

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1395

**KLASIFIKACIJA RESPIRATORNIH BOLESTI NA TEMELJU
ZVUČNIH ZAPISA AUSKULTACIJE PLUĆA KORIŠTENJEM
METODA STROJNOG UČENJA**

Mia Krstičević

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1395

**KLASIFIKACIJA RESPIRATORNIH BOLESTI NA TEMELJU
ZVUČNIH ZAPISA AUSKULTACIJE PLUĆA KORIŠTENJEM
METODA STROJNOG UČENJA**

Mia Krstičević

Zagreb, lipanj 2024.

ZAVRŠNI ZADATAK br. 1395

Pristupnica: **Mia Krstičević (0036540586)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: izv. prof. dr. sc. Alan Jović

Zadatak: **Klasifikacija respiratornih bolesti na temelju zvučnih zapisa auskultacije pluća korištenjem metoda strojnog učenja**

Opis zadatka:

Ispravna i pravovremena dijagnoza respiratornih bolesti važna je za početak liječenja. Respiratorne bolesti moguće je dijagnosticirati temeljem zvučnih zapisa auskultacije digitalnim stetoskopom. Zbog prisutnih smetnji u zapisima i nedostatka vremena, liječnicima bi od značajne pomoći bio automatizirani sustav koji bi visokotočno klasificirao respiratorne bolesti temeljem zvučnih zapisa. Cilj ovog završnog rada je izrada klasifikacijskog modela za nekoliko respiratornih bolesti koji će na temelju zvučnog zapisa auskultacije pluća ustanoviti vrstu oboljenja. Kao skup za učenje modela potrebno je upotrijebiti neki od dostupnih skupova podataka zvučnih zapisa auskultacije pluća, kao što je Respiratory Sound Database, dostupan na web sjedištu Kaggle (<https://www.kaggle.com/datasets/vbookshelf/respiratory-sound-database/data>). Klasifikacijski model treba biti zasnovan na nekom od algoritama strojnog učenja, uključujući algoritme dubokog učenja. Primjeri takvih algoritama su konvolucijske neuronske mreže, povratne neuronske mreže i transformerske arhitekture. U radu je također potrebno razmotriti metode predobrade zvučnih zapisa auskultacije pluća kako bi se poboljšala učinkovitost izgrađenog modela. Izgrađeni model potrebno je ugraditi u jednostavnu web aplikaciju koja će na temelju učitano zapisa odrediti o kojem se oboljenju radi i tu informaciju prikazati korisniku. Implementaciju je potrebno napraviti u programskom jeziku po vlastitom izboru, a za izgradnju modela može se u slučaju nedostatka vlastitih sklopovskih resursa koristiti dostupna web rješenja (npr. Google Colab).

Rok za predaju rada: 14. lipnja 2024.

Zahvaljujem izv. prof. dr. sc. Alanu Joviću na pomoći te obitelji i prijateljima na podršci.

Sadržaj

Uvod	3
1. Respiratorne bolesti	4
1.1. Respiratorni sustav	4
1.2. Digitalni stetoskop	5
2. Podaci	6
2.1. Opis skupa podataka	6
2.2. Predobrada podataka	7
2.3. Poboljšavanje podataka	9
2.4. Ekstrakcija značajki	10
2.5. Podjela podataka	12
3. Duboko učenje	13
3.1. Neuronske mreže	13
3.2. Konvolucijske neuronske mreže	14
3.3. Arhitektura mreže	16
3.4. Učenje modela	17
3.5. Analiza rezultata	18
4. Web aplikacija	20
4.1. Opis i izgled web aplikacije	20
4.2. Tehnologije i alati	21
4.3. Implementacija	22
5. Zaključak	24

Literatura	25
Sažetak	28
Abstract	29

Uvod

Dišni ili respiratorni sustav jedan je od najkompleksnijih i najznačajnijih sustava svih živih bića. Prema podacima iz 2017. godine, u svijetu boluje 544,9 milijuna ljudi od kroničnih respiratornih bolesti s povećanjem od 39% u odnosu na podatke iz 1990-ih [5].

Kao posljedica razvoja tehnologije, a posebice umjetne inteligencije, otvaraju se mnoge prilike i mogućnosti za daljnji razvoj u svim sektorima. Spoj medicine i tehnika strojnog učenja doseže do nove razine znanja i otkrića, značajno olakšavajući rad mnogim liječnicima i znanstvenicima.

Tema ovog završnog rada je klasifikacija respiratornih bolesti na temelju zvučnih zapisa snimljenih digitalnim stetoskopom. Klasifikacija se ostvaruje korištenjem metoda dubokog učenja, sastavnog dijela strojnog učenja. Njome se ostvaruje predikcija diskretnih značajki, u ovom slučaju dijagnoza respiratornih bolesti. Cilj je istražiti različite metode za pripremu i predobradu podataka te razvoj modela kako bi se postigla dobra generalizacija. Za postizanje kvalitetnog modela, koriste se konvolucijske neuronske mreže. Da bi liječnicima uporaba bila jednostavnija i primjenjiva u svakodnevnom radu, ovaj model je implementiran u jednostavnu web aplikaciju.

Uz literaturu i sažetak, rad se sastoji od pet poglavlja. U prvom poglavlju opisuje se dišni sustav te uporaba digitalnog stetoskopa. Drugo poglavlje sadrži opis skupa podataka, predobradu i pripremu podataka za unos u model strojnog učenja. Treće poglavlje objašnjava teoriju iza dubokog učenja i neuronskih mreža. Pritom se prikazuje izgradnja mreže, učenje modela i analiza rezultata. U petom se opisuje funkcionalnost i izgled web aplikacije u koju je implementiran model strojnog učenja.

1. Respiratorne bolesti

1.1. Respiratorni sustav

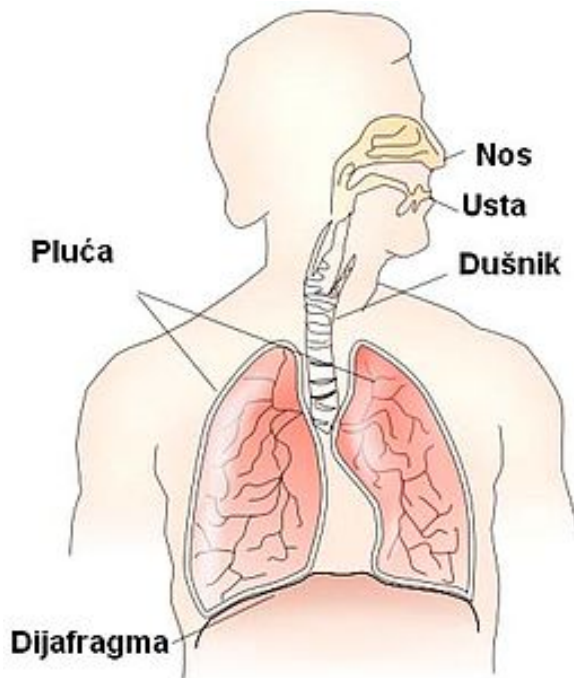
Respiratorni sustav čovjeka je organski sustav koji je zadužen za izmjenu plinova između organizma i okoline. Neke od njegovih glavnih uloga su udisanje (inspiracija), izdisanje (ekspiracija) i filtracija zraka kroz usnu i nosnu šupljinu. Organe za disanje čine: usna i nosna šupljina, ždrijelo, grkljan, dušnik i pluća [4], slika 1.1. Pluća su ključan dio respiratornog sustava, a sastoji se od bronha i bronhiola, dišnih cijevi koje vode zrak iz dušnika (traheje) do alveola. Alveole su sitni mjehurići gdje se odvija razmjena plinova. Pritom se udahnuti kisik apsorbira u krv kroz kapilare, a iz krvi u alveole se izlučuje ugljikov dioksid te izdiše.

Konstantan podražaj s okolinom rezultira čestim infekcijama, osobito kod djece. Mogu biti u rasponu od najjednostavnijih oblika poput prehlade sve do najtežih slučajeva kao što je kronična opstruktivna plućna bolest (KOPB).

KOPB je jedna od najtežih oblika respiratornih bolesti koja dovodi do trajnih oštećenja dišnih puteva. Bolest se s vremenom pogoršava te dolazi do slabije opskrbe tijela kisikom. Simptomi su otežano disanje, umor, napadaji kašlja i osjećaj pritiska u prsima [22]. Astma je kronična opstruktivna bolest dišnih putova, nastala kao kombinacija genetskih i okolišnih čimbenika. Također uzrokuje kratkoću daha, zviždanje u plućima i kašljanje [19]. Uz to, bronhiektazija je kronično stanje karakterizirano oštećenim i abnormalno proširenim bronhima pluća [21].

Od akutnih infekcija, često se javlja bronhitis, upala pluća te infekcija gornjih dišnih (engl. *Upper respiratory tract infection*, URTI) i donjih dišnih puteva (engl. *Lower respiratory tract infection*, LRTI). Bronhitis je upala sluznice bronha i jedan je od preduvjeta za razvoj KOPB-a. Upala pluća je infekcija plućnog tkiva odnosno alveola [20]. Može

biti uzrokovana bakterijama, virusima, gljivicama i slično. Infekcije gornjih dišnih putova su najčešće prehlada, upala sinusa (sinusitis), upala grla (faringitis) i upala grkljana (laringitis), dok su za donje dišne puteve najčešći upala pluća i bronhitis.



Slika 1.1. Respiratorni sustav, preuzeto iz [4]



Slika 1.2. 3M Littmann digitalni stetoskop, preuzeto iz [6]

1.2. Digitalni stetoskop

Jedan od glavnih alata kojim se koriste liječnici za pregled i provjeru dišnih putova je stetoskop. U skupu podataka [1], zvučni zapisi su snimljeni pomoću digitalnog stetoskopa. Naime, to je suvremeni medicinski uređaj koji snima i zabilježava auskultacije pluća, slika 1.2. Pri snimanju zvučnih zapisa za skup podataka [1] korištene su tri različite vrste opreme:

- AKG C417L mikrofoni
- 3M Littmann stetoskop
- WelchAllyn Meditron Master Elite Electronic stetoskop

2. Podaci

Za što veću ispravnost i točnost modela strojnog učenja, potrebno je kvalitetno proučiti i obraditi skup podataka [1]. Skup podataka je preuzet s web stranice Kaggle (<https://www.kaggle.com/datasets/vbookshelf/respiratory-sound-database/data>). Uz zvučne zapise snimljene digitalnim stetoskopom priloženi su i demografski podaci u datoteci *demographic_info.txt* te podaci o dijagnozi pacijenta u *patient_diagnosis.csv*.

2.1. Opis skupa podataka

Skup podataka za ovaj rad se sastoji od 920 zvučnih zapisa .wav formata. Zapisi su vremenske duljine od 10 do 90 sekundi, sakupljenih od 126 pacijenta. Svakom pacijentu je zabilježeno disanje s više različitih lokacija: dušnik, prednje lijevo i desno, stražnje lijevo i desno te bočno lijevo i desno plućno krilo.

Bitna stavka u osluškivanju respiratornih ciklusa jest da se kod respiratornih bolesti poput astme i KOPB-a, često pojavljuje tzv. škripanje i hroptanje. Za svaku identifikacijsku oznaku pacijenta (id) u datoteci *patient_diagnosis.csv* zabilježen je njegov podatak o dijagnozi, a za demografsku strukturu pacijenata priložene su:

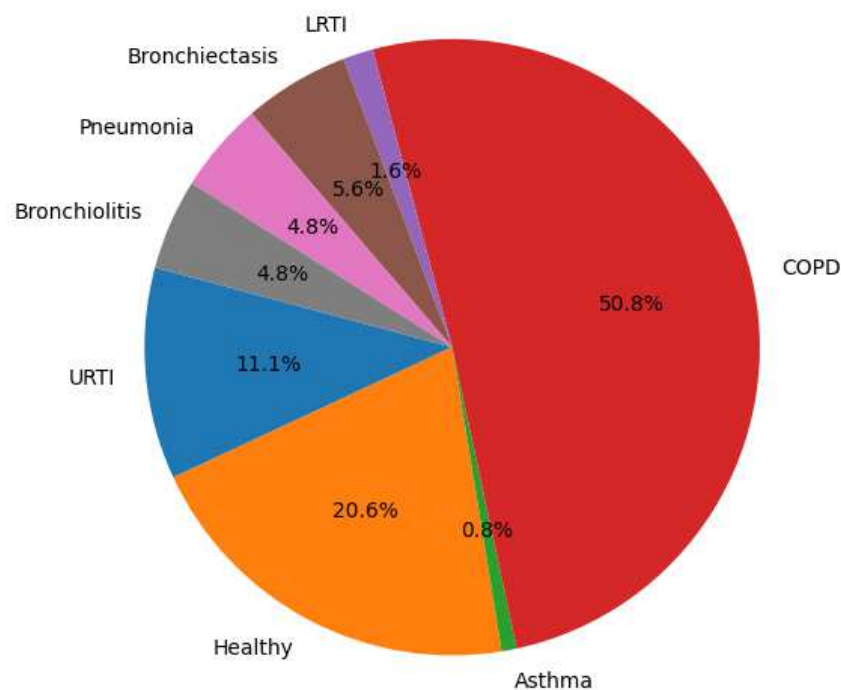
- godina
- spol
- indeks tjelesne mase (odrasle osobe)
- težina djeteta
- visina djeteta

Ciljne klase odnosno dijagnoze su: KOPB, astma, upala pluća, upala gornjih i donjih

dišnih puteva, bronhitis, bronhiektazija i zdrava osoba. Spajanjem ovih dviju datoteka po značajki *patient_id* može se proučiti odnos između parametra i njihova važnost za implementaciju u klasifikacijski model.

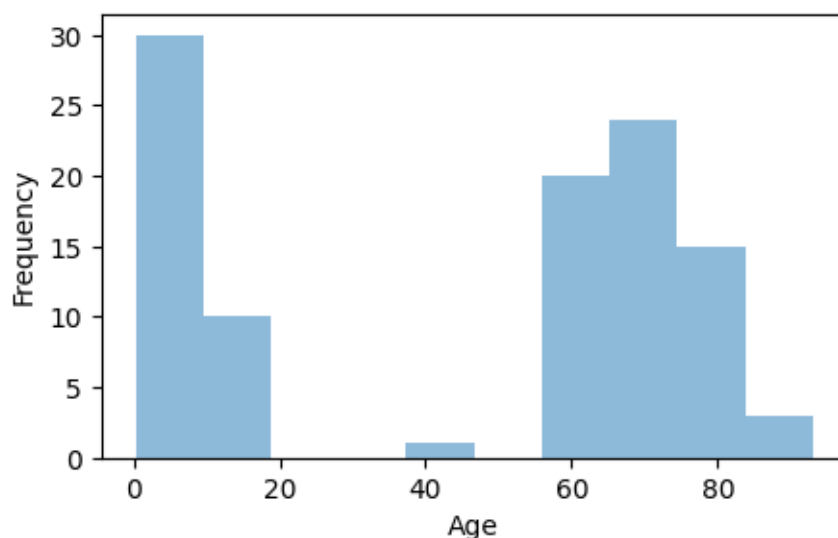
2.2. Predobrada podataka

Potrebno je proučiti distribuciju ciljnih klasa. Analizom kružnog grafikona prikazanog na slici 2.1, može se zaključiti da od sveukupno 126 pacijenta, više od polovice boluje od KOPB-a, zatim slijedi 20% zdravih osoba i 11,1% osoba zaraženih upalom donjih dišnih puteva. Za sve ostale klase postoje zvučni zapisi za manje ili jednako sedam osoba. To je premalo podataka za adekvatno učenje modela, zbog čega će se od šest klasi zadržati tri kako bi se postigle bolje predikcije modela. Odabrane su klase KOPB-a, bronhiektazije i zdravih osoba.



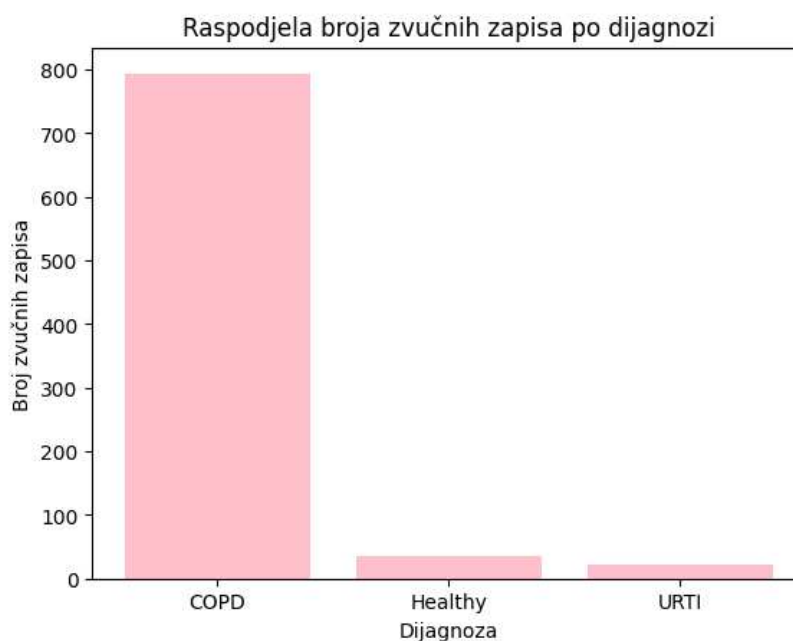
Slika 2.1. Kružni grafikon - distribucija ciljnih klasa

Također je potrebno analizirati utjecaj određenih parametara na izlaznu vrijednost. Što se bolje izdvoje značajni podaci, to će performanse modela biti bolje. Proučava se povezanost između spola, godina, težine i visine u odnosu na dijagnozu. U podacima ima čak 44 djece (35%) od kojih za dvoje nema podataka o visini. Tri pacijenta nemaju podatke o BMI-u ili visini i težini djeteta. Ove pacijente bi se trebalo ukloniti, ali s



Slika 2.2. Histogram godina

obzirom da težina i visina nemaju značajan utjecaj na dijagnozu, to neće biti potrebno. Dodatno, spol nema utjecaja na dijagnozu. Značajan doprinos čine podaci o starosti pacijenata. Naime, starije osobe najčešće boluju od KOPB-a, dok djecu i adolescente češće zahvate upalni procesi, alergijske astme i slično. Iz histograma starosti prikazanog na slici 2.2, može se zaključiti da nedostaju podaci za osobe između 20 do skoro 40 godina te za osobe od 45 do 55 godina. Iz navedenih razloga, nijedna od ovih značajki neće biti korisna za klasifikaciju.



Slika 2.3. Stupčasti dijagram zvučnih zapisa

Na slici 2.3. vidi se da je skup podataka izrazito nebalansiran gdje od 920 zvučnih zapisa, 794 pripada osobama kojima je dijagnosticiran KOPB. Upravo iz ovog razloga, nužno je provesti poboljšavanje podataka kako bi se pokušalo doći do što bolje generalizacije modela.

Neposredno prije poboljšavanja podataka, potrebno je kodirati oznake odnosno izlazne klase. To je potrebno jer neuronska mreža ne može raditi s oznakama u izvornom obliku, već prima numerički ulaz. Postoji više tehnika za kodiranje, a za ovu klasifikaciju se koristi *label encoder*. Ova tehnika pretvara kategoričke varijable u numerički format, pri čemu se vrijednosti kreću od nula do n elemenata (varijabla n je broj različitih klasa). U ovom radu oznake su kodirane uz funkciju *LabelEncoder* iz Pythonove knjižnice scikit-learn.

2.3. Poboljšavanje podataka

Poboljšavanje skupa podataka (engl. *data augmentation*) je postupak generiranja novih podataka iz starih kako bi se poboljšala kvaliteta modela i smanjila prenaučенost. Najčešće se primjenjuje u obradi slika i signala. Kod zvučnih zapisa, koriste se različite metode poput dodavanja šuma, promjena tempa i visine tona, promjena brzine i slično. Ovim tehnikama postiže se varijabilnost među podacima kako bi ih se moglo dalje koristiti u izvedbi [7].

Za implementaciju ovih metoda koristi se Pythonova knjižnica Librosa koja služi za obradu i manipulaciju audio datoteka. Prikaz metoda za poboljšavanje podataka dan je na slici 2.4. Funkcija *add_noise* dodaje originalnom audio zapisu nasumični šum u kombinaciji s faktorom šuma. Dobiveni podatak se na posljetku normalizira. Pomicanje zvuka ili *sound_shift* uzima kao parametar *shift_factor* i pomiče signal ulijevo ili udesno. Naposljetku, *pitch* mijenja visinu tona tj. frekvenciju audio signala.

Međutim, umjesto proširivanja cijelog skupa podataka, primjenjuje se selektivno poboljšavanje (engl. *selective augmentation*) nad manje zastupljenim klasama. Nije potrebno generirati nove podatke za izlaznu klasu KOPB, s obzirom da ona predvladava u skupu. Ovakav pristup je rezultirao boljim performansama i većoj preciznosti.

```

#noise
def add_noise(audio_data, noise_factor):
    noise = np.random.normal(0, 1, len(audio_data))
    augmented_data = audio_data + noise_factor * noise
    augmented_data = augmented_data / np.max(np.abs(augmented_data))

    return augmented_data

#shift
def sound_shift(audio_data, sampling_rate, shift_factor):
    shift = np.random.randint(sampling_rate * shift_factor)
    augmented_data = np.roll(audio_data, shift)

    if shift > 0:
        augmented_data[:shift] = 0
    else:
        augmented_data[shift:] = 0
    return augmented_data

#pitch
def pitch(audio_data, sampling_rate, pitch_factor):
    return librosa.effects.pitch_shift(audio_data, sr= sampling_rate, n_steps= pitch_factor)

```

Slika 2.4. Poboljšavanje podataka, prilagođena verzija kôda iz [8]

2.4. Ekstrakcija značajki

Jedan od ključnih koraka u pripremi podataka za unos u model je izdvajanje ili ekstrakcija značajki. Zvučni zapisi sami po sebi ne mogu služiti kao ulazna vrijednost, već se trebaju preraditi. Potrebno je iz njih izvući najkorisnije značajke.

```

def extract_features_2d(audio_data, sample_rate):
    target_shape = (128, 128)

    # Spektrogram
    spectrogram = np.abs(librosa.stft(audio_data))
    spectrogram_db = librosa.power_to_db(spectrogram, ref=np.max)
    spectrogram_normalized = (spectrogram_db - np.mean(spectrogram_db)) / np.std(spectrogram_db)
    spectrogram_reshaped = spectrogram_normalized[..., np.newaxis]
    spectrogram_resized = cv2.resize(spectrogram_reshaped, target_shape[::-1])

    # MFCC
    mfcc = librosa.feature.mfcc(y=audio_data, sr=sample_rate)
    mfcc_normalized = (mfcc - np.mean(mfcc)) / np.std(mfcc)
    mfcc_resized = cv2.resize(mfcc_normalized, target_shape[::-1])

    # Chroma
    chroma = librosa.feature.chroma_stft(y=audio_data, sr=sample_rate)
    chroma_normalized = (chroma - np.mean(chroma)) / np.std(chroma)
    chroma_resized = cv2.resize(chroma_normalized, target_shape[::-1])

    output = np.concatenate((spectrogram_resized, mfcc_resized, chroma_resized), axis=-1)

    return output

```

Slika 2.5. Izdvajanje značajki, kôd konzultiran s [10]

Zvuk je mehanički val koji se prenosi kroz medij, uzrokujući oscilacije čestica [15]. Rezultat toga je promjena u tlaku zraka, što ljudsko uho percipira kao zvuk. Uz pomoć

biblioteke Librosa, iz svake .wav datoteke izdvoje se audio podaci i frekvencija uzorkovanja (*sample_rate*), koji se zatim koriste kao parametri za funkcije izdvajanja značajki.

Izdvajanje značajki prikazano je na slici 2.5. Među izdvojenim značajkama su spektrogram, MFCC (Mel-frekvencijski kepralni koeficijenti) i kromagram (engl. *chroma*). Spektrogram je vizualna reprezentacija promjene frekvencije zvuka kroz vrijeme [9]. Ostvaruje se Fourierovom transformacijom. To je matematička formula koja pretvara signal iz vremenske domene u frekvencijsku. Uvelike olakšava digitalnu obradu zvuka jer se mogu identificirati i analizirati različite frekvencijske komponente.

$$F(\omega) = \int_{-\infty}^{\infty} f(x)e^{-i\omega x} dx$$

$F(\omega)$ je Fourierova transformacija funkcije $f(x)$, ω je frekvencijska varijabla, a $e^{-i\omega x}$ je kompleksna eksponencijalna funkcija.

Uz spektrogram, mel-frekvencijski kepralni koeficijenti (MFCCs) su također od velike važnosti jer precizno opisuju ukupan oblik spektra zvuka. Kromagram preslikava cijeli spektar zvuka u 12 različitih polutonova oktave (ili *chroma*). Koristan je jer pruža informacije o harmoniji, pritom zanemarivajući apsolutne frekvencije.

Za dobivanje spektrograma, MFCC i kromagrama također je korištena knjižnica Librosa. Za spektrogram se prvotno provede kratkotrajna Fourierova transformacija (engl. *short-time Fourier transform*, STFT). STFT omogućava analizu signala kroz vrijeme i brži je u odnosu na klasičnu Fourierovu transformaciju, što je važno za efikasno učenje modela. Od dobivenih vrijednosti se izračuna apsolutna vrijednost spektrograma. Time se dobiva amplituda svake frekvencijske komponente, a gubi se fazna komponenta. To je korisno jer faza nema značajan utjecaj pri učenju modela. Zatim se amplituda pretvara u decibele radi lakše interpretacije rezultata.

Za svaku značajku je provedena normalizacija kako bi se osiguralo da vrijednosti padaju unutar istog raspona. Također, kako bi podaci bili jednakih veličina, svaka značajka je promjenjena na veličinu (128, 128). Na kraju se sve značajke spajaju funkcijom *concatenate* iz knjižnice NumPy.

2.5. Podjela podataka

Tijekom učenja, model na ulazu dobiva parove odnosno pripadni vektor i oznaku (engl. *label*). Zato se, poboljšavanjem podataka i ekstrakcije značajki, svaki dobiveni vektor sprema u listu *images*, dok se svaka dijagnoza za tog pacijenta sprema u listu *labels*. Kako bi se osigurala kompatibilnost s TensorFlow Kerasom, liste *images* i *labels* pretvaraju se u NumPyove nizove pomoću funkcije *np.array*. Proširenjem podataka, sveukupno ima 1304 zvučnih zapisa gdje svaki od njih ima vektor veličine (128, 384).

Nizovi *images* i *labels* se zatim dijele na skupove kako bi se spriječila prenaučенost. Prenaučenost se dogodi kada se model previše prilagodi podacima na kojima uči zbog čega neće dobro klasificirati nove zapise. Kako bi se to izbjeglo, koristi se metoda unakrsne provjere. Podaci se dijele na tri skupa: skup za učenje, skup za validaciju ili provjeru i skup za testiranje. Uobičajeno se podaci raspoređuju tako da je najviše podataka namijenjeno skupu za učenje (oko 70%), dok se za svaki od skupova za validaciju i testiranje odvaja oko 15% podataka, slika 2.6. Isto tako, ključno je nasumično dodavati podatke u skupove kako bi svaki skup imao ravnomjeran udio klasa. To je posebno bitno kada je skup podataka neuravnotežen kao u ovom slučaju. Zato se u funkciji *train_test_split* navodi parametar *random_state*.

```
train_img, temp_img, train_var, temp_var = train_test_split(images, labels, test_size=0.3, random_state=42)
val_img, test_img, val_var, test_var = train_test_split(temp_img, temp_var, test_size=0.5, random_state=42)

print(train_img.shape, train_var.shape)
print(test_img.shape, test_var.shape)
print(val_img.shape, val_var.shape)
```

Slika 2.6. Podjela na skup za učenje, validaciju i testiranje

Skup za učenje u ovom radu ima 912 zvučnih zapisa, dok skup za validaciju i za testiranje ima svaki po 196 zvučnih zapisa. Ovako pripremljeni podaci su konačno spremni za učenje modela dubokog učenja.

3. Duboko učenje

Duboko učenje je područje unutar umjetne inteligencije i ujedno jedna od grana strojnog učenja [16]. Radi po principu da stroj na temelju velikih količina podataka identificira obrasce te donosi nove zaključke. Na temelju tih zaključaka stroj kontinuirano uči.

Također, primjenjuje se u raznim područjima poput računalnog vida, obradi slike i zvuka, obradi jezika i mnogim drugim područjima. Pri implementaciji modela dubokog učenja koristi se arhitektura umjetnih neuronskih mreža. No, nedostatak dubokog učenja su zahtjevi za velikim resursima, kao što su velika količina memorije i brza procesorska snaga. U okviru ovog rada, upravo radi računalnih resursa, korištena je platforma Google Colab za izgradnju modela i provođenje vrednovanja.

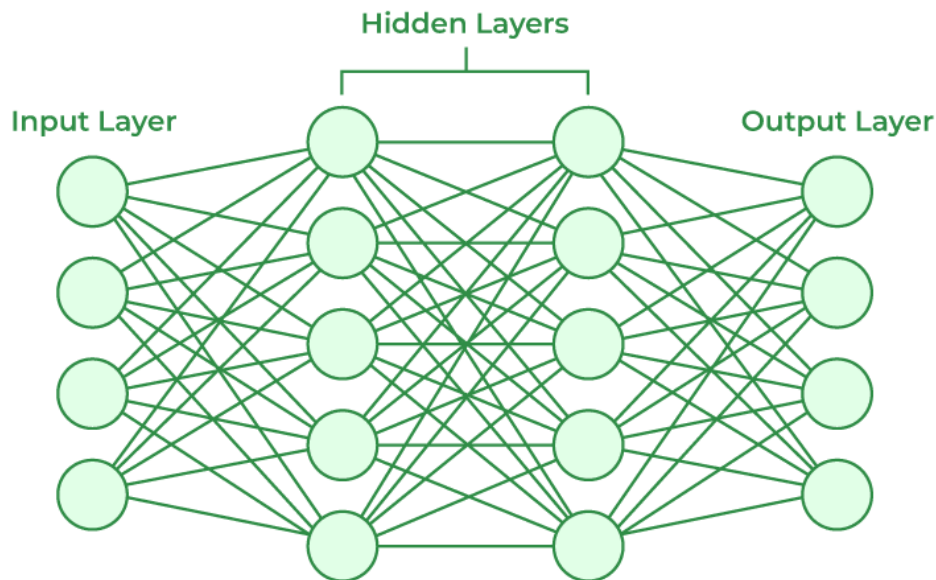
3.1. Neuronske mreže

Umjetna neuronska mreža je skup međusobno povezanih jednostavnih obradbenih jedinica koje se nazivaju neuroni [18]. Njihova funkcija je jednostavna imitacija bioloških neurona, osnovnih jedinica živčanog sustava. Uloga biološkog, pa tako i umjetnog neurona je primanje, obrada i prijenos informacija.

Umjetna neuronska mreža sastoji se od ulaznog sloja (engl. *input layer*) koji prima ulazne podatke, izlaznog sloja (engl. *output layer*) koji sadržava izlazne podatke (klase) i skrivenih slojeva (engl. *hidden layers*) kojih može biti jedan ili više, slika 3.1. Skriveni slojevi se nalaze između ulaznog i izlaznog sloja te obrađuju podatke tako da uzmu najznačajnije informacije na temelju kojih uče.

Svi neuroni su međusobno povezani tako da svaka konekcija ima određenu težinu (engl. *weights*). Težine određuju koliki je utjecaj jednog neurona na drugi i ključne su za proces učenja. Prilikom učenja modela, bitno je istaknuti da se ove težine mijenjaju kako

bi mreža mogla što preciznije odrediti izlaznu klasu. Podaci se proslijeđuju u sljedeći sloj mreže tako da se za svaki neuron u novom sloju izračuna vrijednost temeljena na težinama iz prethodnog sloja. Prilikom računanja, uzimaju se u obzir težine neurona koje su povezane s novim. U izlaznom sloju te vrijednosti služe za određivanje pripadnosti pojedinoj klasi.



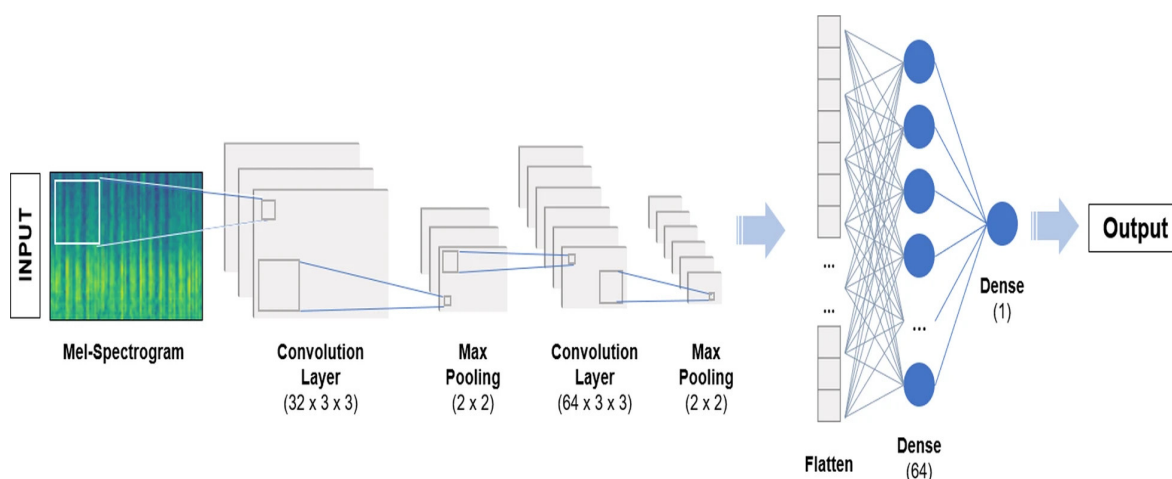
Slika 3.1. Neuronska mreža, preuzeto s [16]

3.2. Konvolucijske neuronske mreže

Za učenje modela u ovom radu koriste se konvolucijske neuronske mreže (engl. *Convolutional Neural Network*, CNN). One su jedan od modela dubokog učenja i izvrsne za izvođenje kompleksnih zadataka. Posebno su korisne u analizi vizualnih podataka poput slika, a pokazala se i visoka razina uspješnosti ovih mreža za obradu zvuka. Glavni koncept CNN-a je korištenje konvolucije, matematičke funkcije koja kombinira ulazni podatak s jezgrom (engl. *kernel*), pritom stvarajući novi izlaz. Na taj način se iz podataka izdvajaju značajke na temelju kojih sustav uči.

Mreža se sastoji od tri sloja: konvolucijski sloj, sloj sažimanja (engl. *Pooling layer*) i potpuno povezani sloj (engl. *Fully-connected layer*). Na slici 3.2. prikazan je tipični primjer konvolucijske neuronske mreže.

Prvi sloj i primarni čimbenik u izgradnji mreže je konvolucijski sloj. Funkcionira tako



Slika 3.2. Konvolucijska neuronska mreža, preuzeto s [23]

što primjenjuje konvoluciju na ulazne podatke odnosno analizira svaku sliku segment po segment koristeći jezgru. Pritom je veličina filtra ili jezgre uvijek manja od same slike. Na početku obrade, naglasak je na izdvajanju sitnijih dijelova i detalja koji će se u zadnjim slojevima koristiti u formiranju šire slike. U ovom radu, slika predstavlja spektrogram, MFCC i chroma značajke kombinirane zajedno. Ovaj pristup dopušta mreži da efikasno prepozna složene uzorke i karakteristike. Konvolucijskih slojeva može biti jedan ili više, a često se koristi više slojeva radi postizanja veće preciznosti.

S obzirom da se u ovom procesu računa s matricama (obje slike i jezgra), može doći do velike potrošnje računalnih resursa i velikog broja parametra. Treba smanjiti dimenzionalnost podataka i spriječiti korištenje velikih količina memorije. Upravo tu ulogu ima sloj sažimanja. Postoje više metoda sažimanja, primjerice sažimanje najvećom vrijednošću (engl. *max pooling*) i sažimanje prosječnom vrijednošću (engl. *avg pooling*). Kod sažimanja najvećom vrijednošću, matrica se podijeli u segmente prema zadanoj veličini, a iz svakog segmenta se vraća maksimalna vrijednost. S druge strane, sažimanje prosječnom vrijednošću uzima iz svakog dijela prosječnu vrijednost. No, obično se uzima sažimanje najvećom vrijednošću jer se pokazalo da daje najbolji rezultat. Ovim slojem se povećava invarijantnost na translaciju, što omogućuje mreži bolju generalizaciju.

Potpuno povezani sloj je posljednji sloj u neuronskoj mreži i služi kako bi izlazne klase bile u skladu s ulaznim podacima. U ovom sloju se postiže potpuna povezanost između neurona, što omogućuje modelu da klasificira ili predviđa podatke. To se ostvaruje uz primjenu aktivacijskih funkcija i suma težina.

3.3. Arhitektura mreže

Pri implementaciji ovog modela korištene su biblioteke Tensorflow i Keras. One omogućuju brzo i jednostavno definiranje modela. Model se sastoji od pet konvolucijskih slojeva, pet slojeva sažimanja, sloja za izravnavanje (*Flatten*), dva sloja za ispuštanje (*Dropout*) i tri potpuno povezana sloja (*Dense*), slika 3.3.

```
modelCNN = tf.keras.Sequential([
    tf.keras.layers.Conv2D(64, 5, activation='relu', padding='same', input_shape=(128, 384, 1)),
    tf.keras.layers.MaxPooling2D(pool_size=(1, 2)),
    tf.keras.layers.Conv2D(128, 3, activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(pool_size=(1, 2)),
    tf.keras.layers.Conv2D(128, 3, activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(pool_size=(1, 2)),
    tf.keras.layers.Conv2D(256, 3, activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(pool_size=(1, 2)),
    tf.keras.layers.Conv2D(256, 3, activation='relu', padding='same'),
    tf.keras.layers.GlobalMaxPooling2D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(3, activation='softmax')
])
```

Slika 3.3. Model CNN

Konvolucijski sloj na početku uzima kao argument manji broj filtara. Ti filtri se postupno povećavaju, počevši od 64, zatim 128 i naposljetku 256 filtera. S obzirom na to da se u svakom sloju provodi konvolucija, u prvim slojevima se izdvajaju sitni detalji iz značajki. Kasniji slojevi formiraju širu sliku, a zbog mnogo različitih kombinacija tih detalja, nužno je povećavati broj filtera. Uz broj filtera, navodi se i veličina jezgre koja je potrebna za konvoluciju.

Također je potrebno kao parametar navesti aktivacijsku funkciju. Duboko učenje obavlja vrlo zahtjevne zadatke, koji se ne mogu riješiti tradicionalno algoritamski i kojima je raspon podataka nelinearan. Zato aktivacijske funkcije imaju ključnu ulogu u skrivenim slojevima jer unose nelinearnost u model, čime se postiže veća preciznost i točnost [12]. Postoji više aktivacijskih funkcija, a ovdje su korištene ReLU i Softmax. ReLU (engl. *Rectified linear unit*) je najpopularnija, prvenstveno zbog toga što je jednos-

tavna i brza za izračunavanje. Ona vraća maksimum funkcije ili nulu ako je maksimum negativan. U zadnjem sloju je korištena aktivacijska funkcija Softmax, koja je oblik sigmoidne funkcije. Izvrsna je za klasifikaciju jer normalizira i raspoređuje vrijednosti za svaku klasu.

3.4. Učenje modela

Neposredno prije učenja modela, potrebno je postaviti njegovu konfiguraciju, slika 3.4. Čine ga optimizator, funkcija gubitka i metrika. U ovom slučaju, za optimizaciju se koristi optimizator Adam (engl. *Adaptive Moment Estimation optimizer*). Adam dinamički izračunava stopu učenja za svaku težinu, a računa se na temelju prethodnih vrijednosti [13]. Time se optimiraju težine unutar neuronske mreže, što rezultira poboljšanom brzinom učenja. Glavna uloga optimizatora je minimizacija greške i prilagodba parametara modela.

Funkcija gubitka mjeri razliku između stvarnih izlaza i njihovih predikcija. Poželjno je da razlika bude što manja. Za funkciju gubitka koristi se kategorička unakrsna entropija za rijetke podatke (*sparse_categorical_crossentropy*), jer su oznake cjelobrojne, a ne jednojedinичno kodirane (engl. *One-Hot Encoding*).

```
modelCNN.compile(  
    optimizer=tf.keras.optimizers.Adam(),  
    loss = 'sparse_categorical_crossentropy' ,  
    metrics = ['accuracy'])  
  
EPOCHS = 30  
history = modelCNN.fit(  
    train_img,  
    train_var,  
    validation_data = (val_img, val_var),  
    epochs=EPOCHS,  
    callbacks=tf.keras.callbacks.EarlyStopping(verbose=1, patience=2),  
)
```

Slika 3.4. Optimizacija i učenje modela

Metrika je mjera za praćenje pouzdanosti modela. Također je značajan i broj epoha u procesu učenja. Epoha je varijabla koja određuje koliko će se puta algoritam izvesti nad

cijelim skupom podataka. Veličina grupe (engl. *batch size*) je količina uzoraka koja se šalje kroz mrežu prije nego što se ažuriraju parametri.

U ovom radu, model uči nad 30 epoha, a veličina grupa je po zadanom odnosno 32. Na slici 3.5. vidi se kako model iz epohe u epohu uči odnosno točnost se povećava, a gubitak se smanjuje. Uz pomoć funkcije *save_model* iz knjižnice Tensorflow, model se sprema kako bi se kasnije mogao implementirati u web stranicu.

```
Epoch 1/30
29/29 [=====] - 31s 510ms/step - loss: 0.9808 - accuracy: 0.5833 - val_loss: 0.8623 - val_accuracy: 0.6071
Epoch 2/30
29/29 [=====] - 7s 248ms/step - loss: 0.9197 - accuracy: 0.6107 - val_loss: 0.8802 - val_accuracy: 0.6071
Epoch 3/30
29/29 [=====] - 7s 238ms/step - loss: 0.8680 - accuracy: 0.6162 - val_loss: 0.7896 - val_accuracy: 0.6071
Epoch 4/30
29/29 [=====] - 7s 241ms/step - loss: 0.7719 - accuracy: 0.6524 - val_loss: 0.7103 - val_accuracy: 0.6786
Epoch 5/30
29/29 [=====] - 7s 247ms/step - loss: 0.6515 - accuracy: 0.7259 - val_loss: 0.5750 - val_accuracy: 0.7806
Epoch 6/30
29/29 [=====] - 7s 250ms/step - loss: 0.5749 - accuracy: 0.7204 - val_loss: 0.5955 - val_accuracy: 0.7653
Epoch 7/30
29/29 [=====] - 7s 248ms/step - loss: 0.5556 - accuracy: 0.7445 - val_loss: 0.5709 - val_accuracy: 0.7755
Epoch 8/30
29/29 [=====] - 7s 252ms/step - loss: 0.4722 - accuracy: 0.7961 - val_loss: 0.4286 - val_accuracy: 0.8265
Epoch 9/30
29/29 [=====] - 7s 251ms/step - loss: 0.4537 - accuracy: 0.8158 - val_loss: 0.6081 - val_accuracy: 0.7959
Epoch 10/30
29/29 [=====] - 7s 245ms/step - loss: 0.3901 - accuracy: 0.8355 - val_loss: 0.3382 - val_accuracy: 0.8622
```

Slika 3.5. Učenje modela

3.5. Analiza rezultata

Tijekom evaluacije rezultata, postignuta je točnost (engl. *accuracy*) od 91,84% i gubitak (engl. *loss*) od 20,57%. Točnost predstavlja omjer točnih predikcija u odnosu na ukupan broj podataka koji se predviđaju.

Uz mjeru točnosti koja se dobila prilikom testiranja podataka, važno je uzeti u obzir nebalansiranost skupa podataka. Zato se računa mjera F1 (engl. *F1-score*) kako bi se dobio detaljniji uvid u rezultate modela. Mjera F1 pokazuje točnost modela, a bitna je kada je skup podataka nebalansiran [14]. Kombinira preciznost (engl. *precision*) i odziv (engl. *recall*), koristeći harmonijsku sredinu. S obzirom da je korišten skup podataka izrazito nebalansiran, bilo je potrebno isprobati različite pristupe obradi podataka i modele mreže.

```

7/7 [=====] - 0s 65ms/step - loss: 0.2057 - accuracy: 0.9184
Loss: 0.2057
Accuracy: 0.9184
7/7 [=====] - 1s 62ms/step
F1-score: 0.8740

```

	precision	recall	f1-score	support
0	0.68	0.83	0.75	18
1	0.96	0.92	0.94	114
2	0.92	0.94	0.93	64
accuracy			0.92	196
macro avg	0.86	0.90	0.87	196
weighted avg	0.92	0.92	0.92	196

Slika 3.6. Rezultati testiranja

Prvotno su se testirale klase s najvećim udjelom podataka (KOPB, zdrava osoba, URTI), no isprobavanjem različitih klasa, pokazalo se da model najbolje izdvaja značajke za klase KOPB-a, bronhiektazije i zdravih osoba. Također je bilo potrebno odrediti optimalan broj iteracija za augmentaciju podataka u manjinskim klasama. Ispitivane su različite metode ekstrakcije značajki, što je rezultiralo zamjenom mel-spektrograma običnim spektrogramom uz kombiniranje ostalih audio značajki. Testirani su i različiti hiperparametri u izgradnji arhitekture mreže. To uključuje isprobavanje različitih broja slojeva te veličina i broja jezgri.

Kao rezultat kombiniranja i isprobavanja različitih mogućnosti, mjera F1 doseže 0,87. Na slici 3.6. se također vidi iznos mjere F1 za svaku klasu te vidimo da unatoč tomu što je skup podataka nebalansiran, uspješno su izdvojene i učene značajke za svaku klasu. Ukoliko bi se u model uvelo svih šest klasa, bilo bi potrebno prikupljati dodatne podatke za ostale klase.

4. Web aplikacija

4.1. Opis i izgled web aplikacije

Da bi liječnici mogli jednostavno rukovati modelom klasifikacije zvučnih zapisa, izrađena je web stranica. Sastoji se od poslužiteljske i klijentske strane. Stranica ima mogućnost učitavanja zvučnog zapisa u formatu .wav, slika 4.1. Klikom na gumb 'Pošalji', audio zapis se obrađuje i klasificira te se prikazuje njena predikcija.



Slika 4.1. Početna stranica

Priložene su slike web aplikacije prije (slika 4.1.) i poslije (slika 4.2.) predikcije.



About

Ispravna i pravovremena dijagnoza respiratornih bolesti važna je za početak liječenja. Respiratorne bolesti moguće je dijagnosticirati temeljem zvučnih zapisa auskultacije digitalnim stetoskopom. Zbog prisutnih smetnji u zapisima i nedostatka vremena, liječnicima bi od značajne pomoći bio automatizirani sustav koji bi visokotočno klasificirao respiratorne bolesti temeljem zvučnih zapisa. Upravo iz ovih razloga, ovdje možete testirati model strojnog učenja.

@FER - Završni rad

Slika 4.2. Prikaz predikcije unutar web aplikacije

4.2. Tehnologije i alati

Za izgradnju poslužiteljske strane korišten je programski jezik Python i radni okvir Flask, a za klijentsku stranu korišten je prezentacijski jezik HTML (engl. *HyperText Markup Language*). Flask je izabran radi jednostavne uporabe i lake integracije modela strojnog učenja s web aplikacijom. Također, za ugađen i estetičan izgled stranice korišten je CSS (engl. *Cascading Style Sheets*), dok je za dinamičnost stranica korišten JavaScript.

4.3. Implementacija

Klijentska strana je izrađena tako da sadrži jednostavnu navigacijsku traku, spremnik koji sadrži ulaz za zvučni zapis i sliku, kontejner za opis stranice i podnožje (engl. *footer*). Dio kôda (slika 4.3.) koji kao ulaz prima datoteku istovremeno provjerava je li datoteka u formatu .wav. Osim toga, odabirom audio datoteke, funkcija *updateFileName()* prikazuje ime datoteke koja je odabrana. Klikom na gumb 'submit', aktivira se funkcija *submitForm()* i forma se šalje na poslužiteljsku stranu pomoću metode *fetch*.

```
<header class="main wide" id="home">
  <div class="grid-container dark-blue">
    <div class="input-container">
      <label for="fileInput">Odaberi zvučni zapis:</label>
      <input type="file" id="fileInput" name="fileInput" accept="audio/wav" onchange="updateFileName()">
      <input type="submit" value="Pošalji" onclick="submitForm()">
      <output id="outputField"></output>
    </div>
    
  </div>
</header>
```

Slika 4.3. Klijentska strana

Na poslužiteljskoj strani, koristi se ruta `/predict` za obradu zahtjeva HTTP POST, slika 4.4. Prvo se učitava zvučna datoteka koja je poslana u zahtjevu, a za njenu obradu koristi se knjižnica Librosa. Zatim se iz zvučnog zapisa izdvajaju značajke pomoću funkcije *extract_features_2d*. Značajke se izdvajaju jednako kao i u predobradi podataka, poglavlje 2.4. Jednom kada se izdvoje značajke, potrebno je pretvoriti izlaz u NumPy niz (metoda *array*).

Model se prethodno dohvaća pomoću funkcije *load_model*. Funkcijom *predict*, model vraća predikciju za svaku od izlaznih klasa. Korištenjem *argmax*, odabire se ona klasa s najvećom vjerojatnošću, pri čemu se vraća indeks predviđene klase. Kako bi se ispisao naziv izlazne klase, uz pomoć *label_encoder*, povezuje se indeks klase s pripadnom oznakom.

Uz naziv predviđene klase, izračunava se i vjerojatnost da je odabrana klasa istinita. Izražava se mjerom preciznosti (engl. *confidence score*). Naposljetku, iznos vjerojatnosti se zaokružava na pet decimala. Odgovor na zahtjev se vraća u obliku JSON formata.

```
@app.route('/predict', methods=['POST'])
def predict():
    file = request.files['audio']
    audio_data, sample_rate = librosa.load(file)

    output = extract_features_2d(audio_data, sample_rate)
    output = np.array([output])

    prediction = model.predict(output)

    predicted_class = np.argmax(prediction, axis=1)[0]
    predicted_label = label_encoder.inverse_transform([predicted_class])[0]

    predicted_probability = float(prediction[0][predicted_class])
    probability = round(predicted_probability, 5)

    return jsonify({
        'prediction': predicted_label,
        'probability': probability
    })

if __name__ == '__main__':
    app.run(debug=True)
```

Slika 4.4. Poslužiteljska strana

5. Zaključak

Cilj ovog rada je izgradnja modela strojnog učenja kojim je moguće klasificirati respiratorne bolesti na temelju zvučnih zapisa. U radu se opisuje predobrada zvučnih datoteka. Također je objašnjena teorija dubokog učenja i konvolucijskih neuronskih mreža. Detaljno je opisan postupak izgradnje modela te proces implementacije istog u web stranicu.

Kroz ovaj rad sam shvatila koliko je bitno kvalitetno pripremiti skup podataka za učenje modela. U moru podataka, bitno je izdvojiti one koje su značajne i korisne za daljnju uporabu. Uz to, ključno je isprobavati različite hiperparametre modela kako bi se postigao što precizniji rezultat. Tijekom izrade ovog rada, stekla sam široko znanje o umjetnoj inteligenciji, posebice o dubokom učenju. Također, matematička podloga koju sam stekla tijekom svog obrazovanja mi je uvelike pomogla u razumijevanju algoritama i principa strojnog učenja. Smatram da je područje umjetne inteligencije izuzetno korisno i zanimljivo, s velikim potencijalom za daljnji razvoj.

Također, područje biomedicine me oduvijek zanimalo i motivirana sam nastaviti istraživati u tom smjeru kao inženjer. Smatram da je spoj programiranja i medicine iznimno koristan i da se može još razvijati. Danas postoje brojne primjene tehnologije u medicini, poput obrade slika, razvoja medicinskih uređaja, robotike i slično.

Literatura

- [1] Respiratory Sound Database, (29.siječnja, 2019), <https://www.kaggle.com/datasets/vbookshelf/respiratory-sound-database/data>, Pristupljeno 20.02.2024.
- [2] Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2019.
- [3] Simon J.D. Prince, *Understanding Deep Learning*, 2023.
- [4] Wikimedia, D. P. (29.siječnja, 2024). Dišni sustav https://hr.wikipedia.org/wiki/Di%C5%A1ni_sustav, Pristupljeno 21.04.2024.
- [5] *Prevalence and attributable health burden of chronic respiratory diseases, 1990–2017*, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7284317>, Pristupljeno 10.05.2024.
- [6] Littmann CORE digitalni stetoskop. <https://www.medicaldirect.hr/littmann-core-digitalni-stetoskop.html>, Pristupljeno 01.06.2024.
- [7] Hallaj, P. (21. rujna, 2023). Data augmentation in Python <https://medium.com/@pouyahallaj/data-augmentation-benefits-and-disadvantages-38d8201aead>, Pristupljeno 20.05.2024.
- [8] Ma, E. (10. prosinca, 2021). Data augmentation for audio <https://medium.com/@makcedward/data-augmentation-for-audio-76912b01fdf6>, Pristupljeno 20.05.2024.

- [9] GeeksforGeeks. (24. listopada, 2023). Audio classification using spectrograms. https://www.geeksforgeeks.org/audio-classification-using-spectrograms/?ref=ml_lbp, Pristupljeno 25.05.2024.
- [10] B, A. (30. ožujka, 2022). Audio signal feature extraction for analysis - Athina B Medium.<https://athina-b.medium.com/audio-signal-feature-extraction-for-analysis-507861717dc1>, Pristupljeno 30.04.2024.
- [11] Nandi, P. (2. kolovoza, 2022). CNNs for audio Classification - towards data science. Medium. <https://towardsdatascience.com/cnns-for-audio-classification-6244954665ab>, Pristupljeno 20.04.2024.
- [12] GeeksforGeeks. (3. ožujka, 2024). Activation functions in Neural Networks. <https://www.geeksforgeeks.org/activation-functions-neural-networks/>, Pristupljeno 28.05.2024.
- [13] Gupta, A. (23. siječnja, 2024). A comprehensive guide on optimizers in deep learning. <https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/>, Pristupljeno 28.05.2024.
- [14] Kundu, R. (10. travnja, 2024). F1 Score in Machine Learning: Intro and Calculation. <https://www.v7labs.com/blog/f1-score-guide>, Pristupljeno 29.05.2024.
- [15] Wikimedije, D. P. (27. travnja, 2024). Zvuk. <https://hr.wikipedia.org/wiki/Zvuk>, Pristupljeno 1.6.2024.
- [16] GeeksforGeeks. (26. svibnja, 2024). Introduction to deep learning. <https://www.geeksforgeeks.org/introduction-deep-learning/>, Pristupljeno 1.6.2024.
- [17] News-Medical. (16. ožujka, 2023). What can COVID-19 Teach us about Respiratory Disease? <https://www.news-medical.net/health/What-can-COVID-19-Teach-us-about-Respiratory-Disease.aspx>, Pristupljeno 1.6.2024.
- [18] FER. Umjetne Neuronske Mreže. [https://www.fer.unizg.hr/_download/repository/UI_12_UmjetneNeuronskeMreze\[1\].pdf](https://www.fer.unizg.hr/_download/repository/UI_12_UmjetneNeuronskeMreze[1].pdf), Pristupljeno 12.5.2024.

- [19] Pliva Zdravlje. Astma. <https://www.plivazdravlje.hr/bolest-clanak/bolest/191/Astma.html>, Pristupljeno 28.5.2024.
- [20] Pliva Zdravlje. Upala pluća. <https://www.plivazdravlje.hr/bolest-clanak/bolest/108/Upala-pluca.html>, Pristupljeno 28.5.2024.
- [21] Mišković, A. (3. veljače, 2020). Bronhiektazije – uzroci, simptomi, dijagnoza i liječenje . <https://krenizdravo.dnevnik.hr/zdravlje/bolesti-zdravlje/bronhiektazije-uzroci-simptomi-dijagnoza-i-lijecenje>, Pristupljeno 30.5.2024.
- [22] Pliva Zdravlje. Astma. <https://www.plivazdravlje.hr/aktualno/clanak/27604/KOPB-simptomi-dijagnoza-i-lijecenje.html>, Pristupljeno 30.5.2024.
- [23] Convolutional neural network structure. Illustration of the CNN. https://www.researchgate.net/figure/Convolutional-neural-network-structure-Illustration-of-the-CNN-architecture-employed-in_fig8_363068363, Pristupljeno 5.6.2024.

Sažetak

Ovaj rad opisuje implementaciju modela strojnog učenja u jednostavnu web stranicu. Model klasificira respiratorne bolesti na temelju zvučnih zapisa auskultacije pluća snimljene digitalnim stetoskopom. Koristi se arhitektura konvolucijskih neuronskih mreža. Također je prikazana predobrada i izdvajanje značajki iz audio datoteka. Rezultati su dani u obliku točnosti, preciznosti, odziva i mjere F1. Postignuta je točnost od 91,84% s mjerom F1 od 87,4%.

Ključne riječi: klasifikacija, model, učenje, zvučni zapis, skup podataka, respiratorne bolesti

Abstract

This Bachelor's thesis describes the implementation of a machine learning model and its integration into a simple web page. The model classifies respiratory diseases based on sound recordings from lungs auscultation measured by a digital stethoscope. Convolutional neural networks are used as a model. Preprocessing and feature extraction from audio files are also presented. The results are given in the form of accuracy, precision, recall and F1-score. An accuracy of 91,84% and F1-score of 87,4% were achieved.

Keywords: classification, model, training, sound recording, dataset, respiratory diseases