od 0 do 255 i traži koordinate graničnog pravokutnika na temelju najveće konture kako bi se izvorna slika obrezala.


```

def crop_image(img):
    gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
    gray = cv2.GaussianBlur(gray, (3, 3), 0)
    thresh = cv2.threshold(gray, 45, 255, cv2.THRESH_BINARY)[1]
    thresh = cv2.erode(thresh, None, iterations=2)
    thresh = cv2.dilate(thresh, None, iterations=2)
    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    contour_areas = [cv2.contourArea(cnt) for cnt in contours]
    max_contour = contours[contour_areas.index(max(contour_areas))]
    left, top, right, bottom = cv2.boundingRect(max_contour)
    padding = 0
    cropped_img = img[top-padding:top+bottom+padding, left-padding:left+right+padding].copy()
    return cropped_img

```

Slika 4.1: Funkcija za preobradu slika

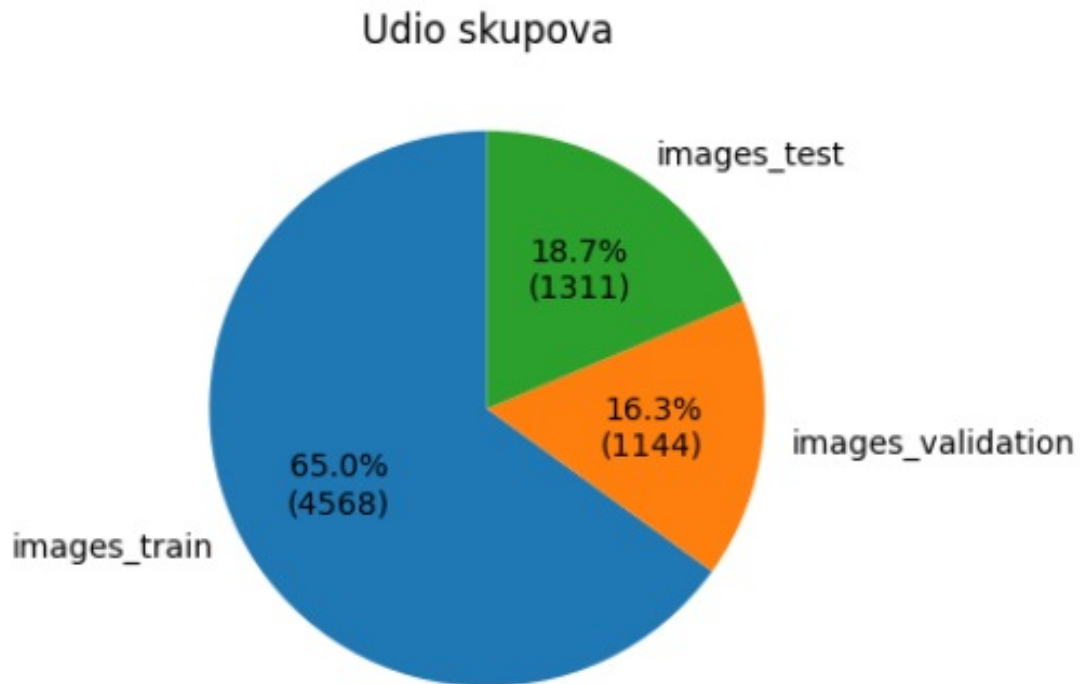
Podatke je potrebno podijeliti u 3 skupa opisana u poglavlju 3.1: skup za učenje, skup za validaciju i skup za testiranje. Preuzeti skup podataka već je podijeljen na skup za učenje koji sadrži 5712 slike i skup za testiranje koji sadrži 1311 sliku. Budući da preuzeta zip datoteka nema odvojen skup za validaciju, potrebno je iz skupa za testiranje izdvojiti 20% slika i stvoriti skup za validaciju pomoću funkcije `train_test_split` iz `scikit-learn` biblioteke. Na slici 4.2 dan je isječak koda kojim su iz skupa za učenje izdvojeni podaci potrebni za stvaranje skupa za validaciju, a histogram na slici 4.3 prikazuje udio svakog od 3 skupa.

```

train_files, validation_files = train_test_split(image_files, test_size=0.2, random_state=123)

```

Slika 4.2: Stvaranje skupa za validaciju



Slika 4.3: Prikaz udjela skupova podataka

Potrebno je za svaki od direktorija *Training*, *Validation* i *Testing* koji sadrže podatke iz istoimenog skupa, stvoriti listu koja će sadržavati slike iz pripadnog skupa i također listu s odgovarajućim labelama za svaki skup slika. Liste sa slikama nazvane su `images_train`, `images_validation` i `images_test`, dok su liste koje sadrže labele nazvane `label_train`, `label_validation` i `label_test`. Nakon stvaranja i punjenja lista slikama i labelama, potrebno je primjeniti funkciju `shuffle` koja sprječava potencijalne pristranosti i obrase koji mogu postojati u redosljedu podataka. Miješanjem podataka se, osim sprječavanja pristranosti, postiže i poboljšanje izvedbe modela upravo zbog uklanjanja mogućih uzoraka među podacima. Liste sa slikama i labelama koje pripadaju istom skupu potrebno je miješati skupa kako bi se održala korespondencija između slika i odgovarajućih oznaka. Zatim je potrebno sve liste pretvoriti u NumPy nizove pomoću funkcije `numpy.array`, jer framework Keras očekuje da ulazni podaci budu u formatu NumPy niza.

```

images_train, label_train = shuffle(images_train, label_train, random_state=123)
images_validation, label_validation = shuffle(images_validation, label_validation, random_state=123)
images_test, label_test = shuffle(images_test, label_test, random_state=123)

images_train = np.array(images_train)
images_validation = np.array(images_validation)
images_test = np.array(images_test)
label_train = np.array(label_train)
label_validation = np.array(label_validation)
label_test = np.array(label_test)

```

Slika 4.4: Miješanje podataka i pretvorba u NumPy objekt

Naposlijetku je potrebno napraviti normalizaciju slika i oznaka. Podaci o slici predstavljeni su vrijednostima piksela u rasponu od 0 do 255. Dijeljenjem podataka slike sa 255.0 vrijednost piksela se skalira na vrijednost između 0 i 1. Time se osigurava da sve ulazne značajke odnosno sve vrijednosti piksela imaju sličan raspon, te se sprječava dominacija određenih značajki procesom učenja. Binarizacija oznaka podrazumjeva pretvorbu oznaka iz kategoričkog oblika u numerički. Pretvorba kategoričkih oznaka u binarni radi se uz pomoć klase `LabelBinarizer` iz `scikit-learn` biblioteke, a postupak se naziva binarizacija oznaka. Svaka se oznaka pretvara u binarni niz duljine jednake broju klasa, a vrijednost 1 je dodijeljena odgovarajućoj klasi, dok su ostali elementi u nizu postavljeni na 0. Na ovaj način se osigurava da su ulazni podaci u odgovarajućem formatu i rasponu za učinkovito učenje modela neurinske mreže.

4.2. Izrada jednostavnijeg modela

Prvi primjer prikazuje jednostavniji model konvolucijske neuronske mreže. Pri izgradnji mreže korišten je `Keras Sequential` model koji je namijenjen gradnji mreže sloj po sloj, te podaci teku od jednog sloja do drugog zadanim redoslijedom dok konačno ne dođu do izlaznog sloja [21]. Mreža sadrži 10 slojeva. Početni sloj mreže je konvolucijski sloj a njegovi parametri su broj filtara koji je postavljen na 16, od kojih svaki otkriva različite značajke na ulaznoj slici, zatim veličina filtra koja je postavljena na 3×3 , aktivacijska funkcija `ReLU` koja je korištena kroz cijelu mrežu unutar svih slojeva, te dodatno prvi sloj ima parametar kojim se definira veličina ulaza u model. Ulazi u ovaj model veličine su $224 \times 224 \times 3$. Visina i širina ulazne slike su veličine 224×224 piksela, a brojka 3 označava da ulazne slike imaju 3 kanala u boji koji predstavljaju RGB vrijednosti (engl. *red green blue*), to jest za svaki piksel u slici postojat će tri vrijednosti koje odgovaraju intenzitetu kanala crvene, zelene i plave boje. Nakon svakog konvolucijskog sloja dolazi sloj sažimanja čija je uloga upisana u poglavlju

3.3.2. Broj filtra se u zadnjem konvolucijskom sloju povećao sa 16 na 64. Povećanje broja filtara u dubljim slojevima uobičajena je praksa za dobivanje apstraktnijih prikaza značajki. Nakon 3 konvolucijska i 3 sloja sažimanja dolazi sloj izravnavanja koji pretvara višedimenzionalne vektore u jednodimenzionalni vektor. Sloj ispadanja (engl. *dropout*) ima zadatac spriječiti prenaucenost modela tako što tijekom treniranja mreže nasumično odabire podskup ulaznih jedinica i postavlja njihove vrijednosti na nulu s određenom stopom ispadanja koja se šalje kao parametar sloju. Stopa ispadanja je najčešće vrijednosti između 0,2 i 0,5, a ona ukazuje na udio jedinica koje su postavljene na 0. Ispadanje sprječava da se mreža počne previše oslanjati na određene pojedinačne jedinice i tako poboljšava svojstvo generalizacije. Na kraju dolaze i dva potpuno povezana sloja (engl. *dense layers*) koji su temeljna komponenta neuronske mreže jer su odgovorni za učenje i mapiranje odnosa između ulaznih značajki i izlaznih oznaka. Prvi potpuno povezani sloj sadrži 64 neurona, dok drugi sadrži 4 neurona. Broj neurona u zadnjem potpuno povezanom sloju odgovara broju klasa u danom problemu klasifikacije. Funkcija aktivacije korištena u zadnjem sloju je softmax. Softmax funkcija na kraju daje kao rezultat distribuciju vjerojatnosti po klasama, normalizira izlaz modela tako da zbroj vjerojatnosti bude 1. To nam omogućuje tumačenje izlaznih vrijednosti kao vjerojatnost da ulaz pripada svakoj klasi. Ovaj jednostavniji model sastoji se od 2 781 268 parametara, odnosno težina. Slika 4.5 prikazuje slojeve modela i njegovu implementaciju.

```
model = Sequential()

model.add(Conv2D(16, (3, 3), activation='relu', input_shape=(224, 224, 3)))
model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(16, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))

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

model.add(Flatten())
model.add(Dropout(0.2))
model.add(Dense(64, activation='relu'))
model.add(Dense(4, activation='softmax'))
```

Slika 4.5: Implementacija jednostavnijeg modela

4.3. Izrada složenijeg modela

Za razliku od jednostavnijeg modela, složeniji model sastoji se od 16 slojeva, a sadrži dodatna 2 konvolucijska i 2 sloja sažimanja, te dodatni sloj ispadanja i potpuno povezani sloj. Osim promjene u arhitekturi mreže, to jest broju slojeva, promjena se učituje i u ostalim hiperparametrima mreže kao što su broj filtara u konvolucijskom sloju i broj neurona u potpuno povezanom sloju. Dodatni sloj ispadanja služi za prevenciju pretreniranja (prenaučenosti) modela. Složenije mreže povećavaju kapacitet modela, imaju veći kapacitet učenja uzoraka i odnosa unutar podataka, imaju poboljšanu izvedbu na složenijim zadacima poput prepoznavanja slika ili govora i mogu poboljšavati generalizaciju modela. S druge strane, složeniji model može dovesti do prenaučnosti mreže što je moguće spriječiti korištenjem nekih tehnika poput slojeva za ispadanje. Na slici je prikazana implementacija modela. Ovakav složeniji model sastoji se od 3 802 276 težina.

```
model = Sequential()

model.add(Conv2D(16, kernel_size=(3,3), activation='relu', input_shape=(224, 224, 3)))
model.add(MaxPooling2D(pool_size=(2,2)))

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

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

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

model.add(Conv2D(256, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(4, activation='softmax'))
```

Slika 4.6: Implementacija složenijeg modela

4.4. Evaluacija rezultata

4.4.1. Učenje modela

Prilikom izgradnje modela potrebno je definirati optimizator (engl. *optimiser*), funkciju gubitka (engl. *loss function*) i metriku (engl. *metrics*) to jest mjeru uspješnosti modela. Optimizator je algoritam koji poboljšava izvedbu modela i također utječe na

točnost i brzinu treniranja modela tako što prilagođava atribute neuronske mreže kao što su težine i stope učenja. Stopa učenja je hiperparametar modela koji kontrolira prilagođavanje to jest promjene parametara modela tijekom učenja. Optimizator korišten pri izgradnji oba modela je Adam koji održava stopu učenja za svaki parametar u mreži [9]. Funkcija gubitka je matematička funkcija koja mjeri pogrešku ili odstupanje između predviđenog izlaza mreže i pravih izlaznih oznaka skupa za učenje. Ona omogućuje praćenje rezultata modela tijekom učenja i daje uvid u točnost s kojom je model u mogućnosti klasificirati podatke. Za klasifikacijske probleme s više izlaza koristi se kategorična unakrsna entropija (engl. *categorical cross-entropy*). Mjera učinkovitosti modela je točnost odnosno preciznost na skupu podataka za validaciju.

```
optimizer = Adam(learning_rate=0.0001)
model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
```

Slika 4.7: Prikaz funkcije optimizacije, gubitka i preciznosti

Modeli su izgrađeni i trenirani u Jupyter Notebooku. Proces učenja svakog od modela sastoji se od 20 epoha, s veličinom poduzorka za učenje (engl. *batch size*) postavljenom na 32. Slika prikazuje treniranje složenijeg modela.

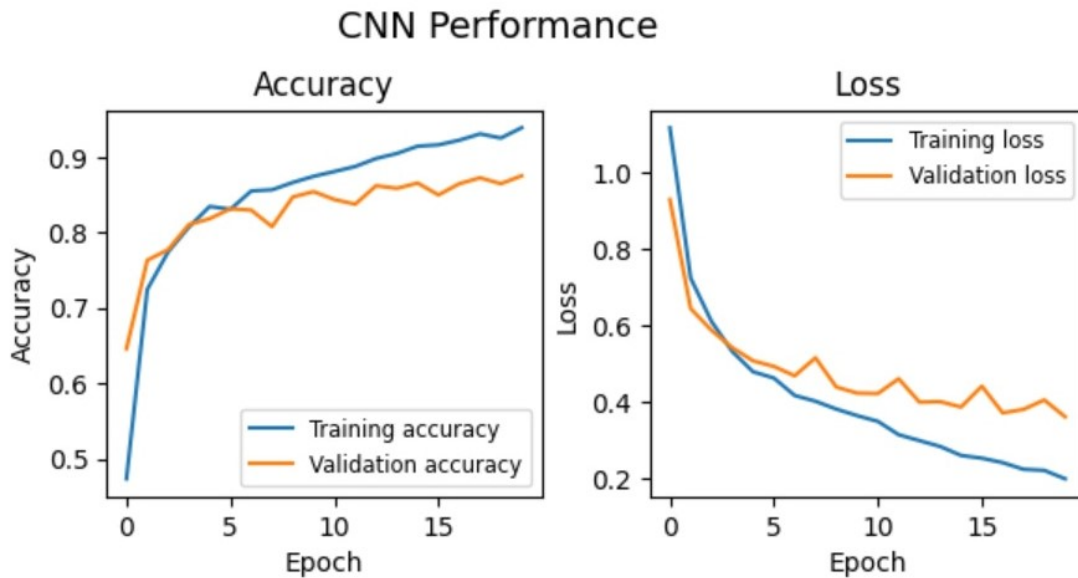
```
Epoch 7/20
142/142 [=====] - 122s 857ms/step - loss: 0.4030 - accuracy: 0.8516 - val_loss: 0.3775 - val_accuracy: 0.8619
Epoch 8/20
142/142 [=====] - 141s 995ms/step - loss: 0.3443 - accuracy: 0.8765 - val_loss: 0.3307 - val_accuracy: 0.8846
Epoch 9/20
142/142 [=====] - 111s 780ms/step - loss: 0.3225 - accuracy: 0.8871 - val_loss: 0.3153 - val_accuracy: 0.8916
Epoch 10/20
142/142 [=====] - 111s 779ms/step - loss: 0.2855 - accuracy: 0.8948 - val_loss: 0.3373 - val_accuracy: 0.8820
Epoch 11/20
142/142 [=====] - 112s 790ms/step - loss: 0.2397 - accuracy: 0.9189 - val_loss: 0.2888 - val_accuracy: 0.9021
Epoch 12/20
142/142 [=====] - 112s 787ms/step - loss: 0.2246 - accuracy: 0.9257 - val_loss: 0.4254 - val_accuracy: 0.8462
Epoch 13/20
142/142 [=====] - 110s 778ms/step - loss: 0.1844 - accuracy: 0.9407 - val_loss: 0.2510 - val_accuracy: 0.9248
Epoch 14/20
142/142 [=====] - 110s 778ms/step - loss: 0.1636 - accuracy: 0.9422 - val_loss: 0.2688 - val_accuracy: 0.9143
Epoch 15/20
142/142 [=====] - 115s 808ms/step - loss: 0.1553 - accuracy: 0.9442 - val_loss: 0.2663 - val_accuracy: 0.9056
Epoch 16/20
142/142 [=====] - 110s 773ms/step - loss: 0.1339 - accuracy: 0.9541 - val_loss: 0.3086 - val_accuracy: 0.8934
Epoch 17/20
142/142 [=====] - 110s 774ms/step - loss: 0.1089 - accuracy: 0.9641 - val_loss: 0.2477 - val_accuracy: 0.9301
Epoch 18/20
142/142 [=====] - 110s 776ms/step - loss: 0.1019 - accuracy: 0.9647 - val_loss: 0.2313 - val_accuracy: 0.9371
Epoch 19/20
142/142 [=====] - 109s 770ms/step - loss: 0.0851 - accuracy: 0.9698 - val_loss: 0.2311 - val_accuracy: 0.9292
Epoch 20/20
142/142 [=====] - 113s 799ms/step - loss: 0.0740 - accuracy: 0.9746 - val_loss: 0.2432 - val_accuracy: 0.9327
```

Slika 4.8: učenje složenijeg modela

4.4.2. Funkcija preciznosti i funkcija gubitka

Funkcija preciznosti daje nam uvid u točnost modela nakon treniranja. Jednostavniji model postiže točnost od 98,31% na podacima iz skupa za učenje, 88,46% na podacima

iz skupa za validaciju i 87,95% na podacima iz skupa za testiranje. Funkcija gubitka za jednostavniji model iznosi 6,99% gubitka na skupu za učenje, 35,78% gubitka na skupu za validaciju i 36,8% gubitka na skupu za testiranje. Grafički prikaz funkcija preciznosti i gubitka na skupu za testiranje dan je u nastavku.

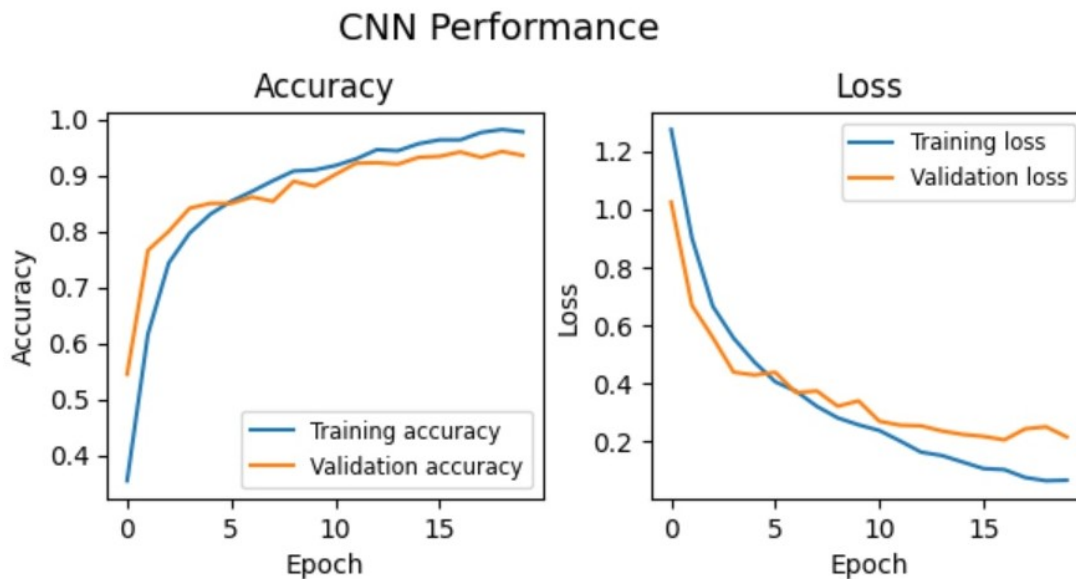


Slika 4.9: Funkcije preciznosti i gubitka za jednostavniji model

Složeniji model daje nešto bolje predikcije, odnosno postiže veću točnost i manji gubitak. Na testnom skupu podataka točnost složenijeg modela prelazi 93%, a funkcija gubitka je nešto manja od 20%. Točni iznosi preciznosti i gubitka na sva tri skupa, kao i grafički prikazi funkcija na testnom skupu dani su u nastavku.

```
143/143 [=====] - 25s 174ms/step - loss: 0.0199 - accuracy: 0.9956
36/36 [=====] - 10s 274ms/step - loss: 0.2432 - accuracy: 0.9327
41/41 [=====] - 14s 330ms/step - loss: 0.1935 - accuracy: 0.9382
Train Loss: 0.019926374778151512
Train Accuracy: 0.9956217408180237
Validation Loss: 0.24318447709083557
Validation Accuracy: 0.932692289352417
Test Loss: 0.1934806853532791
Test Accuracy: 0.9382150769233704
```

Slika 4.10: Prikaz rezultata složenijeg modela



Slika 4.11: Funkcije preciznosti i gubitka za složeniji model

Iz gore prikazanih grafova vidljivo je da složeniji model brže uči. Jednostavniji model svoj maksimum postiže u 19. epohi učenja dok složeniji model svoj maksimum postiže nešto ranije, u 17. epohi učenja.

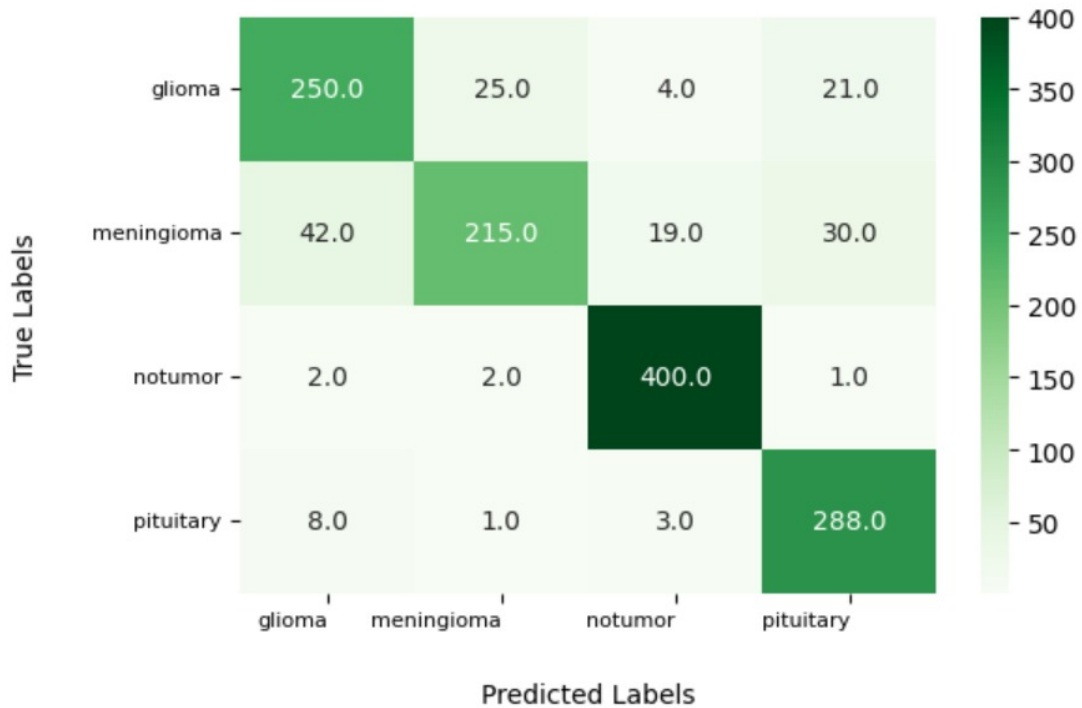
4.4.3. Matrica konfuzije

Matrica konfuzije je vizualni prikaz različitih ishoda predviđanja i rezultata problema klasifikacije. Veličina matrice je $N \times N$ gdje N predstavlja broj mogućih izlaza modela, to jest broj klasa u klasifikacijskom problemu. Pomoću standardne mjere točnosti nemamo uvid u točne performanse klasifikatora. Primjerice model bi mogao pogrešno klasificirati neke kategorije više nego druge, ali to ne možemo vidjeti pomoću standardne mjere točnosti. Također ukoliko postoji neravnoteža među podacima, odnosno ako neka klasa ima više instanci podataka od ostalih, model bi mogao imati visoku stopu točnosti za najmnogobrojniju klasu, dok za ostale nema dobru mogućnost predviđanja. U takvim situacijama korisno je izraditi matricu konfuzije [20].

Postoji nekoliko kombinacija predviđenih i stvarnih vrijednosti klasifikatora. Dijagonalni elementi matrice predstavljaju prave pozitivne (ispravno predviđene) instance za svaku klasu. Nediagonalni elementi predstavljaju pogrešne klasifikacije. Zbroj svakog retka predstavlja ukupan broj instanci u stvarnoj klasi, a zbroj svakog stupca predstavlja ukupan broj instanci predviđenih za svaku klasu.

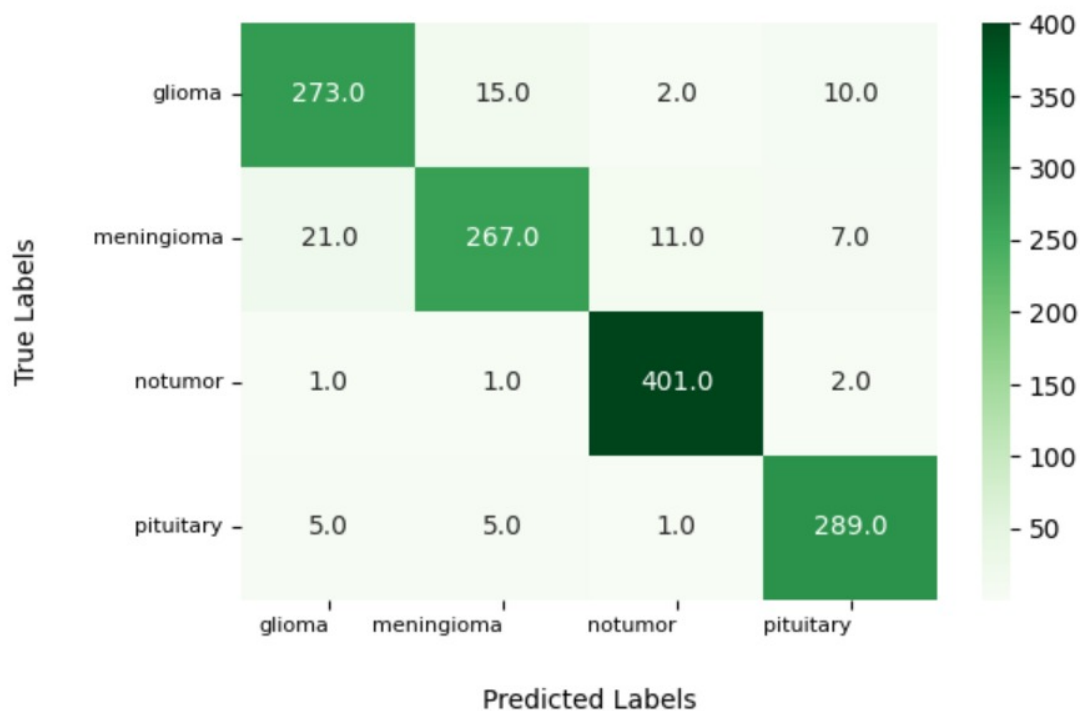
Testni skup sadrži ukupno 1311 slika od kojih 300 pripada oznaci *glioma*, 306 pripada oznaci *meningioma*, 405 pripada oznaci *no tumor* i 300 pripada oznaci *pitutary*.

Prva matrica je matrica konfuzije jednostavnijeg modela. Vidljivo je da model najbolje klasificira slike s oznakom *no tumor* s točnih 400 primjera, dok ostale kategorije klasificira s točnostima u rasponu od 215 do 288. Broj točno klasificiranih slika dakle ovisi i o veličini uzorka pojedine kategorije, klasa s najviše točnih primjera je ujedno i klasa s najviše podataka. Povećanjem broja slika u ostalim klasama dovelo bi do smanjenja pogrešnih klasifikacija.



Slika 4.12: Matrica konfuzije za jednostavniji model

Složeniji model kao što je očekivano bolje klasificira primjere, i predstavlja pogodniju implementaciju klasifikatora moždanih oboljenja. Kao i kod jednostavnijeg modela, klasa s najviše točno klasificiranih primjera je *no tumor*, dok su ostale vrijednosti brojeva točno klasificiranih primjera u rasponu od 267 do 289.



Slika 4.13: Matrica konfuzije za složeniji model

Mjera kojom možemo prikazati točnost modela u problemima klasifikacije naziva je *F1 score*. Računa se kao harmonijska sredina preciznosti (engl. *precision*) i odziva (engl. *recall*). Odziv je mjera sposobnosti modela da ispravno identificira pozitivne instance (tumore) među svim stvarnim pozitivnim instancama, odnosno to je omjer broja točno predviđenih i ukupnog broja stvarnih pozitivnih slučajeva kad je tumor prisutan. Preciznost mjeri točnost modela u ispravnom predviđanju pozitivnih slučajeva. Drugim riječima, preciznost izračunava omjer točno predviđenih pozitivnih instanci prema ukupnom broju instanci predviđenih kao pozitivnih. Formula pomoću koje se računa *F1 score* je $F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$ [14].

$$F1 \text{ Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

Slika 4.14: Formula za izračunavanje F1-score, preuzeto iz [14]

```

41/41 [=====] - 6s 143ms/step
      precision    recall  f1-score   support

   glioma         0.91     0.91     0.91        300
 meningioma       0.93     0.87     0.90        306
   notumor       0.97     0.99     0.98        405
  pituitary       0.94     0.96     0.95        300

 micro avg         0.94     0.94     0.94       1311
 macro avg         0.94     0.93     0.93       1311
weighted avg         0.94     0.94     0.94       1311
 samples avg         0.94     0.94     0.94       1311

```

Slika 4.15: F1-score za složeniji model

5. Zaključak

Ovaj rad opisuje i prikazuje izgranju i učenje modela čiji je zadatak napraviti točnu klasifikaciju moždanih oboljenja na temelju snimki magnetskih rezonanci. U radu je objašnjena teorija koja stoji iza principa dubokog učenja, kao i teoretsko pojašnjenje konvolucijskih neuronskih mreža i njihova primjena. Izrađena su dva modela kako bi se pokazale razlike u rezultatima pri korištenju jednostavnijeg i složenijeg modela. Složeniji model daje veću preciznost i manju grešku, te je samim time bolji izbor za korištenje u medicinske svrhe sa stajališta medicinske dijagnostike zbog točnijih klasifikacija.

Izgrađeni modeli su primjer praktične primjene modela dubokog učenja. Primjena dubokog učenja se svakim danom sve više širi i već je našlo svoju uporabu u različitim ljudskim djelatnostima poput računalnog vida, obradi prirodnog jezika odnosno klasifikaciji teksta, financija, jer pružaju pomoć pri analizi velikih količina financijskih podataka i stvaraju predviđanja na temelju njih, robotike, autonomnih sustava, gdje omogućuju sustavima poput samovozećih automobila bolju percepciju okoline i donošenje odluka, pa tako i u medicini.

Poboljšanje modela može se postići povećavanjem skupa podataka, pogotovo u kategorijama s manje podataka kao na primjer *glioma*, dodatnim povećavanjem složenosti modela (dodavanjem novih slojeva, povećavanjem broja filtara i sl.), dodatnom preobradom slika kako bi se povećala raznolikost podataka u skupu za učenje (primjerice dodavanje šuma ili okretanje slika), kao i mnogim drugim tehnikama.

LITERATURA

- [1] P. Antoniadis. Hidden layers in a neural network, 2022. URL <https://www.baeldung.com/cs/hidden-layers-neural-network>. 2022-11-05.
- [2] M. Banoula. Classification in machine learning: What it is & classification models, 2023. URL https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning#what_is_classification_algorithm. 2023-05-23.
- [3] N. Bhargav. Neurons in neural networks, 2022. URL <https://www.baeldung.com/cs/neural-networks-neurons>. 2022-10-26.
- [4] J. Brownlee. 4 types of classification tasks in machine learning, 2019. URL <https://machinelearningmastery.com/types-of-classification-in-machine-learning/>. 2019-08-19.
- [5] Cancer.net. Brain tumor: Statistics, 2023. URL <https://www.cancer.net/cancer-types/brain-tumor/statistics>. 2023/03.
- [6] Mayo Clinic. Brain tumor, 2023. URL <https://www.mayoclinic.org/diseases-conditions/brain-tumor/symptoms-causes/syc-20350084#dialogId4207612>. 2023-04-21.
- [7] Mayo Clinic. Brain tumor, 2023. URL <https://www.mayoclinic.org/diseases-conditions/brain-tumor/symptoms-causes/syc-20350084>. 2023-04-21.
- [8] O. G. Evans. Parts of the brain: Anatomy, structure & functions, 2023. URL <https://www.simplypsychology.org/anatomy-of-the-brain.html>. 2023-05-15.

- [9] A. Gupta. A comprehensive guide on optimizers in deep learning, 2023. URL <https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/>. 2023-05-18.
- [10] hopkinsmedicine. Brain anatomy and how the brain works, 2023. URL <https://www.hopkinsmedicine.org/health/conditions-and-diseases/anatomy-of-the-brain>. 2023/03.
- [11] IBM. What is deep learning?, 2023. URL <https://www.ibm.com/topics/deep-learning>. 2023-05-20.
- [12] IBM. What is a neural network?, 2023. URL <https://www.ibm.com/topics/neural-networks>. 2023-05-20.
- [13] National Cancer Institute. Adult central nervous system tumors treatment (pdq®)patient version, 2021. URL <https://www.cancer.gov/types/brain/patient/adult-brain-treatment-pdq>. 2021-08-06.
- [14] R. Kundu. F1 score in machine learning: Intro & calculation, 2022. URL <https://www.v7labs.com/blog/f1-score-guide>. 2022-12-16.
- [15] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, i Jun Zhou. *A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects*. IEEE, 2021. URL <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9451544>.
- [16] M. Mishra. Convolutional neural networks, explained, 2020. URL <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>. 2020-04-26.
- [17] S. Saha. A comprehensive guide to convolutional neural networks — the eli5 way, 2018. URL <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way>. 2018-12-15.
- [18] T. Shah. About train, validation and test sets in machine learning, 2017. URL <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>. 2017-12-06.

- [19] Simplilearn. What is keras: The best introductory guide to keras, 2023. URL <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras>. 2023-04-01.
- [20] simplilearn. What is a confusion matrix in machine learning?, 2023. URL <https://www.simplilearn.com/tutorials/machine-learning-tutorial/confusion-matrix-machine-learning>. 2023-02-16.
- [21] tutorialspoint. Keras - models, 2017. URL https://www.tutorialspoint.com/keras/keras_models.htm. 2017-12-06.
- [22] Wikipedia. Multi-label classification, 2023. URL https://en.wikipedia.org/wiki/Multi-label_classification. 2023-03-03.

Klasifikacija moždanih tumora metodama dubokog učenja na temelju snimki magnetske rezonancije

Sažetak

Ovaj rad prikazuje primjenu dubokog učenja, odnosno konvolucijskih neuronskih mreža u rješavanju klasifikacijskog problema u području medicinske dijagnostike. Prikazana je izgradnja dva modela neuronskih mreža i analizirane su razlike u njihovoj građi koje utječu na performanse i dobivene rezultate. Rezultati učenja neuronskih mreža dani su u obliku funkcija gubitka i preciznosti, i matrice konzucije. Postignuta je točnost klasifikacije četiri klase moždanih tumora od preko 93% koristeći konvolucijski model.

Ključne riječi: Umjetna inteligencija, duboko učenje, konvolucijske neuronske mreže, klasifikacija slika

Brain tumor classification using deep learning methods based on magnetic resonance imaging

Abstract

This paper shows the application of deep learning, respectively convolutional neural networks in solving the classification problem in the field of medical diagnostics. It shows the building of the two models of neural networks, and analyses the differences in their structure, which affect the performance and the obtained results. The learning results of the neural networks are given in the form of loss and precision functions, and the confusion matrix. The classification accuracy of four classes of brain tumors of over 93% was achieved using the convolutional model.

Keywords: Artificial intelligence, deep learning, convolutional neural networks, image classification