

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
INFORMATIKOS INSTITUTAS  
PROGRAMŲ SISTEMŲ KATEDRA

# **Objektų identifikavimas iš audio srauto**

## **Object identification from audio stream**

Magistro baigiamasis darbas

Atliko: Donatas Kurapkis (parašas)

Darbo vadovas: lekt. Irus Grinis (parašas)

Darbo recenzentas: Linas Petkevičius (parašas)

Vilnius 2018

## Turinys

Padėkos .....	4
Santrauka.....	5
Summary .....	6
Įvadas .....	7
1. Tiriamasis objektas .....	7
2. Tyrimo aktualumas ir praktinis pritaikymas.....	7
3. Darbo tikslai ir uždaviniai .....	9
4. Tyrimo metodika .....	9
4.1. Literatūros analizė .....	10
4.2. Duomenų rinkimo metodika .....	10
4.3. Praktinės dalies įgyvendinimo metodika .....	10
5. Naudojami įrankiai .....	10
1. Teorinė dalis.....	11
1.1. Bendrieji skaitmeninės informacijos apdorojimo metodai ir algoritmai.....	11
1.1.1. Signalai ir jų tipai .....	11
1.1.2. Furje transformacijos .....	12
1.1.3. Morlet bangelių transformacija.....	13
1.2. Filtrai .....	14
1.2.1. Filtrų klasifikacija .....	14
1.3. Sistemos mokymosi algoritmai .....	15
1.3.1. Sprendimų medžio .....	15
1.3.2. Dirbtiniai neuroniniai tinklai.....	16
1.3.3. Giluminis mokymasis .....	19
1.4. Automobilio variklio skleidžiamo garso struktūra.....	21

1.5. Praktiniai darbai tiesiogiai susieti su magistro darbo uždaviniu .....	21
2. Praktinė dalis .....	25
2.1. Teorinės darbo prielaidos .....	25
2.2. Duomenų rinkimas .....	25
2.2.1. Įrašų sandara.....	27
2.3. Sistemos modelis.....	28
2.4. Duomenų transformavimas (spektrinių charakteristikų gavimas) .....	28
2.4.1. Audio įrašų skaidymas .....	29
2.4.2. Audio įrašų spektrinių charakteristikų gavimas.....	29
2.5. Duomenų klasifikavimas.....	30
2.5.1. Pradinių duomenų klasifikavimas su [ELI] „Audfprint“ programa. ....	30
2.5.2. Klasifikavimas su „Keras“ giliojo mokymosi biblioteka.....	33
2.5.3. Tyrimo rezultatai.....	45
2.6. Tobulinimo galimybės .....	46
2.7. Potencialūs sistemos trūkumai .....	46
Išvados .....	47
Šaltiniai .....	48
Santrumpos.....	53
Priedai .....	54

## **Padėkos**

Norėčiau padėkoti visiems įmonės Baltic Amadeus darbuotojams, kurie padėjo surinkti didžiąją dalį mokslinio darbo tyrimui reikalingų duomenų.

## **Santrauka**

Darbe sprendžiamas motorinių transporto priemonių variklių skleidžiamo garso klasifikavimo uždavinys. Tikslas – sukurti kompiuterizuotą sistemą gebančią identifikuoti konkrečią motorinę transporto priemonę.

Darbo eigoje apžvelgiami egzistuojantys audio srauto apdorojimo algoritmai. Aptariami skirtingi sistemų mokymosi algoritmai. Apibūdinami darbai tiesiogiai susiję su darbo tematika.

Praktinėje darbo dalyje yra išbandomas viena iš jau egzistuojančių audio srauto atpažinimo programų. Detaliai aprašomas sukurtasis giliojo mokymosi (angl. Deep learning) modelis. Pateikiami sukurtojo modelio testų rezultatai ir išvados.

## **Summary**

This thesis deals with the classification of engine sound propulsion by motor vehicles. The purpose is to create a computerized system which is capable of identifying a specific motor vehicle engine sound.

In thesis theory part is overviewing existing audio stream processing algorithms. Discussed about different machine learning algorithm. Describes other works directly related to thesis subject.

In practical part is trying out one of already existing audio sound recognition programs. Later providing detailed description of developed deep learning model, tests results and conclusion of it.

# **Įvadas**

Keičiantis informacinių technologijų pasauliui – keičiasi ir žmogus. Dirbtiniu intelektu paremti algoritmai perima vis daugiau žmogaus darbų. Viena iš aktualiausių ir labiausiai plėtojamų sričių yra objektų atpažinimas.

## **1. Tiriamasis objektas**

Darbo tiriamasis objektas yra techninių objektų – motorinių transporto priemonių variklių skleidžiami garso įrašai, jų analizė, bei konkretaus automobilio variklio skleidžiamo garso atpažinimas, kitų motorinių transporto variklių skleidžiamų garsų aplinkoje, naudojantis egzistuojančiais audio/video srautą apdorojančiais algoritmais.

Motorinių transporto priemonių skleidžiami garsai pasižymi:

- plačiu garsiniu spektru;
- skleidžiamas garsas girdimas kitų motorinių transporto priemonių aplinkoje;
- garsas yra netolygus dėl viso rinkinio įvairių priežasčių: skirtingos variklio apsukos, transmisijos keliamas triukšmas, išmetimo sistemos specifikos;
- priklauso nuo šių veiksnių: aplinkos temperatūros, variklio susidėvėjimo, kuro, tepalų kokybės, oro drėgnumo ir kt.

Garso informacija dažnai gali būti naudojama įvairiems tikslams pasiekti, kurių svarbiausias yra objekto identifikavimas. Kartu tai yra raktas į socialiai orientuotą, efektyvų ir natūralų robotų bendravimo būdą su žmonėmis [SSS+12].

Bendroji šiame darbe sprendžiama problema yra minėtų garso objektų identifikavimas naudojantis vadinamaisiais šablonais (angl. Patterns).

Šablonas šiame darbe apibrėžiamas kaip konkrečios motorinės transporto priemonės variklio skleidžiamas garso įrašas. Įrašė girdima tik konkrečios motorinės transporto priemonės variklio skleidžiamas garsas. Įrašas sudaromas aplinkoje, kurioje idealiu atveju nėra pašalinių garso skleidžiančių objektų.

## **2. Tyrimo aktualumas ir praktinis pritaikymas**

Plėtojant objektų atpažinimo algoritmus, stengiamasi sukurti kuo geriau veikiančią programinę įrangą, kuri gebėtų atpažinti ir klasifikuoti objektus taip kaip tai gali padaryti žmogus, kitaip tariant siekiama sukurti dirbtinį intelektą, kuris gebėtų „mąstyti“.

Dažniausiai automobiliai yra atpažįstami naudojantis vaizdo apdorojimo algoritmais ir tam skirta technika. Naudojantis tokiais atpažinimo algoritmais yra kuriamos automatinės sistemos, kurios padeda identifikuoti objektus, juos klasifikuoti, sekti. Puikus pavyzdys yra paprastas automobilio numerio atpažinimas: kompiuterinė sistema analizuoja video srauto duomenis taikydama įvairius vaizdo apdorojimo algoritmus kurių pavyzdžiai yra: krašto atitikmens (angl. edge matching), dalink ir valdyk paieška (angl. divide-and-conquer search), palyginimas pilkų atspalvių skalėje (angl. greyscale matching), gradiento atitikmens (angl. gradient matching), mąstelio invariantiškos sąvybės transformacija (angl. scale-invariant feature transform (SIFT)) ir kitus. Minėtais algoritmais yra išgaunamas automobilio numeris, sulyginamas su duomenų bazėje esančiu ir automobilis įleidžiamas į teritoriją.

Šiame darbe bus tiriamas kitas būdas, kaip būtų galima identifikuoti automobilį – pagal jo variklio skleidžiamo garso šabloną. Ši alternatyva galėtų būti netgi pigesnė, nei naudojantis vaizdo atpažinimo, kadangi naktinio matymo ir aukštos kokybės vaizdo kameros nėra pigios lyginant su aukštos kokybės mikrofonais.

Objektų identifikacija naudojantis audio srautu nėra naujiena. Egzistuoja daugybė algoritmų analizuojančių audio srautą: Furje transformacijos (FT), Gauso mišinių modelis (angl. Gaussian mixture model), paslėptų Markovo modelių (angl. hidden Markov model) ir kiti. Kasdien naudojamos balso atpažinimo – identifikavimo programos tokios kaip „Google Speech Recognition API“, kuri yra naudojama Chrome naršyklėje nuo 25 versijos. Literatūroje galima surasti daugybę audio srauto analizavimo pavyzdžių. Šio darbo idėja buvo paskatinta remiantis šaltiniais [OE], [CE09]. Minėtuose darbuose autoriai pagal tam tikrą audio muzikos šabloną bando atpažinti dainas. Programoje yra analizuojamas audio srautas pasitelkiant „fingerprints“. „Fingerprints“ – šiuo konkrečiu atveju yra tam tikros maišos funkcijos (angl. *hash*) nuo audio srauto spektro. Spektras gaunamas pasitelkus FFT (angl. Fast Furje Transformation) algoritmą. Kaip nebūtų keista pabandžius šią programą su variklių skleidžiamais garsais irgi matomas tam tikras rezultatyvumas.

Tai nėra vieninteliai darbai, kuriuose bandoma atpažinti tam tikrus objektus audio sraute. Prie jų galima paminėti tokius projektus: [WAN], [KHS], [BWA]. Minėtuose projektuose yra analizuojamas audio srauto panaudojimas kompiuterinėje regoje.

Objektų identifikavimas iš audio srauto gali būti itin sėkmingai panaudojamas miestuose – siekiant užtikrinti saugumą vaizdo kameromis neapdrėpiamuose plotuose. Taip pat gali būti naudojamas kaip papildomas objekto atpažinimas kartu su vaizdo srautu. Toks derinys galėtų



išplėsti objektų atpažinimo galimybes, kadangi vaizdo kameros turi ribotą matomumo kampą, o tuo tarpu mikrofonus galima paslėpti dėl garso bangų savybės užlinkti už kliūties.

Praktikoje kuriamoji sistema galėtų būti pritaikyta šiose srityse:

- Išmaniųjų vartų automatikai (nenaudojant jokių nuotolinio valdymo pultelių) – garažo, kiemo vartai, būtų automatiškai atidaromi atpažinus konkrečią motorinę transporto priemonę;
- Automobilių paieškos sistemoje – jeigu jūsų automobilis būtų pavogtas, jums užtektų įkelti jo variklio skleidžiamą garsą į paieškos sistemą. Mikrofonus būtų galima išdėstyti prie kiekvienos sankryžos. Jie siųstų duomenis į centrinį serverį, kuris ieškotų jūsų pateiktojo šablono;
- Išmaniosioms minoms (perimetro apsaugai) – sistema galėtų automatiškai deaktivuoti užminuotą minų lauką toms motorinėms transporto priemonėms, kurios yra sąrašė.

Taip pat verta paminėti elektromobilius, kurie dabar neskleidžia jokio garso. Tačiau anksčiau ar vėliau jie turės skleisti tam tikrą garsą [HUG].

### **3. Darbo tikslai ir uždaviniai**

Darbo tikslas yra sukurti veikiančios kompiuterizuotos sistemos prototipą, kuris audio sraute gebėtų identifikuoti konkrečią motorinę transporto priemonę, pagal jos variklio skleidžiamą garsą, kitų motorinių transporto priemonių skleidžiamuose variklių garsuose, naudojantis ieškomosios motorinės transporto priemonės variklio garso šablonu (angl. Pattern).

Siekiant šio tikslo buvo sprendžiami uždaviniai:

- Atlikti žinomų objektų identifikavimo algoritmų, modelių ir sistemų, bei atitinkamos literatūros analizę.
- Surinkti motorinių transporto priemonių variklių skleidžiamus garsus, reikalingus sistemai apmokyti.
- Sukurti paieškos modelį, gebantį identifikuoti tam tikros motorinės transporto priemonės variklio skleidžiamą garsą.

### **4. Tyrimo metodika**

Tyrimo atlikimo metodika susidėjo iš šių punktų:

- Literatūros analizė;
- Duomenų rinkimo;
- Praktinės dalies įgyvendinimo.

#### **4.1. Literatūros analizė**

Šioje dalyje buvo apžvelgti egzistuojantys audio srauto apdorojimo algoritmai, sistemų mokymosi algoritmai, bei tiesiogiai su darbo tema susiję darbai.

#### **4.2. Duomenų rinkimo metodika**

Duomenys reikalingi praktinės dalies įgyvendinimui buvo renkami atsitiktinėmis dienomis realiomis sąlygomis (girdimi aplinkos garsai). Duomenų rinkimas rinkimo metu buvo fiksuojami šie parametrai:

- Apytikslė oro temperatūra.
- Apytikslė variklio temperatūra.
- Komentarai.

#### **4.3. Praktinės dalies įgyvendinimo metodika**

- Teorinių darbo prielaidų.
- Duomenų transformavimas.
- Sistemos modelio sukūrimo.
- Modelio testų.

### **5. Naudojami įrankiai**

Dėl lengvai išmokstamos, bei plačiai naudojamos dirbtinio intelekto programose – praktinės dalies įgyventinimui buvo pasirinkta python programavimo kalba.

Dirbtinio intelekto programai sukurti pasirinktas „*Keras*“ sistemos karkasas (angl. framework). Šis karkasas supaprastina „*TensorFlow*“ ir „*Theano*“ bibliotekų naudojimą.

Taip pat buvo naudojama populiari *Anaconda* python programavimo aplinka, veikianti įvairiose operacinėse sistemose.

Duomenų surinkimui buvo naudojamas paprastas mobilusis telefonas *Samsung Galaxy xcover 2*. Mobiliojo telefono įrašymo įrenginys pasirinktas dėl plataus paplitimo visuomenėje t.y., norint pasinaudoti sukurta programa užtenka telefonu įrašyti motorinės transporto priemonės variklio skleidžiamą garsą. Darbo metu daroma prielaida, jog visų mobiliųjų telefonų garso įrašymo kokybė yra panaši, su kitais telefonų modeliais motorinių transporto priemonių variklių skleidžiami garsai nebuvo įrašinėjami.

# 1. Teorinė dalis

Šioje darbo dalyje yra nagrinėjami teoriniai aspektai reikalingi praktinės dalies įgyvendinimui. Taip pat apžvelgiami praktiniai darbai tiesiogiai susiję su magistro darbo uždaviniu.

## 1.1. Bendrieji skaitmeninės informacijos apdorojimo metodai ir algoritmai

Absoliuti dauguma šiandieninės informacijos apdorojimo algoritmų skirta skaitmeninei informacijai. Norint gauti duomenis apie mus dominantį reiškinį, priklausomai nuo jo prigimties ir turimų stebėjimo priemonių, galimi du signalų apdorojimo variantai:

- Betarpiškai apdoroti.
- Originalų signalą pakeisti kitos rūšies signalu – pavyzdžiui, į garso plokštę atėjęs signalas yra analoginis ir tam, kad jį būtų galima apdoroti CPU pagalba jį reikia skaitmeninti.

Tam tikra darbo dalis bus skiriama esamų skaitmeninės informacijos apdorojimo algoritmų apžvalgai.

### 1.1.1. Signalai ir jų tipai

Signalai skirstomi į keturias grupes [LL97]:

- analoginiai signalai (*tolydaus laiko ir tolydžiosios amplitudės*);
- diskretiniai signalai (*diskretaus laiko ir tolydžiosios amplitudės*);
- kvantuotieji signalai (*tolydaus laiko ir diskrečiosios amplitudės*);
- skaitmeniniai signalai (*diskretaus laiko ir diskrečiosios amplitudės*).

Analoginiai signalai yra aprašomi konkrečiu laiko intervalu. Kaip pavyzdį analoginį signalą galima paminėti: mikrofono išėjimo signalą, temperatūros, slėgio jutiklių išėjimo signalus. Šie signalai paprastai atspindi realių fizikinių reiškinų elektrinį modelį. Dar kitaip vadinami tolydaus laiko ir tolydžiosios amplitudės signalais.

Diskretieji signalai yra apibrėžti tik tam tikrais diskrečiais laiko momentais. Jie gali būti gauti iš analoginių signalų, atskaitant jo reikšmes diskretizacijos žingsniu [PVE]. Iš esmės diskretus signalas yra skaičių rinkinys. Tokio signalo pavyzdys gali būti reguliarus temperatūros stebėjimas (kas valandą, parą) meteorologinėje tarnyboje. Kartais jie vadinami diskretaus laiko ir tolydžiosios amplitudės signalais.

Kvantuotieji signalai (tolydaus laiko ir diskrečiosios amplitudės) gali įgyti tik tam tikras iš anksto apibrėžtas reikšmes (pvz., 1,2,3). Tokie signalai gaunami tolydujį signalą kvantuojant pagal amplitudę ir jį pakeičiant laiptine stačiakampe funkcija. Tokių signalų galimų amplitudės reikšmių skaičius yra ribotas.

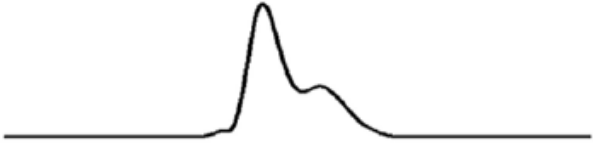
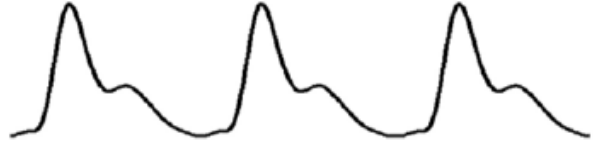

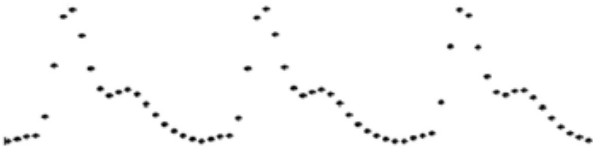
Skaitmeniniai signalai yra kvantuojami pagal amplitudę ir atskaitomi diskrečiais laiko momentais. Jie gaunami iš analoginių signalų analoginiu-kodiniu keitikliu (A/K), juos kvantuojant pagal amplitudę ir diskretizuojant pagal laiką. Kartais jie vadinami diskretaus laiko ir diskrečios amplitudės signalais.

Toliau teorinėje dalyje yra nagrinėjami skaitmeniniai signalai. Vienas iš plačiausiai naudojamų metodų apdorojant skaitmeninius signalus yra Furje transformacijos ir skaitmeniniai filtrai.

### **1.1.2. Furje transformacijos**

Furje transformaciją galima suskaidyti į keturias dalis, priklausomai nuo analizuojamo signalo tipo. Pastarieji gali būti dviejų tipų: tolydieji arba diskretieji, o šie gali būti periodiniai arba neperiodiniai. Kombinuojant šiuos signalų rūšis gaunamas toks transformacijų sąrašas [SMI]:

- Tolydus – neperiodinis (Furje transformacija).
- Tolydus – periodinis (Furje eilutė).
- Diskretus – neperiodinis (diskretaus laiko Furje transformacija).
- Diskretus – periodinis (diskrečioji Furje transformacija).

Transformacijos tipas	Signalų tipas
<b>Furje transformacija</b> neperiodinis tolydus signalas	
<b>Furje eilutė</b> periodinis tolydus signalas	
<b>Diskretaus laiko Furje transformacija</b> diskretus neperiodinis signalas	
<b>Diskrečioji Furje transformacija</b> diskretus periodinis signalas	

1 pav. Furje transformacijų tipai ir jų signalų pavyzdžiai [SMI]

Bendrai tariant Furje transformacijos skirtos sudėtingos formos signalą išskleisti į keletą kitų tipų signalų, taip suteikiant galimybę daryti vienokias ar kitokias išvadas apie signalą.

### 1.1.3. Morlet bangelių transformacija

Morlet transformacija – tai bangelės sudarytos iš kompleksinės eksponentės padaugintos iš Gauso lango funkcijos. Ši transformacija yra glaudžiai susijusi su žmogaus garso ir vaizdo suvokimu [MAL].

Morlet transformacijos formulė [TAN]:

$$\psi(t) = \frac{1}{\sqrt[4]{\pi}} \left( e^{j\omega_0 t} - e^{-\frac{\omega_0^2}{2} t^2} \right) e^{-\frac{t^2}{2}}$$

Kur  $\omega_0$  yra dažnis ir  $\sigma$  yra nešėjo matas, o  $t$  – laikas, kur  $t > 0$ .

Morlet transformacija yra viena iš populiariausių kompleksinių bangelių (angl. Complex wavelet) transformacijų naudojamų praktikoje [TAN].

Transformacija naudojama:

- Medicinoje [TAN].
- Muzikos atpažinime [MPV].
- Kvantinėje mechanikoje [ASH].

## **1.2. Filtrai**

Kartais taip nutinka, jog skaitmeninis signalas būna sugadintas, t.y. jame būna didelis triukšmas, arba kitaip paveiktas aplinkos. Kartais tiesiog reikia išgauti tam tikrą informaciją. Tokiu atveju į pagalbą pasitelkiami skaitmeniniai filtrai. Pastarieji yra vieni svarbiausių signalų apdorojimo elementų. Skaitmeniniai filtrai plačiai paplitę tokiose srityse kaip:

- Kalbos signalų apdorojimas;
- Skaitmeninio ryšio realizacija;
- Elektros prietaisų valdymas;
- Vaizdų perdavimas ir apdorojimas.

Filtrai gali padėti atskirti (išfiltruoti) vertingą informacijos dalį nuo nevertingos (pavyzdžiui ieškant tik tam tikro dažnio informacijos, visa kita informacija būtų nenaudojama) ir taip pasitarnauti įgyvendinant sistemos prototipą.

Bendrai sąvoka filtras yra suvokiama kaip tam tikros operacijos pritaikymas signalui norint gauti tik aktualią informaciją, pavyzdžiui: įėjimo signalui pritaikius žemųjų dažnių filtrą jame bus išfiltruoti tik žemi dažniai t.y. išėjimo signale neliks aukštų dažnių.

### **1.2.1. Filtrų klasifikacija**

Skaitmeniniai filtrai gali būti realizuoti dviem būdais (2 pav.): naudojantis sąsūkos operacija, arba naudojantis rekursija.

Šios dvi filtrų realizacijos dar skirstomos į tris dalis, pagal tai kam yra naudojamas filtras (2 pav.): laiko domeną, dažnio domeną ir vartotojo apibrėžtus filtrus.



2 pav. Filtrų klasifikacija [SMI]

### 1.3. Sistemos mokymosi algoritmai

Mokymosi algoritmai yra skirstomi į tris dalis [DIR16]:

- su mokytoju (Supervised Learning) – reikalingi įėjimo, bei tikslo duomenys.
- be mokytojo (Unsupervised Learning) – algoritmui suteikiama visiška savaiminio mokymosi laisvė. Nėra duodami jokie nurodymai.
- hibridinis (Hybrid Learning) – algoritmas nėra pilnai apmokomas surasti teisingą rezultatą. Pavyzdys: mokymasis žaisti žaidimą žaidžiant prieš oponentą.

Egzistuoja nemažai skirtingų sistemų mokymosi su mokytoju algoritmų:

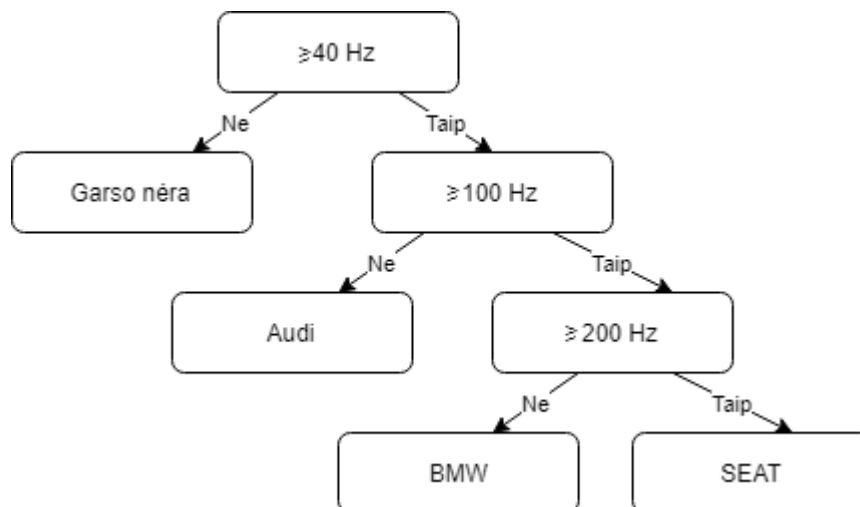
- sprendimų medžio;
- dirbtiniai neuroniniai tinklai;
- giluminis mokymasis;
- atraminių vektorių metodas;
- atsitiktinis miško klasifikatorius.

#### 1.3.1. Sprendimų medžio

Sprendimų medžiai (angl. Decision tree (DT)) – yra neparimetriniai prižiūrimi mokymosi algoritmai naudojami klasifikacijai ir regresijai. DT mokymosi algoritmai naudoja sprendimų medžius kaip prognozuojamus modelius, kurie susiveda į pastabas apie nagrinėjamą objektą. Šis metodas yra plačiai naudojamas duomenų analizėje (angl. Data mining) [RM]. Sprendimų medžio tikslas yra sukurti modelį, kuris nuspėtų ieškomo kintamojo reikšmę pasinaudodamas keletu įvesties kintamųjų.

DT būtų galima pritaikyti mokslo tiriamojo darbo problemai spręsti. Kadangi sprendimų medžio algoritmas yra prižiūrimasis, tokiu atveju galima turėti tam tikrą atskirų transporto priemonių skleidžiamų audio įrašo bazę, o tam tikra funkcija  $F(x)$  gebėtų tuos garsus klasifikuoti.

Tarkime mašinos skleidžia harmoninius signalus. Tokiu atveju sprendimų medis galėtų būti panašus į 3 pav. pavaizduotą sprendimų medžio diagramą.



3 pav. Sprendimų medžio pavyzdys

#### 1.3.1.1. Taikymo pavyzdžiai

DT – gali būti sėkmingai panaudojamas ir audio srauto apdorojime. Pavyzdžiui: kalbos atpažinime [AMU01], kalbos klaidų atpažinime, muzikos atpažinime. Darbe „A Decision-Tree-Based Algorithm for Speech/Music Classification and Segmentation“ [LRU09] yra pateikiami atitinkami pavyzdžiai.

DT algoritmas taip pat yra realizuotas Python scikit-learn bibliotekoje, todėl implementuojant šį algoritmą užtektų tiesiog kviesiti minėtosios bibliotekos turimus metodus.

#### 1.3.2. Dirbtiniai neuroniniai tinklai

Dirbtiniai neuroniniai tinklai (angl. Artificial neural networks (ANN)) – mokymosi algoritmai, kurie stengiasi imituoti biologinius neuroninius tinklus.

Tradicinis skaitmeninių signalų apdorojimas (angl. Digital signal processing) remiasi algoritmais, kurie keičia vienos rūšies duomenis kitos rūšies duomenimis, turi statinius keitimo parametrus. Neuroniniai tinklai gali padėti parinkti tinkamus parametrus algoritmams. Rezultate programos algoritmas gebėtų duoti tikslesnius atsakymus.

ANN gali būti dviejų tipų [MAC03]:



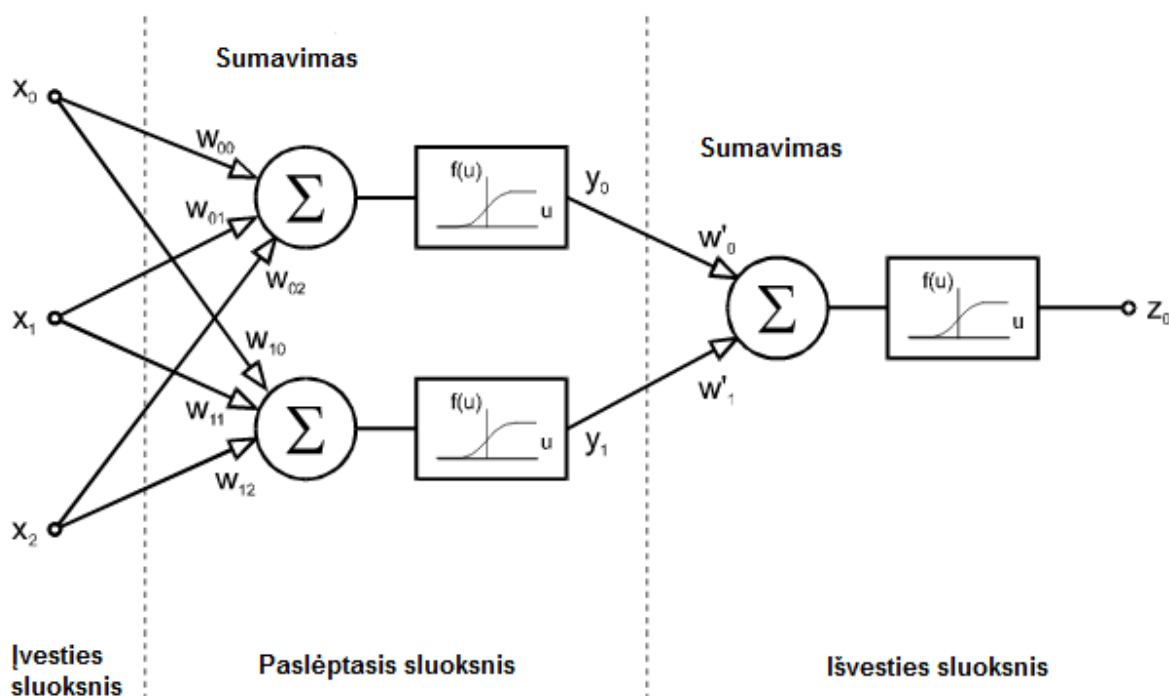
- Vienpusio ryšio (angl. Feed-forward) – tinkluose įvesties signalas gali keliauti tik į vieną pusę. Išėjties rezultatai tiesiogiai priklauso nuo įvesties. Šio tinklo apmokymo algoritmas yra realizuotas Python Pybrain bibliotekoje;
- Grįžtamojo ryšio (angl. Feedback) – tinklai pasižymi signalo galimybe keliauti į abi puses. Šio tipo neuroniniai tinklai yra žymiai sudėtingesni, nei vienpusio ryšio, kadangi yra dinaminiai ir jų būsena nuolat keičiasi iki tol kol jie pasiekia pusiausvyros tašką. Įvesties signalui pasikeitus būtina vėl viską perskaičiuoti iki pusiausvyros taško.

Dažniausiai neuroninių tinklų architektūrą sudaro tryjų tipų sluoksniai:

- Įvesties – dar kitaip vadinamas pavyvioji (angl. Passive) kadangi šiame sluoksnyje duomenys nėra modifikuojami.
- Paslėptieji – aktyvieji sluoksniai, nes yra modifikuojami duomenys.
- Išvesties – aktyvusis sluoksnis, nes yra modifikuojami duomenys.

Įvesties sluoksnyje įvesties duomenims pritaikomi svoriai ir jie sumuojami. Po to gauti rezultatai keliauja į pirmą palėptą sluoksnį, kur vėl bus sumuojami su tam tikrais svoriais ir taip toliau. Kiekvienas neuronas po svartinio sumavimo pateikia rezultatą sigmoidinei funkcijai:

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

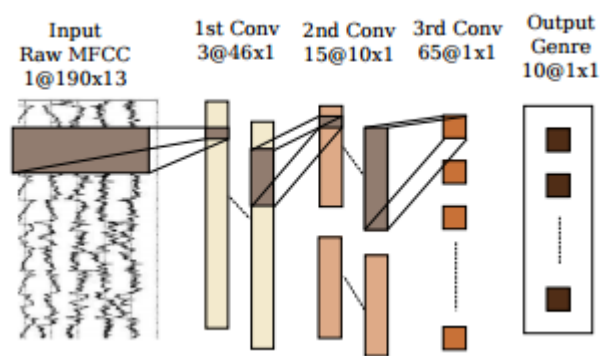


4 pav. Tipinis neuroninių tinklų modelis [NIE93]

### 1.3.2.1. Taikymo pavyzdžiai

Praktinis ANN pritaikymas yra beribis. Juos galima naudoti analizuojant bet kokius duomenis, siekiant simuliuoti žmogaus smegenų veiklą. Audio srauto analizavime gali būti pritaikytas kaip automatizuotas muzikos klasifikatorius, nustatant garso kokybę [NIE93], taip pat gali padėti įgyvendinti tiriamąjį darbą.

Vienas iš praktinių pavyzdžių, kai neuroninių tinklų viduje yra panaudojamos kitos operacijos, pavyzdžiui sąsūkos minimi [LCC10]. Darbo tikslas buvo pademonstruoti, jog audio srautui apdoroti galima naudoti tuos pačius metodus kaip ir vaizdo apdorojime. Autoriai naudodamiesi sąsūkos neuroniniais tinklais (angl. Convolutional neural network (CNN)) pademonstravo automatinį muzikos klasifikavimo algoritmą pagal muzikos žanrą. Sukonstruota CNN architektūra yra pavaizduota 5 pav.



5 pav. CNN architektūra muzikiniam šablonui gauti naudojantis MFCC [LCC10].

Architektūra sudaryta iš penkių sluoksnių, įskaičiuojant įvesties ir išvesties sluoksnius. Pirmasis sluoksnis yra MFCC – Melo dažnių Cepstral koeficientų algoritmas (angl. el Frequency Cepstral Coefficient) . Antrasis yra sąsūkos sluoksnis iš trijų skirtingų branduolių, tačiau vienodo ilgio. Trečiasis ir ketvirtasis labai panašus į antrą tik su kitokiu branduolio funkcijos dydžiu. Penktasis sluoksnis yra išvesties.

Į CNN tinklą patenka duomenys apdoroti MFCC algoritmu. Šis algoritmas yra plačiai naudojamas automatiniam kalbos atpažinimui. Naudojantis juo yra išgaunami unikalūs audio įrašo komponentai (detalės), kurie gerai tinka lingvistinio turinio identifikavimui [LCC10].

### 1.3.3. Giluminis mokymasis

Vieni iš populiariausių sistemų mokymosi algoritmų yra giluminio mokymosi metodai (angl. Deep learning (DL)). Šie algoritmai susideda iš daugybės paslėptų sluoksnių, kuriuose yra neuroniniai tinklai.

DL – paremtas algoritmų rinkiniu, kurie bando modeliuoti aukšto lygio abstrakcijas duomenyse, naudojant gilius grafus su keletu apdorojimo sluoksnių, sudarant keletą ne linijinių transformacijų [GBC16][DYU].

Bendrai gylusis mokymasis kaip ir kiti mokymosi algoritmai kategorizuojamas į tris stambias dalis [DYU]:

- neprižiūrimus;
- prižiūrimus;
- hibridinius.

Giluminis mokymasis Python kalboje turi nemažai bibliotekų tokių kaip:

- Caffe.
- Theano.
- TensorFlow.
- Lasagne.
- Keras.

#### 1.3.3.1. Caffe

Caffe – giliojo mokymosi (angl. Deep learning) karkasas sukurtas atsižvelgiant į greitį ir modulumą [CAF]. Ši biblioteka suteikia [CAF]:

- Galimybė aprašyti modelius ir optimizacijas konfigūracijos failuose be statinių reikšmių kūrimo (angl. Hardcoded).
- Greitas perėjimas nuo CPU į GPU ir atvirkščiai. Užtenka pakeisti vieną požymį.
- Greitai plečiamas programinis kodas.
- Skaičiavimo greitis – Caffe gali apdoroti 60 mln. paveikslėlių per dieną su vienu NVIDIA K40 GPU. Tai yra 1 ms/paveikslėlį.
- Atvirojo kodo (angl. Open source).

Visi šie privalumai skamba gana neblogai, tačiau pateiktieji pavyzdžiai kaip naudotis šia biblioteka yra gana sudėtingi ir neintuityvūs. Taip pat nėra arba sudėtinga rasti kur minėtoji biblioteka būtų taikoma realiose sistemose.

### 1.3.3.2. Theano

Theano – dar vienas giliojo mokymosi karkasas sukurtas universitetų grupės [BRO]. Šis karkasas pasižymi šiomis savybėmis [THE]:

- Standartiškai veikia su CPU ir GPU.
- Veikia windows, OS X, Linux aplinkose.
- Gali atpažinti nestabilias matematinės išraiškas ir jas apskaičiuoti daug stabiliau.
- Reikalauja veikiančio C kompiliatoriaus.

Vienas didžiausių šio karkaso minusų: aktyvus karkaso tobulinimas yra nutrauktas.

### 1.3.3.3. TensorFlow

TensorFlow – kitas giliojo mokymosi karkasas sukurtas Google korporacijos. Šis karkasas yra vienas populiariausių pasaulyje. Apie tai galima spręsti iš svetainėje [www.tensorflow.org](http://www.tensorflow.org) minimų kompanijų vardų kurios naudoja TensorFlow.

Karkasas yra aktyviai tobulinamas. Laikas nuo laiko išleidžiama nauja versija, klaidų pataisymai, bei naujos funkcijos (produktai), tokie kaip TensorFlow.js [TEN]. Karkasas pasižymi šiomis savybėmis:

- Standartiškas CPU ir GPU palaikymas.
- Statistikos įrankiu – įdiegus TensorFlow kartu gaunamas ir TensorBoard, kuris yra skirtas įvairių grafikų vizualizacijai, bei jūsų mokymosi algoritmo progreso stebėjimui realiu laiku.
- Gana sudėtingas, turintis daug įvairių funkcijų.

### 1.3.3.4. Lasagne

Lasagne – giliojo mokymosi karkasas stipriai sujungtas su Theano karkasu [LYG]. Kuris kaip ir anksčiau buvo minėta yra nebetobulinamas. Lasagne karkasas pasižymi šiomis savybėmis [LYG]:

- Kaip ir kiti veikia su CPU ir GPU.
- Docker šablono palaikymu.

### 1.3.3.6. Keras

„Keras“ – tai giliojo mokymosi karkasas skirtas darbui su „Theano“ ir „TensorFlow“ giliojo mokymosi bibliotekomis. „Keras“ biblioteka buvo sukurta, tam kad pagreitintų programų kūrimo procesą. Ši biblioteka suteikia [KER]:

- Paprastą ir greitą prototipavimą.
- Palaiko sąsūkos ir rekurentinius tinklus, bei jų kombinacijas.
- Palaiko betkokias sujungimo schemas: daugybinius įvesties išvesties modelius, sluoksnių dalinimąsi (angl. Layer sharing), modelių dalinimąsi ir kt. Tai reiškia, kad „Keras“ yra tinkamas bet kokio giliojo mokymosi modelio sukūrimui pradendant nuo atminties tinklo baigiant neuronine tiuringo mašina.
- Veikia naudojant CPU ir GPU.
- Prototipavimo pavyzdžiai yra lengvai suprantami.

Išvados: praktinės dalies atlikimui buvo pasirinktas „Keras“ giliojo mokymosi karkasas dėl šių priežasčių:

- Prototipavimo pavyzdžiai pasirodė lengvai suprantami ir nesudėtingi.
- Gali veikti kartu su „TensorFlow“, kuris yra vienas populiariausių pasaulyje.
- Nesudėtingas įdiegimas windows aplinkoje.

## 1.4. Automobilio variklio skleidžiamo garso struktūra

Motorinių transporto priemonių variklių skleidžiami garsai priklauso nuo daugybės faktorių:

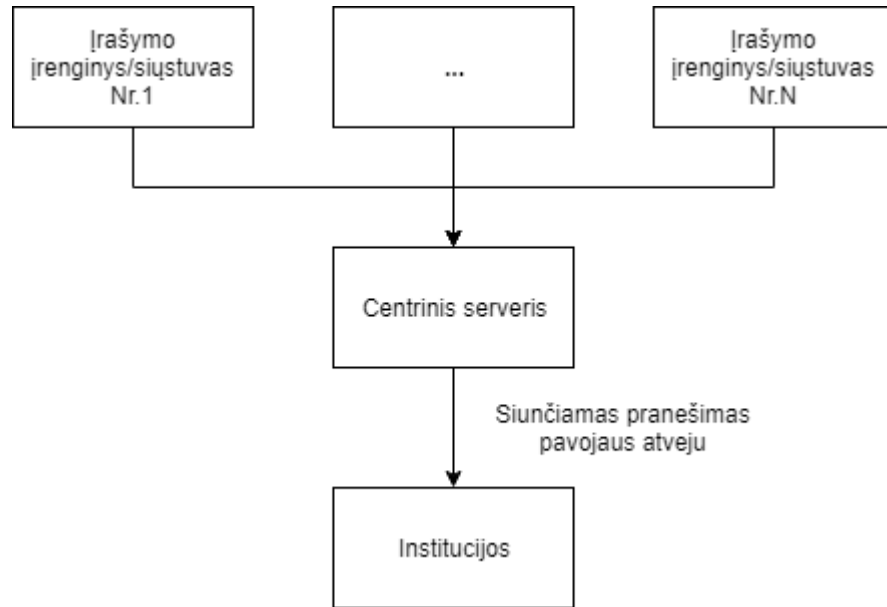
- variklio darbinio tūrio;
- cilindrų išsidėstymo, jų kiekio;
- temperatūros;
- medžiagų iš kokių pagamintas;
- naudojamos variklio alyvos.

## 1.5. Praktiniai darbai tiesiogiai susieti su magistro darbo uždaviniu

Panašiausias darbo temai praktinė realizacija yra [WHI]. Giliojo mokymosi biblioteka Tensorflow pasitelkiama nelegalių miškų kirtimų atpažinimui. Kaip įmonės Rainforest Connection įkūrėjas Topher White atskleidžia praktinėje realizacijoje panaudoti paprasti mobilieji telefonai. Jie siunčia mikrofono įrašytus duomenis į centrinį serverį, kuriame pasitelkiant Tensorflow biblioteką realiu laiku yra atpažįstama nelegali veikla miškuose:

- Grandininių pjūklų garsai.
- Kitos miško technikos skleidžiami garsai.

Kadangi tai yra privatus projektas, tikslūs algoritmai nėra žinomi. 6 pav. pateikiama sistemos realizacija sprendžiant pagal pasakojimą. Ant medžių yra pakabinami mobilūs telefonai, kurie įrašinėja garsą ir siunčia į centrinę sistemą. Atpažinus nelegalią veiklą yra siunčiamas pavojaus signalas atitinkamoms tarnyboms.



6 pav. Galimos realizacijos [WHI] diagrama

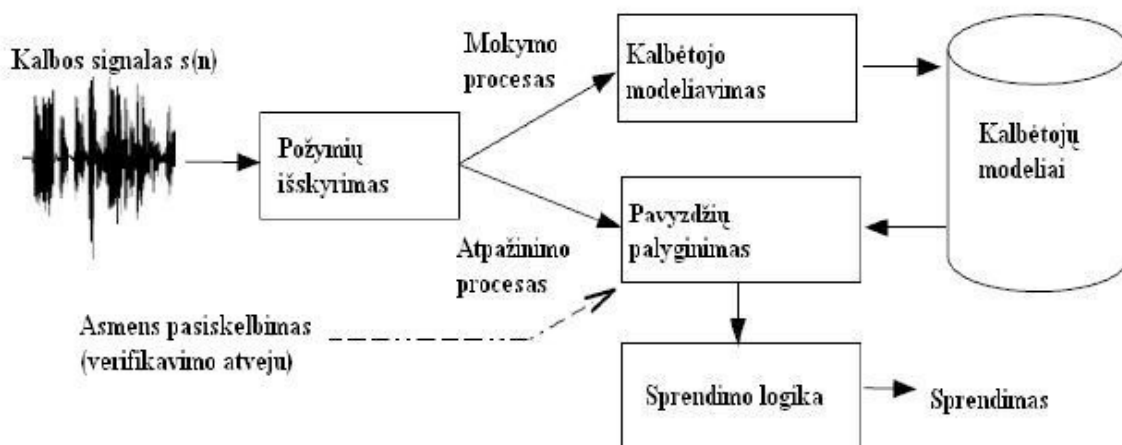
Kitas panašus darbas nagrinėjama temai yra [SSS+12]. Darbe yra bandoma atpažinti žmogaus veiklas pagal jų garsą.



7 pav. Veiklos atpažinimas [SSS+12]

7 pav., pateikta kaip vyksta žmogaus veiklos atpažinimas. Pirmame lygyje audio srautas yra sudalinamas į daug nedidelių dalių. Naudoja prižiūrimą (angl. Supervised) mokymosi algoritimą paremtą atsitiktiniu miško klasifikatoriumi (angl. Random Forest classifier). Taip pat jie pabrėžia, kad tam tikrame kadro (angl. frame) lygyje informacija nevisada būna teisingai suklasifikuojama, dėl įvairių pašalinių garsų.

Plačiausiai audio srauto apdorojime yra paplitusi programinė įranga, kuri apdoroja – klasifikuoja muzikos failus, bei randa panašius pagal šabloną, kuris yra algoritmų įeities parametras. Pavyzdžiui: asmens atpažinimas pagal balsą [KAM09]. Darbe, apdorojant audio srautą, naudojamos FFT, DFT, ADFT transformacijos, neuroniniai tinklai ir kiti.



8 pav. Automatinės kalbančiojo atpažinimo sistemos struktūra [KAM09]

[ELI] python kalba parašyta sistema geba sukurti audio įrašų duomenų bazę ir vėliau pagal pateiktą audio šabloną ieškoti atitikmens (panašaus tipo darbai [WAN], [KHS]). Programos tikslumas su muzikos įrašais yra gana tikslus, tačiau pabandžius programą su motorinių

transporto priemonių audio įrašais tikslumas dingsta. Atsižvelgiant į tai, jog ši programa turi nemažai parametrizuojamų kintamųjų, būtų galima pritaikyti vieną ar net kelis algoritmus, kurie geba mokytis tam, kad būtų išsiaiškinti optimaliausi parametrai.

Taip pat egzistuoja daug įvairių atvirojo kodo bibliotekų, kurios geba analizuoti audio srautą, pateikti analizės vizualizacijas. Viena iš tokių bibliotekų yra [GIA] parašyta python kalba gebanti klasifikuoti audio failus. Ši programa susideda iš šių dalių [GIA]:

- Duomenų bazės su komentarais – saugomi suklasifikuoti audio failai, prižiūrimam mokymuisi reikalingi duomenys.
- Požymių išgavimo (angl. Feature extraction) – šioje dalyje yra išgaunami tam tikri požymiai tiek iš laiko domeno tiek iš dažnio domeno.
- Klasifikacijos – naudojamas prižiūrimasis mokymasis. Optimaliems parametrams gauti galima naudoti SVM algoritmą su kainos parametru arba skaičių artimiausių kaimynų naudojamą kNN klasifikatorių. Gautąjį klasifikacijos modelį galima išsaugoti į failą.
- Regresijos – modeliai, kurie sujungia audio požymius į realių reikšmių kintamuosius taip pat gali būti apmokyti naudojantis prižiūrimuoju mokymu.
- Segmentacijos – galimi prižiūrimas ir neprižiūrimas mokymai.
- Vizualizacijos – teikia galimybę vizualiai pamatyti sąryšius tarp audio įrašų.

Ši biblioteka taip pat gali dirbti realiuoju laiku įrašinėdama fiksuoto dydžio audio segmentus ir klasifikuodama juos.

Dar vienas įdomus audio srauto apdorojimo būdas publikuojamas moksliniame darbe „Fansmitter: Acoustic Data Exfiltration from (Speakerless) Air-Gapped Computers“ [GSD+]. Nors šiame darbe pagrindinė tema yra duomenų saugumas, tačiau jame taip pat puikiai atsispindi audio srautą apdorojančių algoritmų svarba dabartiniame pasaulyje, bei tai, jog pats audio srautas gali nešti labai daug vertingos informacijos. Konkrečiai šiuo atveju autoriai pasinaudodami kompiuterio aušinimo ventiliatoriaus sukeliama garsu į netoliese esantį audio imtuvą sugebėjo perduoti informaciją. Tam, kad kompiuterio naudotojui būtų sudėtingiau išgirsti arba suvokti, jog su aušintuvu kažkas negerai autoriai naudojo žemus dažnius 140 – 170 Hz. Nepaisant esančio natūralaus foninio triukšmo buvo pasiekta apie 900 bitų per valandą sparta.



## 2. Praktinė dalis

Šioje darbo dalyje apibūdinama reikalingų duomenų rinkimo specifiška, surinktų duomenų specifiška. Aprašomi žingsniai, kurie buvo atlikti prieš pateikiant duomenis mokymuisi. Taip pat aprašomas sukurtasis mokymosi algoritmo modelis, bei atlikti eksperimentai.

### 2.1. Teorinės darbo prielaidos

Praktinio darbo metu daromos šios teorinės prielaidos:

- Kiekvienos motorinės transporto priemonės variklio skleidžiamas garsas skiriasi.
- Kitų mobiliųjų telefonų garso įrašymo kokybė yra panaši arba geresnė.
- Siekiant sumažinti duomenų imtį, darbo metu nagrinėjami tik dyzelinių motorinių transporto priemonių variklių skleidžiami garsai.
- Audio įrašai įrašinėjami automobilių varikliams dirbant laisvomis apsukomis.

### 2.2. Duomenų rinkimas

Mokslo tiriamojo darbui reikalingi duomenys (motorinių transporto priemonių variklių skleidžiami garsai) buvo renkami 2016 m. – 2018 m. laikotarpyje, mobiliojo telefono *Samsung Galaxy xcover 2* (toliau įrašomasis įrenginys) pagalba. Minėtuju laikotarpiu buvo surinkti 47 skirtingų motorinių transporto priemonių variklių skleidžiami garsai.

Visi audio įrašai pasižymi šiomis savybėmis:

- Įrašo trukmė apie 1 minutę.
- Formatas m4a.
- Bit rate 123 kbps.
- 44100 Hz.
- 32 bit.

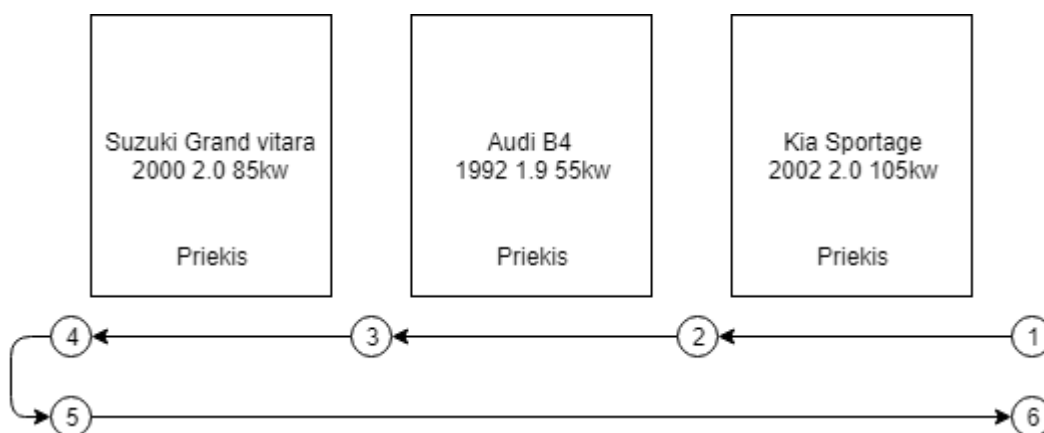
Kartu su įrašais buvo fiksuojami papildomi duomenys:

- Apytikslė lauko temperatūra Celsijaus laipsniais.
- Įrašo data.
- Apytikslė variklio temperatūra Celsijaus laipsniais.
- Papildomi veiksniai (pavyzdžiui, pro šalį važiavo kita mašina).

Surinktųjų duomenų sąrašas pateikiamas priedų skyrelio 2 lentelėje. Šią lentelę sudaro stulpeliai iš anksčiau minėtų papildomų duomenų pridėdant įrašo numerį bei įrašo pavadinimą.

Visi įrašai vadinasi **Balsas\***, čia \* įrašo identifikacinis numeris. Toks pavadinimas yra numatytasis įrašančiojo įrenginio operacinėje sistemoje. Darbo metu šis pavadinimas nebuvo keičiamas. Lentelėje nuline variklio temperatūra yra laikoma, jog variklis yra aplinkos temperatūros ir tik ką buvo užvestas. Darbine variklio temperatūra laikome tokią temperatūrą, prie kurios variklis dirba optimaliu režimu.

Duomenų rinkimo metu taip pat buvo įrašyta keletas įrašų kuomet girdimi keli motorinių transporto priemonių variklių skleidžiami garsai. Įrašymo pozicijos pateiktos 3 paveikslėlyje.



9 pav. Eksperimentinio įrašo eiga.

Įrašų pradžia prasideda nuo 9 paveikslėlyje pažymėto pirmo burbuliuko ir baigiasi šeštajame. Pakeliui yra sustojama prie kiekvieno automobilio ir palaukiama keletą sekundžių. Taip pat jie pasižymi tokiomis pačiomis savybėmis kaip ir visi surinktieji audio įrašai, bei:

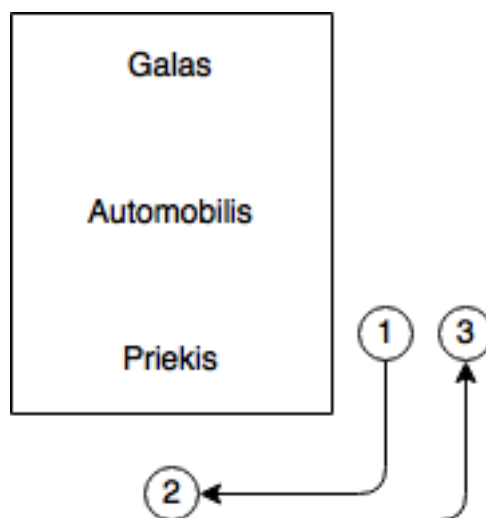
- Įrašo trukmė yra nėra apibrėžta minutės intervale.
- Veikia vieno automobilio variklis.
- Veikia dviejų automobilių varikliai.
- Veikia visų trijų automobilių varikliai.
- Neveikia nei vienas automobilio variklis.

Visi surinktieji audio įrašai vėliau buvo konvertuoti į wav tipo failus, naudojantis internete rastu audio įrašų konverteriu *online-audio-converter.com*. Šie įrašai bus naudojami vėlesnėse darbo dalyse. Jie pasižymi tokiomis savybėmis:

- Įrašo trukmė apie 1 minutę.
- Formatas wav.
- Bit rate 3072 kbps.
- 96 kHz.
- 32 bit.

### 2.2.1. Įrašų sandara

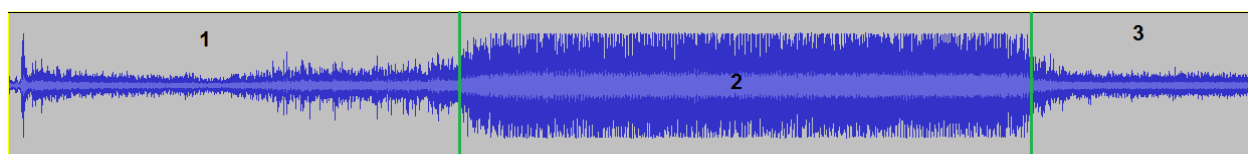
Kiekvienas įrašas sudarytas iš trijų dalių. Įrašymo pozicijos pavaizduotos 10 paveikslėlyje.



10 pav. Įrašymo pozicijos.

Audio įrašas prasideda pirmoje pozicijoje, vėliau po tam tikro laiko tarpo įrašomasis įrenginys pernešamas į antrąją poziciją, vėl po tam tikro laiko tarpo pernešamas į trečiąją poziciją. Visose trijose pozicijose stengiamasi įrašymo įrenginį palikti vienodą laiko tarpą.

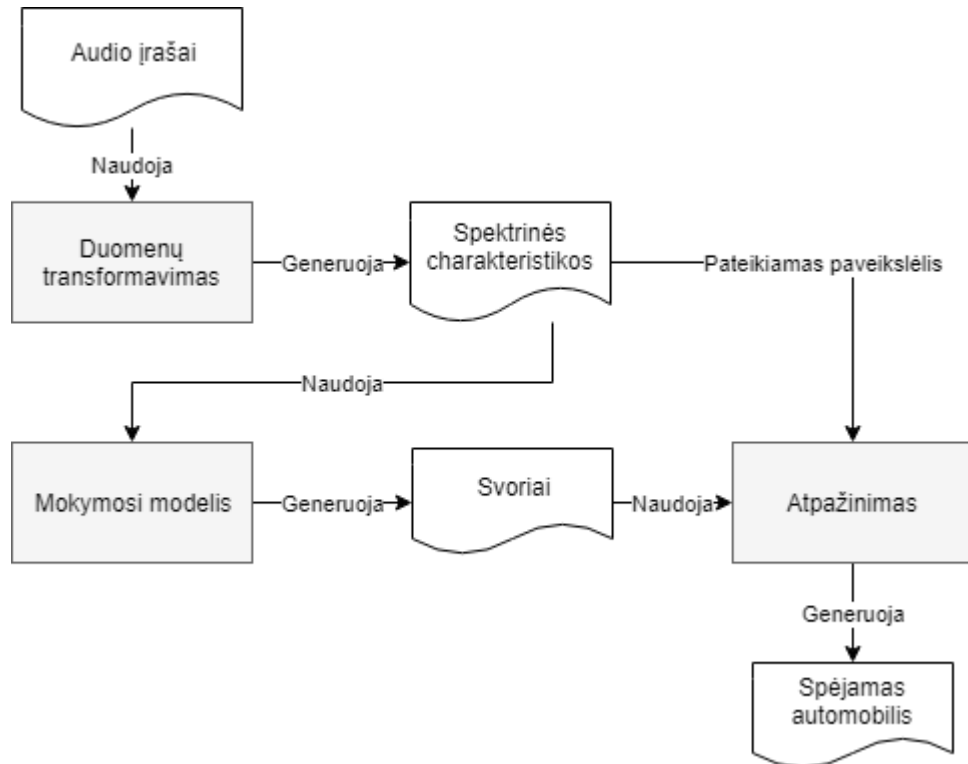
Šios trys įrašymo pozicijos pasirinktos darant prielaidą, jog realioje sistemoje mikrofonas bus nukreiptas į motorinės transporto priemonės priekines variklio grotelles, o pravažiuojant pro mikrofoną pasieks 1 ir 3 pozicijas.



11 pav. Audio įrašo takelis.

Įrašymo pozicijų atitiktis audio sraute pavaizduota 11 paveikslėlyje. Jame matomos 1 – 3 dalys atitinka 1 – 3 įrašymo pozicijas pavaizduotas 1 paveikslėlyje. 1 ir 3 įrašo dalys vizualiai yra panašios, kadangi kaip ir minėta anksčiau – pradžios ir pabaigos pozicijos sutampa. Žalia linija yra atskirta antros pozicijos pradžia ir pabaiga. Taip pat antroje pozicijoje matoma, jog variklio skleidžiami dažniai yra gerokai didesni. Taip yra todėl, kad įrašomasis įrenginys yra tiesiai prie priekinių automobilio grotelių ir garsas sklinda tiesiogiai.

### 2.3. Sistemos modelis



12 pav. Sistemos modelis.

12 paveikslėlyje pavaizduodas siūlomas sistemos modelis, kuris susideda iš trijų dalių:

- Duomenų transformavimo – įvesties parametrai yra audio įrašai iš kurių gaunamos spektrinės charakteristikos.
- Mokymosi modelio – įvesties parametrai yra anksčiau gautosios spektrinės charakteristikos. Mokymosi rezultatas yra svoriai.
- Atpažinimo programos – kaip įvesties parametrai pateikiami mokymosi metu sugeneruoti svoriai, bei spektrinės charakteristikos paveikslėlis. Išvestis: spėjamas automobilis.

Kadangi darbų, kuriuose būtų bandoma atpažinti automobilį iš variklio skleidžiamo garso nebuvo rasta, šis modelis remiasi tik logine seka.

### 2.4. Duomenų transformavimas (spektrinių charakteristikų gavimas)

Prieš pateikiant duomenis giliojo mokymosi algoritmui juos būtina transformuoti. Duomenų transformavimas susideda iš:

- Audio įrašų skaidymo.
- Audio įrašų spektrinių charakteristikų gavimas.

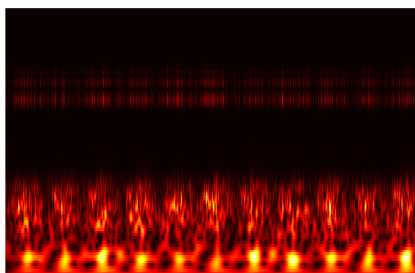
### 2.4.1. Audio įrašų skaidymas

Paprastai ruošiant audio įrašus sistemų apmokymui jie būna suskaidomi į tam tikros trukmės įrašus. Šiame darbe įrašai skaidomi 400ms ilgio. Failai buvo skaidomi automatiškai parašius nedidelę python programėlę, bei naudojantis ffmpeg platforma [FFM].

### 2.4.2. Audio įrašų spektrinių charakteristikų gavimas

Spektrinių charakteristikų analizė labai panaši į Furje analizę. Signalą dalina į jo sudedamąsias dalis, kol tuo tarpu Furje transformacija signalą sudalina į skirtingo dažnio sinuso bangas. Taigi spektrinių charakteristikų analizė mums leidžia pamatyti bendresnį vaizdą.

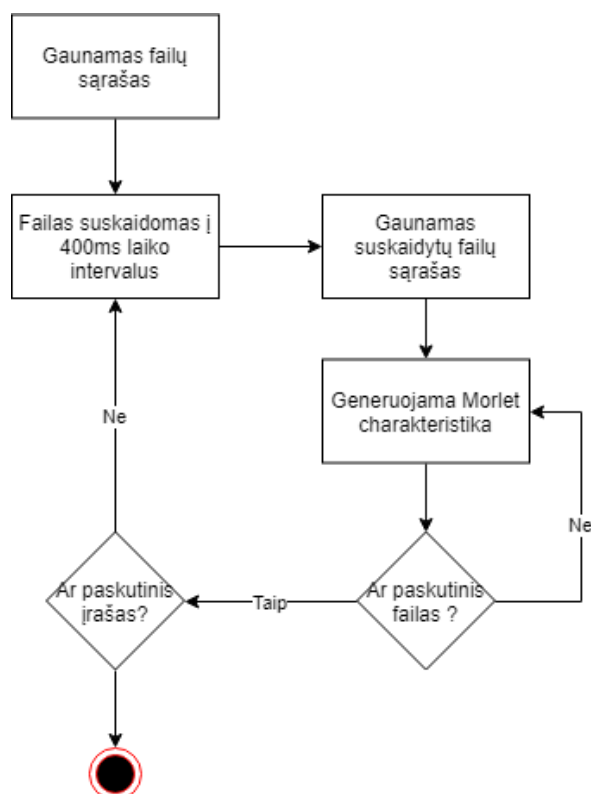
Naudojantis ObsPy biblioteka [OBS] iš suskaidytų failų Morlet transformacijos metu buvo gaunami spektrinių charakteristikų paveikslėliai. Paveikslėlio pavyzdys pateiktas 13 pav.



13 pav. Vieno iš įrašyto automobilių skleidžiamo variklio garso spektrinė charakteristika.

Pateiktajame paveikslėlyje nesunkiai galima įžiūrėti tam tikrą intervalų susiskaidymą, kuris yra matomas visuose surinktųjų audio įrašų spektrinėse charakteristikose. Šios charakteristikos yra vaizduojamos 3D formatu, kuris įtraukia laiką, amplitudę ir dažnį.

Spektrinių charakteristikų gavimo programos modelis pateiktas 14 paveikslėlyje.



14 pav. Supaprastinta spektrinių charakteristikų generavimo schema.

## 2.5. Duomenų klasifikavimas

Šiame skyrelyje aprašomas surinktųjų duomenų klasifikavimas. Skyrelio tematika:

- Pradinių duomenų klasifikavimas su „Audfprint“ programa [ELI].
- Gautųjų spektrinių charakteristikų klasifikavimu naudojantis „Keras“ giliojo mokymosi (angl. deep learning) biblioteka.

### 2.5.1. Pradinių duomenų klasifikavimas su [ELI] „Audfprint“ programa.

Šios dalies tikslas yra įsitikinti, jog dabartinė sistema negeba atpažinti motorinių transporto priemonių variklių skleidžiamų garsų, naudojantis metodais sukurtais muzikos atpažinimui.

[ELI] sukurtoji programa – paremta Landmark-based audio fingerprinting. Ji gana tiksliai geba atpažinti muzikinius audio įrašus, jeigu toks įrašas yra programos duomenų bazėje. Šiuo atveju duomenų bazė laikomas failas, kuriame yra įrašomi konkretaus audio failo identifikatoriai (angl. *Features*). Duomenų bazė gali būti sudaryta iš daugiau negu vieno audio įrašo. Pagrindinės programos funkcijos yra šios:

- Sukurti duomenų bazę.
- Sujungti/papildyti duomenų bazes.
- Ieškoti atitikimų duomenų bazėje pagal pateiktą audio įrašą.

Duomenų baze šiame skyriuje laikomas failas, kurio plėtinys yra *pklz* [ELI]. Testams atlikti buvo naudojami 2.2. Skyrelyje surinkti duomenys. Iš 2.2.1 skyrelyje minimų įrašymo pozicijų buvo iškirpta atitinkamai po 10s, 20s, 10s audio įrašų. Iš kiekvieno audio failo gaudavosi trys įrašai. Šie intervalai minėtose pozicijose buvo pasirenkami atsitiktinai. Įrašai buvo karpomi naudojantis [AUD] programa. Gautųjų failų struktūra:

Balsas<N>\_<M>.wav

Kur N – įrašo identifikacinis numeris, M – sekundžių skaičius.

Viso buvo atlikti keturi testai:

- Pirmasis – sukurtoji duomenų bazė lyginama su tuo pačiu audio įrašu naudotu sukurti duomenų bazę.
- Antrasis – pirmame žingsnyje pirmoji duomenų bazė lyginama su M – tuoju audio įrašu, toliau lyginama įprastai nuo pirmojo iki M – 1 (jeigu turime failų masyvą nuo 1 iki 30, tai atliekame postūmį per vieną į dešinę, taip 30-as masyvo elementas tampa pirmuoju ir t.t.).
- Trečiasis – šis testas yra analogiškas, tik anksčiau minėtasis postūmis atliekamas per vieną į kairę pusę. Pavyzdžiui, jeigu turime masyvą nuo 1 iki 30 ir atliekame minėtąjį postūmį, tai tokiu atveju pirmasis elementas taps 30-tuoju, o 30-asis taps 29-tuoju.
- Ketvirtasis – duomenų bazės lyginamos su atsitiktiniais audio įrašais.

Nr.	Duomenų bazės pavadinimas	Testo numeris			
		1	2	3	4
		Audio įrašas	Audio įrašas	Audio įrašas	Audio įrašas
1.	Balsas001_10.pklz	Balsas001_10.wav	Balsas<M>.wav	Balsas001_20.wav	Balsas011_10.wav
2.	Balsas001_20.pklz	Balsas001_20.wav	Balsas001_10.wav	Balsas001_10_2.wav	Balsas025_20.wav
...	...	...	...	...	...
N	Balsas<M>.pklz	Balsas<M>.wav	Balsas<M-1>.wav	Balsas<M>.wav	Balsas<T>.wav

1 lentelė. Testų (duomenų bazių) rinkinys sukurtas su [ELI] programa.

Pirmoje lentelėje N yra testų skaičius. M – duomenų bazės ir audio įrašo identifikacinis numeris, kur  $M > 0$ . T – audio įrašo identifikacinis numeris.

### Duomenų bazių sukūrimas

Prieš pradėdant testus buvo sukurta kiekvieno audio failo duomenų bazės atitinkamo naudojantis parašyta programa *database\_creator.sh*. Programa kiekvienam nurodyto katalogo failui sukuria atitinkamą duomenų bazę. Duomenų bazės sukūrimo pavyzdys pateiktas žemiau:

```
python audfprint/audfprint.py new -h 20 -r 96000 --dbase "
audfprint/databases/experiment/10_20_10/$filename.pklz" "$file"
>> isvestis.log
```

15 pav. Duomenų bazių sukūrimo skriptas

Čia *audfprint.py* – [ELI] programa, leidžianti atlikti 2.5.1. skyrelio pradžioje minėtas funkcijas.

### Testų rezultatai

Pirmajam testui atlikti buvo sukurta programa pavadinimu *matcher\_one.sh*. Atitikmenų ieškojimo pavyzdys pateiktas žemiau:

```
for file in *;
do
    filename=$(basename "$file")
    extension="${filename##*.}"
    filename="${filename%.*}"
    echo "$file"
    echo $filename

    python /home/kd/Desktop/audfprint/audfprint.py match -h 20
-r 96000 --dbase
"/home/kd/Desktop/audfprint/databases/experiment/10_20_10/$filen
ame.pklz" "$file" >> ~/Desktop/matcher_same.log
done
```

16 pav. Programos *matcher\_one.sh* išeities kodas.

Programa lygina sukurtąsias duomenų bazes *Balsas<N>.pklz* su įrašais *Balsas<N>.wav*, kur *N* – įrašo identifikacinis numeris. Testas kaip ir buvo tikėtasi davė teigiamą rezultatą, t.y. kiekviena duomenų bazė atitiko audio įrašą iš kurio ji buvo pagaminta. Viso buvo rasta atitikmenų 235.

Antrasis ir trečiasis testai iš esmės buvo beveik vienodi. Tik vienu atveju postūmis buvo per vieną į dešinę kitu – per vieną į kairę. Šie testai buvo sugalvoti darant prielaidą, jog gretimi audio įrašai yra panašūs ir turėtų būti rasta atitikmenų. Šiems testams atlikti buvo sukurtos programos *matcher\_two.sh* ir *matcher\_third.sh*. Darbo rezultatai talpinami failuose *matcher\_two.log* ir *matcher\_third.log*. Šių testų rezultatai buvo ganėtinai prasti. Vos vienas kitas audio įrašas sugebėjo atitikti. Minimalūs atitikimai buvo surasti tarp skyrelyje 2.2.1. aprašytų įrašų dalinimo pozicijų. Viso buvo surasti 53 ir 61 atitikmuo.



Paskutiniajame teste buvo atsitiktine tvarka imami audio įrašai ir lyginami su einamąja duomenų baze. Šis testas taip pat rado atitikimų, tačiau dar mažiau negu ankstesnieji. Viso buvo rastas tik 21 atitikimas. Programos išeities kodas pateikiamas *matcher\_rand.sh* faile, o darbo rezultatai *matcher\_rand.log*.

Lyginant visus šiuos testus pirmasis buvo tiksliausias, kadangi kiekvienas audio įrašas atitiko savo duomenų bazę. Žemiau pateikiami kiekvieno testo rezultatų teigiami pavyzdžiai:

Pirmasis:

```
Matched Balsas001_10.wav 9.9 sec 7838 raw hashes as Balsas001_10.wav at 0.0 s with
2390 of 2878 common hashes at rank 0
```

Antrasis:

```
Matched ../experiment/10_20_10/Balsas011_20.wav 19.9 sec 17460 raw hashes as
Balsas011_10.wav at 0.4 s with 40 of 4331 common hashes at rank 0
```

Trečiasis:

```
Matched ../experiment/10_20_10/Balsas069_10.wav 10.0 sec 15275 raw hashes as
Balsas069_20.wav at 0.3 s with 34 of 6073 common hashes at rank 0
```

Ketvirtasis:

```
Matched ../experiment/10_20_10/Balsas051_20.wav 19.9 sec 8145 raw hashes as
Balsas047_20.wav at 6.9 s with 78 of 15547 common hashes at rank 0
```

Antrasis, trečiasis ir ketvirtasis testai laikomi nesėkmingais, nes lyginant su pirmuoju testu rastų atitikmenų suma buvo nereikšminga. Todėl apibendrinant rezultatus galima pastebėti, jog [ELI] programos egzistuojantis algoritmas nėra tinkamas motorinių transporto priemonių skleidžiamo garso atpažinimui.

### 2.5.2. Klasifikavimas su „Keras“ giliojo mokymosi biblioteka

Naudojantis „Keras“ biblioteka šiame skyrelyje buvo suklasifikuoti spektrinių charakteristikų duomenys gauti 2.4.2 skyrelyje. Tikimasi, jog galutinis rezultatas bus geresnis nei [ELI] programos.

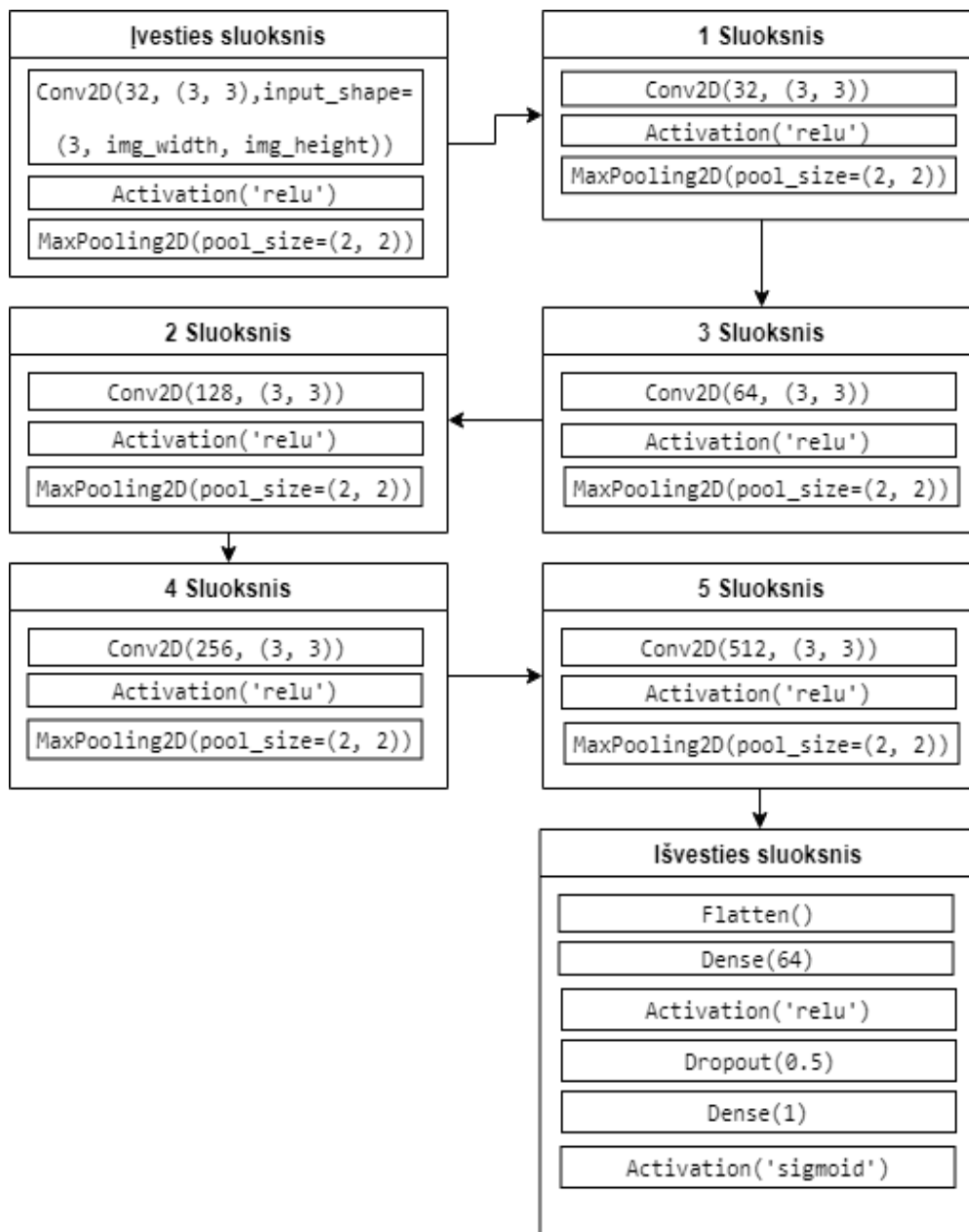
Tinklo mokymosi pradžioje duomenys gauti iš 2.4.2 skyrelio buvo padalinti į dvi grupes:

- Mokomieji – duomenys, kurie bus naudojami neuroninio tinklo mokymuisi.
- Testavimo – duomenys, kurie bus naudojami neuroninio tinklo validacijai.

### 2.5.2.1. Mokymosi modelis

Giliojo mokymosi sistemos modelis pavaizduotas 17 paveikslėlyje. Modelis sudarytas iš:

- Įvesties sluoksnio.
- Penkių sąsūkos sluoksnių.
- Išvesties sluoksnio.



17 pav. Sukurtasis mokymosi modelis

Branduolio dydis (angl. Kernel size) – visuose sluoksnuose yra 3x3. Šis dydis apibrėžia lango dydį, kuris imamas iš įvesties paveikslėlio.

Poslinkis (angl. Stride) – modelyje yra 1x1 numatytoji reikšmė. Šis poslinkis apibrėžia kokių greičiu branduolio dydis judės per paveikslėlį į dešinę ir į apačią. Šiuo atveju jis judės per 1 px.

MaxPooling2D – labai populiarus sprendimas sąsūkos operacijų tinkluose [VEL]. Ši operacija leidžia sumažinti paveikslėlių dydį. Numatytasis nustatymas yra 2x2 (angl. Pool size). Iš gautųjų sąsūkos operacijos metu reikšmių imama po 2x2 dydžio matrica ir iš jos išrenkama didžiausia reikšmė yra išvesties parametras. Tarkime, jeigu turime paveikslėlį 4x4 iš jo gautume 2x2.

#### **Įvesties sluoksnis:**

Šiuo atveju įvesties sluoksnis privalo turėti tas pačias paveikslėlių dimensijas kaip ir tinklo apmokymui skirti duomenys. Šio modelio atveju įvesties sluoksnis turi RGB paveikslėlį su priskirtais aukščio ir pločio parametrais. Jie apibrėžiami vėliau.

#### **Paslėptieji sluoksniai:**

Buvo naudojami penki paslėptieji sluoksniai, kuriems taikoma sąsūkos operacija. Šie sluoksniai yra visi panašūs tik skiriasi filtrų (literatūroje dar kartais vadinami kanalais (angl. Channels)) skaičius, kuris su kiekvienu sluoksniu didėja. Kiekviename sluoksnyje vyksta tos pačios operacijos kaip ir pirmajame įvesties sluoksnyje.

#### **Išvesties sluoksnis:**

Flatten – aprašytojo modelio atveju iš trijų dimensijų yra padaroma viena dimensija.

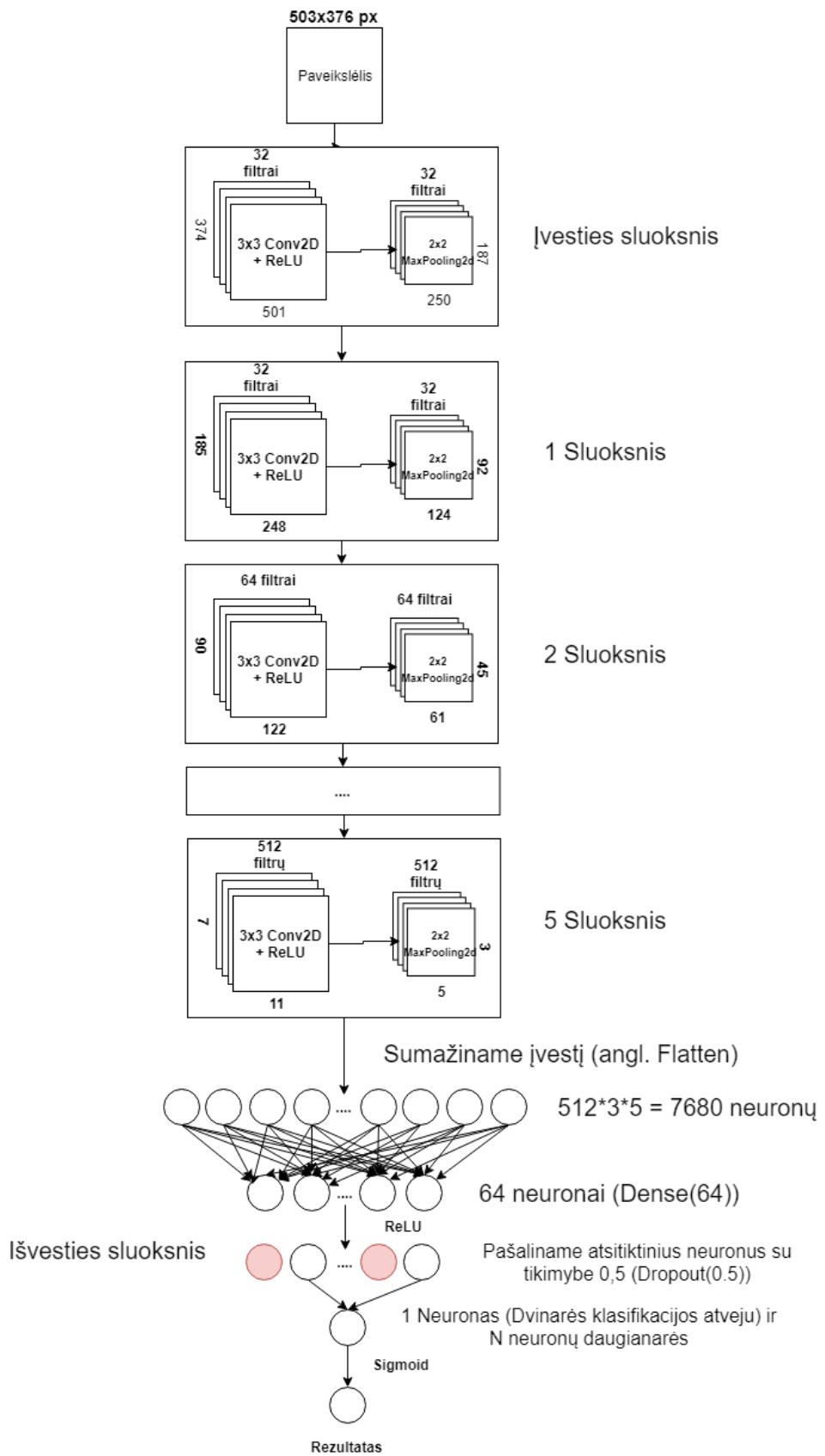
Dense – parametras naudojamas tvirtam neuronų tarp skirtingų sluoksnių sukabinimui (angl. Tight coupled)

Dropout – parametras skirtas išvengti mokymosi metu persotinimo (angl. Overfitting). Su tikimybe  $P$  (modelio atveju 0,5) yra panaikinami neuroninio tinklo sąryšiai, bei sunaikinamas pats neuronas.

Activatijos funkcija ReLu (angl. Rectified Linear Units) pasirinkta dėl:

- Dažniausiai giliojo mokymosi modeliuose [REP].
- Paprasta išvestinė  $f(x) = \max(x, 0)$ . Kas reiškia, kad funkcija gražins  $x$  jeigu  $x > 0$  arba  $x = 0$ , ir 0 jeigu  $x < 0$  [XWT+15].

Minėtojo modelio paaiškinamoji diagrama pateikta 18 paveikslėlyje.



18 pav. Sukurtojo mokymosi modelio architektūra.

### 2.5.2.2. Modelio apmokymas ir testavimas

Sukurtasis modelis buvo apmokomas su priedų skyrelyje 2 lentelėje X pažymėtais audio įrašais (jų spektrinėmis charakteristikomis). Visi kiti X nepažymėti įrašai į mokymasi nepateko ir buvo palikti modelio patikrinimui su nematytais duomenimis (konfuzijos (statistinio reikšmingumo) matricių generavimui).

Mokymasis buvo suskaidytas į dvi dalis:

- Apmokymas klasėmis – kai kiekviena skirtinga motorinė transporto priemonė turėjo savo klasę. Viso buvo 47 skirtingos klasės. Šiuo atveju 17 ir 18 paveikslėliuose išvesties metu būtų 47 neurorai (Dense(47)).
- Dvinaris – mokymosi duomenys suskaidyti į dvi klases: ieškomojo automobilio klasė ir klasė kita.

Visų paveikslėlių pateiktų mokymuisi ir testavimui dydžiai:

- Plotis – 376 px.
- Aukštis – 503 px.

Visiems testams buvo naudojamas gradiento nusileidimo metodo atmaina RMSProp [RUD]. Šis metodas darbe nėra nagrinėjamas.

Tikslumo metrikos (angl. accuracy):

- Klasių tikslumas –  $K.argmax(y\_true)$  ima didžiausią spėjimo reikšmę ir tikrina su atitinkamu rinkiniu [CRO].

```
def categorical_accuracy(y_true, y_pred):  
    return K.cast(K.equal(K.argmax(y_true, axis=-1), K.argmax(y_pred, axis=-1)),  
K.floatx())
```

- Dvinaris tikslumas –  $K.round(y\_pred)$  reiškia, kad riba yra 0,5, ir viskas kas aukščiau traktuojama kaip tiesa. Programinis išeities kodas pateikiamas žemiau [CRO] .

```
def binary_accuracy(y_true, y_pred):  
    return K.mean(K.equal(y_true, K.round(y_pred)), axis=-1)
```

Visų testų metų buvo skaičiuojami dydžiai:

- acc/loss – yra atitinkamai tikslumo ir paklaidos kreivės generuojamos iš mokymosi duomenų imties.
- val\_acc/val\_loss – yra atitinkamai tikslumo ir paklaidos kreivės generuojamas iš validavimo imties duomenų.

Testų metu siekiama, jog tikslumo kreivė artėtų ir taptų vienetine funkcija, o paklaidų kreivė artėtų ir taptų nuline.

## Testai

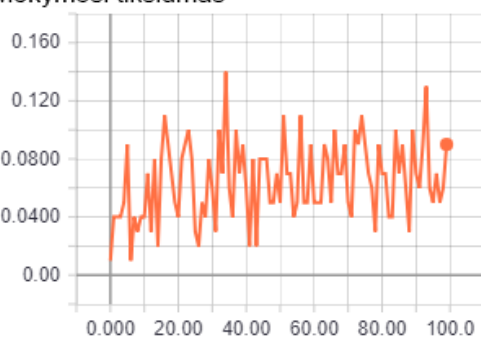
### 1. Apmokymas su klasėmis

Šio apmokymo metu buvo atlikti du testai. Abu testai pasižymi šiais parametrais:

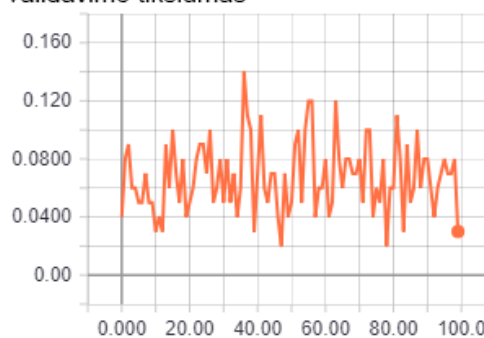
- Mokymosi imtis = 100 – iš kiekvienos klasės imama po 100 paveikslėlių.
- Validavimo imtis = 100 – iš kiekvienos klasės imama po 100 paveikslėlių.
- Epochų skaičius 100.
- Batch size = 5 – kiek vienu metu paimama paveikslukų.
- Paklaidos funkcija = categorical\_crossentropy. Ši funkcija darbe detaliam nenagrinėjama.

Pirmasis testas buvo vykdomas 100 epochų, o antrasis 500.

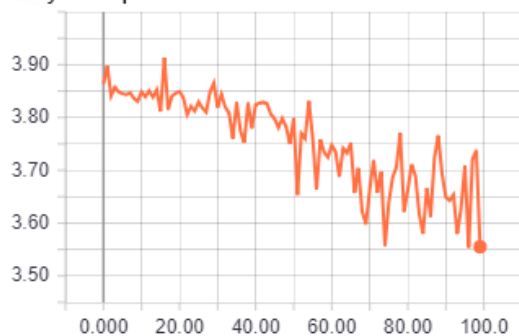
Mokymosi tikslumas



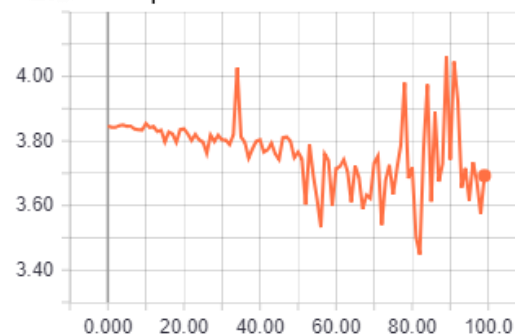
Validavimo tikslumas



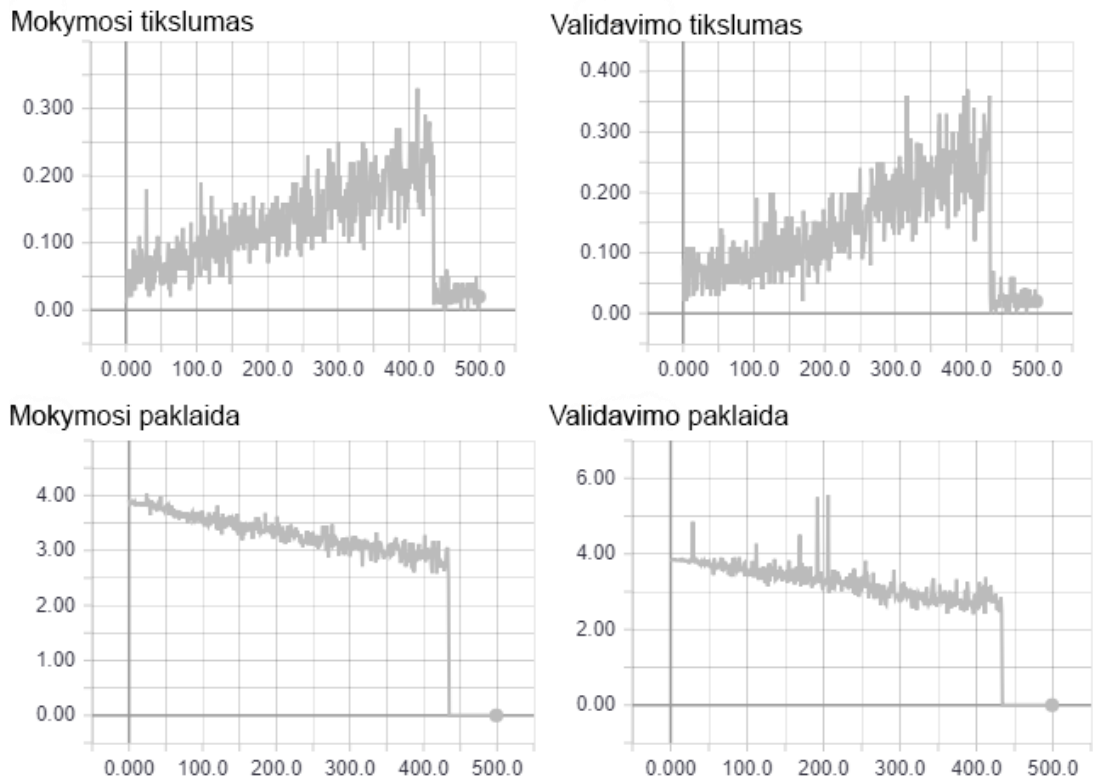
Mokymosi paklaida



Validavimo paklaida



19 pav. Klasių apmokymas 100 epochų



20 pav. Klasių apmokymas 500 epochų.

Abu testai duoda teigiamą rezultatą. Galime matyti, jog sukurtasis modelis mokosi, tačiau 19 pav. atveju jo tikslumas siekia tik 0.08%, o 20 pav. atveju jis pakyla iki 30%, tačiau ties 450 – tąja epocha staigiai krenta žemyn. Taip galėjo nutikti dėl šių priežasčių:

- Gradiento funkcija įgyja neleistinas reikšmes (NaN).
- Per didelio mokymosi greičio.
- Duomenų trūkumo.
- Per mažo mokymosi modelio.

Išvados: klasių apmokymui reikia daugiau duomenų, bei didesnio epochų skaičiaus. Iš esmės modelis su klasėmis nėra tinkamas realiai automobilio paieškos sistemai, dėl:

- Reikalingo didelio mokymosi duomenų kiekio.
- Mokymosi laiko.

## 2. Dvinarė klasifikacija

Šio apmokymo metu buvo atliktos keturios testų poros po tris testus poroje. Visi testai pasižymi šiais parametrais:

- Epochų skaičius 30.

- Batch size = 20 – kiek vienu metu naudojama paveikslėlių tinklo mokymuisi.
  - Paklaidos funkcija = binary\_crossentropy. Ši funkcija darbe detaliau nenagrinėjama.
- Trys testų poros turėjo tokias mokymosi imtis:

- Mokymosi imtis = 763.
- Validavimo imtis = 763.

Likusioji ketvirtoji pora turėjo tokias imtis:

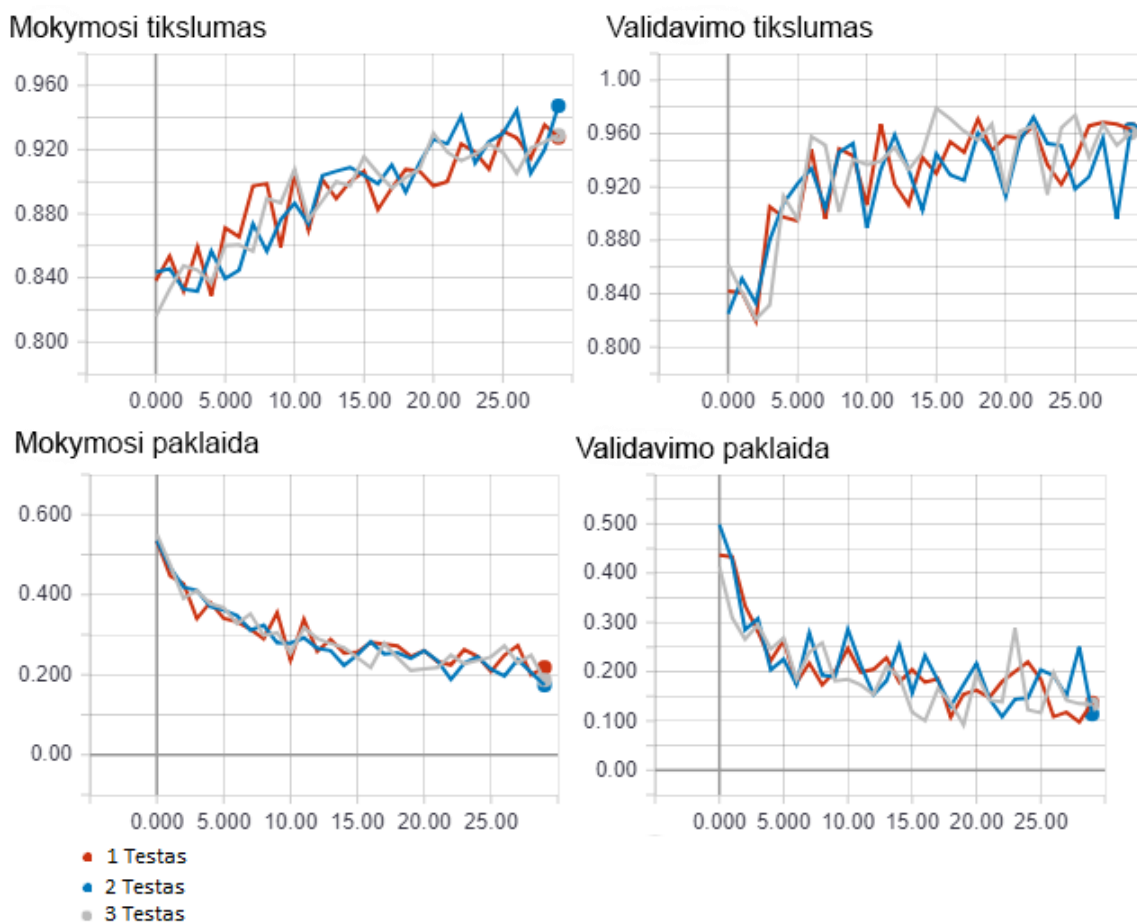
- Mokymosi imtis = 3700.
- Validavimo imtis = 3700.

Imtys skiriasi dėl to, kad tiriamojo objekto audio įrašų (autorius automobilio) buvo surinkta nepalyginamai daugiau negu kitų motorinių transporto priemonių.

Mokymosi metu duomenys buvo sugrupuoti į dvi klases:

- Ieškomasis automobilis.
- Kita – visi kiti spektrinės charakteristikos paveikslėliai išskyrus ieškomojo automobilio.

### 2.1. Audi b4 1992 55kw ir Kita (1 testas)

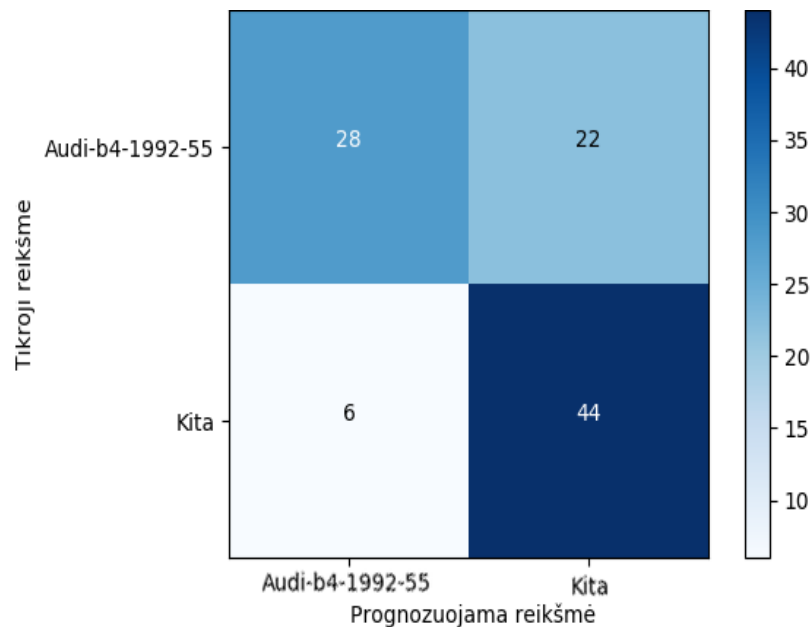


21 pav. Audi b4 1992 55kw ir *Kita* modelio tikslumo ir paklaidos kreivės.



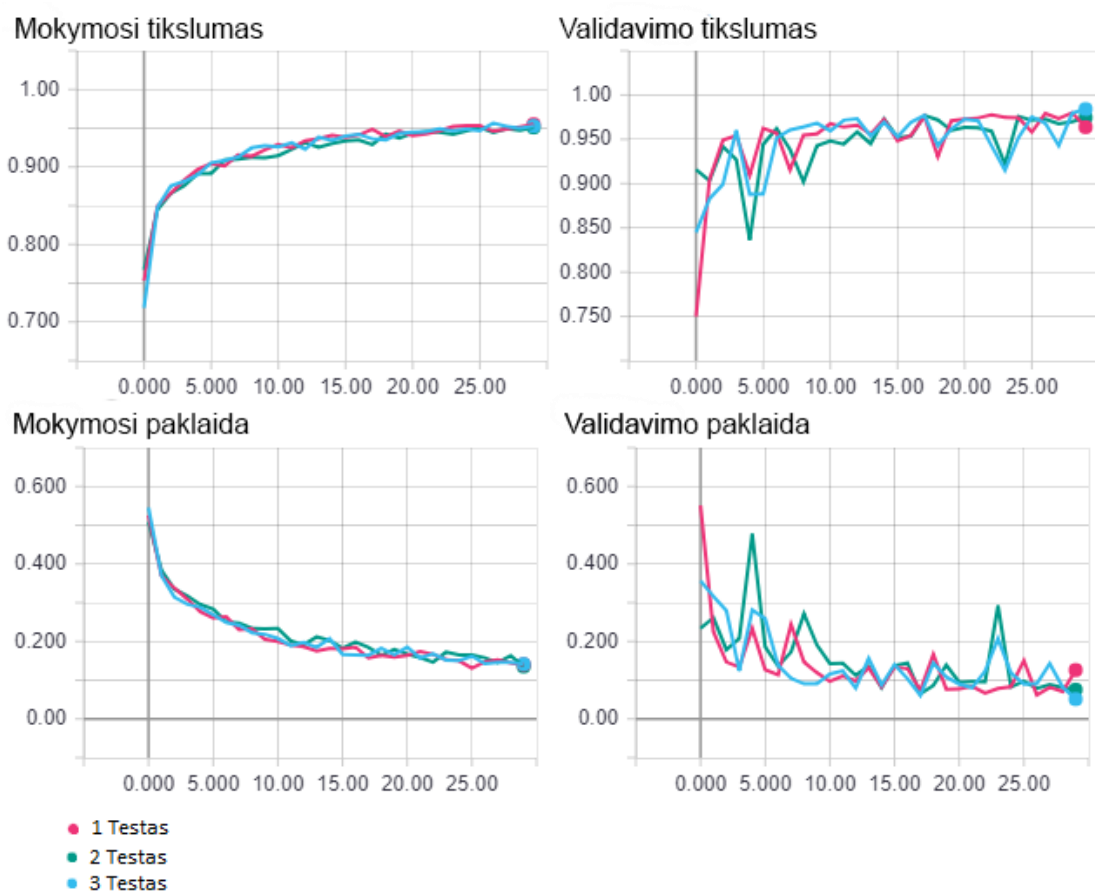
21 paveikslėlyje pateikiama standartinės „Keras“ karkaso tikslumo charakteristikos apmokymo metu. Viso 3 testai su tuo pačiu duomenų rinkiniu. Kiekviena mokymosi ir validavimo duomenų imtis sudarė 763 paveikslėliai. Iš kreivių matyti, jog mokymasis yra netolidus t.y. vienoje epochoje tikslumas kyla, o kitoje staigiai krenta. Toks netolidumas gali atsirasti dėl per mažo mokymosi duomenų skaičiaus.

22 paveikslėlyje pateikta klasifikavimo lentelė (angl. confusion matrix). Ji buvo sugeneruota po modelio apmokymo, tinklui pateikus suklasifikuoti 100 nematytų paveikslėlių, atitinkamai kiekvienai klasei po 50. Kaip matyti modelio tikslumas nėra didelis, tačiau teikia vilčių.



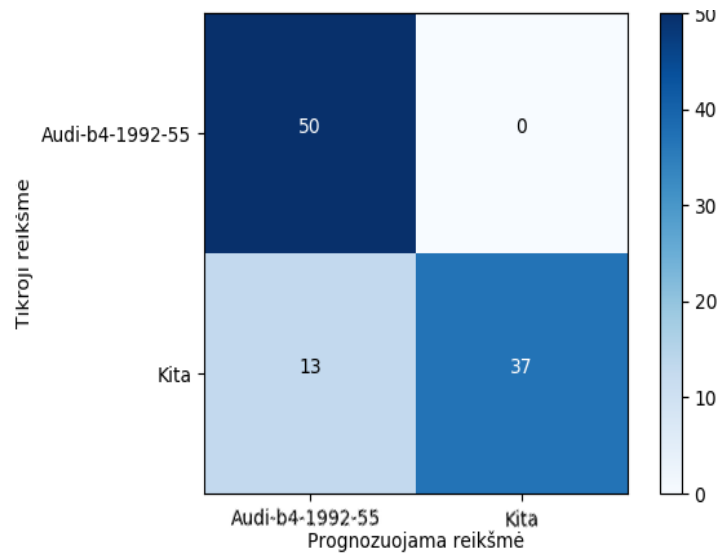
22 pav. Klasifikavimo lentelė

## 2.2. Audi b4 1992 55 kw ir Kita (2 testas)



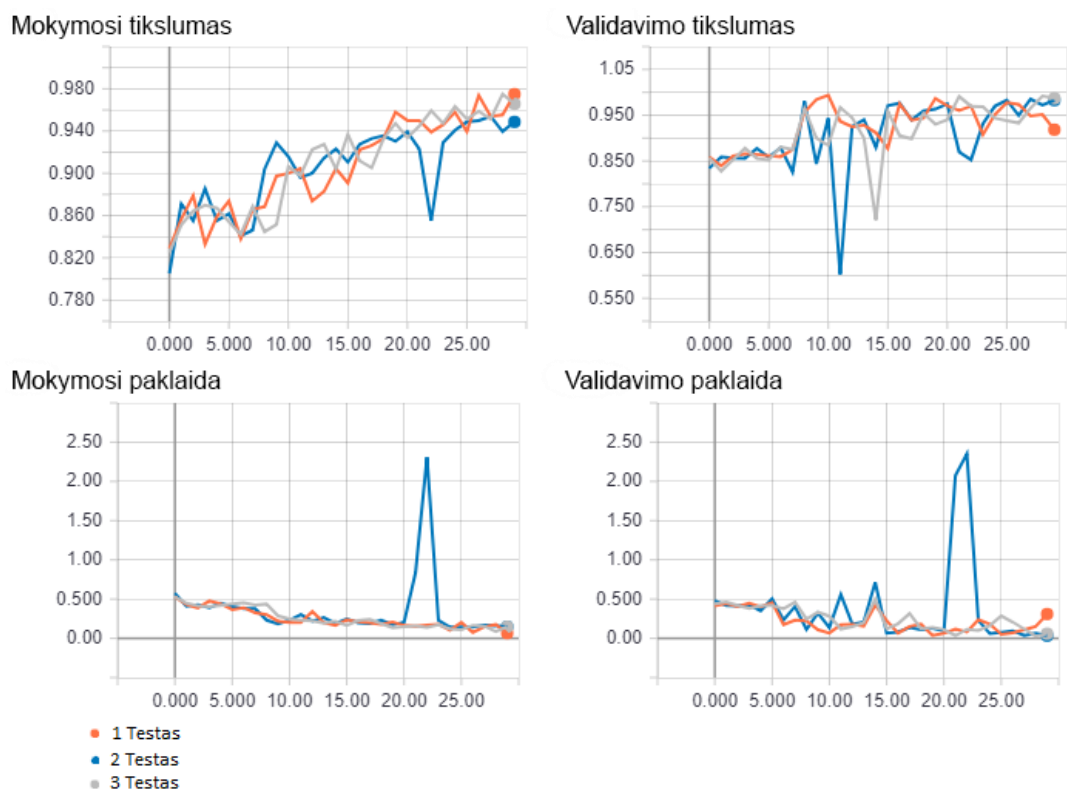
23 pav. Audi b4 1992 55kw ir Kita modelio tikslumo ir paklaidos kreivės.

Antrasis atliktas testas buvo analogiškas pirmajam, tik su didesniu duomenų kiekiu: mokymosi ir validavimo metu buvo paimti 3700 paveikslėlių. Kaip ir buvo tikėtasi mokymosi tikslumo ir paklaidos kreivės tapo glodesnės 23 pav. Klasifikavimo lentelėje pateiktoje 24 paveikslėlyje matomas 100% tikslumas ieškomojo automobilio atžvilgiu, tačiau fiksuojamas kritimas klasės *kita* atpažinime lyginant su 22 paveikslėlyje pateiktais duomenimis.



24 pav. Klasifikavimo lentelė.

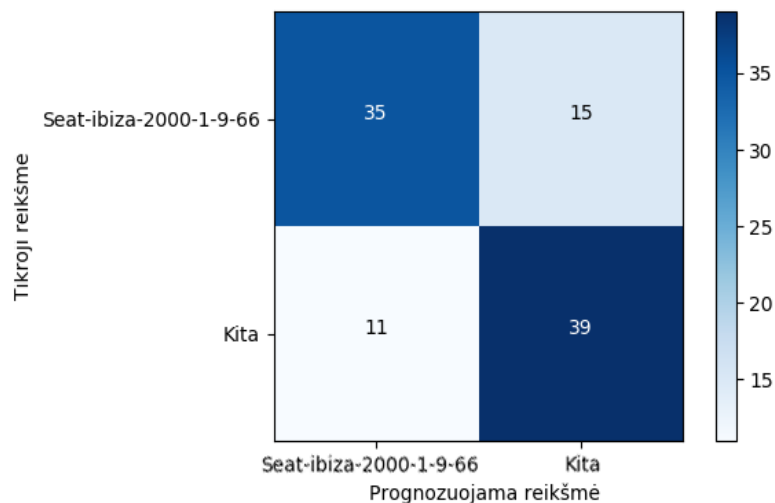
### 2.3. Seat Ibiza 2000 66kw ir Kita



25 pav. Seat Ibiza 2000 66kw ir Kita modelio tikslumo ir paklaidos kreivės.

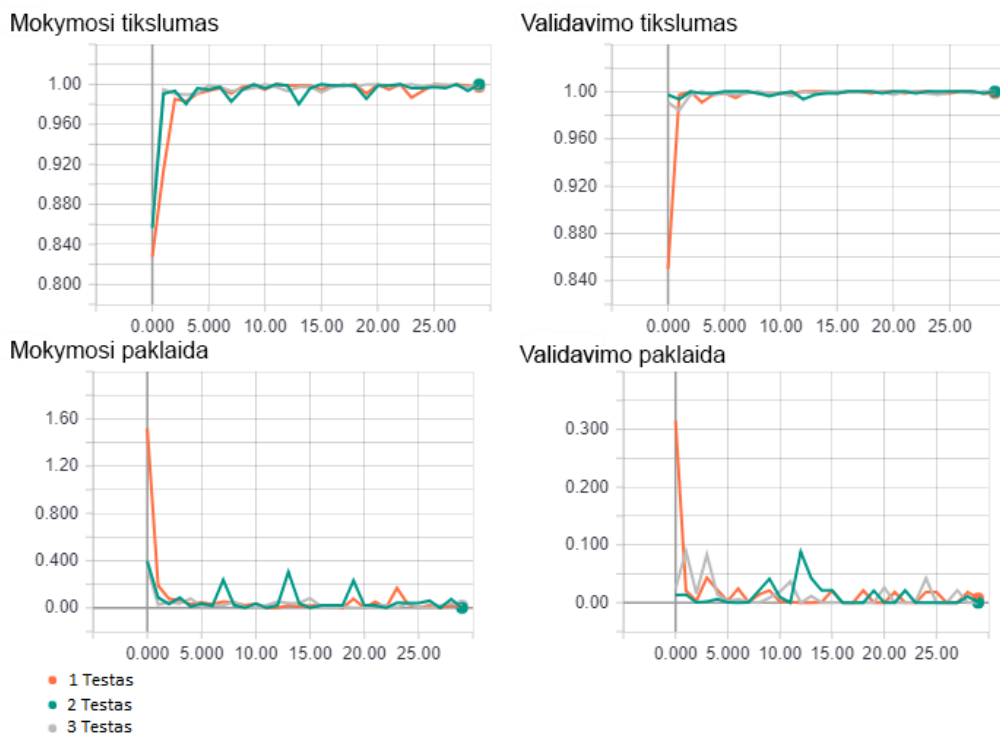
Trečiojo testo metu ieškomasis automobilis buvo *Seat Ibiza* 2000m., 1.9l., 66kw. Testo metu taip pat atlikti trys testai, kurių tikslumo ir paklaidos rezultatai pateikti 25 pav. Šio testo antrajame teste (mėlyna kreivė) galime matyti didžiulį paklaidos ir tikslumo šuolį ties 23 epocha. Taip nutikti galėjo dėl tuo metu įvykusio persotinimo (angl. overfitting) arba dėl mirusių

neuronų. Anot kreivių šis modelis pasiekė beveik 98% tikslumą. Patikrinus šį modelį su nematytais duomenimis 26 pav., gautasis tikslumas buvo tik 78%. Išvada: mokymosi metu reikia daugiau duomenų.



26 pav. Klasifikavimo lentelė.

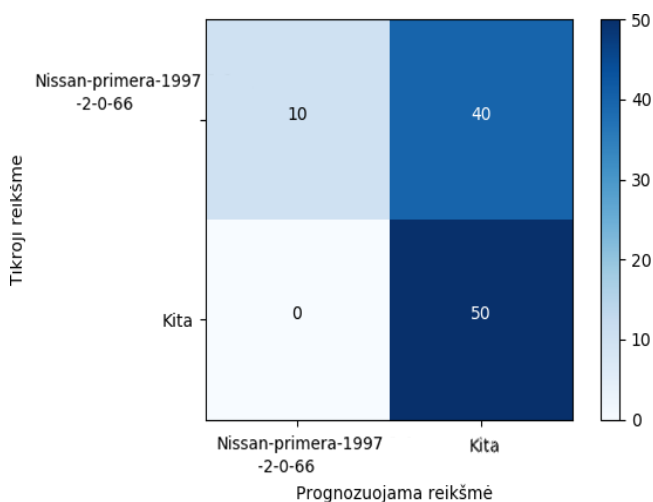
#### 2.4. Nissan primera 1997 66kw ir Kita



27 pav. Nissan primera 1997 66kw ir Kita modelio tikslumo ir paklaidos kreivės.

Ketvirtojo testo metu ieškomasis automobilis buvo Nissan primera 1997m, 2.0l, 66kw. Tikslumo ir paklaidos kreivės pateiktos 27 pav. Klasifikavimo lentelė su nematytais duomenimis 28 pav.

Šiuo atveju tikslumas iš 27 pav., pasiektas 100% tačiau klasifikavimo lentelės atveju tikslumas tesiekia tik 60%. Taip pat tikslumo ir paklaidos kreivės labai greitai pasiekia savo tikslą kas nėra gerai. Taip pat galima daryti prielaidą, jog egzistuoja duomenų trūkumas, bei galbūt šio automobilio atveju nėra tinkamas optimizacijos algoritmas. Sukurtojo modelio atveju nėra naudojamas *BatchNormalization*, kuris galėtų pagelbėti šioje situacijoje.



28 pav. Klasifikavimo lentelė.

### 2.5.3. Tyrimo rezultatai

- Tinklo apmokymui reikia daugiau duomenų. Tai indikuoja 20, 25, 27 paveikslėliai, kuriuose matomi staigūs kreivių šuoliai.
- Didesnis mokymosi duomenų kiekis duoda žymiai stabilesnę mokymąsi, bei tinklas yra geriau apmokomas. Tai atsispindi skirtumuose tarp 22 ir 24 paveikslėlių.
- Dvinarė klasifikacija parodė geresnius rezultatus nei daugiaklasė.
- Sukurtas giliojo mokymosi modelis, kuris parodė gana neblogus rezultatus atpažįstant Audi b4 1992m., 1,9l., 55 kw automobilį, bei Seat Ibiza 2000m., 1,9l., 66kw automobilį, tačiau sunkiai atpažino automobilį Nissan Primera 1997m., 2,0l., 66 kw. Vertėtų pastebėti, jog Seat yra Audi ir Volkswagen grupės narys, todėl gali būti, jog jų varikliai turi panašumų, ir visą tai atsispindėjo mokymosi rezultatuose. Tuo tarpu Nissan yra Japonijos gamintojas.

- Greičiausiai modelio rezultatai galėtų būti pagerinti turint geresnį įrašymo įrenginį, kuris galėtų įrašyti ultragarsą. Šiuo atveju buvo naudojamas paprastas telefono mikrofonas.
- Taip pat vertėtų paminėti, kad darbo metu sukurtoji *GitHub* repozitorija<sup>1</sup> sulaukė nors ir labai nedidelio, tačiau paskatinančio susidomėjimo.

## 2.6. Tobulinimo galimybės

Sukurtoji sistema galėtų būti patobulinta šiais aspektais:

- Spektrinių charakteristikų generavimo greičiu – išsaugant spektrines charakteristikas skaitiniu formatu, bei jų generavimui naudojant GPU. To pasekoje reikėtų keisti giliojo mokymosi įvesties sluoksnį.
- Modelio patobulinimas:
  - Pridėti BatchNormalization()
  - Apsvarstyti kiekviename sluoksnyje dropout funkcijos naudojimą. Tai turėtų padėti sumažinti mokymosi metu patiriamą persipildymą (angl. overfitting).
  - Išbandyti kitus optimizacijos algoritmus, bei kitas aktyvacijos funkcijas.
  - Modeliui pateikti audio įrašų dalis suskaidytus į mažesnius intervalus.
  - Didesnis modelis galėtų veikti geriau.
- Implementuoti tiesioginį audio srauto skaitymą iš mikrofono.

## 2.7. Potencialūs sistemos trūkumai

Sukurtoji sistema neveiktų šiais atvejais:

- Automobilio variklio garsas pasikeistų neatpažįstamai – pavyzdžiui būtų didinama variklio galia, darbinis tūris.
- Sugedus automobilio varikliui.

---

<sup>1</sup> <https://github.com/KaroDievas/car-sound-classification-with-keras>

## Išvados

Darbo metu buvo apžvelgti egzistuojantys audio srauto apdorojimo algoritmai bei jų panaudojimo būdai. Apžvelgti sistemų mokymosi algoritmai – jų skirtumai ir panašumai. Taip pat buvo apžvelgti tiesiogiai su magistro darbo uždaviniu susiję darbai bei jau egzistuojančios sistemos. Panašiausias pristatytas darbe [WHI].

Egzistuojanti muzikos atpažinimo programa „Audprint“ [ELI] nėra tinkama motorinių transporto priemonių variklio skleidžiamo garso atpažinimui.

Automobilių variklių skleidžiamo garso atpažinimui buvo pasiūlytas sistemos modelis, kuris sugebėjo atpažinti autoriaus automobilį. Tačiau bandant atpažinti kitus automobilius rezultatai buvo prastesni.

Sukurtasis mokymosi modelis reikalauja kur kas daugiau mokymosi duomenų nei buvo surinkta. Vienos minutės variklio skleidžiamo garso įrašo nepakankama tinklo apmokymui. Tačiau jau apmokytajam tinklui atpažinimui užtektų ir 400ms.

Didesnis mokymosi duomenų kiekis duoda stabilesnį mokymąsi.

Tinklo daugiaklasinis apmokymas reikalauja žymiai daugiau laiko ir jo tikslumas yra žymiai mažesnis lyginant su dvinariu apmokymu. Todėl realioje sistemos realizacijoje turėtų būti naudojamas dvinaris apmokymas.

## Šaltiniai

- [AMU01] Yali Amit and Alejandro Murua. Speech Recognition Using Randomized Relational Decision Trees. 2001. Prieiga per internetą: <<https://galton.uchicago.edu/~amit/Papers/sound1.pdf>>
- [ASH] Morlet wavelets in quantum mechanics. John Ashmead. Prieiga per internetą: <<https://arxiv.org/ftp/arxiv/papers/1001/1001.0250.pdf>>
- [BRO] Introduction to the Python Deep Learning Library Theano. Jason Brownlee. Prieiga per internetą: <<https://machinelearningmastery.com/introduction-python-deep-learning-library-theano/>>
- [BWA] Mark A. Bartsch, Member, IEEE and Gregory H. Wakefield, Member, IEEE. Audio Thumbnailing of Popular Music Using Chroma-Based Representations. Prieiga per internetą: <<http://musicweb.ucsd.edu/~sdubnov/CATbox/Reader/ThumbnailingMM05.pdf>>
- [CAF] Caffe. Deep learning framework. Prieiga per internetą: <<http://caffe.berkeleyvision.org/>>
- [CE09] Courtenay Cotton and Daniel P. W. Ellis. Finding similar acoustic events using matching pursuit and locality-sensitive hashing. 2009. Prieiga per internetą: <<http://www.icsi.berkeley.edu/pubs/ai/findingsimilar10.pdf>>
- [CRO] How does Keras calculate accuracy? Dries Cronje. Prieiga per internetą: <<https://www.quora.com/How-does-Keras-calculate-accuracy>>
- [DIR16] Dirbtinio intelekto metodų taikymas kraštų aptikimui. Darius Dirvanauskas. 2016. Prieiga per internetą: <<https://core.ac.uk/download/pdf/51698688.pdf>>
- [DYU] Li Deng and Dong Yu. Deep Learning Methods and Applications. Prieiga per internetą: <<http://research.microsoft.com/pubs/209355/DeepLearning-NowPublishing-Vol7-SIG-039.pdf>>



[ELI] Dan Elis. Audfprint. Prieiga per internetą: <<http://www.ee.columbia.edu/%7Edpwe/resources/matlab/audfprint/>>

[FFM] A complete, cross-platform solution to record, convert and stream audio and video. Prieiga per internetą: <<https://www.ffmpeg.org>>

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press. 2016. Prieiga per internetą: <<http://www.deeplearningbook.org>>

[GIA] Theodoros Giannakopoulos. pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. Prieiga per internetą: <<http://journals.plos.org/plosone/article/asset?id=10.1371/journal.pone.0144610.PDF>>

[GSD+] Mordechai Guri, Yosef Solewicz, Andrey Daidakulov, Yuval Elovici Ben-Gurion University of the Negev Cyber Security Research Center. Fansmitter: Acoustic Data Exfiltration from (Speakerless) Air-Gapped Computers. Prieiga per internetą: <<https://arxiv.org/ftp/arxiv/papers/1606/1606.05915.pdf>>

[HUG] Hybrid and Electric Cars Must Make Noise by 2020. Prieiga per internetą: <<http://www.thedrive.com/news/18820/hybrid-and-electric-cars-must-make-noise-by-2020>>

[KAM09] Juozas Kamarauskas. Asmens atpažinimas pagal balsą. 2009.

[KHS] Yan Ke, Derek Hoiem, Rahul Sukthankar, School of Computer Science, Carnegie Mellon. Intel Research Pittsburgh. Computer Vision for Music Identification. Prieiga per internetą: <<http://www.cs.cmu.edu/~yke/musicretrieval/cvpr2005-mr.pdf>>

[LYG] Lygsagne. Prieiga per internetą: <<https://lasagne.readthedocs.io/en/latest/user/installation.html>>

[LL97] A.Lipeika, J.Lipeikienė. Diskretinio laiko signalų ir sistemų analizė. Vilnius, 1997.

- [LCC10] Tom LH. Li\*, Antoni B. Chan\*and Andy HW. Chun\*. Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network, 2010. Prieiga per internetą: <[http://www.iaeng.org/publication/IMECS2010/IMECS2010\\_pp546-550.pdf](http://www.iaeng.org/publication/IMECS2010/IMECS2010_pp546-550.pdf)>
- [LRU09] Yizhar Lavner and Dima Ruinskiy. A Decision-Tree-Based Algorithm for Speech/Music Classification and Segmentation, 2009. Prieiga per internetą: <<http://asmp.eurasipjournals.springeropen.com/articles/10.1155/2009/239892>>
- [MAC03] MacKay, David, J.C. Information Theory, Inference, and Learning Algorithms. Cambridge University Press. ISBN 9780521642989. 2003. <<http://www.inference.phy.cam.ac.uk/itprnn/book.pdf>>
- [MAL] Time-Frequency Dictionaries. Stephane Mallat. Prieiga per internetą: <[https://cnx.org/contents/SkfT37\\_1@2/Time-Frequency-Dictionaries](https://cnx.org/contents/SkfT37_1@2/Time-Frequency-Dictionaries)>
- [MPV] Piano Transcription using Wavelet Decomposition and Neural Networks. Esben Madsen, Johnni Thomsen Pedersen and Louise Baldus Vestergaard. Prieiga per internetą: <[http://vbn.aau.dk/files/16328474/piano\\_transcription.pdf](http://vbn.aau.dk/files/16328474/piano_transcription.pdf)>
- [NIE93] Lars Bramsløw Nielsen. A Neural Network Model for Prediction of Sound Quality. TECHNICAL UNIVERSITY OF DENMARK, 1993. Prieiga per internetą: <[http://www.mp3-tech.org/programmer/docs/LA\\_report\\_53.pdf](http://www.mp3-tech.org/programmer/docs/LA_report_53.pdf)>
- [OBS] Python Toolbox for seismology/seismological observatories. Prieiga per internetą: <<https://docs.obspy.org>>
- [OE] James P. Ogle and Daniel P.W. Ellis. Fingerprinting to identify repeated sound events in long-duration personal audio. Prieiga per internetą: <<http://www.ee.columbia.edu/~dpwe/pubs/OgleE07-pershash.pdf>>
- [PVE] Paolo Prandoni and Martin Vetterli. Signal Processing for Communications. Prieiga per internetą: <<http://www.sp4comm.org/webversion/livre.html>>

[REP] Visualizing parts of Convolutional Neural Networks using Keras and Cats. Erik Reppel. Prieiga per internetą: <<https://hackernoon.com/visualizing-parts-of-convolutional-neural-networks-using-keras-and-cats-5cc01b214e59>>

[RM] Rokach, Lior; Maimon, O. (2008). Data mining with decision trees: theory and applications. World Scientific Pub Co Inc. Prieiga per internet: <<https://doc.lagout.org/Others/Data%20Mining/Data%20Mining%20with%20Decision%20Trees%20Theory%20and%20Applications%20%282nd%20ed.%29%20%5BRokach%20%26%20Maimon%202014-10-23%5D.pdf>>.

[RUD] An overview of gradient descent optimization algorithms. Sebastian Ruder. Prieiga per internetą: <<http://ruder.io/optimizing-gradient-descent/index.html#rmsprop>>

[SMI] Steven W. Smith, Ph.D. The Scientist and Engineer's Guide to Digital Signal Processing. Prieiga per internetą: <<http://www.dspguide.com/pdfbook.htm>>

[SSS+12] Johannes A. Stork, Luciano Spinello, Jens Silva, Kai O. Arras. Audio-Based Human Activity Recognition Using Non-Markovian Ensemble Voting. In Proc. of the IEEE International Symposium on Robot and Human Interactive Communication (RoMan), Paris, France, 2012. Prieiga per internetą: <<http://www2.informatik.uni-freiburg.de/~spinello/storkROMAN12.pdf>>

[TAN] Wavelet Transforms in Time Series Analysis. Andrew Tangborn. Prieiga per internetą: <[https://www.atmos.umd.edu/~ekalnay/syllabi/AOSC630/Wavelets\\_2010.pdf](https://www.atmos.umd.edu/~ekalnay/syllabi/AOSC630/Wavelets_2010.pdf)>

[TEN] TensorFlow. Prieiga per internetą: <<https://www.tensorflow.org/>>

[THE] Theano. Prieiga per internetą: <<http://deeplearning.net/software/theano/introduction.html>>

[VEL] Deep learning for complete beginners: convolutional neural networks with keras. Petar Veličković. Prieiga per internetą:

<<https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html>>

[WAN] Avery Li-Chun Wang. An Industrial-Strength Audio Search Algorithm. Prieiga per internetą: <<http://www.ee.columbia.edu/~dpwe/papers/Wang03-shazam.pdf>>

[WHI] The fight against illegal deforestation with TensorFlow. Prieiga per internetą: <<https://www.blog.google/topics/machine-learning/fight-against-illegal-deforestation-tensorflow/>>

[WSK] Vehicle Sound Signature Recognition by Frequency Vector Principal Component Analysis. Huadong Wu, Mel Siegel, Pradeep Khosla. Prieiga per internetą: <<http://repository.cmu.edu/cgi/viewcontent.cgi?article=1550&context=isr> >

[XWT+15] Empirical Evaluation of Rectified Activations in Convolution Network. Bing Xu, Naiyan Wang, Tianqi Chen, Mu Li. 2015. Prieiga per internetą: <<https://arxiv.org/pdf/1505.00853.pdf>>

## Santrumpos

Irašomasis įrenginys – samsung xcover 2;

FT – Furje transformacija (angl. Fourier transform)

DFT – diskreti Furje transformacija (angl. Discrete Fourier transformation)

DTFT – Diskretaus laiko Furje transformacija (angl. Discrete time Fourier transformation)

DFS – diskrečioji Furje eilutė (angl. Discrete Furje series)

FFT – greita Furje transformacija (angl. Fast Fourier transformation)

VIPF – vienetinio impulso perdavimo funkcija

IIR – begalinio impulso atsako (angl. Infinite Impulse Response)

ANN – intelektualūs neuroniniai tinklai (angl. Artificial neural network)

DT – sprendimų medžiai (angl. Decision tree).

DL – gylusis mokymasis (angl. Deep learning).

SVM – pagalbinės vektorių mašinos (angl. Support vector machines)

RL – mokymosi stiprinimo (angl. Reinforcement learning)

kNN – k-Artimiausių kaimynų algoritmas (angl. k-Nearest Neighbors algorithm)

RFC – Atsitiktinis miško klasifikatoriaus (angl. Random Forest classifier)

MFCC – Melo dažnių keistro koeficientų algoritmas (angl. el Frequency Cepstral Coefficient)

## Priedai

Nr.	Irašo pavadinimas	Data	°C		Markė	Modelis	Metai	Darbinis tūris (l)	Galia (KW)	Papildomi veiksniai	
			Oro	Variklio							
1	Balsas001	2015-12-30	-8	0	Audi	B4	1992	1.9	55		X
2	Balsas002	2015-12-31	-12	0	Audi	B4	1992	1.9	55		X
3	Balsas003	2016-01-03	-14	0	Audi	B4	1992	1.9	55		X
4	Balsas004	2016-01-04	-17	0	Audi	B4	1992	1.9	55		X
5	Balsas005	2016-01-05	-22	0	Audi	B4	1992	1.9	55		X
6	Balsas006	2016-01-06	-24	0	Audi	B4	1992	1.9	55		X
7	Balsas007	2016-01-08	-17	0	Audi	B4	1992	1.9	55		X
8	Balsas008	2016-01-09	-8	0	Audi	B4	1992	1.9	55		X
9	Balsas009	2016-01-11	-4	0	Audi	B4	1992	1.9	55		X
10	Balsas010	2016-01-15	-11	0	Audi	B4	1992	1.9	55		X
11	Balsas011	2016-01-15	-17	70	Audi	B4	1992	1.9	55		X
12	Balsas012	2016-01-16	-14	0	Audi	B4	1992	1.9	55		X
13	Balsas013	2016-01-17	-7	0	Audi	B4	1992	1.9	55		X
14	Balsas014	2016-01-18	-9	65	Audi	B4	1992	1.9	55		X
15	Balsas015	2016-01-19	-11	0	Audi	B4	1992	1.9	55		X
16	Balsas016	2016-01-19	-9	60	Audi	B4	1992	1.9	55		X
17	Balsas017	2016-01-20	-8	0	Audi	B4	1992	1.9	55		X
18	Balsas018	2016-01-20	-8	60	Audi	B4	1992	1.9	55		X
19	Balsas019	2016-01-23	-9	0	Audi	B4	1992	1.9	55		X
20	Balsas020	2016-01-23	-9	70	Audi	B4	1992	1.9	55		X
21	Balsas021	2016-01-29	3	70	Audi	B4	1992	1.9	55		X
22	Balsas022	2016-01-31	2	0	Audi	B4	1992	1.9	55	Vėjas fone	X

23	Balsas023	2016-02-04	1	70	Audi	B4	1992	1.9	55		X
24	Balsas024	2016-02-05	1	60	Audi	B4	1992	1.9	55		X
25	Balsas025	2016-02-06	2	60	Audi	B4	1992	1.9	55		X
26	Balsas026	2016-02-07	6	0	Audi	B4	1992	1.9	55		X
27	Balsas027	2016-02-11	-1	0	Audi	B4	1992	1.9	55		X
28	Balsas028	2016-02-12	0	70	Audi	B4	1992	1.9	55		X
29	Balsas029	2016-02-14	1	70	Audi	B4	1992	1.9	55	pakeisti tepalai	X
30	Balsas030	2016-02-18	-4	60	Audi	B4	1992	1.9	55		X
31	Balsas031	2016-02-18	2	70	Audi	B4	1992	1.9	55		X
32	Balsas032	2016-02-19	2	0	Audi	B4	1992	1.9	55		X
33	Balsas033	2016-02-20	2	70	Audi	B4	1992	1.9	55		X
34	Balsas034	2016-02-23	2	70	Audi	B4	1992	1.9	55		X
35	Balsas035	2016-02-24	1	60	Audi	B4	1992	1.9	55		X
36	Balsas036	2016-02-25	1	60	Audi	B4	1992	1.9	55		X
37	Balsas037	2016-02-26	1	70	Audi	B4	1992	1.9	55		X
38	Balsas038	2016-02-27	1	60	Audi	B4	1992	1.9	55		X
39	Balsas039	2016-02-27	2	65	Audi	B4	1992	1.9	55		X
40	Balsas040	2016-02-27	2	65	Audi	B4	1992	1.9	55	išminta sankaba	X
41	Balsas041	2016-02-28	2	0	Audi	B4	1992	1.9	55		X
42	Balsas042	2016-02-28	0	65	Audi	B4	1992	1.9	55		X
43	Balsas043	2016-02-29	-1	65	Audi	B4	1992	1.9	55		X
44	Balsas044	2016-03-02	0	60	Audi	B4	1992	1.9	55	pro šali važiavo mašina	X
45	Balsas045	2016-03-04	-1	65	Audi	B4	1992	1.9	55		X
46	Balsas046	2016-03-05	0	0	Audi	B4	1992	1.9	55		X
47	Balsas047	2016-03-05	1	65	Audi	B4	1992	1.9	55		X

48	Balsas048	2016-03-08	4	80	Audi	B4	1992	1.9	55		X
49	Balsas049	2016-03-15	1	65	Audi	B4	1992	1.9	55		X
50	Balsas050	2016-04-07	15	65	Audi	B4	1992	1.9	55		X
51	Balsas051	2016-04-08	13	60	Audi	B4	1992	1.9	55		X
52	Balsas052	2016-04-09	10	60	Audi	B4	1992	1.9	55		X
53	Balsas053	2016-04-11	10	60	Audi	B4	1992	1.9	55		X
54	Balsas054	2016-04-12	13	65-70	Audi	B4	1992	1.9	55		X
55	Balsas055	2016-04-13	8	60	Audi	B4	1992	1.9	55		X
56	Balsas056	2016-06-12	13	70	Audi	B4	1992	1.9	55		X
57	Balsas057	2016-06-20	17	80	Audi	B4	1992	1.9	55		X
58	Balsas058	2016-06-21	16	70	Audi	B4	1992	1.9	55		X
59	Balsas059	2016-08-03	13	0	Audi	B4	1992	1.9	55		X
60	Balsas060	2016-08-03	19	60	Audi	B4	1992	1.9	55		X
61	Balsas061	2016-08-04	13	0	Audi	B4	1992	1.9	55		X
62	Balsas062	2016-08-04	20	90	Audi	B4	1992	1.9	55		X
63	Balsas063	2016-08-07	15	0	Audi	B4	1992	1.9	55		X
64	Balsas064	2016-08-07	18	90	Audi	B4	1992	1.9	55		X
65	Balsas065	2016-08-08	13	0	Audi	B4	1992	1.9	55		X
66	Balsas066	2016-08-08	23	80	Audi	B4	1992	1.9	55		X
67	Balsas067	2016-08-09	21	100	Audi	B4	1992	1.9	55		X
68	Balsas068	2016-08-10	14	0	Audi	B4	1992	1.9	55		X
69	Balsas069	2016-08-10	16	100	Audi	B4	1992	1.9	55		X
70	Balsas070	2016-08-12	18	90	Audi	B4	1992	1.9	55		X
71	Balsas071	2016-08-16	10	0	Audi	B4	1992	1.9	55		X
72	Balsas072	2016-08-19	21	100	Audi	B4	1992	1.9	55		X



73	Balsas073	2016-08-20	15	0	Audi	B4	1992	1.9	55		X
74	Balsas074	2016-08-22	16	0	Audi	B4	1992	1.9	55		X
75	Balsas075	2016-08-22	24	100	Audi	B4	1992	1.9	55		X
76	Balsas076	2016-08-23	21	90	Audi	B4	1992	1.9	55		X
77	Balsas077	2016-08-24	14	0	Audi	B4	1992	1.9	55		X
78	Balsas078	2016-08-24	21	100	Audi	B4	1992	1.9	55		X
79	Balsas079	2016-08-29	15	0	Audi	B4	1992	1.9	55		X
80	Balsas080	2016-09-05	13	0	Audi	B4	1992	1.9	55		X
81	Balsas081	2016-09-18	16	0	Kia	Sportage	2003	2.0	103	Veikia tik Kia.	
82	Balsas082	2016-09-18	16	0	Kia	Sportage	2003	2.1	103	Veikia tik Kia.	X
83	Balsas083	2016-09-18	16	0	Audi	B4	1992	1.9	55	Veikia Audi	
84	Balsas084	2016-09-18	16	0	Audi	B4	1992	1.9	55	Veikia Audi	X
85	Balsas085	2016-09-18	16	0	Suzuki	Grand Vitara	2000	2.0	85	Veikia Suzuki	
86	Balsas086	2016-09-18	16	0	Suzuki	Grand Vitara	2000	2.0	85	Veikia Suzuki	X
87	Balsas087	2016-09-18	16	0						Veikia: 81,83,85	
88	Balsas088	2016-09-18	16	0						Veikia: 81,83,86	
89	Balsas089	2016-09-18	16	0						Veikia: 81,85	
90	Balsas090	2016-09-18	16	0						Veikia:81,83	
91	Balsas091	2016-09-18	16	0						Veikia: 83,85	
92	Balsas092	2016-09-18	16	0						tuščias garsas	
93	Balsas093	2016-09-18	16	0						tuščias garsas	
94	Balsas094	2016-11-25	-4	0	Audi	B4	1992	1.9	55		X
95	Balsas095	2016-12-06	0	0	Audi	B4	1992	1.9	55		X
96	Balsas096	2016-12-06	-3	60	Audi	B4	1992	1.9	55		X
97	Balsas097	2016-12-19	2	60	Audi	B4	1992	1.9	55		X

98	Balsas098	2016-12-20	2	60	Audi	B4	1992	1.9	55		X
99	Balsas099	2016-12-21	1	60	Audi	B4	1992	1.9	55		
100	Balsas100	2016-12-22	-1	60	Audi	B4	1992	1.9	55		
101	Balsas101	2017-01-08	-14	0	Nissan	Primera	1997	2.0	66		
102	Balsas102	2017-01-15	-6	0	Opel	Vectra	2000	2.0	60		X
103	Balsas103	2017-01-15	-6	0	Opel	Vectra	2000	2.0	60		X
104	Balsas104	2017-01-17	3	0	Lancia	Delta	2009	2.0	121		X
105	Balsas105	2017-01-18	3	0	Audi	A4	1999	2.5	110		X
106	Balsas106	2017-01-18	3	0	Audi	A4	1999	2.5	110		X
107	Balsas107	2017-02-02	4	0	Seat	Ibiza	2000	1.9	66		
108	Balsas108	2017-02-10	2	0	Peugot	307	2005	1.6	80		X
109	Balsas109	2017-02-10	2	0	Volvo	V40	2004	1.9	85		X
110	Balsas110	2017-03-15	4	0	Renault	Laguna	2007	1.9	96		X
111	Balsas111	2017-03-15	4	0	Peugot	307	2002	2.0	79		X
112	Balsas112	2017-03-15	4	0	VW	Tiguan	2015	2.0	85		X
113	Balsas113	2017-03-15	4	0	BMW	X5	2006	3.0	190		X
114	Balsas114	2017-03-15	4	0	Opel	Omega	2002	2.5	110		X
115	Balsas115	2017-03-15	4	0	Seat	Alhambra	2005	1.9	85		X
116	Balsas116	2017-03-15	4	0	VW	Touran	2004	1.9	77		X
117	Balsas117	2017-03-15	4	0	Audi	A6	2012	2.0	130		X
118	Balsas118	2017-03-15	4	0	Opel	Vectra	1999	2.0	74		X
119	Balsas119	2017-03-15	4	0	BMW	x3	2016	2.0	140		X
120	Balsas120	2017-03-15	4	0	Renault	Clio	2008	1.5	52		X
121	Balsas121	2017-03-15	4	0	Volvo	XC70	2006	2.2	136		X
122	Balsas122	2017-03-15	4	0	Mitsubishi	Carisma	2002	1.9	75		X

123	Balsas123	2017-03-15	4	0	Toyota	Corola	2003	2.0	85		X
124	Balsas124	2017-03-15	4	0	Volvo	XC90	2003	2.4	120		X
125	Balsas125	2017-03-15	4	0	Toyota	Rav4	2014	2.2	110		X
126	Balsas126	2017-03-15	4	0	BMW	530	2012	3.0	190		X
127	Balsas127	2017-03-15	4	0	Kia	Optima	2012	1.7	100		X
128	Balsas128	2017-04-01	6	0	BMW	320	2002	2.0	110		X
129	Balsas129	2017-04-01	6	0	Ford	Galaxy	1998	1.9	66		X
130	Balsas130	2017-04-01	6	0	VW	Passat	2011	2.0	100		X
131	Balsas131	2017-04-01	6	0	Volvo	V50	2006	1.9	81		X
132	Balsas132	2017-04-01	6	0	Chevrolet	Lacete	2008	2.0	103		X
133	Balsas133	2017-04-01	6	0	Audi	A6	1998	2.5	110		X
134	Balsas134	2017-04-01	6	0	Citroen	C3	2003	1.4	50		X
135	Balsas135	2017-04-01	6	0	VW	Passat	2002	1.9	96		X
136	Balsas136	2017-04-01	6	0	Toyota	Yaris	2009	1.4	66		X
137	Balsas137	2017-04-01	6	0	Volvo	V50	2007	2.0	100		X
138	Balsas138	2017-04-01	6	0	Toyota	Rav4	2007	3.0	145		X
139	Balsas139	2017-04-02	8	0	Toyota	Rav4	2007	3.0	145		X
140	Balsas140	2017-04-02	8	0	BMW	325 coupe	2007	3.0	145		X
141	Balsas141	2017-04-02	8	0	BMW	X5	2006	3.0	160		X
142	Balsas142	2017-04-02	8	0	VW	Golf	2000	1.9	85		X
143	Balsas143	2017-04-02	8	0	Audi	A4	1998	1.9	81		X
144	Balsas144	2017-04-02	8	0	Škoda	Octavia	2005	1.9	77		X
145	Balsas145	2017-04-02	8	0	Audi	A6	1997	2.5	103		X
146	Balsas146	2017-04-02	8	0	Opel	Astra	2000	1.7	55		X
147	Balsas147	2018-03-31	9	0	Seat	Ibiza	2000	1.9	66	10 min. Įrašas	X

148	Balsas148	2018-04-02	3	0	Nissan	Primera	1997	2.0	66	10 min. Įrašas	X
149	Balsas149	2018-04-07	9	0	Kia	Sportage	2002	2.0	105	10 min. Įrašas	X
150	Balsas150	2018-04-07	9	0	Suzuki	Grand Vitara	2000	2.0	85	10 min. Įrašas	X

Lentelė 2. Audio įrašai

Paaiškinimai:

X – įrašai kurie pateko į gilųjį mokymąsi