VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
PROGRAMŲ SISTEMŲ BAKALAURO STUDIJŲ PROGRAMA

Emocijų atpažinimas balse naudojant neuroninius tinklus ir signalų spektrogramas

Voice emotion recognition using deep neural networks and signal spectrograms

Bakalauro baigiamasis darbas

Atliko: Mykolas Skrodenis (parašas)

Darbo vadovas: doc. Jonas Matuzas (parašas)

Darbo recenzentas: Tadas Žvirblis (parašas)

Vilnius – 2023

Santrauka

Šiame darbe nagrinėjama balso emocijų atpažinimo problema naudojant signalų spektrogramomas iš anksto apmokytas konvoliucinių neuroninių tinklų modelių architektūras. Ypatingas dėmesys skiriamas septynioms pagrindinėms emocijoms: baimė, džiaugsmas, liūdesys, neutrali būsena, nuostaba, pasišlykštėjimas ir pyktis. Apžvelgiami paprasčiausi neuroniniai tinklai. Pasigilinama į spektrogramas ir duomenų apdorojimo procedūras ir koncoliucinių neuroninų tinklų architektūras. Aprašoma tyrimo eiga, naudotos architektūros įvertinamos naudojant tikslumo, precizižkumo, atkūrimo ir F1 metrikas. Analizuojami darbo rezultatai. Lyginamos tirtos architektūros ir daromos išvados. Pasiūlomi būdai, kuriais, autoriaus nuomone, galima būti pasiekti geresnius rezultatus.

Raktiniai žodžiai: spektrograma, koncoliucinis neuroninis tinklas, modelio architektūra, emocijos

Summary

This work addresses the problem of voice emotion recognition using signal spectrograms in pre-trained convolutional neural network model architectures. Particular attention is paid to seven basic emotions: fear, happyness, sadness, neutral, surprise, disgust and anger. The simplest neural networks are reviewed. Spectrograms and data processing procedures and convolutional neural network architectures are discussed. The research process is described and the architectures used are evaluated using accuracy, precision, recall and F1 metrics. The results are analysed. Comparisons of the architectures studied are made and conclusions are drawn. the author suggest method that he believes could achieve better results.

Keywords: spectrogram, convolutional neural network, model architecture, emotions

TURINYS

Įvadas

Gebėjimas atpažinti ir interpretuoti emocijas yra labai svarbus žmonių bendravimui ir socialinei sąveikai. Emocijos perteikia svarbią informaciją apie asmens psichinę būseną, reakcijas, ketinimus ir yra labai svarbios bendravimui, empatijai bei ryšiui užmegzti. Tačiau atpažinti ir interpretuoti emocijas ne visada lengva, ypač kai verbaliniai ir neverbaliniai signalai yra dviprasmiški, prieštaringi ar nenuoseklūs. Bendraujant tiesiogiai ar netiesiogiai, net ir nebūnant jokiam komunikaciniam vaizdui, žmogaus balsas perteikia perduodamą turinį, ir kartu dažnai išreiškia asmens emocinę būseną [TDL+22]. Tobulėjant technologijoms platėja ir galimybės kur jas galima pritaikyti. Šiuo metu viena labiausiai vystomų ir plėtojamų informatikos mokslo šakų yra gilusis mokymas (angl. deep learning). Tai sritis, kurioje naudojami dirbtiniai neuronų tinklai (angl. neural networks) ir kurioje siūlomi daugelio problemų sprendimai, atsiradę plėtojantis technologijoms [ÇN21].

Emocijų atpažinimas iš balso yra perspektyvi technologija, kuria siekiama automatiškai atpažinti emocijas asmens balse naudojant giliojo mokymosi ir signalų apdorojimo metodus. Balso emocijų atpažinimas gali būti įvairiai taikomas tokiose srityse kaip sveikatos priežiūra, švietimas, pramogos ir saugumas, pavyzdžiui, nustatant ir stebint emocinius sutrikimus, tobulinant kalbos mokymąsi, tobulinant virtualiuosius agentus ir pokalbių robotus, nustatant apgaulingą ar įtartiną elgesį. Tačiau balso emocijų atpažinimo užduotis yra sudėtinga dėl kelių veiksnių, pavyzdžiui, balso aukščio, tono, ritmo ir dialekto kintamumo, taip pat emocijų išraiškos ir intensyvumo skirtumų.

Problemą galima išspręsti giliųjų neuroninių tinklų ir signalų spektrogramų pagalba. Šis perspektyvus balso emocijų atpažinimo metodas gali išmokti sudėtingas garso signalų reprezentacijas, užfiksuoti laiko ir dažnių modelius, kurie yra svarbūs emocijoms atpažinti.

Šio darbo tikslas - sukurti giliuoju neuroniniu tinklu pagrįstą balso emocijų atpažinimo modelį, kuris gali tiksliai ir patikimai atpažinti septynias pagrindines emocijas iš garso signalų, naudodamas signalo spektrogramas kaip įvesties požymius. Klasifikuojamos emocijos yra: baimė, džiaugsmas, liūdesys, neutrali, nuostaba, pasišlykštėjimas ir pyktis. Siekiama išbandyti ir įvertinti įvairias iš anksto apmokytų modelių architektūras.

Norint pasiekti darbo tikslą, reikia įgyvendinti šiuos uždavinius:

- Išanalizuoti jau esamą literatūrą apie emocijų atpažinimą balse naudojant neuroninius tinklus. Išsiaiškinti naujausias žinias apie balso emocijų atpažinimą naudojant giliuosius neuroninius tinklus ir signalų spektrogramas. Įvertinti rekomenduojamų metodų tinkamumą turimam uždavinio sprendimui.
- Sutvarkyti naudojamų duomenų rinkinį suvienodinant garso signalų ilgį.
- Sukurti giliuoju mokymusi pagrįstą balso emocijų atpažinimo modelį naudojant signalų spektrogramas. Išbandyti įvairias iš anksto apmokytas architektūras.
- Pritaikyti augmentaciją turimiems duomenims.
- Įvertinti tiriamų modelių veikimą naudojant įvairius vertinimo rodiklius, tokius kaip tikslumas, preciziškumas, atkūrimo statistika, F1 rodiklis, klasifikavimo lentele.

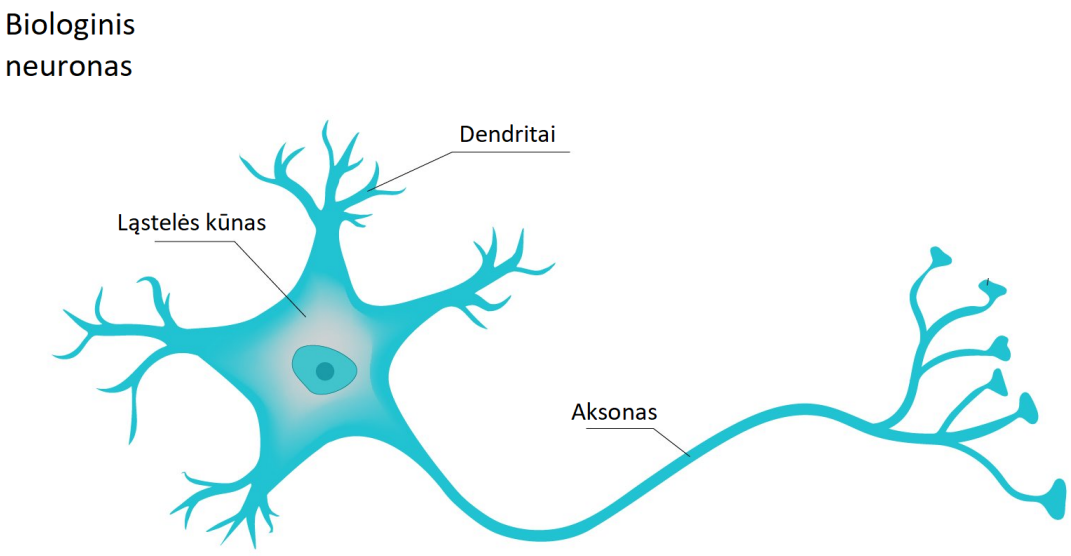
Siekiami šio darbo rezultatai:

- Giliuoju mokymusi pagrįstas modelis, kuris gali atpažinti emocijas iš balso signalų.
- Modelių lyginamoji analizė su visomis tirtomis architektūromis.
- Efektyviausių giliųjų neuroninių tinklų architektūrų, skirtų emocijų atpažinimui balsu, nustatymas naudojant signalų spektrogramas.

Neuroniniai tinklai – tai mašininio mokymosi modelio tipas, sukurtas remiantis žmogaus smegenų struktūra ir funkcijomis. Juos sudaro tarpusavyje sujungti ”neuronai”, kurie gali apdoroti ir perduoti informaciją, jie geba mokytis bei prisitaikyti prie naujų duomenų per procesą, vadinamą mokymu. Norint apmokyti neuroninį tinklą atlikti tam tikrą užduotį, tiksliai, su nedidele klaidų tikimybe, gali prireikti daug mokymo duomenų, daug laiko ir kompiuterinių resursų, tačiau nauda gali būti didžiulė.

1. Neuroniniai tinklai

Žmogaus smegenys sudarytos iš milijardų neuronų, kurie kartu apdoroja ir interpretuoja pojūčių informaciją, generuoja mintis ir elgesį bei valdo kūno funkcijas. Neuronas yra pagrindinis nervų sistemos komunikacijos vienetas, pagrindinė nervų sistemos ląstelė. Kiekvienas neuronas sudarytas iš ląstelės kūno, kelių dešimčių dendritų ir aksono [1]. Ląstelės kūne yra branduolys kuriame užkoduota genetinė informacija, citoplazma ir kitos organėlės. Neurone esantys dendritai priima elektrinius signalus iš kitų neuronų, o ilgesnis, vientisas ląstelės kūno tęsinys vadinamas aksonu, leidžia siųsti elektrinius signalus [Woond].



1 pav. Biologinio neurono pavyzdys [Vad21]

Neuronai ne tik perduoda informaciją, bet ir gali keisti savo ryšius su kitais neuronais, remdamiesi patirtimi. Šis procesas, vadinamas plastiškumu, leidžia smegenims mokytis ir prisitaikyti laikui bėgant. Pavyzdžiui, kai žmogus išmoksta kažką naujo, įgyja naują įgūdį, ryšiai tarp su tuo įgūdžiu susijusių neuronų sustiprėja, todėl ateityje jis gali lengviau atlikti tą įgūdį.

Pagal šį žmogaus smegenų veikimo principą, kuriami kompiuterizuoti neuronai, galintys spręsti įvairiausias problemas. Tai matematinės funkcijos atkartojančios biologinių neuronų veikimą.

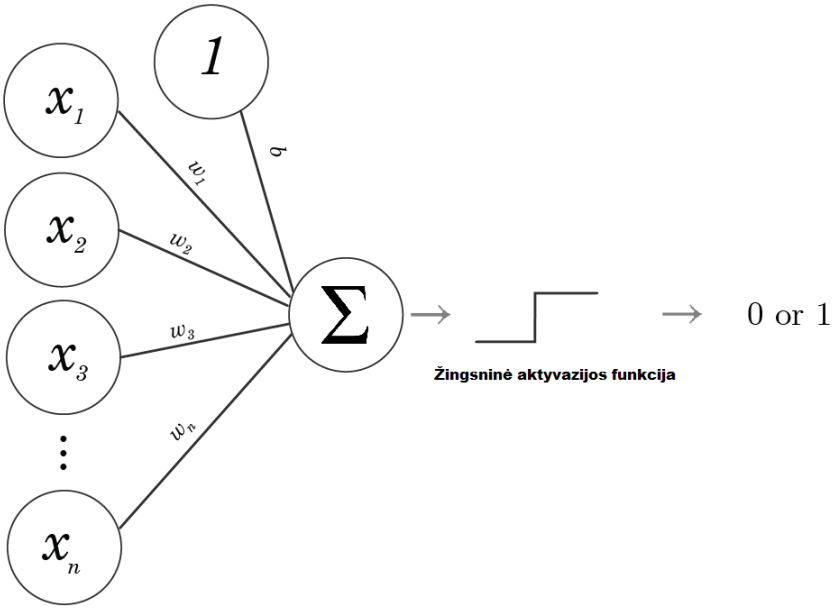
1.1. Perceptronas

Perceptronas yra dirbtinis neuronas, kuris buvo vienas iš pirmųjų sukurtų mašininio mokymosi modelių, panaudotas Franko Rosenblatto tyrimuose 1957 metais [AON+09]. Tai labai paprastas modelis, tačiau jis gali būti naudingas duomenų klasifikavimo problemoms spręsti, kai duomenys yra suskirstyti į dvi kategorijas.

Perceptrono modelį sudaro vienas dirbtinis neuronas, kuriam būdingi svoriai ir laisvasis

narys (angl. bias). Šie parametrai yra koreguojami mokymo metu, siekiant išmokti tinkamai atskirti dvi kategorijas. Svoriai nurodo, kiek kiekvienas įvesties požymis įtakoja galutinį rezultatą, o laisvasis narys leidžia modeliui veikti lanksčiau, nes jis gali būti pritaikytas prie skirtingų duomenų savybių.

Perceptrono veikimo principas yra gana paprastas. Norėdamas klasifikuoti įvestį, perceptronas apskaičiuoja įvesties požymių svertinę sumą, pridedant laisvąjį narį. Tai reiškia, kad kiekvienas įvesties požymis yra sudauginamas su atitinkamu svoriu, o rezultatai yra sudedami kartu ir prie galutinės sumos pridedamas laisvasis narys.



2 pav. Perceptrono modelio atvaizdavimas su binarine aktyvacijos funkcija

Šiame procese (2 paveikslėlyje pavaizduotas perceptrono modelis) įvesties duomenys x1, x2 ... xn yra svertai w1, w2 ... wn, priskirti atitinkamoms įvestims, ir b yra laisvasis narys.

$$net = \sum_{i=1}^{n}(w_i * x_i) + b$$

Tada rezultatas yra praeinamas per aktyvavimo funkciją, kuri gali būti paprasta binarinė funkcija, leidžianti įvertinti, ar įvestis priklauso vienai ar kitai kategorijai. Jei rezultatas, gautas po aktyvavimo funkcijos taikymo, yra 1, tai reiškia, kad įvestis priklauso antrajai kategorijai.

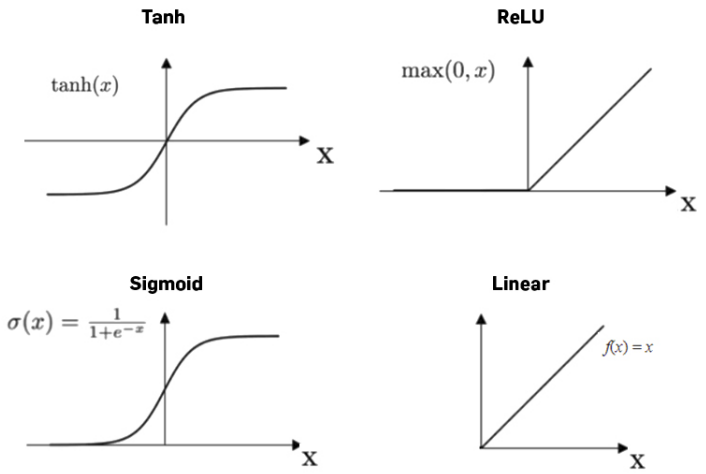
$$f(x) = \begin{cases} 1, & \text{jei } net \geq 1 \\ 0, & \text{kitu atveju} \end{cases}$$

Nepaisant savo paprastumo, perceptrono idėja padėjo pagrindą tolesniam dirbtinių neuronų tinklų ir kitų galingesnių mašininio mokymosi modelių vystymui. Šiandien daugelis šiuolaikinių mašininio mokymosi sistemų, pvz., daugiasluoksnių perceptronų ar konvoliucinių neuroninių tinklų, yra pagrįsti panašiais principais, nors jie yra žymiai sudėtingesni ir gali spręsti daug platesnį uždavinių spektrą.

Perceptronai taip pat yra naudingi pradedantiesiems mašininio mokymosi mokslininkams ir inžinieriams, nes jie suteikia aiškų supratimą apie pagrindinius mašininio mokymosi principus ir procesus. Jų paprastumas leidžia greitai suprasti, kaip veikia dirbtiniai neuronai ir kaip jie gali būti mokomi sprendžiant realaus pasaulio problemas. Nors perceptronas gali būti pernelyg ribotas sudėtingesniems uždaviniams spręsti, jis vis tiek yra puikus pagrindas tolesniam mašininio mokymosi tyrinėjimui.

1.2. Aktyvacijos funkcijos

Aktyvavimo funkcijos yra vienas iš svarbiausių dirbtinių neuroninių tinklų komponentų. Jos atlieka lemiamą vaidmenį paverčiant įvesties duomenų svertinę sumą išvesties rezultatu, kuris naudojamas prognozavimui arba klasifikavimui. Yra daug aktyvacijos funkcijų, įskaitant tiesinę, ReLU, sigmoidinę, ELU, tanh ir kitas. [KNS+21]



3 pav. Aktyvacijos funkcijų pavyzdžiai [AI]

- **Sigmoidinė aktyvavimo funkcija:** Sigmoidinė aktyvavimo funkcija bet kokią reikšmę gali atvaizduoti į intervalą nuo 0 iki 1. Dėl to ją patogu naudoti modeliuose, kai vienas iš išėjimų yra prognozuoti tikimybę.

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

- **Tanh:** $\tanh(x)$ Atvaizduoja tik reikšmes intervale $(-1, 1)$. Ji dažnai naudojama palaidotuose neuroninio tinklo sluoksniuose ir užtikrina didesnius gradientus didelėms įvesties reikšmėms.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

- **Dalimis tiesinis vienetas(Relu):** Funkcija grąžina 0, jei įvestis yra neigiama, o bet kokia teigiama x reikšmė yra grąžinama.

$$relu(x) = max(0, x)$$

- **Tiesinė (angl. linear):** Tiesinė funkcija, dar vadinama tapatybės funkcija, yra funkcija, kuri nekeičia įvesties

$$f(x) = x$$

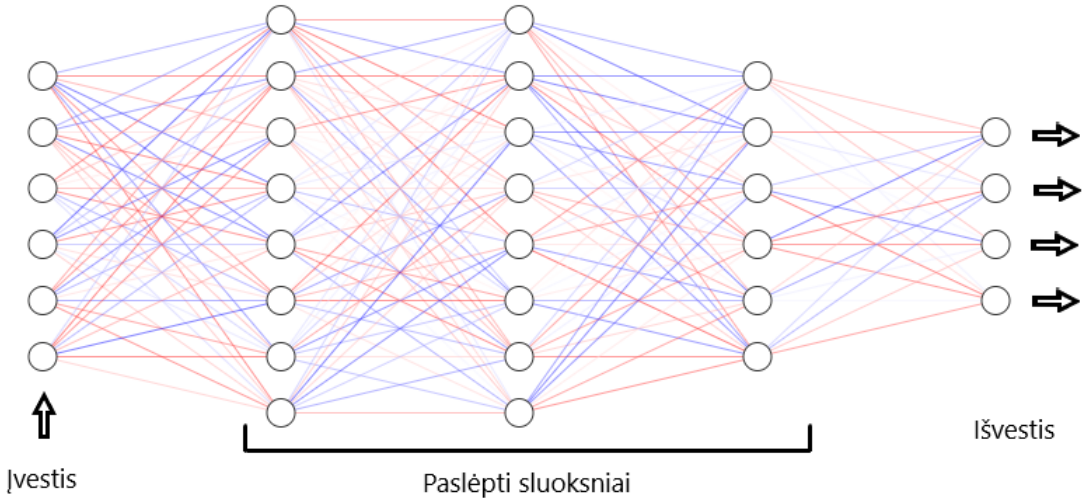
  Ši funkcija grąžina įvestį tokią, kokia ji yra, ir nekeičia jos elgsenos.
- **Eksponentinis normalizavimas (Softmax funkcija):** Neuroninio tinklo išėjimai $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ įvedami į aktyvavimo funkciją, kuri sukuria tikimybių pasiskirstymą.

$$\text{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}, i = 1, 2, \ldots, N$$

Aktyvinimo funkcijos atlieka esminį vaidmenį neuroninių tinklų efektyvumui, daro įtaką jų gebėjimui išmokti sudėtingus modelius ir ryšius. Šios funkcijos, į tinklo sprendimų priėmimo procesą įtraukdamos netiesiškumą, leidžia neuroniniams tinklams suprasti sudėtingus netiesinius duomenų ryšius.

1.3. Gilusis neuroninis tinklas

Giliojo neuroninio tinklo struktūra yra sudėtingesnė nei paprastų dirbtinių neuroninių tinklų (ANN) struktūra, nes ji apima keletą tarpusavyje sujungtų dirbtinių neuronų sluoksnių. Dėl savo sudėtingumo ir gebėjimo efektyviai mokytis bei apibendrinti duomenis, gilieji neuroniniai tinklai buvo naudojami pasiekiant puikius rezultatus įvairių užduočių sprendime, tokiose srityse kaip vaizdų klasifikavimas, mašininis vertimas, žaidimų strategija ir kt.

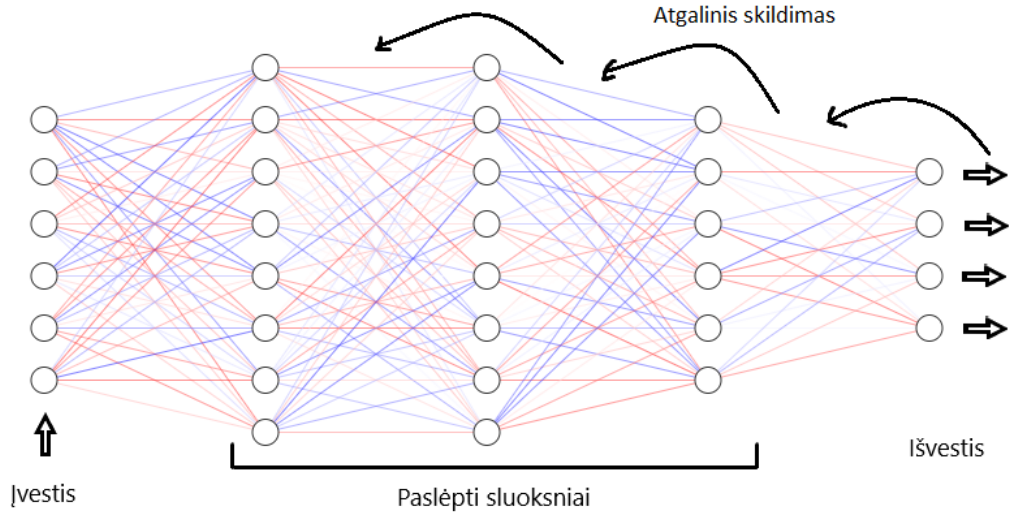


4 pav. Giliojo neuroninio tinklo modelio iliustracija

Sluoksniuotas apdorojimas yra gilaus neuroninio tinklo principo pagrindas, kai kiekvienas sluoksnis pradeda savo užduotį su gautais duomenimis ir perduoda juos sekančiam sluoksniui. Informacija juda iš įvesties sluoksnio į išvesties sluoksnį, nes sluoksniai išdėstyti vienas ant kito. Kiekviename sluoksnyje yra dirbtiniai neuronai, sujungti su tarpusavio sluoksniais. Šioms neuronų jungtims priskiriami svoriai, kurie per mokymą tiksliai sureguliuojami, kad būtų užtikrintas

optimalus tinklo veikimas. Skirtingi svorių dydžiai nurodo kiekvieno neurono individualų poveikį rezultatui.

Atgalinio skleidimo algoritmas (angl. backpropagation) yra esminis metodas, kuris naudojamas giliųjų neuroninių tinklų mokymui. Šis algoritmas yra neuroninio tinklo optimizavimo strategijos pagrindas. Konkrečiau, atgalinio skleidimo algoritmas veikia taip: pirmiausia, įvestis yra skleidžiama per visus tinklo sluoksnius, kiekvienas sluoksnis pritaiko savo svorius o paskui taiko aktyvavimo funkciją. Po šio etapo gaunamas numatomas išėjimas. Tada yra apskaičiuojama nuostolių funkcija, kuri nustato skirtumą tarp numatytų išėjimų ir tikrosios reikšmės. Šis skirtumas atspindi tinklo prognozavimo klaidą. Galų gale, šios klaidos yra atgaliai skleidžiamos per tinklą, pradedant nuo išėjimo sluoksnio ir baigiant įvesties sluoksniu. Šiuo metodu yra apskaičiuojami gradientai (arba pokyčiai) kiekvienam svoriui, atsižvelgiant į jų indėlį į bendrą klaidą. Šie gradientai yra naudojami svoriams atnaujinti taip, kad sumažėtų nuostolių funkcijos vertė. Šis procesas yra iteratyviai kartojamas per visą mokymo rinkinį, kol tinklo prognozavimo klaida tampa pakankamai maža.



5 pav. Atgalinio sklidimo iliustracija

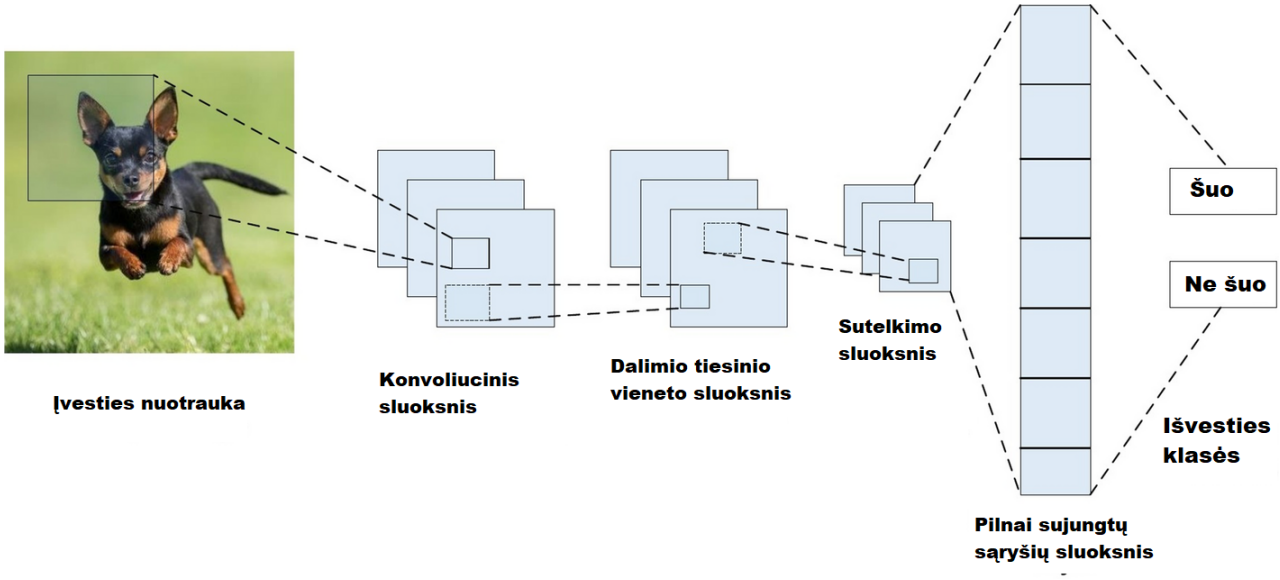
Tai sudėtingas procesas, bet jis yra pagrindas, kuris leidžia neuroniniams tinklams mokytis iš duomenų ir tobulinti savo prognozavimo gebėjimus. Sėkmingai apmokius neuroninį tinklą, jis gali būti naudojamas nepažintų duomenų prognozavimui, kas yra esminė daugumos šiuolaikinių dirbtinio intelekto aplikacijų dalis.

1.4. Konvoliucinis neuroninis tinklas

Konvoliucinis neuroninis tinklas (angl. „„, (CNN)) yra vienas iš giliųjų mokymosi modelių, kuris yra specifiškai projektuotas ir optimaliai pritaikytas vizualių duomenų, tokių kaip vaizdai ir video, apdorojimui. Šis modelis ypač plačiai naudojamas daugelyje sričių, pradedant nuo kompiuterinės regos (veido ir objektų atpažinimas, automatinis vaizdų generavimas), baigiant medicinos vaizdų ir satelitų nuotraukų analize.

Pagrindinis CNN bruožas, išskiriantis jį iš kitų neuroninių tinklų, yra jo gebėjimas automatiškai ir adaptatyviai mokytis hierarchines vaizdo savybes iš išvesties. Šis gebėjimas atsiranda

dėl dviejų pagrindinių CNN komponentų - konvoliucinių ir sutelkimo (angl. pooling) sluoksnių.



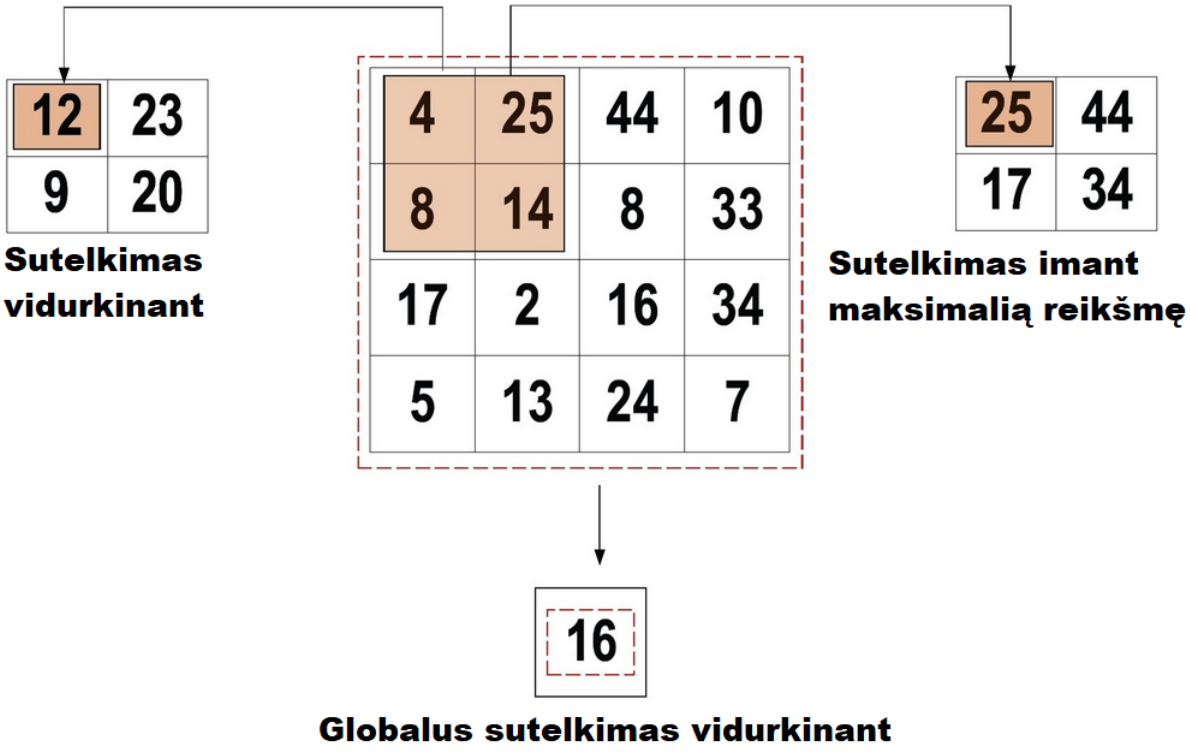
6 pav. Konvoliucinio neuroninio tinklo pavyzdys [AZH+21]

Konvoliucinis neuroninis tinklas paprastai susideda iš konvoliucinio sluoksnio su dalimio tiesinio vieneto (angl. Rectified Linear Unit ("Relu")) funkcija ir sutelkimo funkcijos. Konvoliuciniame sluoksnyje ieškomi atitinkantys šablonai ir svarbiems pikseliams suteikiama didesnė reikšmė. Pirmiausia parenkamas filtro branduolys (angl. kernel), jis nusako kokia paveikslėlyje atvaizduota savybė ieškoma. Filtrai yra maži vietiniai modeliai, kurie išmokstami iš duomenų ir naudojami tokioms įvesties vaizdo savybėms, kaip briaunoms, kampams ir tekstūroms, išgauti. Išvestimi gaunamas požymių žemėlapių rinkinys, kuriame išryškinami filtrais aptikti požymiai.

Prafiltruotai matricai pritaikoma dalimio tiesinio vieneto funkcija

$$f(x) = \max(0,x)$$

Panaudojus šia funkcija, visos nenaudingos langelių reikšmės yra nuliuojamos. Galiausiai pritaikoma sutelkimo funkcija. Pagrindinė sutelkimo funkcijos užduotis - atrinkti požymių žemėlapius [AZH+21]. Siekiama sumažinti gauta rastų požymių matrica neprarandant jau rastų požymių. Šis procesas sumažina neuroninio tinklo skaičiavimą nepakenkdamas rezultatui.



7 pav. Sutelkimo funkcojos pavyzdys [AZH+21]

Po konvoliucinių sluoksnių konvoliucinius neuroninius tinklus paprastai apima vieną ar daugiau visiškai sujungtų sluoksnių, kurie yra panašūs į tradicinio neuroninio tinklo sluoksnius [7]. Visiškai sujungti sluoksniai kaip įvestį priima konvoliucinių sluoksnių išvestį ir naudoja ją prognozėms atlikti.

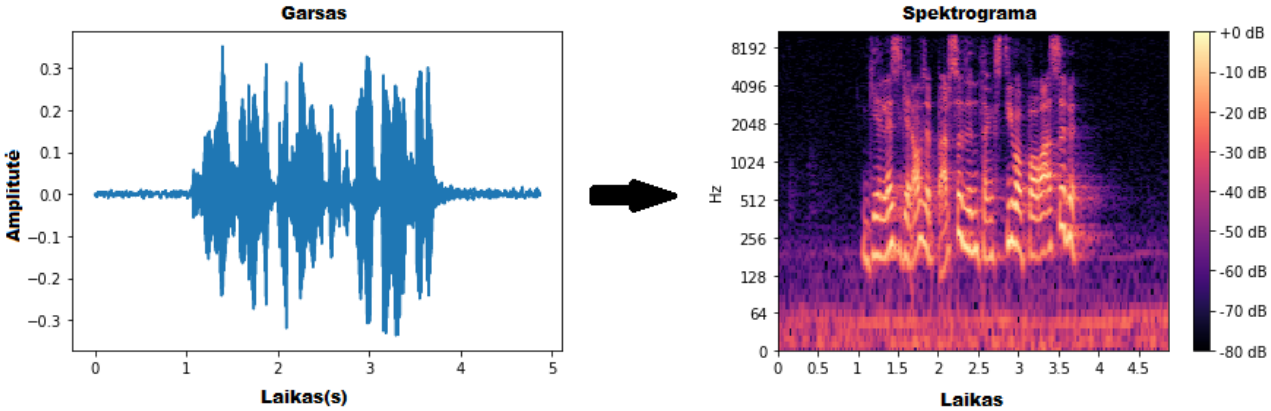
Konvoliuciniai neuroniniai tinklai plačiai naudojami vaizdų klasifikavimui, objektų aptikimui ir kitoms užduotims, kuriose reikia apdoroti duomenis, turinčius tinklinę struktūrą. Jie geba išmokti hierarchinių duomenų atvaizdavimų, todėl veiksmingai iš įvesties duomenų išgauna aukšto lygio požymius.

2. Emocijų balse atpažinimo modelis

Emocijų balse atpažinimo modeliui sukurti buvo naudoti konvoliuciniai neuroniniai tinklai. Duomenys modeliui apmokyti pateikti garsinio failo (wav.) formatu. Iš turimų garso duomenų sudarytos spektrogramos ir joms pritaikytas konvoliucinių neuroninių tinklų modelis. Pagrindinė modelio užduotis gebėti atpažinti viena iš septynių emocijų, kurios yra: baimė, džiaugsmas, liūdesys, neutrali, nuostaba, pasišlykštėjimas ir pyktis.

2.1. Spektrogramos

Spektrograma - tai vizualus signalo dažnių spektro, kaip laiko funkcijos, atvaizdavimas. Ji dažnai naudojama signalų, pavyzdžiui, garso signalų ar laiko eilučių duomenų, dažnių turiniui analizuoti.



8 pav. Duomenų failo pavertimas į spektrogramą

Garsinio failo konvertavimas į spektrogramą yra įprastas pirminio apdorojimo etapas atliekant daugelį garso analizės užduočių, įskaitant kalbos ir emocijų atpažinimą. Šio proceso metu neapdoroti garso duomenys paverčiami naudingesniu formatu, kuriame galima užfiksuoti modelius, kurie gali būti svarbūs sprendžiant užduotį. Verčiant garsą į spektrogramą gaunamas dvimatis signalo vaizdas. X ašis paprastai rodo laiką, y ašis - dažnį, o kiekvieno taško intensyvumas (arba spalva vaizdiniuose atvaizduose) rodo tam tikro dažnio amplitudę arba garsumą tam tikru laiko momentu.

Konvertuotam audio failui taikoma Furjė transformacija. Tai matematinis metodas, kuriuo laiko funkcija arba signalas išskaidomas į jį sudarančius dažnius. Garso duomenims dažnai naudojama trumpojo laiko Furjė transformacija (angl. Short-Time Fourier Transform (STFT)), kuri apskaičiuoja Furjė transformaciją trumpuose, persidengiančiuose laiko eilutės languose. Spektrograma apskaičiuojama imant STFT rezultato kvadratinį dydį [DV90]. Taip gaunamas realiosios vertės dažnių intensyvumo laike rinkinys, kurį galima rodyti kaip vaizdą. Gautas spektras nubraižomas kaip laiko funkcija, kai vertikalioji ašis rodo dažnį, o horizontalioji - laiką [8]. Kiekviename taške esančio grafiko spalva arba intensyvumas rodo signalo amplitudę (arba stiprumą) tuo dažniu ir laiku.

Spektrogramos paveiksliuką galima naudoti kaip įvestį įvairiems modeliams, pavyzdžiui, konvoliuciniams neuroniniams tinklams, kurie gali veiksmingai apdoroti tokius duomenis. Taip pat spektrogramos gali būti naudingos nustatant signalo modelius ar savybes, kurios nėra lengvai matomos pirminiuose duomenyse. Jos, be kita ko, dažnai naudojamos kalbos atpažinimo ir garso klasifikavimo užduotyse.

Norėdami sukurti spektrogramą kodu, naudojamos tokios bibliotekos kaip "Numpy", "Scipy" ir "Librosa". Pavyzdžiui, naudodamiesi "Librosa" biblioteka pythone, galite naudoti funkcijas "stft" arba "melspectrogram", kad atliktumėte Furjė transformaciją ir sukurtumėte garso signalo spektrogramos atvaizdavimą. Gautas išvesties rezultatas bus dvimatis skaitinių reikšmių masyvas, kurį galima naudoti kaip įvestį gilaus mokymosi modeliui. Dirbant su STFT arba mel-spektrogramomis klasifikacijos tikslumas beveik nesiskiria [CFC+17]. Todėl kuriant modelį naudosime mel-spektrogramas

2.2. Duomenys

Projekte "Mokslo pieva" atliekant tyrimą "Išmokykime kompiuterį jausti (atpažinti emocijas žmogaus balse)", sukurti mokymo garso įrašai. Duomenys saugomi (wav.) formatu, atitinkamose emocijų klasėse: baime, džiaugsmas, liūdesys, neutrali, nuostaba, pasišlykštėjimas ir pyktis. Šie duomenys sudaro 10 skirtingų žmonių, kiekvienas iš jų sako po 100 skirtingų frazių, kurias jie išreiškia įvairiomis emocijomis. Balso įrašai, pateikti šiame projekte, nėra vienodo ilgio. Ilgiausias įrašas yra 8.16 sekundžių ilgio, trumpiausias 2.36 sekundžių, bendras visų garso duomenų ilgių vidurkis 4.11 sekundės. Todėl prieš pradedant mokymą, duomenų apdorojimas yra būtinas, siekiant užtikrinti, kad visi balsų įrašai būtų tinkamai apdoroti ir tinkami naudojimui mokymo procese. Iš viso duomenų yra apie 1465. Septynioms skirtingoms klasėms tai nėra daug duomenų

Duomenų apdorojimo etapai gali apimti signalo apdorojimą, kad visi įrašai turėtų vienodą ilgį. Tai gali būti pasiekiama pritaikant signalo išlyginimo, mažinant ar padidinant garso signalo ilgį taip, kad jis būtų suderintas su nustatytais parametrais. Taip užtikrinama, kad visi mokymo duomenys turės vienodą trukmę, nepriklausomai nuo pradinio įrašo ilgio.

Kai duomenys yra tinkamai apdoroti ir sugrupuoti pagal emocijų klases, jie gali būti naudojami kaip mokymo duomenys kompiuterio modeliui, kuris gali mokytis atpažinti emocijas žmogaus balse. Toks modelis gali būti labai naudingas ir turi daug potencialo įvairiose srityse, pvz., kalbos analizėje, emocinėje kompiuterinėje sąveikoje arba netgi įrankiuose, skirtuose atpažinti ir analizuoti emocines išraiškas.

Visų šių veiksmų rezultatas yra patikimi mokymo duomenys, kuriuos galima naudoti mokant kompiuterį atpažinti ir interpretuoti žmogaus balsu išreikštas emocijas.

2.3. Signalų preprocesinimas

Pirminis signalų apdorojimas - tai neapdorotų duomenų paruošimas tolesnei analizei ar apdorojimui. Jis gali apimti įvairias užduotis, pavyzdžiui, duomenų filtravimą, kad būtų pašalintas triukšmas, duomenų transformavimą į kitokį atvaizdavimą arba duomenų atrankos mažinimą, kad

būtų sumažintas imčių skaičius. Konkretūs taikomi pirminio apdorojimo veiksmai priklauso nuo duomenų savybių ir tolesnių užduočių reikalavimų.

Duomenims patiektiems duomenų bazėje reikalinga apdorojimas, kadangi įrašytos frazės yra skirtingo ilgio. Skirtingos trukmės garso įrašų duomenų rinkinį tvarkyti gali būti sudėtinga, nes tai gali turėti įtakos gilaus mokymosi modelio veikimui. Vienas iš sprendimų yra įrašų užpildymas. Trumpesnius garso įrašus galima užpildyti tyla arba pakartoti paskutinį signalo pavyzdį iki fiksuoto ilgio. Šį metodą paprasta įgyvendinti, tačiau jis gali būti neidealus, jei ilgio skirtumas yra labai didelis. Šioje duomenų bazėje esantys garso įrašai nėra idealiai skirtingo dydžio, tačiau skirtumas tarp ilgiausio įrašo ir trumpiausio yra nedidelis, kad turėtų didelę įtaką. Taigi visi garso duomenys verčiami į ilgiausio įrašo ilgį. Trumpesniems įrašai užpildomi tyla, kitaip tariant prirašomi nuliai.

Galimos ir alternatyvos tokios kaip:

- Sutrumpinimas: Ilgesnius garso įrašus galite sutrumpinti iki fiksuoto ilgio. Šį metodą taip pat paprasta įgyvendinti, tačiau iš signalo gali būti pašalinta svarbi informacija.
- Laiko ištempimas: galite reguliuoti garso įrašų ilgį ištempdami arba suspausdami laiko ašį, išlaikydami garso aukštį. Šis metodas gali būti skaičiavimo požiūriu brangesnis, tačiau jis leidžia išsaugoti signale esančią informaciją ir yra tinkamesnis ilgesniems įrašams.
- Duomenų papildymas: Taikant įvairius laiko ištempimo ir (arba) aukščio keitimo metodus galima sukurti naujus pavyzdžius. Taikant šį metodą galima padidinti duomenų rinkinio dydį, todėl modelis gali būti geriau apibendrintas.
- Pirminis apdorojimas: Garso signalus reikės iš anksto apdoroti atliekant tokias operacijas, kaip normalizavimas, perrinkimas ir, jei reikia, triukšmo pašalinimas. Taip bus užtikrinta, kad garso signalai būtų nuoseklaus formato ir kad būtų pašalinta visa nereikšminga informacija.
- Segmentavimas: Garso signalus reikės suskirstyti į mažesnius, persidengiančius langus. Tai būtina, nes Furjė transformacija, naudojama spektrogramai sukurti, gali būti taikoma tik baigtiniam signalo segmentui.

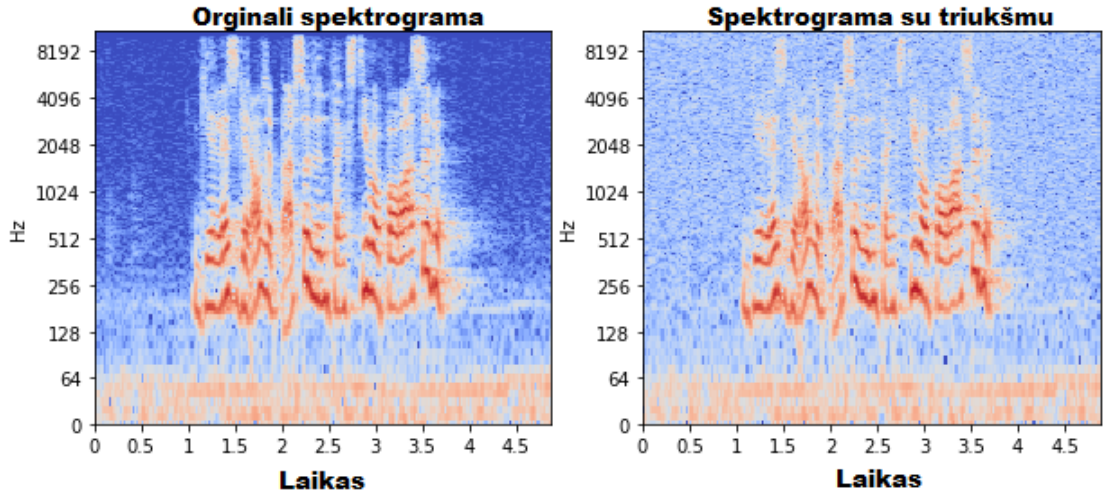
Galiausiai spektrogramas reikės paruošti gilaus mokymosi modeliui, pakeičiant jų dydį ir normalizuojant iki vienodo dydžio bei suskirstant jas į mokymo, patvirtinimo ir testavimo rinkinius. Be to, svarbu atsižvelgti į signalų apdorojimui reikalingus skaičiavimo išteklius ir laiką, nes šis procesas gali būti skaičiavimo požiūriu brangus.

2.4. Duomenų augmentacijos

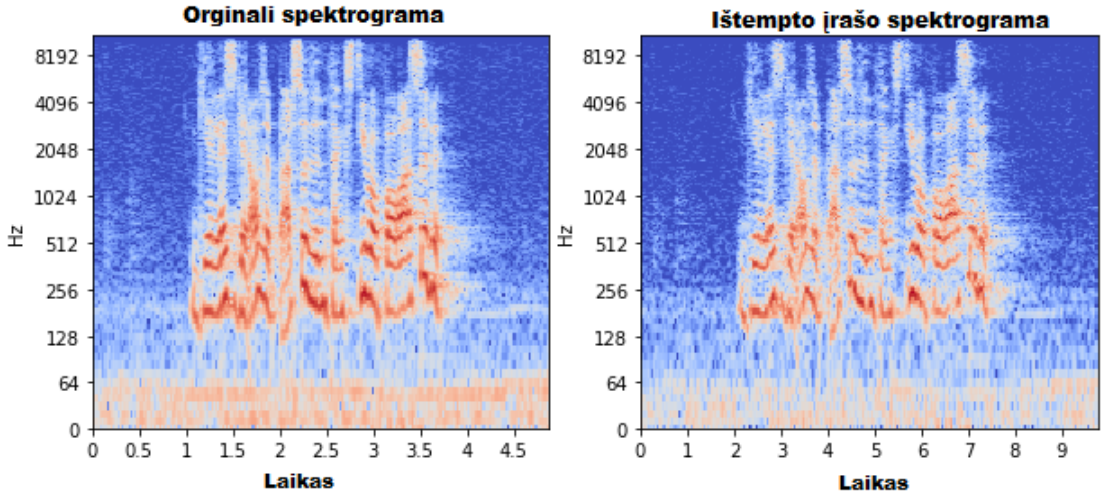
Dažna problema mokinant modelį su tam tikrais duomenimis yra persimokymas (angl. over-fitting). Tai reiškinys kuomet modelis gerai išmoksta atpažinti mokymo duomenis, tačiau pateikus kitokius testavimo duomenis modelio veikimo tikslumas sumažėja ir nesugeba tiksliai klasifikuoti naujų duomenų. Modelis pateikus mažą arba labai didelius duomenų rinkinius pradeda mokytis triukšmo (angl. noise) ar nereikšmingos informacijos. Norint išvengti persimokymo, svarbu naudoti užduočiai gerai pritaikytą modelį ir pakankamą mokymo duomenų kiekį, kad būtų galima užfiksuoti atitinkamus duomenų dėsningumus.

Vienas iš būdų neleisti modeliui persimokyti yra duomenų augmentacijos. Tai metodas, kai iš esamų duomenų pavyzdžių dirbtinai sukuriami papildomi duomenų pavyzdžiai, siekiant pagerinti mašininio mokymosi modelio veikimą. Paveiksliukų atpažinimo uždaviniuose, pakeisti duomenys būna pakeisti įvairiai: šiek tiek pasukti per kuria nors ašį, pritraukti, sumažintas raiškumas, pakeistos spalvos ar paslinkti į kuria nors pusę. Šis būdas paprastai naudojamas vaizdų atpažinimo uždaviniuose, siekiant padidinti mokymo duomenų aibės dydį ir įvairovę.

Augmentacijos metodus galima taikyti balso duomenims, siekiant atpažinti emocijas naudojant spektrogramas. Norint taikyti balso duomenų papildymą naudojant spektrogramas, vienas iš metodų yra papildomų spektrogramų generavimas taikant transformacijas pradinėms spektrogramoms. Pavyzdžiui, į originalias spektrogramas galima pridėti triukšmo [9] arba jas ištempti ar suspausti laiko dimensijoje [10]. Taip pat galite pabandyti sujungti kelias spektrogramas ir sukurti naują sintetinę spektrogramą.



9 pav. Duomenų augmentacija pridedant triukšmo



10 pav. Duomenų augmentacija ištempiant įrašą

Šias papildytas spektrogramas galima naudoti emocijų atpažinimo mašininio mokymosi

modeliui apmokyti. Taip padidinus mokymo duomenų aibės dydį ir įvairovę, modeliui gali pavykti išmokti patikimesnių ir labiau apibendrinančių emocijų atpažinimo požymių.

2.5. Giliojo mokymosi modelio kūrimas

Yra daug skirtingų gilaus mokymosi modelių, kuriuos galima naudoti emocijoms atpažinti naudojant spektrogramas, tačiau šiai problemai spręsti yra naudojamas konvoliucinis neuroninis tinklas (CNN), kuris yra neuroninio tinklo tipas, ypač gerai tinkantis apdoroti duomenis, turinčius tinklinę struktūrą, pavyzdžiui, vaizdą.

Spektrogramos pateikiamos kaip vaizdai ir naudojant konvoliucininį neuroninį tinklą ieškoma atitinkami požymiai. Taip konvoliucini neuroninį tinklą galima išmokyti klasifikuoti spektrogramas pagal tai, kokią emociją jos atspindi.

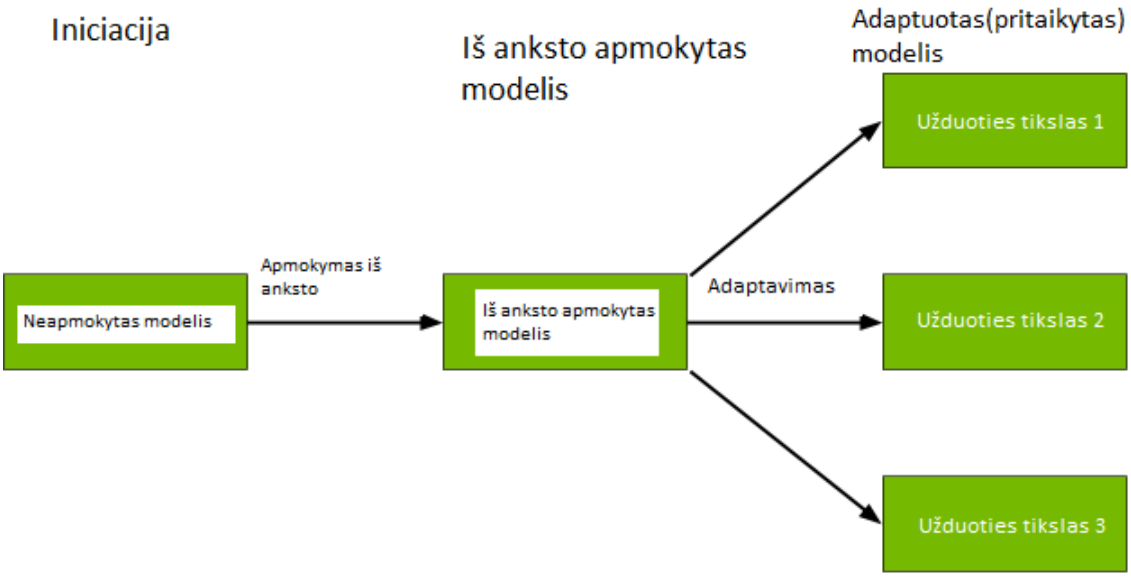
Žingsniai kuriant modelį:

- Išankstinis apdorojimas: Prieš pradedant mokyti modelį, svarbu iš anksto apdoroti spektrogramų duomenis normalizuojant reikšmes ir suskirstant juos į mokymo, patvirtinimo ir testavimo rinkinius.
- Konvoliucinių neuroninių tinklų architektūros projektavimas: Konvoliucinių neuroninių tinklų architektūra priklausys nuo konkrečių duomenų savybių ir užduoties. Įprastas metodas - naudoti kelis konvoliucinius sluoksnius, po kurių seka vienas ar daugiau visiškai sujungtų sluoksnių. Konvoliuciniai sluoksniai gali būti naudojami požymiams iš spektrogramos išgauti, o visiškai sujungti sluoksniai gali būti naudojami požymiams klasifikuoti.
- Mokymas: Konvoliucinių neuroninių tinklų modelį galima apmokyti naudojant iš anksto apdorotus spektrogramos duomenis. Dažniausiai modelio parametrams optimizuoti naudojamas stochastinis gradientinis nusileidimas (SGD) arba jo variantas, pvz. "Adam".
- Hiperparametrų derinimas: konvoliucinių neuroninių tinklų modelio našumą galima pagerinti derinant hiperparametrus, pavyzdžiui, mokymosi greitį, partijos dydį ir filtrų skaičių konvoliuciniuose sluoksniuose. Tai galima padaryti taikant tokius metodus, kaip tinklelio paieška arba atsitiktinė paieška.
- Vertinimas: konvoliucinių neuroninių tinklų modelio veikimą galima įvertinti lyginant modelio prognozes su tikrosiomis bandomosios aibės etiketėmis. Įprasti konvoliucinių neuroninių tinklų modelio, skirto garso klasifikavimui, našumo vertinimo rodikliai yra tikslumas, preciziškumas, atkūrimo statistika (angl. Recall) ir F1 balas.
- Tikslus derinimas: Gavus gerą modelį, jį galima tikslinti tęsiant mokymą su mažesniu mokymosi greičiu arba taikant tokius metodus kaip perkėlimo mokymasis.

Norint pasiekti geriausius rezultatus reik išbandyti įvairias architektūras ir įvertinti jų našumą, kad būtų galima pasirinkti geriausią.

2.6. Iš anksto apmokytas modelis

Iš anksto apmokytas modelis (angl. Pre-trained model) - tai modelis, kuris buvo apmokytas naudojant didelį duomenų rinkinį ir kurį vėliau galima tiksliai pritaikyti konkrečiai užduočiai. Iš anksto parengtų modelių naudojimo idėja yra ta, kad jie jau yra išmokę naudingų savybių iš didelio duomenų rinkinio ir gali būti naudojami kaip pradinis taškas naujai užduočiai atlikti, taip sutaupant laiko ir kompiuterinių išteklių, reikalingų modeliui iš naujo parengti [HZD+21; QSX+20].



11 pav. Iš anksto apmokyto modelio pavyzdys [JD20]

Yra keletas iš anksto parengtų modelių, skirtų įvairioms užduotims, pavyzdžiui, vaizdų klasifikavimui, objektų aptikimui, natūralios kalbos apdorojimui ir kalbos atpažinimui. Šiuos modelius galima lengvai pasiekti naudojant įvairias gilaus mokymosi bibliotekas, pavyzdžiui, TensorFlow, Keras, PyTorch ir kitas.

Naudojant iš anksto apmokytus modelius, svarbu atsižvelgti į išankstinio mokymo duomenų rinkinio ir duomenų rinkinio, kuriame modelis bus tikslinamas, panašumą. Jei abu duomenų rinkiniai labai skiriasi, tikslus derinimas gali būti neveiksmingas, todėl gali būti geriau mokyti modelį iš naujo.

Apskritai, iš anksto apmokyti modeliai yra naudinga gilaus mokymosi priemonė, nes jie gali sutaupyti laiko ir skaičiavimo išteklių ir dažnai gali būti geras atspirties taškas naujai užduočiai atlikti.

Naudojant iš anksto apmokytą modelį, pavyzdžiui, modelį, apmokytą pagal "ImageNet" duomenų rinkinį, gali būti geras būdas pradėti spręsti gilaus mokymosi užduotį. ImageNet duomenų rinkinys yra didelė pažymėtų vaizdų kolekcija, kuri plačiai naudojama vaizdų klasifikavimo modeliams mokyti. Naudodami modelį, kuris jau buvo apmokytas pagal šį duomenų rinkinį, kaip pradinį tašką, galite sutaupyti laiko ir potencialiai pagerinti savo modelio našumą, ypač jei užduotis, kurią sprendžiate, yra panaši į užduotį, pagal kurią buvo apmokytas iš anksto apmokytas

modelis.

Tačiau svarbu nepamiršti, kad naudojant iš anksto apmokytą modelį taip pat gali kilti ”paradoksas”, jei modelis nėra gerai pritaikytas nagrinėjamai užduočiai. Tokiu atveju modelis gali būti išmokęs savybių, kurios nėra svarbios sprendžiamam uždaviniui, o tai gali pabloginti rezultatus. Paprastai verta įvertinti iš anksto apmokyto modelio veikimą sprendžiant konkrečią užduotį ir naudojant konkrečius duomenis, kad įsitikintumėte, jog jis tinka jūsų poreikiams.

2.7. Giliojo mokymosi architektūros

Konvoliucinių tinklų architektūros projektavimas yra svarbus žingsnis norint sukurti modelį, kuris duotų gerus rezultatus. Konvoliucinių tinklų architektūra dažniausiai priklauso nuo konkrečių duomenų savybių ir užduoties. Įprastas metodas - naudoti kelis konvoliucinius sluoksnius, po kurių seka vienas ar daugiau visiškai sujungtų sluoksnių. Konvoliuciniai sluoksniai gali būti naudojami požymiams iš spektrogramos išgauti, o visiškai sujungti sluoksniai gali būti naudojami požymiams klasifikuoti. Galima padidinti modelio gylį, kad jis išmoktu sudėtingesniu požymius ir išgautu geresnius rezultatus, tačiau tai ne visada veikia ir reikalauja daugiau resursų. Sutelkimo imant maksimalią reikšmę sluoksnio naudojimas padeda sumažinti perteklinį pritaikymą ir padidinti modelio gebėjimą apibendrinti. Pilnai sujungtų sąryšių sluoksnis po konvoliucinių sluoksnių požymiams klasifikuoti. Paprastai naudojamas vienas ar daugiau visiškai sujungtų sluoksnių. Šie sluoksniai gali būti naudojami prognozėms atlikti pagal konvoliucinių sluoksnių išskirtus požymius. Taip pat dažnai siekiant išvengti persimokymo naudojamas išmetimas. Tai gilaus mokymosi metodas, naudojamas siekiant išvengti persimokymo atsitiktinai išbraukiant vienetus (neuronus) mokymo metu. Ji veikia nustatant tikimybę (išmetimo normą) kiekvienam vienetui, o per kiekvieną mokymo iteraciją kai kurie vienetai ”iškrenta” arba yra ignoruojami. Dėl to modelis tampa atsparesnis konkretiems mokymo duomenims, nes priverčia modelį per daug nesiremti viena funkcija ar neuronu. Paprastai šis metodas taikomas visiškai sujungtiems sluoksniams, tačiau jį galima taikyti ir konvoliuciniams ir pasikartojantiems sluoksniams. Tai paprastas, bet galingas metodas, kurį galima naudoti kartu su kitais reguliarizavimo metodais, tokiais kaip L1 ir L2 svorio mažinimas ir ankstyvas sustabdymas. Naudojant atmetimą svarbu nustatyti tinkamą atmetimo normą. Įprasta vertė yra 0,5, o tai reiškia, kad per kiekvieną mokymo iteraciją kiekvienas neuronas turi 50% šansą būti išmestas. Tačiau optimali vertė gali skirtis priklausomai nuo duomenų rinkinio ir konkrečios modelio architektūros. Testavimo etape išmetimas paprastai išjungiamas ir naudojami visi neuronai. Taip yra todėl, kad testavimo etape siekiama panaudoti visą informaciją, kurią modelis išmoko mokymo metu, prognozuojant nematytus duomenis.

Šiuolaikiniai konvoliuciniai neuroniniai tinklai, toki kaip EfficientNet, VGG ir DenseNet, naudoja aukštai optimizuotus architektūros modelius, kurie gali efektyviai atlikti įvairias užduotis su aukštu tikslumu.

- EfficientNet. Šis konvoliucinio neuroninio tinklo modelis yra paremtas idėja, kad modelio dydis, gylis ir rezoliucija turi būti kartu keičiami, siekiant optimalaus našumo. Tai leidžia modeliui efektyviai mokytis ir taip pasiekti geresnius rezultatus su mažesniu skaičiavimų kiekiu.

- VGG modeliai (VGG16 ir VGG19) yra giliai sukrauti konvoliuciniai neuroniniai tinklai, kurių ypatybė yra tai, kad jie naudoja tik 3x3 konvoliucinius filtrus ir 2x2 maksimalaus suderinimo (max-pooling) operacijas. Ši architektūra yra gana paprasta, su daug sluoksnių, bet su mažesnėmis filtrų dydžio vertėmis, leidžiančiomis modeliui išmokti gilesnes savybes iš duomenų.
- DenseNet modeliai yra unikalūs dėl savo sluoksnių jungimo būdo. Kiekvienas sluoksnis yra tiesiogiai susijęs su kiekvienu ankstesniu sluoksniu, o jo išėjimai tampa įėjimais į kiekvieną sekantį sluoksnį. Šis "tankus" jungimo būdas leidžia propaguoti gradientus per visą tinklą, kas palengvina mokymą ir gali pagerinti bendrą našumą.

Šie trys modeliai parodė, kad konvoliucinių neuroninių tinklų architektūros projektavimas yra esminis veiksnys, lemiantis tinklo našumą. Nors kiekvienas modelis turi savo unikalias ypatybes ir stiprybes, jie visi naudoja konvoliucinių

2.8. Nuostolių apskaičiavimas

Atpažįstant emocijas naudojant spektrogramas, nuostolių funkcija yra matas, rodantis, kaip gerai modelis sugeba klasifikuoti spektrogramas pagal jose vaizduojamas emocijas. Nuostolių funkcija naudojama modeliui optimizuoti mokymo metu, nukreipiant modelį į tikslesnes prognozes.

Viena iš įprastų emocijų atpažinimo nuostolių funkcijų yra kategorinis kryžminės entropijos nuostolis, kuris apibrėžiamas taip:

$$Loss = \begin{cases} (y \log(p) + (1-y)\log(1-p)), & \text{jei, } M = 2 \\ \sum_{c=1}^{M} y_{o,c} \log(p_{o,c}), & \text{jei, } M > 2 \end{cases}$$

M - klasių skaičius.

log - natūralusis logaritmas.

y - dvejetainis rodiklis (0 arba 1), jei klasės etiketė c yra teisinga stebinio o klasifikacija.

p - prognozuojama tikimybė, kad stebinys o priklauso c klasei.

Nuostoliai apskaičiuojami kiekvienam mokymo duomenų aibės bandiniui, o tada išvedamas viso duomenų aibės vidurkis. Taip pat nuostolių funkcija skaičiuojama ir validavimo duomenims siekiant gauti kuo geresnius rezultatus.

2.8.1. Klasifikavimo lentelė

Klasifikavimo matrica yra naudinga priemonė konvoliucinio neuroninio tinklo modelio, skirto emocijų atpažinimui balse naudojant spektrogramas, našumui įvertinti.

Kelių klasių klasifikavimo problemos sumaišymo matrica paprastai yra n x n dydžio, kur n yra klasių skaičius. Eilutėse pateikiamos tikrosios etiketės, o stulpeliuose - prognozuojamos etiketės. Klasifikavimo matrica turėtų būti sukurta lyginant prognozuojamas etiketes su tikrosiomis etiketėmis. Matricoje turėtų būti suskaičiuotas ir įrašytas teisingų teigiamų, teisingų neigiamų, klaidingai teigiamų ir klaidingai neigiamų prognozių skaičius. Iš klasifikavimo matricos galima

apskaičiuoti kitus našumo rodiklius, pvz., tikslumą, F1 balą ir t.t. Rezultatai turėtų būti interpretuojami atsižvelgiant į konkrečias duomenų rinkinio savybes ir užduoties tikslus. Svarbu pažymėti, kad painiavos matrica yra tik viena iš metrikų, kuri turėtų būti naudojama modelio našumui įvertinti, ji turėtų būti naudojama kartu su kitomis metrikomis. Be to, geras modelis turėtų ne tik turėti gerus našumo skaičius, bet ir gerai veikti su nematytais duomenimis bei būti atsparus įvairiems triukšmams ir įvesties duomenų pokyčiams.

3. Tyrimai

Šiame skyriuje detaliai aptariamas atliktas tyrimas, kuriant emocijas balse atpažįstantį modelį, naudotos neuroninių tinklų modelių architektūros ir jų vertinimas. Kiekviena naudota modelio architektūra buvo pasirinkta atsižvelgiant į specifinį sprendimo uždavinį ir turimus duomenis.

Modeliai mokomi 25 epochas keičiant mokymosi greiti siekiant išgauti geriausius rezultatus. Kadangi duomenų rinkinys nėra didelis modeliams apmokyti nereikia labai daug mokymosi iteracijų. Kiekvieno modelio rezultatai yra atidžiai analizuojami, o gauti rezultatai palyginami tarpusavyje. Analizuojamos modelių klasifikavimo gebėjimo stiprybės.

Be to, aptariami potencialūs tyrimo metodų trūkumai, iššūkiai, su kuriais susiduriama dirbant su duomenimis ir mokant modelius, bei galimybės jų tobulinimui ateityje.

3.1. Darbo aplinka ir technologijos

Tyrimas buvo atliktas naudojant lokalaus kompiuterio resursus, kuris buvo paruoštas atlikti sudėtingus skaičiavimus, reikalingus giliosios mokymosi modeliams apmokyti. Konkretus darbo aplinkos pasirinkimas buvo Jupyter Lab (python), kuris yra populiari mokslinių tyrimų platforma, leidžianti vykdyti interaktyvų programavimą ir analizę. Pagrindinės naudotos bibliotekos modelio kūrimui buvo:

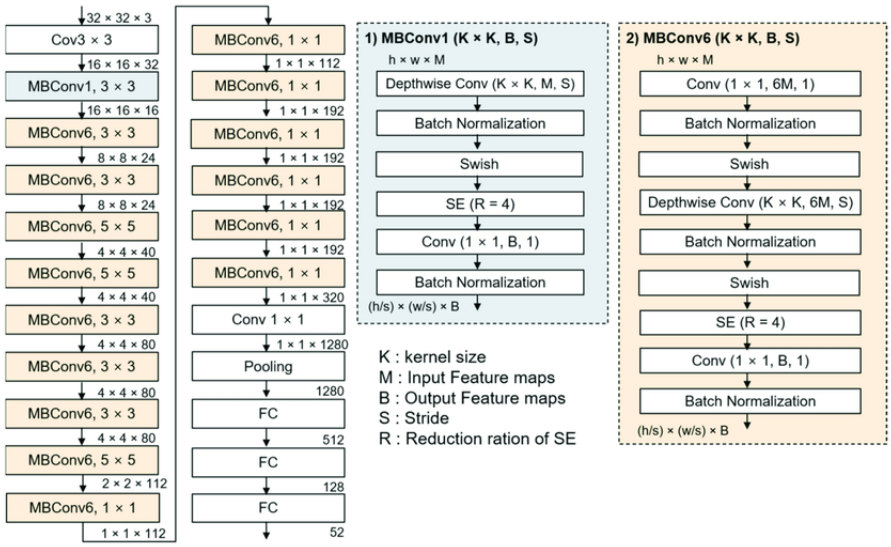
- Tensorflow
- numpy
- matplotlib
- seaborn
- sklearn.metrics
- PIL
- librosa
- soundfile

Sukurti modeliui naudota konvoliuciniai neuroniniai tinklai ir spektrogramos. Vaizdai verčiami dvimačiu figūrų masyvu. Kadangi tai yra dvimatis masyvas, jis suteikia daugiau informacijos nei vienmatės MFCC funkcijos. Nors mokymosi greitis yra lėtesnis nei gilaus mokymosi naudojant MFCC, kuris yra vienmačiai duomenys, naudojant GPU galima padidinti gilaus mokymosi greitį [LK20]. Mokinant modeli naudojant CPU laiko sąnaudos ženkliai padidėja, taigi norint įveikti šiuos aparatinės įrangos apribojimus, naudojamas grafinis procesorius (NVIDIA GeForce GTX 1080). Remiantis [Ten15], pagal `tensorflow_gpu-2.10.0` veikiančias versijas, pritaikytos cuDNN 8.1 ir CUDA 11.2 leidžiančios dirbti naudojant lokalų GPU.

3.2. EfficientNet

Tai ”Google” pristatyta 2019 m. EfficientNet modelių kovoliucinių tinklų šeimai priklaisanti architektūra. EfficientNet modeliai yra labai efektyvūs, nes, palyginti su kitais CNN modeliais, jie užtikrina geresnį tikslumo ir skaičiavimo išteklių naudojimo kompromisą. EfficientNet šeimą

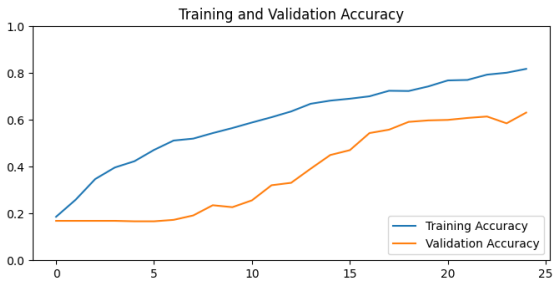
sudaro aštuoni modeliai, nuo EfficientNetB0 (bazinis modelis) iki EfficientNetB7. Kiekvienas vėlesnis modelis yra padidinta ankstesniojo modelio versija, pasižyminti didesniu tikslumu, tačiau reikalaujanti daugiau skaičiavimo išteklių. Taip pat egzistuoja patobulinta EfficientNetV2 architektūra turinti nuo B0 iki B3 ir dar tris papildomus modelius L, M ir S reiškiančius atitinkamai mažą, vidutinį ir didelį variantus.



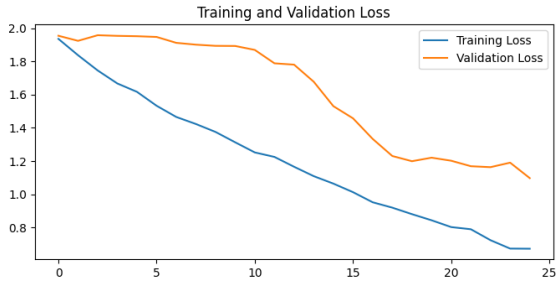
12 pav. EfficientNetB0 architektūros pavyzdys [GFC+21]

Emocijų balse atpažinimui išbandyti keli modeliai: EfficientNetB2, EfficientNetV2B2 ir EfficientNetV2S

3.2.1. EfficientNetB2



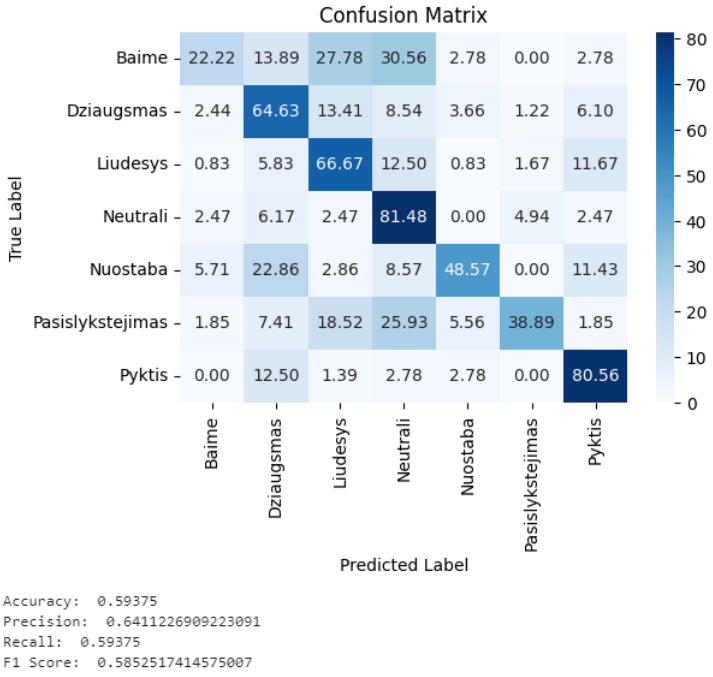
13 pav. Mokymosi ir validacijos tikslumas



14 pav. Mokymosi ir validacijos nuostoliai

Modelį sudarė 7,710,857 apmokomi parametrai. Mokymo proceso metu modelis buvo pakankamai gerai išmokytas. Mokant modelį pasiektas tikslumas (angl. training accuracy) yra aukštas - 82.3%. Tačiau, kai modelis buvo testuojamas su nematytų pavyzdžių duomenimis, tikslumas (angl. accuracy) pakilo tik iki 59,3%. Tinklo didžiausias tikslumas pasiektas ties paskutine 25 epocha, vadinasi pridėjus daugiau epochų ar keičiant mokymosi greiti galima pasiekti šiek tiek geresnius rezultatus

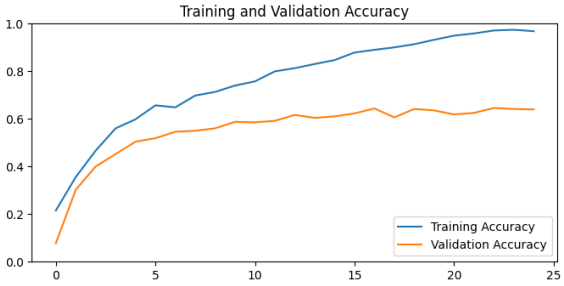
Analizuojant klasifikavimo lentelę, matyti, kad sunkiausiai atpažinti buvo baimė, pasišlykštėjimas ir nuostaba.



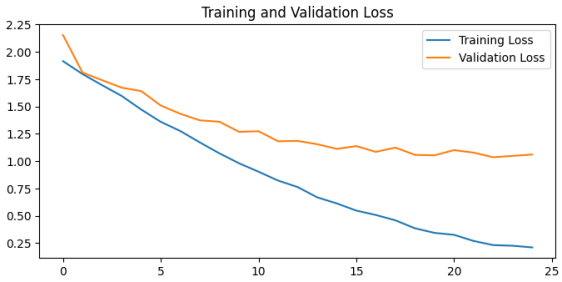

```
Accuracy:  0.59375
Precision:  0.6411226909223091
Recall:  0.59375
F1 Score:  0.5852517414575007
```

15 pav. EfficientNetB2 Klasifikavimo lentelė

3.2.2. EfficientNetV2B2



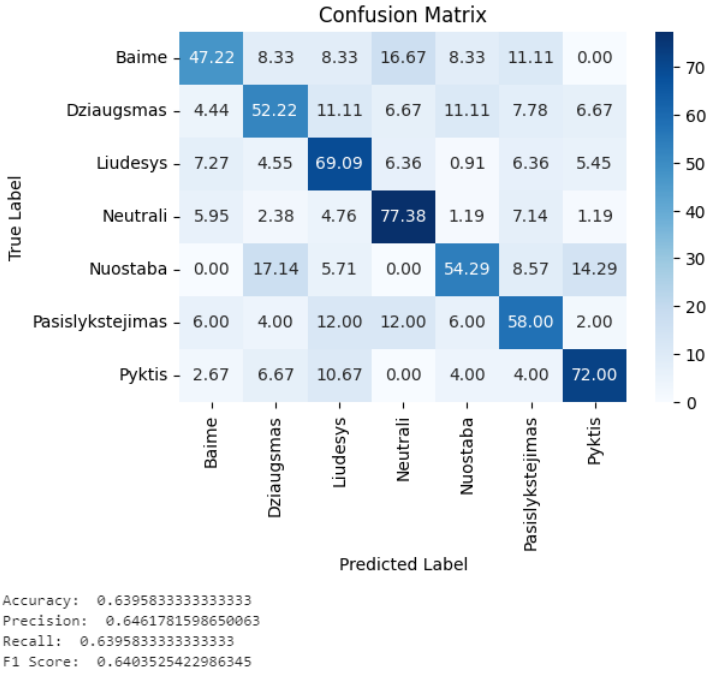
16 pav. Mokymosi ir validacijos tikslumas



17 pav. Mokymosi ir validacijos nuostoliai

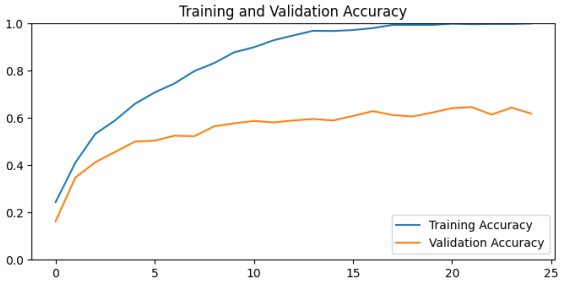
Modelio sudarė 8,696,949 apmokomi parametrai, o EfficientNetV2B2 architektūra, pagrįsta šiuo modeliu, davė geresnius rezultatus. Su mokymosi duomenimis modelis pasiekė 97.4% tikslumą per 24 epochas. Didžiausias validacijos tikslumas buvo 64.5% pasiekta 23 epochoje.

Klasifikavimo lentelė atskleidžia, kad bendras modelio tikslumas siekia 63.9%. Sunkiausiai atpažinti buvo emocijos, susijusios su baimės, džiaugsmo, pasišlykštėjimo ir nuostabos emocijomis. Tai rodo, kad šios emocijos gali turėti tam tikrą panašumą arba yra iššūkis modeliui atpažinti jas iš turimų duomenų.

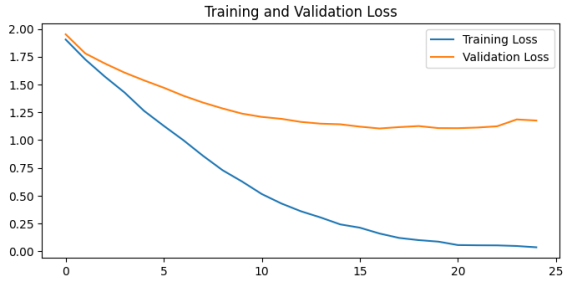


18 pav. EfficientNetV2B2 Klasifikavimo lentelė

3.2.3. EfficientNetV2S



19 pav. Mokymosi ir validacijos tikslumas

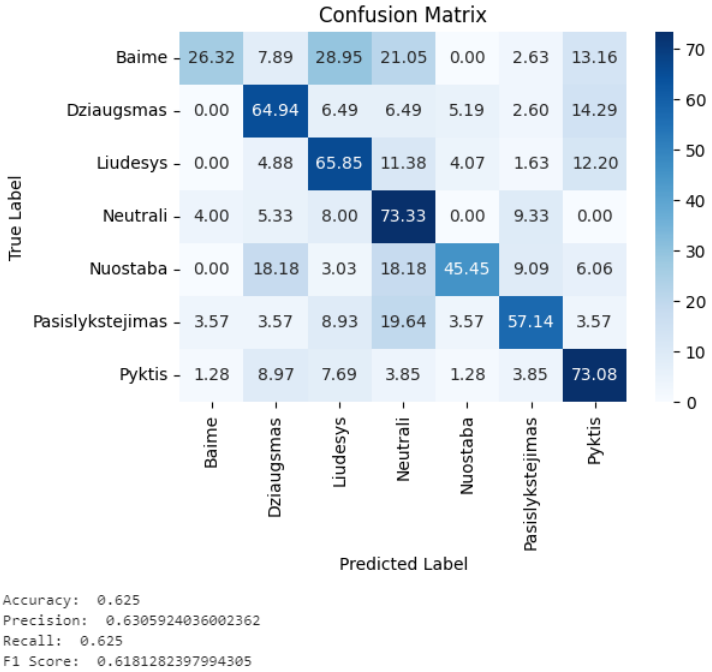


20 pav. Mokymosi ir validacijos nuostoliai

Paskutinį apmokytą EfficientNet modelį sudarė 20,340,327 mokomieji parametrai, kurių yra žymiai daugiau nei kituose modeliuose. Tai rodo, kad šis modelis yra gilesnis ir sudėtingesnis. Su mokymosi duomenimis pasiektas aukštasis tikslumas - 100% po 25 epochų, rodydamas, kad modelis išmoko mokymo duomenis beveik iki tobulumo.

Su validacijos duomenimis tikslumas pasiekė 64.5%, kas yra lygiai tas pats rezultatas kaip ir EfficiencyNetV2B2 modeliui. Tai gali reikšti, kad šie du modeliai pasiekė panašius rezultatus atpažįstant nematytus duomenis.

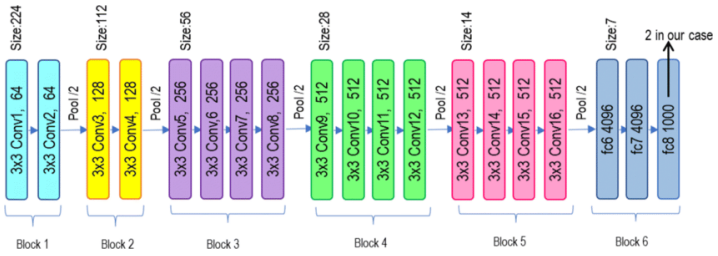
Bendras modelio tikslumas, matuojamas klasifikavimo lentelėje, yra 62.5%. Sunkiausiai atpažįstamos emocijos, susijusios su baimės, pasišlykštėjimo ir nuostabos išraiškomis. Tačiau džiaugsmą modelis atpažino geriau nei ankstesni modeliai. Taip pat matosi, kad modelis dažniau spėja pykti.



21 pav. EfficientNetV2S Klasifikavimo lentelė

3.3. VGG19

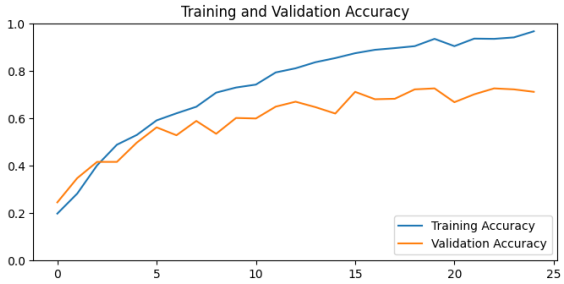
Konvoliucinio neuroninio tinklo modelis, vadinamas VGG19 , yra gerai žinomas dėl savo paprastumo ir pasikartojančių projektavimo modelių. Jį sudaro 19 sluoksnių, iš kurių 16 konvoliucinių sluoksnių išgauna požymius, o 3 visiškai sujungti sluoksniai galiausiai klasifikuoja vaizdus. Kiekviename konvoliuciniame sluoksnyje naudojami maži 3x3 filtrai, o toliau seka ReLU aktyvavimo funkcija. Siekiant sumažinti požymių žemėlapių dimensiškumą ir padaryti tinklą lengviau valdomą skaičiavimo požiūriu, keliuose architektūros taškuose įgyvendinamas maksimalus sutelkimas.



22 pav. VGG19 Modelio architektūra [KQ22]

VGG19 konvoliucinio neuroninio tinklo modelis, turintis 20,027,975 mokomuosius parametrus, yra sudėtingas ir galingas instrumentas vizualiniam atpažinimui. Šio modelio dydis yra palyginamas su EfficientNetV2S, tačiau jo veikimas skiriasi. Po 24-os epochos, modelio tikslumas su validacijos duomenimis siekė 72.3%, o su mokymo duomenimis - net 96.8%.

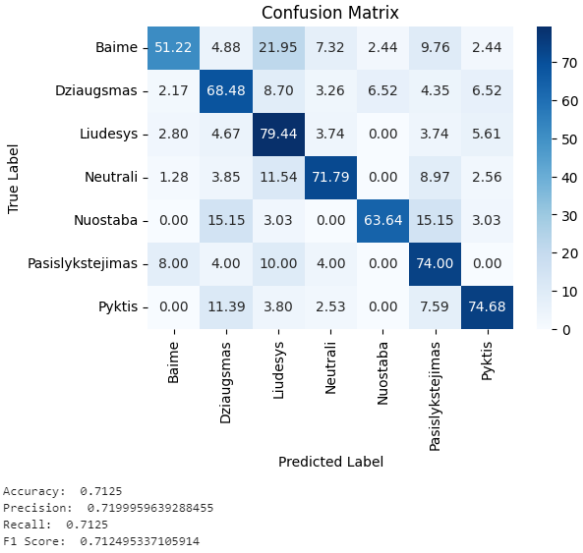
Tačiau modelis susidūrė su kai kuriomis problemomis, pavyzdžiui, atpažindamas baimę. Ši emocija buvo tiksliai klasifikuota tik 51.2% atvejų iš validacijos duomenų. Modelis dažnai maišydavo liūdesį su baime.



23 pav. Mokymosi ir validacijos tikslumas

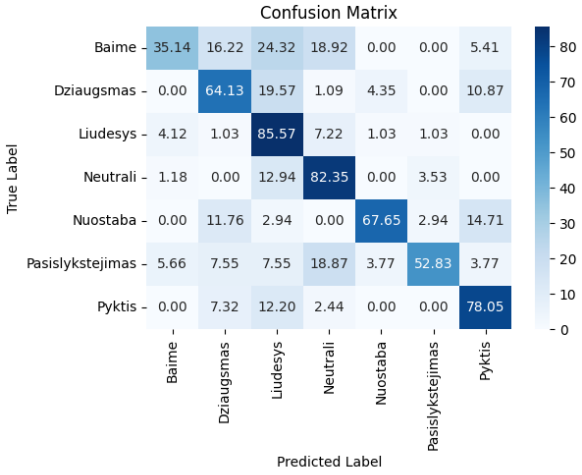
24 pav. Mokymosi ir validacijos nuostoliai

Nepaisant minėtų problemų, VGG19 modelis vis dar pasirodė esąs gana efektyvus, pasiekdamas 71.5% tikslumą, kuris yra žymiai aukštesnis nei su visais EfficientNet apmokytai modeliai. Tai rodo, kad VGG19, nepaisant savo paprastumo, gali išgauti gerus rezultatus sprendžiant sudėtingas klasifikacijos problemas.




```
Accuracy:  0.7125
Precision:  0.7199959639288455
Recall:  0.7125
F1 Score:  0.712495337105914
```

25 pav. VGG19 Klasifikavimo lentelė




```
Accuracy:  0.7083333333333334
Precision:  0.7199141183356415
Recall:  0.7083333333333334
F1 Score:  0.7015742637653574
```

26 pav. VGG16 Klasifikavimo lentelė

Tyrimo eigoje buvo apmokytas ir VGG16 modelis. Pagrindinis jų skirtumas - sluoksnių skaičius. 13 konvoliucinių sluoksnių ir 3 visiškai sujungti sluoksniai sudaro 16 VGG16 sluoksnių, o 16 konvoliucinių sluoksnių ir 3 visiškai sujungti sluoksniai sudaro 19 VGG19 sluoksnių.

Atliekant tyrimą nustatyta, kad VGG16 modelio tikslumas buvo 70,8%, tai yra tik vienu procentu mažesnis už VGG19 modelio tikslumą. Tai rodo, kad VGG19 modelis nėra žymiai efektyvesnis už VGG16 modelį, nors turi daugiau sluoksnių. Taip pat jų klasifikavimo prognozės gan skiriasi.

Papildomi konvoliuciniai sluoksniai VGG19 modelyje galėjo padėti jam išmokti atpažinti sudėtingesnius vaizdinius signalus, susijusius su pasišlykštėjimu ir baime, todėl atrodo, kad jis geriau atpažįsta šias emocijas. Tuo tarpu VGG16 modeliui geriau sekėsi atpažinti neutralias ir liūdesio emocijas.

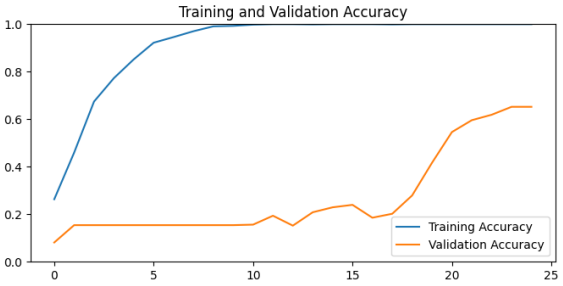
3.4. ResNet50

Dar vienas tiriamas konvoliucinio neuroninio tinklo modelis ResNet50, kuris priklauso ResNet (angl. ”Residual Network”) modelių šeimai. Architektūros pavadinime 50 nurodo šio modelio sluoksnių skaičių. ResNet50 yra labai populiarus modelis dėl savo efektyvumo ir gero našumo įvairiose užduotyse, tokiuose kaip vaizdų klasifikavimas taigi išbandomas ir emocijų atpažinimo balse naudojant spektrogramas uždavinyje.

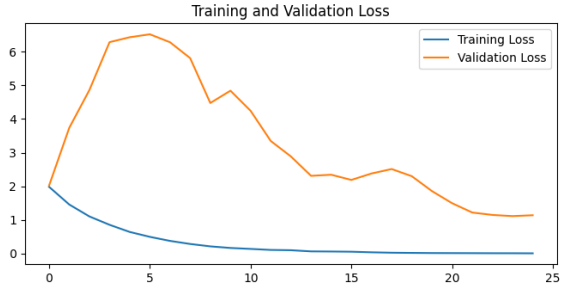
Kaip ir kitos ResNet architektūros kurios yra ResNet-18, ResNet-34, ResNet-101, ResNet-110, ResNet-152, ResNet-164, ResNet-1202, ResNet50 naudoja ”rezidualinius blokus” arba ”identitetus” perduodamus blokus, kurie leidžia mokymo signalui tiesiogiai praeiti per kelių sluoksnių grandinę. Tai palengvina mokymąsi giliuose tinkluose ir mažina išnykimo gradiento problemą.

ResNet50 modelis sudarytas iš 50 sluoksnių, suskirstytų į 4 blokus. Kiekviename bloke yra kelios konvoliucinės ir pasikartojančios sluoksnio operacijos, o blokas baigiasi ”rezidualiniu” ryšiu, leidžiančiu mokymo signalui praeiti per visą bloką nesikeičiant.

Tai yra galingas modelis, kuris gali būti pritaikytas įvairioms užduotims, o jo architektūra gali būti lengvai pritaikyta pagal konkrečius poreikius.



27 pav. Mokymosi ir validacijos tikslumas

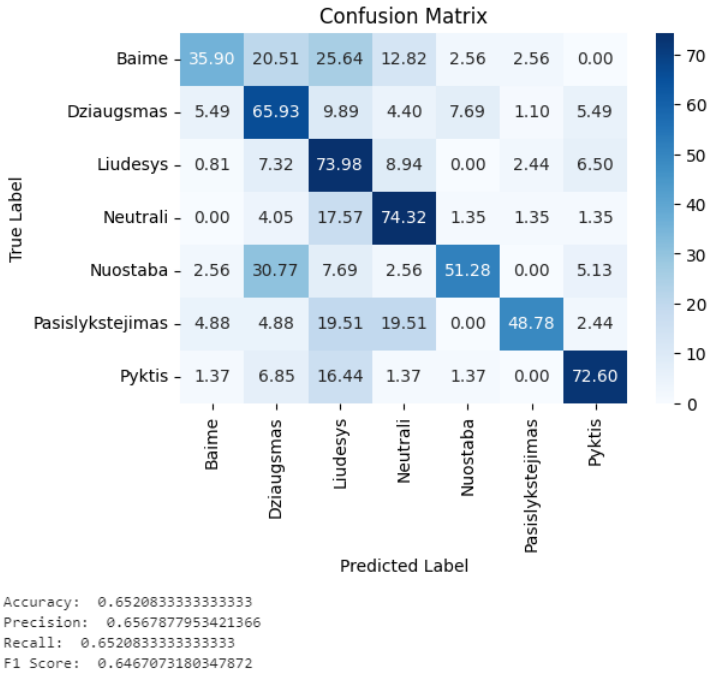


28 pav. Mokymosi ir validacijos nuostoliai

ResNet50 yra didlė ir gili konvoliucinio neuroninio tinklo architektūra, kurios dydį nusako 23,548,935 parametrų skaičius. Ši sudėtinga struktūra, kurią sukuria šie parametrai, leidžia modeliui išmokti ir atskirti sudėtingus požymius duomenyse.

Mokymo procesas, vykęs per 25 epochas, parodė greitą pažangą, su mokymosi tikslumu pasiekusiu 100% - tai rodo, kad modelis puikiai išmoko atpažinti mokymo rinkinio požymius. Tačiau vertinant modelį naudojant validavimo duomenų rinkinį, tikslumas stabilizavosi 10 epochų metu, siekdami 15.8%, o nuostolių funkcija išaugo iki 5 epochos, po to pradėjo mažėti. Per paskutinius 5 mokymo epizodus validacijos funkcija staigiai pradėjo augti, pasiekdama net 65.1% tikslumą.

Detalus klasifikavimo lentelės peržiūrėjimas rodo, kad modelis gana efektyviai atpažino džiaugsmo, liūdesio, pykčio ir neutralumo emocijas. Tačiau baimės emocija sukėlė daugiausiai problemų, dažnai ją klaidingai susiejant su džiaugsmo, liūdesio ir neutralumo emocijų klasėmis. Šie rezultatai rodo ResNet50 galimybes ir jo veiksmingumą sprendžiant emocijų atpažinimo užduotis, naudojant spektrogramas.

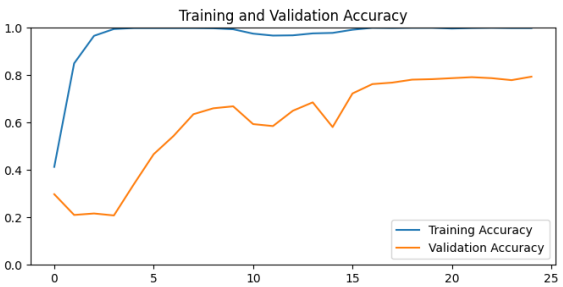


29 pav. ResNet50 Klasifikavimo lentelė

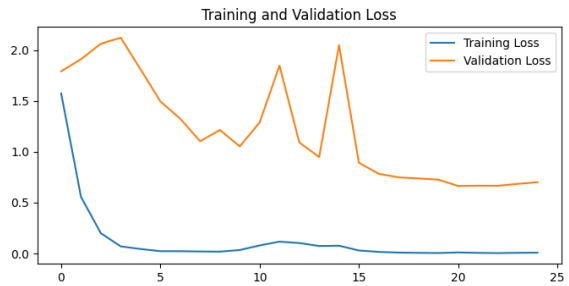
3.5. DenseNet169

Paskutinė tiriama konvoliucinių neuroninių tinklų architektūra yra DenseNet169. Bendras tinklo sluoksnių skaičius žymimas skaičiumi ”169” žodyje DenseNet169. Pagrindinis ”DenseNet” principas yra tas, kad tankiojo bloko viduje kiekvienas sluoksnis yra tiesiogiai sujungtas su kiekvienu kitu sluoksniu [HLV+17]. Mažiau parametrų, geresnis pakartotinis funkcijų naudojimas ir geresnis našumas daugybėje lyginamųjų duomenų rinkinių - visa tai yra šio tankaus sujungimo privalumai. Dar viena ”DenseNet ”169 ir kitų ”DenseNet” architektūrų ypatybė - ”butelio kaklelių sluoksnių” naudojimas[HLV+17]. Šie sluoksniai padeda sumažinti įvesties ir išvesties ryšių kiekį ir apdorojimo sudėtingumą. Perjungiant kelis tankius blokus, ”DenseNet169” papildomai naudoja pereinamuosius sluoksnius su telkimo operacijomis, kad pakeistų požymių žemėlapio dydžius.

Kadangi šis modelis naudoja mažiau nulinio blokavimo operacijų ir sukuria mažiau parametrų nei kiti gilaus mokymosi modeliai, jo architektūra pasirodo esanti efektyvi skaičiavimų ir atminties panaudojimo požiūriu. Be to, ji padeda išspręsti nykstančio gradiento problemą, su kuria dažnai susiduriama giliuosiuose neuroniniuose tinkluose, todėl šiuos tinklus paprasčiau optimizuoti.



30 pav. Mokymosi ir validacijos tikslumas



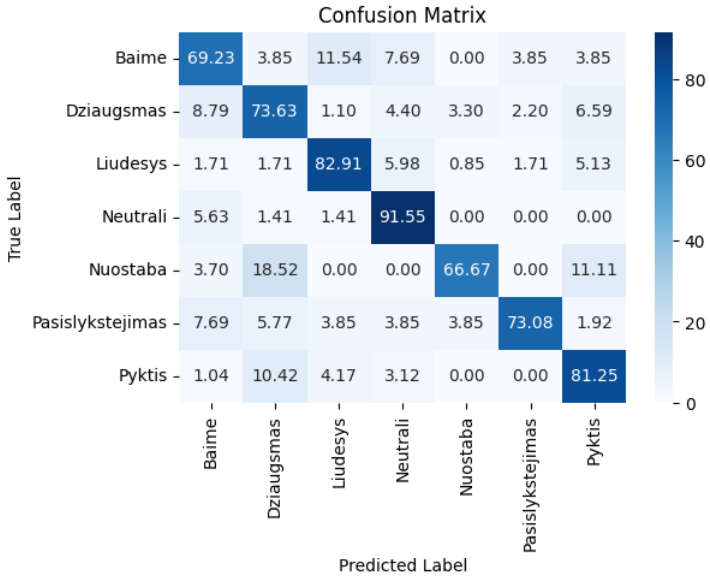
31 pav. Mokymosi ir validacijos nuostoliai

Šio modelis turėjo 12,496,135 mokamuosius parametrus. Apmokant DenseNet169 modelį

su emocijų atpažinimo užduotimi naudojant spektrogramas, per 3 epochas pasiekė 100% tikslumą su mokomaisiais duomenimis. Toks aukštas tikslumas rodo, kad modelis sugeba puikiai prisitaikyti prie mokymosi duomenų ir efektyviai atpažinti emocijas. Tačiau, kaip ir visada, svarbu atsiminti, kad mokomieji duomenys nėra viskas – modelis turi būti efektyvus ir naudojant nežinomus, anksčiau neregėtus duomenis.

Validavimo tikslumas taip pat greitai augo ir per 25 epochas buvo pasiektas 79.3% tikslumas. Tai yra didžiausias tikslumas iš visų mokytų modelių šiame tyrime, rodo, kad DenseNet169 yra labai stiprus kandidatas emocijų atpažinimui naudojant spektrogramas.

Pažvelgus į klasifikavimo lentelę matyti, kad nuostabos ir baimės emocijos buvo sudėtingiausios modeliui atpažinti. Tačiau net šios klaidos yra gana mažos, ir DenseNet169 modelis vis tiek sugebėjo atlikti solidų darbą spėjant šias emocijas.



32 pav. DenseNet169 Klasifikavimo lentelė

3.6. Tyrimo išvados

Per šį tyrimą buvo išmokytos ir palygintos įvairios giliųjų neuroninių tinklų architektūros, skirtos emocijų atpažinimui iš spektrogramų. Modeliai buvo išmokomi naudojant skirtingus tinklus, įskaitant EfficientNetB2, EfficientNetV2B2, EfficientNetV2S, VGG19, ResNet50 ir DenseNet169.

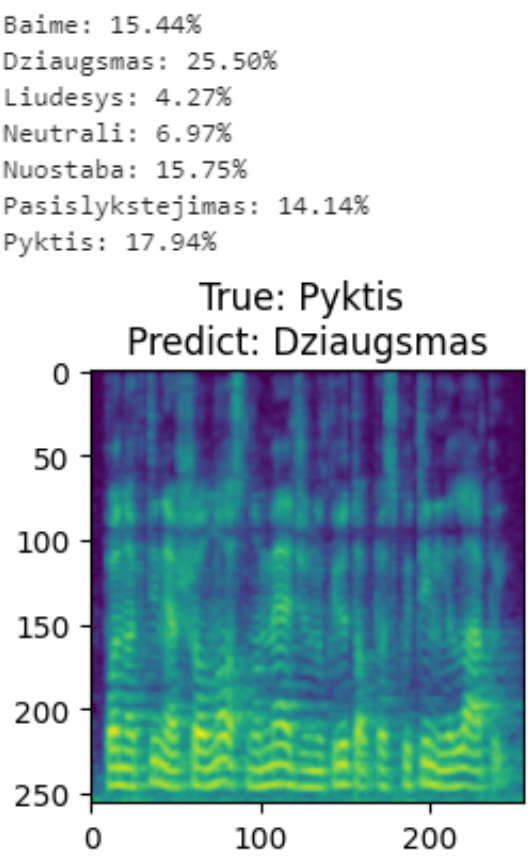
Žinant duomenų rinkinio mažą kiekį, modeliai parodė gerus rezultatus, tačiau DenseNet169 architektūra parodė aukščiausią tikslumą, preciziją, atkūrimo statistiką ir F1 statistiką tarp visų modelių. DenseNet169 modelis pasiekė 79,3% tikslumą, 80,5% preciziją, 79,3% atkūrimo statistiką ir 79,6% F1 statistiką [1].

Visi modeliai daug klydo atpažindami baimė. Šia emocija dažnai maišydavo su liūdesiu ir neutralia būsena. Taip yra turbūt dėl to, kad šių emocijų garso signalai turi panašių bruožų. Taip pat nuostaba dažnai maišoma su džiaugsmu arba pykčiu, pasišlykštėjimas su neutralia emocija. Šios klaidos neigiamai veikia modelių atpažinimo rezultatams. Viena iš sunkaus mokymosi priežasčių

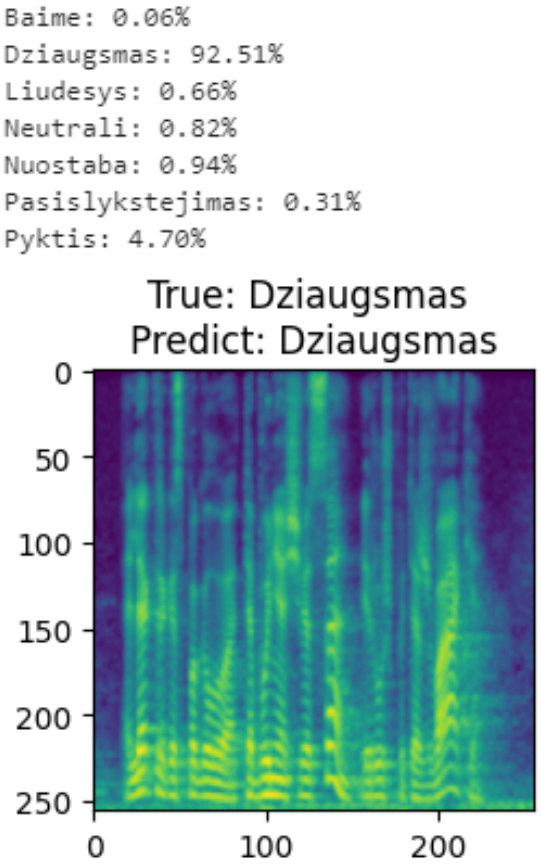
Modelių metrikos	EfficientNetB2	EfficientNetV2B2	EfficientNetV2S	VGG19	ResNet50	DenseNet169
Tikslumas	0.593	0.639	0.625	0.713	0.652	0.793
Precizižkumas	0.641	0.646	0.631	0.72	0.657	0.805
Atkūrimo statistika	0.593	0.639	0.625	0.713	0.652	0.793
F1 statistika	0.585	0.64	0.618	0.712	0.646	0.796

1 lentelė. Skirtingų modelio architektūrų metrikų palyginimas

gali būti turimas duomenų rinkinys. Duomenys sudarė 10 neprofesionalių aktorių balso įrašai, taigi emocijos gali būti ne visiškai tiksliai suvaidintos ir būti panašios į kitas emocijas. Modelio spėjimų tikimybes galima pamatyti [33] ir [34]. Paveikslėliuose matosi kai kurių spėjimų neužtikrintumas, kai visos emocijos gauna po panašia tikimybę. Taip ir užtikrintą spėjimą, kai modelis daugiau nei 90% užtikrintas savo spėjimu.



33 pav. Klasių spėjimo tikimybės



34 pav. Klasių spėjimo tikimybės

Rezultatai ir išvados

Šio darbo metu buvo sukurti ir išmokinti gilūs konvoliuciniai neuroniniai tinklai, kurie pagal balso spektrogramas atpažintų septynias pagrindines emocijas: baimę, džiaugsmą, liūdesį, neutralią būseną, nuostabą, pasišlykštėjimą ir pyktį. Kadangi balso emocijų atpažinimas yra sudėtinga problema, įvairūs modeliai buvo išbandyti siekiant rasti tinkamiausią sprendimą.

Konvoliucinių neuroninių tinklų modeliai, pasirinkti eksperimentams, įtraukė EfficientNetB2, EfficientNetV2B2, EfficientNetV2S, VGG19, ResNet50 ir DenseNet169. Kiekviena iš šių architektūrų buvo iš anksto apmokyta, suteikiant pradinį gebėjimą atpažinti vizualinius požymius, o tada buvo toliau mokyta naudojant mūsų spektrogramų duomenis.

Modelių veikimo vertinimas buvo atliktas pagal kelių metrikų, įskaitant tikslumą, preciziškumą, atkūrimo statistiką ir F1 statistiką, vertinimus. Šios metrikos leidžia įvertinti, kaip gerai modelis atpažino kiekvieną emociją. DenseNet169 modelis pasiekė geriausius rezultatus tarp visų išbandytų modelių, pasiekdamas 79,3% tikslumą. Blogiausią rezultatą parodė EfficientNetB2 - 59.3% tikslumą. Tai gali būti todėl nes šis mažiausias turintis tik 7,710,857 apmokomų parametrų.

Nepaisant šio aukšto tikslumo, visi modeliai susidūrė su sunkumais atpažindami tam tikras emocijas. Pavyzdžiui, baimės emocija buvo dažnai painiojama su liūdesiu ar neutralia būsena. Be to, neutralios emocijos dažnai buvo klaidingai identifikuojamos kaip nuobodulio, o džiaugsmo emocija buvo painiojama su pykčiu.

Atsižvelgiant į šiuos rezultatus, būtina atlikti tolesnius tyrimus, siekiant gerinti modelių gebėjimą atpažinti emocijas. Vienas iš galimų problemos sprendimo būdų yra apmokyti modelius su didesniu ir įvairesniu duomenų rinkiniu. Svarbu, kad šie rinkiniai apimtų platų asmenų spektrą, įskaitant skirtingas amžiaus grupes, lytis, kalbas ir tautybes. Tai leistų užtikrinti modelių atpažinimo gebėjimo universalumą. Be to, svarbu atsižvelgti į skirtingas socialines, kultūrines ir individualias emocijų raiškos ypatybes.

Šaltiniai

[AI] AI Wiki. *Activation Function - AI Wiki* [https://machine-learning.paperspace.com/wiki/activation-function]. [Sine anno]. [žiūrėta 2023-05-20].

[AON+09] M. k. Alsmadi, K. B. Omar, S. A. Noah, I. Almarashdah. Performance Comparison of Multi-layer Perceptron (Back Propagation, Delta Rule and Perceptron) algorithms in Neural Networks. Iš: *2009 IEEE International Advance Computing Conference*. 2009, p. 296–299. Prieiga per internetą: https://doi.org/10.1109/IADCC.2009.4809024.

[AZH+21] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili ir kiti. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*. 2021-03, tomas 8, numeris 1, p. 53. ISSN 2196-1115. Prieiga per internetą: https://doi.org/10.1186/s40537-021-00444-8.

[CFC+17] K. Choi, G. Fazekas, K. Cho, M. B. Sandler. A Comparison on Audio Signal Preprocessing Methods for Deep Neural Networks on Music Tagging. *CoRR*. 2017, tomas abs/1709.01922. Prieiga per internetą: http://arxiv.org/abs/1709.01922.

[ÇN21] A. N. Çayır, T. S. Navruz. Effect of Dataset Size on Deep Learning in Voice Recognition. Iš: *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. 2021, p. 1–5. Prieiga per internetą: https://doi.org/10.1109/HORA52670.2021.9461395.

[DV90] P. Duhamel, M. Vetterli. Fast Fourier transforms: a tutorial review and a state of the art. *Signal processing*. 1990, tomas 19, numeris 4, p. 259–299.

[GFC+21] S. Gang, N. Fabrice, D. Chung, J. Lee. Character Recognition of Components Mounted on Printed Circuit Board Using Deep Learning. *Sensors*. 2021-04, tomas 21, p. 2921. Prieiga per internetą: https://doi.org/10.3390/s21092921.

[HLV+17] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger. Densely Connected Convolutional Networks. Iš: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, p. 2261–2269. Prieiga per internetą: https://doi.org/10.1109/CVPR.2017.243.

[HZD+21] X. Han, Z. Zhang, N. Ding, Y. Gu ir kiti. Pre-trained models: Past, present and future. *AI Open*. 2021, tomas 2, p. 225–250. ISSN 2666-6510. Prieiga per internetą: https://doi.org/https://doi.org/10.1016/j.aiopen.2021.08.002.

[JD20] L. Jin, M. Demoret. *Jump-start AI Training with NGC Pretrained Models On-Premises and in the Cloud*. 2020. [žiūrėta 2023-01-15]. Prieiga per internetą: https://developer.nvidia.com/blog/jump-start-ai-training-with-ngc-pretrained-models-on-premises-and-in-the-cloud/.

[KNS+21] F. Kamalov, A. Nazir, M. Safaraliev, A. K. Cherukuri, R. Zgheib. Comparative analysis of activation functions in neural networks. Iš: 2021-11, p. 1–6. Prieiga per internetą: https://doi.org/10.1109/ICECS53924.2021.9665646.

[KQ22] A. Khattar, S. Quadri. "Generalization of convolutional network to domain adaptation network for classification of disaster images on twitter". *Multimedia Tools and Applications*. 2022-09, tomas 81. Prieiga per internetą: https://doi.org/10.1007/s11042-022-12869-1.

[LK20] K. H. Lee, D. H. Kim. Design of a Convolutional Neural Network for Speech Emotion Recognition. Iš: *2020 International Conference on Information and Communication Technology Convergence (ICTC)*. 2020, p. 1332–1335. Prieiga per internetą: https://doi.org/10.1109/ICTC49870.2020.9289227.

[QSX+20] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*. 2020-10, tomas 63, numeris 10, p. 1872–1897. ISSN 1869-1900. Prieiga per internetą: https://doi.org/10.1007/s11431-020-1647-3.

[TDL+22] L. Trinh Van, T. Dao Thi Le, T. Le Xuan, E. Castelli. Emotional Speech Recognition Using Deep Neural Networks. *Sensors*. 2022, tomas 22, numeris 4. ISSN 1424-8220. Prieiga per internetą: https://doi.org/10.3390/s22041414.

[Ten15] Tensorflow. *Tensorflow source$_w indows$*. 2015. [žiūrėta 2023-01-10]. Prieiga per internetą: https://www.tensorflow.org/install/source_windows#gpu.

[Vad21] P. Vadapalli. *Biological Neural Network: Importance, Components Comparison*. 2021. [žiūrėta 2023-01-15]. Prieiga per internetą: https://www.upgrad.com/blog/biological-neural-network/.

[Woond] A. Woodruff. *What is a neuron?* n.d. [žiūrėta 2023-01-15]. Prieiga per internetą: https://qbi.uq.edu.au/brain/brain-anatomy/what-neuron.